



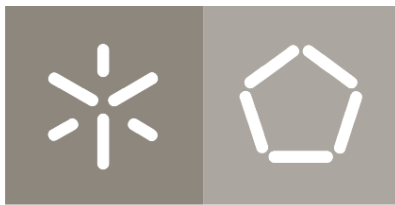
Universidade do Minho  
Escola de Engenharia

Tiago Marrucho Veloso

Gestão de filas de espera no Serviço de  
Urgência

Tiago Marrucho Veloso  
Gestão de filas de espera no Serviço de Urgência





**Universidade do Minho**

Escola de Engenharia

Tiago Marrucho Veloso

## Gestão de filas de espera no Serviço de Urgência

Tese de Mestrado

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia Biomédica

Trabalho realizado sob a orientação do

**Professor Doutor António Carlos da Silva Abelha**  
**Universidade do Minho**

Outubro de 2011

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Braga, \_\_\_\_/\_\_\_\_/\_\_\_\_\_

Assinatura: \_\_\_\_\_



## Agradecimentos

Este espaço é dedicado às pessoas que deram a sua contribuição para que esta dissertação fosse concretizada. A todos eles deixo aqui o meu sincero agradecimento.

Primeiro gostaria de agradecer ao meu orientador Doutor António Abelha, pela forma como orientou o meu trabalho, por toda a ajuda, orientação, disponibilidade e pelos seus conselhos que guiaram o desenvolvimento do meu trabalho. Estou grato também pela liberdade de ação que me admitiu. Esta foi decisiva para que este trabalho contribuisse para o meu desenvolvimento pessoal

Ao Professor Doutor José Machado agradeço a atenção e disponibilidade mostrada ao longo do projeto.

Agradeço aos meus amigos pelo constante apoio ao longo deste ano. Em especial os meus colegas de curso, pela companhia e ajuda constante ao longo da minha vida académica.

À minha família pelas condições que me ofereceram ao longo destes cinco anos. O constante apoio que me deram foi essencial para me fazer crescer pessoal e academicamente.

## **Title**

Management of waiting lines in the Emergency Department

## **Abstract**

The constant evolution of information technology allows its application in health care to become increasingly widespread. Its use allows obtaining economic, effective and rational health care with quality, as well as a better organization of the health care system at any level.

The Emergency Department appears with an increasingly important role in the National Health Services. In this department, the waiting times are an important factor in patient satisfaction. An Emergency Department where patients are seen quickly is considered a good Emergency Department. The presentation of waiting times in the Emergency Department improves the process of waiting for the patients, while still allowing the Emergency Department to administrate these waiting times and take action if the waiting times aren't normal and are negative to the patients. Another important aspect in the Emergency Department is the patient call and the annunciation messages to patients, these are sometimes not clearly perceptible. This may occur for example, due to the state of the voice of the person who makes the call or because the room where the call is made presents a lot of noise. In order to solve this situation, computer synthesized speech that rules out some of these variables can be used, automatizing the process of calling patients and reading messages.

The main objective of this thesis includes the development of a functional desktop application that allows the presentation of information about waiting times for the patients in the Emergency Department, and makes patient calls and announces messages using computer synthesized speech.

In this thesis, the Windows Presentation Forms (WPF) technology was used to develop a desktop application that displays the average waiting times and the time that the last patient waited, both obtained in relation to the time required for a patient to be triaged and the 5 categories of the Manchester Triage System. The application developed also allows the presentation of videos and written messages, makes patient calls and announces messages. For patient calling and message announcing, a Text-To-Speech or TTS is used in order to transform the text into a sound message.

Using the ASP.NET technology an application that allows the management of the desktop application features was also developed. This application can manage videos and the insertion, modification and deletion of written messages and messages to be read.

In conclusion, an application was developed that matches the existing needs in the Emergency Department that facilitates processes like calling patients, providing relevant information to the patients and displaying waiting times.



## Titulo

Gestão de filas de espera no Serviço de Urgência

## Resumo

A constante evolução das tecnologias da informação permite que a sua aplicação na área da saúde seja cada vez mais vasta. A sua utilização permite a obtenção de cuidados de saúde com qualidade, económicos, efetivos e racionais, bem como uma melhor organização do sistema de cuidados de saúde a qualquer nível.

Com um papel cada vez mais importante no Sistema Nacional de Saúde surgem os Serviços de Urgência, nestes, os tempos de espera são um factor importante na satisfação dos pacientes. Um serviço de urgência em que os pacientes são atendidos rapidamente é considerado um bom serviço de urgência. A apresentação dos tempos de espera num serviço de urgência faz com que o processo de espera dos pacientes seja melhorado, permitindo ainda ao serviço de urgência monitorizar estes tempos e tomar medidas caso estes não sejam normais e prejudiquem os pacientes. Outro aspecto importante no serviço de urgência é a chamada de pacientes e anunciação de mensagens aos utentes, estas por vezes não são perfeitamente perceptíveis. Isto pode ocorrer por exemplo, pelo estado da voz da pessoa que efetua a chamada ou por a sala onde é efectuada a chamada apresentar muito barulho. De modo a resolver esta situação pode utilizar-se voz sintetizada por computador descartando assim essas variáveis, uniformizando então os processos de chamada de pacientes e leitura de mensagens.

O objetivo principal desta dissertação compreende o desenvolvimento de uma aplicação Desktop funcional que permita apresentar informação sobre tempos de espera aos pacientes no serviço de urgência, efetue a chamada dos pacientes e anuncie mensagens utilizando síntese de voz computadorizada.

Nesta dissertação recorreu-se a tecnologia *Windows Forms Presentation* (WPF) para desenvolver uma aplicação Desktop que permitisse apresentar tempos de espera médios e o tempo de espera que o último paciente esperou, ambos são obtidos em relação ao tempo necessário para um paciente ser triado e para as 5 categorias diferentes do Sistema de Triagem de Manchester. Esta aplicação desenvolvida permite ainda apresentar vídeos e mensagens escritas, realizar a chamada dos pacientes e realizar leitura de mensagens. Na chamada de pacientes e leitura de mensagens é utilizado um Sistema *Text-to-Speech* ou TTS de modo a poder transformar o texto numa mensagem sonora.

Foi ainda desenvolvido utilizando tecnologia ASP.NET uma aplicação Web que permite gerir funcionalidades da aplicação Desktop. Esta aplicação permite a gestão de vídeos e a inserção, alteração e eliminação de mensagens escritas e mensagens destinadas a ser lidas.

Em conclusão, foi desenvolvida uma aplicação que corresponde às necessidades existentes num serviço de urgência, permitindo facilitar processos de chamada de pacientes, fornecer informações relevantes aos pacientes e apresentar aos pacientes tempos de espera.

## Índice

Agradecimentos .....	i
Title .....	ii
Abstract .....	ii
Titulo.....	iv
Resumo.....	iv
Índice .....	v
Notação e Terminologia .....	viii
Notação .....	viii
Abreviações.....	viii
Lista de Figuras .....	x
Lista de Tabelas .....	xi
Lista de Códigos.....	xii
Capítulo 1 .....	1
1. Introdução .....	1
1.1. Problema e Motivação.....	2
1.2. Objectivos.....	3
1.3. Estrutura .....	3
Capítulo 2 .....	5
2. Enquadramento .....	5
2.1. Serviço de Urgência .....	5
2.1.1. Infraestrutura Informática.....	6
2.2. Triagem.....	7
2.2.1. Sistema de Triagem de Manchester.....	7
2.3. Tempo de Espera .....	9
Capítulo 3 .....	12
3. Tecnologia .....	12
3.1. .NET <i>framework</i> .....	12
3.1.1. VB, C# e as Linguagens .NET.....	13

3.1.2.	Linguagem Intermédia .....	13
3.1.3.	<i>Common Language Runtime</i> .....	14
3.1.4.	Biblioteca de Classes .....	15
3.1.5.	<i>ASP.NET</i> .....	16
3.1.6.	<i>WPF (Windows Presentation Foundation)</i> .....	18
3.1.7.	ADO.NET .....	20
3.1.8.	Visual Studio .....	21
<b>3.2.</b>	<b>Outras Linguagens utilizadas .....</b>	<b>22</b>
<b>3.3.</b>	<b>Base de Dados .....</b>	<b>25</b>
3.3.1.	Oracle .....	26
<b>3.4.</b>	<b>Síntese de Voz .....</b>	<b>26</b>
<b>Capítulo 4</b> .....	<b>29</b>	
<b>4.</b>	<b>Trabalho Elaborado .....</b>	<b>29</b>
<b>4.1.</b>	<b>Aplicação Desenvolvida .....</b>	<b>29</b>
4.1.1.	Funcionalidades da Aplicação desenvolvida .....	29
4.1.2.	Tabelas Criadas e Modelo Relacional .....	30
4.1.3.	Arquitetura Da Aplicação .....	33
4.1.4.	Interface .....	34
4.1.5.	Funcionalidades da Aplicação .....	36
<b>4.2.</b>	<b>Aplicação de Configuração .....</b>	<b>46</b>
4.2.1.	Interface .....	46
4.2.2.	Arquitetura .....	47
4.2.3.	Gestão de Vídeos .....	48
4.2.4.	Inserir Novas Mensagens De Rodapé .....	49
4.2.5.	Inserir Novas Mensagens De Voz .....	50
4.2.6.	Apagar Mensagens De Rodapé e De Voz .....	52
<b>Capítulo 5</b> .....	<b>54</b>	
<b>5.</b>	<b>Conclusão .....</b>	<b>54</b>
5.1.	Trabalho Futuro .....	55
<b>CAPÍTULO 6</b> .....	<b>56</b>	
<b>6.</b>	<b>Bibliografia .....</b>	<b>56</b>
<b>Anexos</b> .....	<b>59</b>	

Anexo A – Código SQL para a criação das tabelas ED_MENSAGEM_VOZ, ED_MENSAGEM_RODAPE, ED_NOMES e ED_APELIDOS .....	59
--	----

## Notação e Terminologia

### Notação

Ao longo do documento é adoptada a seguinte convenção:

- **Texto em itálico** - utilizado em palavras em língua estrangeira e para destacar um determinado termo ou expressão. Também foi utilizado sempre que se faz referência a uma coluna de uma tabela.
- **Texto em negrito** - é empregado sempre que se pretende realçar um termo ou palavra no meio de um parágrafo. Foi ainda utilizado como forma de destacar nomes de tabelas no meio de um parágrafo.

Utiliza-se o tipo de letra **CONSOLAS** para representar excertos de código. É ainda importante referir que alguns termos não foram traduzidos.

### Abreviações

**ADO.NET** - ActiveX Data Object for .NET

**ASP.NET** - ASP – Active Server Pages for .NET

**ATS** - Australasian Triage Scale

**CIL** - Common Intermediate Language

**CLR** - Common Language Runtime

**CSS** - Cascading Style Sheets

**CTAS** - Canadian Triage and Acuity Scale

**DLL** - Data Definition Language

**ESI** - Emergency Severity Index

**GPT** – Grupo Português de Triagem

**HTML** - HyperText Markup Language

**IDE** - Integrated Development Environment

**IM** – Informática Médica

**ISS** - Internet Information Services

**MTS** - Manchester Triage System

**SGBD** - Sistema de Gestão de Base de Dados

**SNS** – Serviço Nacional de Saúde

**SONHO** – Sistema Integrado de Informática Médica

**SQL** - Structured Query Language

**SUTE** - Serviço de Urgência e Tempos de Espera

**TI** – Tecnologias da Informação

**TTS** – Text-To-Speech

**VB** – Visual Basic

**VS** – Visual Studio

**XAML** - eXtensible Application Markup Language

**XHTML** - eXtensible HTML

**XML** - Extensible Markup Language

**WF** - Windows Forms

**WMP** - Windows Media Player

**WPF** - Windows Presentation Foundation

**WWW** - World Wide Web

Esta dissertação foi redigida ao abrigo do novo acordo ortográfico.

## Lista de Figuras

Figura 1 - Fluxograma para a queixa principal "dor torácica" do STM. Fonte [10] .....	9
Figura 2 - Compilação de Linguagem em .NET. imagem adaptada de [14] .....	15
Figura 3 - Os 3 principais aspectos de uma base de dados com modelo relacional.....	25
Figura 4 - Visão geral de um sistema TSS típico. (adaptado de [27]) .....	27
Figura 5 - Modelo Relacional das tabelas utilizadas no desenvolvimento da aplicação. ....	31
Figura 6 - Estrutura das Tabela Criadas (ED_MENSAGEM_VOZ,ED_MENSAGEM_RODAPE, ED_APELIDOS e ED_NOMES). .....	32
Figura 7 - Arquitetura utilizada no desenvolvimento da aplicação SUTE. ....	33
Figura 8 - Esquema representativo da interface principal. ....	34
Figura 9 - Esquema representativo da interface de chamada.....	35
Figura 10 - Esquema representativo da Interface de Mensagem de Voz.....	36
Figura 11 - Interface Principal da aplicação SUTE em funcionamento.....	37
Figura 12 - Mensagens apresentadas na aplicação em diferentes tempos. ....	41
Figura 13 - Exemplo de uma chamada de um paciente, apresentando a interface de chamada. ..	43
Figura 14 - Exemplo da apresentação de mensagem personalizável, apresentando a interface de leitura de mensagem de voz.....	44
Figura 15 - Página Inicial de Configuração da Aplicação.....	47
Figura 16 – Parte comum em todas as páginas da aplicação de configuração.....	47
Figura 17 - Arquitetura de 3 camadas utilizada na Aplicação Web .....	48
Figura 18 - Página que permite ver os vídeos que são reproduzidos na aplicação.....	49
Figura 19 - Página que permite inserir novas mensagens de rodapé na base de dados .....	50
Figura 20 - Mensagens de erro obtidas pela introdução de informação incorreta. ....	50
Figura 21 - Página que permite inserir novas mensagens de voz na base de dados.....	51
Figura 22 - Mensagens de erro obtidas pelo preenchimento incorreto dos campos.....	52
Figura 23 - Página que permite apagar mensagens de rodapé e mensagens de voz .....	53
Figura 24 - Mensagem de confirmação de eliminação de mensagem .....	53

## Lista de Tabelas

Tabela 1 - Os 10 serviços mais visados nas reclamações no SNS. (Retirado de [3]) .....	5
Tabela 2 - Classificação dos tipos de urgência do STM. ....	8
Tabela 3 - As 10 causas mais mencionadas nas reclamações no SNS. (retirado de [3]) .....	10
Tabela 4 - Algumas classes e <i>namespaces</i> importantes em .NET.....	16
Tabela 5 – Exemplo de conversão de tempos de espera realizado pela função <i>formathoras</i> . .....	39
Tabela 6 - Conteúdo da tabela ED_MENSAGEM_RODAPE contendo 2 mensagens.....	39
Tabela 7 - Conteúdo da tabela ED_MENSAGEM_VOZ. ....	45



## Lista de Códigos

Código 1 – Porção de código em VB 6. ....	13
Código 2 – Porção de código em VB.NET. ....	13
Código 3 - Declaração de um elemento label com o nome Label1 e conteúdo “ <i>Hello World</i> ”. ....	19
Código 4 – Código em VB.NET que altera o conteúdo da Label1 para “Label Alterada”. ....	19
Código 5 - Etiqueta h1 com o texto "Olá Mundo". ....	22
Código 6 - Exemplo de Etiqueta imagem.....	22
Código 7 - Parte de um ficheiro CSS contendo uma regra. ....	23
Código 8 - Exemplo de uma função JavaScript. (w3schools) ....	24
Código 9 – Exemplo da utilização de uma função JavaScript associada ao evento de clicar no botão. (w3schools) ....	24
Código 10 – Parâmetros da função <i>formathoras</i> . ....	39
Código 11 - Parâmetros da função <i>addmsg</i> . ....	40
Código 12 – Parâmetros da função <i>Bannernews</i> . ....	41
Código 13 – Declaração da função <i>getlistvideo</i> . ....	42
Código 14 – Declaração da função <i>video_MediaEnded</i> . ....	42
Código 15 – Parâmetros das funções <i>GetCurrentTime</i> e <i>GetCurrentDate</i> . ....	42
Código 16 - Parâmetros da função <i>getnome</i> . ....	43
Código 17 – Parâmetros da função <i>getapelido</i> . ....	43
Código 18 - Parâmetros da função <i>ChamarVoz</i> . ....	44
Código 19 – Parâmetros da função <i>LerMensagemVoz</i> ....	46

# Capítulo 1

---

## 1. Introdução

Num mundo cada vez mais informatizado e globalizado a Informática Médica (IM) tem vindo a ganhar cada vez mais popularidade e atenção, destacando-se como área prometedora de aplicação das Tecnologias da Informação (TI). A IM é uma junção entre disciplinas científicas teóricas e aplicacionais que utilizam os seus conhecimentos base e TI modernas. A utilização das TI permite proporcionar cuidados de saúde com qualidade, económicos, efetivos e racionais, bem como uma melhor organização do sistema de cuidados de saúde a qualquer nível. Pode-se dizer que a IM é uma disciplina que reconcilia e conecta o mundo das tecnologias computacionais e a abordagem humana. A rápida e constante evolução que ocorre na medicina e das TI proporciona à IM um vasto campo de ação. [1]

O serviço de urgência existe de modo a proporcionar aos pacientes um atendimento rápido em situações de potencial risco para a saúde. Uma urgência é qualquer situação cuja demora de diagnóstico ou de tratamento apresente grave risco ou prejuízo para a vítima, como por exemplo traumatismos graves, intoxicações agudas, queimaduras, crises cardíacas ou respiratórias. Os pacientes em situações graves necessitam de atendimento mais rápido em relação aos menos graves, visto que uma espera prolongada pode comprometer o seu estado de saúde. É então necessário que exista um método que permita classificar a prioridade clínica dos pacientes, por exemplo o Sistema de Triage de Manchester. [2]

Após efetuar o registo de entrada no serviço de urgência o paciente é encaminhado para um gabinete de triagem, onde é submetido a uma observação prévia, com identificação de um conjunto de sintomas ou sinais que permitem atribuir um grau de prioridade clínica no atendimento e um tempo de espera recomendado até à primeira observação médica.

Num serviço de urgência um paciente encontra-se sujeito a tempos de espera, sendo estes constantemente motivos de reclamações devido às esperas prolongadas. Um serviço de urgência que forneça informação sobre os possíveis tempos de espera e tenha em conta o bem estar dos pacientes vai estar dando um passo em frente na melhoria dos serviços que fornece.

A qualidade e a forma como os serviços são prestados num serviço de urgência pode ser melhorada recorrendo à utilização dos novos avanços tecnológicos. Esta melhoria pode envolver o modo como a informação clínica é armazenada, tratada, alcançada e utilizada. É com o objectivo de obter uma melhoria no modo como o Serviço de Urgência presta os serviços que esta dissertação foi realizada.

## **1.1. Problema e Motivação**

A evolução das novas tecnologias permite a melhoria da prestação dos cuidados de saúde, permitindo, por exemplo, armazenar e trocar informação clínica de modo organizado e estruturado.

Quando um paciente recorre ao serviço de urgência tem dificuldade em saber quando vai ser atendido e quantas pessoas estão à sua frente, por isso seria de grande utilidade conseguir fornecer aos pacientes este tipo de informação dinâmica. Informações sobre os tempos de espera e o número de pessoas que não receberam ainda cuidados médicos são importantes. Imaginemos que um paciente é informado que tem de esperar 2 a 3 horas até poder ser atendido, este paciente neste tempo pode então organizar melhor a sua vida pois tem uma referência sobre o tempo de espera a que vai estar sujeito. Como referido no capítulo 2.3 o controlo e apresentação dos tempos de espera possuam aspectos positivos tanto para os pacientes como para o serviço de urgência.

Outro factor a considerar num serviço de urgência é a chamada de pacientes realizada pelos profissionais de saúde de modo a informar os pacientes para que sala se devem dirigir. Esta chamada ocorre numa primeira instância para informar ao paciente que se deve dirigir para a sala de triagem e volta a ocorrer quando depois de triado o paciente é chamado para uma sala de tratamento específica.

A chamada do paciente é um factor importante no serviço de urgência e pode apresentar problemas de percepção por parte dos pacientes. Por exemplo, em situações em que o médico se encontra com limitações vocais (rouquidão, afonia, entre outros) ou a sua pronúncia não é muito correta a chamada do paciente pode ser pouco perceptível para os pacientes. Este tipo de situações podem ser superadas mostrando visualmente a chamada, ou seja o nome do paciente e a sala para a qual este se deve dirigir. Acompanhado da apresentação visual da chamada é utilizada uma voz sintetizada por computador que realiza a chamada automaticamente e sempre do mesmo modo.

Outra necessidade que por vezes surge é a apresentação ou anúncia de mensagens aos pacientes de modo a notificá-los sobre algo importante. As mensagens a ser anunciadas podem ser realizadas do mesmo modo que a chamada dos pacientes, ou seja, utilizando voz sintetizada por computador.

É neste panorama global que se justifica o desenvolvimento de uma aplicação que permita apresentar tempos de espera e realize chamadas e anuncie mensagens num serviço de urgência. A aplicação permite ainda mostrar mensagens escritas e vídeos

## 1.2. Objectivos

O objectivo principal deste trabalho consiste no desenvolvimento de uma aplicação Desktop funcional que forneça informação sobre tempos de espera aos pacientes do serviço de urgência, efetue a chamada dos pacientes e anuncie mensagens utilizando síntese de voz computadorizada.

Compreendidos neste objectivo principal estão:

- Obtenção dos tempos de espera médios e tempo de espera do último paciente atendido para cada uma das 5 categorias diferentes da triagem de Manchester.
- Obtenção dos tempo de espera médio e tempo de espera do último paciente na triagem.
- Criação de uma interface simples e apelativa.
- Adicionar funcionalidades como apresentação de vídeos e mensagens de rodapé.
- Obtenção de uma voz sintetizada por computador que permita realizar a chamada e leitura de mensagens de um modo perceptível.
- Criação de uma aplicação Web que permita gerir funcionalidades da aplicação Desktop. Realizar a gestão de vídeos, mensagens de rodapé e mensagens destinadas para leitura são as funcionalidades que a aplicação Web criada deve englobar.

## 1.3. Estrutura

O facto da presente dissertação envolver um carácter mais prático e voltado para o desenvolvimento de uma aplicação faz com que a estrutura deste documento seja afectada. Consequentemente a parte do documento que envolve levantamento bibliográfico, especificamente os capítulos 2 e 3, não foram alvos de um estudo muito exaustivo e como tal o seu peso no documento final não é significativo. A presente dissertação foi estruturada em cinco capítulos, organizados da seguinte forma:

## **1. Introdução**

Neste capítulo é feita uma rápida descrição da situação do trabalho, uma introdução aos principais conceitos que posteriormente são aprofundados e é apresentada a motivação e objectivos do presente trabalho. É também apresentada uma breve descrição da estrutura da dissertação.

## **2. Enquadramento**

Neste capítulo é fornecida uma visão geral sobre alguns aspectos relevantes no serviço de urgência. São abordados assuntos como a importância do serviço de urgência no Sistema Nacional de Saúde e sua infraestrutura informática. É feita uma breve análise da triagem dos pacientes no serviço de urgência, tendo em especial atenção o Sistema de Triagem de Manchester. É abordada ainda a problemática dos tempos de espera no serviço de urgência.

## **3. Tecnologia**

Neste capítulo é apresentado um conjunto de tecnologias utilizadas/analizadas no desenvolvimento desta dissertação. A grande parte deste capítulo aborda o *framework .NET*, que possui um conjunto de tecnologias necessárias nas quais o projeto desenvolvido nesta dissertação assenta. É dada também importância às bases de dados pois estas desempenham um papel importante no modelo funcional das aplicações. Por último realiza-se uma breve abordagem sobre a tecnologia de síntese de voz que desempenha um papel importante na aplicação desenvolvida.

## **4. Trabalho Elaborado**

O quarto capítulo explora a implementação deste projeto. Este capítulo é dividido em dois subcapítulos. O primeiro aborda o desenvolvimento da aplicação desktop desenvolvida (designada como SUTE, uma forma de abreviar futuras referências) e nele são explicadas as funcionalidades da aplicação, arquitetura e interface. No segundo subcapítulo é descrita em pormenor a página que permite gerir algumas funcionalidades personalizáveis da aplicação SUTE. Para uma melhor compreensão, as explicações são normalmente acompanhadas por imagens da interface desenvolvida, tabelas e excertos de código.

## **5. Conclusão**

O último capítulo desta dissertação resume o trabalho realizado e são identificadas as principais conclusões a retirar. São ainda referidas possíveis melhorias no trabalho desenvolvido.

# Capítulo 2

---

## 2. Enquadramento

### 2.1. Serviço de Urgência

Os serviços de urgência são cada vez mais o primeiro nível de acesso aos cuidados de saúde, isto faz com que representem um papel importante no Serviço Nacional de Saúde (SNS) e sejam motivo de preocupação constante. Os serviços de urgência são grandes consumidores de recursos humanos e financeiros, condicionando, em muitos hospitais, o funcionamento regular de todos os outros serviços de ação médica, do ambulatório e dos meios complementares de diagnóstico e terapêutica. [2]

Segundo o Relatório do Sistema SIM-Cidadão (Tabela 1) o serviço de urgência Geral foi o serviço que apresentou mais reclamações no ano 2010, com um total de 13904 o que representa quase três vezes mais reclamações do que a Medicina Geral e Familiar que se encontra em segundo lugar com 5609 reclamações. [3]

**Tabela 1 - Os 10 serviços mais visados nas reclamações no SNS. (Retirado de [3])**

Serviços	Nº de reclamações em 2009	Nº de reclamações em 2010
Urgência Geral	16.007	13.904
Medicina Geral e Familiar	6772	5609
Urgência – SAP	2020	2087
Consulta Externa	2055	1850
Coordenação	529	1354
Saúde do Adulto	88	1132
Admissão de Doentes	1635	1018
Cuidados de Enfermagem	1060	875
Gestão de Enfermagem	1006	852
Secretaria	744	794
Total	31916	29.475

De modo a compreender melhor a lógica de funcionamento dos serviços de urgência é necessário esclarecer o que são situações urgentes e emergências. Situações urgentes são todas as situações clínicas de instalação súbita, desde as não graves até às graves, com risco de estabelecimento de falência de funções vitais. Situações de emergência são todas as situações clínicas de estabelecimento súbito, em que existe o compromisso de uma ou mais funções vitais. [4]

Os serviços de urgência todos os dias são confrontados com a chegada de pacientes, apresentando estes uma série de vários problemas. O impacto que estes novos pacientes provocam no funcionamento do serviço depende do número de pacientes por atender e do estado clínico dos mesmos. É necessário então um sistema de triagem que permita o atendimento dos utentes de forma justa, uniforme e que este se realize por ordem de necessidade clínica e não por ordem de chegada ao serviço.

Quando um utente chega a um serviço de urgência tem de realizar a admissão, ou seja a inscrição, a qual pode implicar o fornecimento de alguns dados pessoais. Após a admissão o estado do paciente é inicialmente avaliado por um profissional de enfermagem de modo a ser triado.

### **2.1.1. Infraestrutura Informática**

O Sistema Integrado de Informação Hospitalar (SONHO) é uma base estrutural do sistema de informação da saúde em Portugal. O SONHO desempenha um papel essencial ao nível administrativo no registo de informação de gestão de doentes. Possui um módulo urgência, sendo este o único que permite a existência de dados de identificação de doentes que, tendo solicitado o serviço no hospital, não sejam ainda doentes do mesmo (isto é, cujos registos não constem no módulo Identificação). [5]

O módulo de Identificação armazena dados de identificação de doentes e um conjunto de dados clínicos mínimos. Os dados de identificação do paciente devem ser armazenados apenas no módulo de identificação de modo a evitar problemas de incoerência e de redundância, obtendo-se assim um registo centralizado de doentes. As informações presentes neste módulo podem ser acedidos e/ou inseridos a partir de outros módulos, por exemplo o módulo urgência.

Os objectivos do SONHO no serviço de urgência são: [6]

- Registrar todos os pedidos no serviço de urgência, bem como um conjunto de dados mínimos acerca desses episódios.
- Admitir o registo de pacientes que se dirigem ao serviço de urgência, cujos dados não se encontrem presentes no módulo Identificação. Isto evita a sobrecarga do sistema com doentes que só vão esporadicamente ao hospital.

- Permitir a ligação com o modulo Identificação, a fim de transferir ou aceder aos dados de identificação do doente.

No entanto o SONHO encontra-se seriamente comprometido devido à sua infraestrutura desatualizada, porque depende atualmente de uma base de dados Oracle e versões de formulários Oracle desatualizados. Muitos hospitais que utilizam o SONHO procuram alternativas, no entanto são enfrentados problemas de migração. [7]

## 2.2. Triagem

Triagem significa escolher, selecionar, qualificar. De um modo geral, é um processo no qual o paciente vai ser avaliado e observado de modo a lhe ser atribuído um grau de prioridade de atendimento. A triagem é realizada por enfermeiros formados e treinados para tal função. [4] [8]

Um sistema/escala de triagem é uma forma organizada e sistematizada de realizar a triagem. Os sistemas de triagem têm um objetivo principal comum que é ajudar o enfermeiro a identificar rapidamente os utentes em situação de emergência.

Os sistemas de triagem mais utilizados a nível mundial atualmente na avaliação do estado clínico dos pacientes no serviço de Urgência são : *Manchester Triage System* (MTS) , *Canadian Triage and Acuity Scale* (CTAS) , *Australasian Triage Scale* (ATS) e *Emergency Severity Index* (ESI). Estes sistemas de triagem têm em comum a utilização de cinco níveis de prioridade. Segundo o estabelecimento de cinco categorias de urgência apresenta uma melhor discriminação dos utentes que apresentam lesões ou doenças mais graves. Sendo assim uma versão mais válida, justa e sensível, que por exemplo a versão ESI com 3 níveis de urgência. [9]

### 2.2.1. Sistema de Triagem de Manchester

O Sistema de Triagem de Manchester (STM) foi criado e implementado em 1997 na cidade de Manchester (Reino Unido) pelo Grupo de Triagem de Manchester. Este sistema foi implementado em países como Irlanda, Portugal, Espanha, Itália, Suécia, Holanda, Alemanha, Canadá, Japão. [2]

Em Portugal foi implementado no ano 2000 no Hospital Geral de Santo António e o Hospital Fernando Fonseca. Em 2001 foi criado o Grupo Português de Triagem (GPT) uma entidade reconhecida pelo Ministério da Saúde e pelo Grupo de Manchester, que inventiva a implementação do sistema em Portugal e representa os interesses dos autores do sistema.



O objetivo do STM não é estabelecer diagnósticos mas sim identificar factores de gravidade que indiquem a prioridade clínica do paciente de forma objectiva e sistematizada.

### 2.2.1.1. Metodologia da Triagem de Manchester

O método do STM começa pela identificação da queixa principal do utente/paciente seguindo-se a escolha do respectivo fluxograma de decisão. Existem 54 fluxogramas que abrangem todas as situações previsíveis. Na Figura 1 é apresentado o fluxograma para queixa principal “dor torácica”. O fluxograma contém várias questões que devem ser colocadas ao doente. Estas são denominadas de “discriminadores”. As questões têm de ser realizadas seguindo uma ordem, de cima para baixo, uma vez que as questões estão organizadas de modo decrescente quanto ao nível de prioridade.

Os discriminadores podem ser específicos ou gerais. Os primeiros aplicam-se a pequenos grupos ou a casos individuais e tendem a familiarizar-se com características manifestadas pelos utentes (por exemplo, oftalmologia e dor pré-cordial). Os discriminadores gerais aplicam-se a todos os utentes, independentemente da sua doença. Apresentam-se várias vezes ao longo dos fluxogramas (por exemplo dor, hemorragia e temperatura). [10]

O enfermeiro utilizando este sistema consegue classificar o utente em 5 categorias identificadas por um número, nome, cor e tempo máximo de espera até à observação inicial do médico. Na Tabela 2 é possível observar as 5 categorias do STM: Emergente com a cor vermelha, Muito Urgente com a cor laranja, Urgente com a cor amarela, Pouco Urgente com a cor verde e Não Urgente com a cor azul.

Tabela 2 - Classificação dos tipos de urgência do STM.

Nº	Urgência	Cor	Tempo Máximo de espera
1	Emergente	Vermelho	0 minutos
2	Muito Urgente	Laranja	10 minutos
3	Urgente	Amarelo	60 minutos
4	Pouco Urgente	Verde	120 minutos
5	Não Urgente	Azul	240 minutos

Caso o estado clínico do paciente se agrave durante o tempo de espera este deve ser triado novamente para uma prioridade superior à anterior. [2]

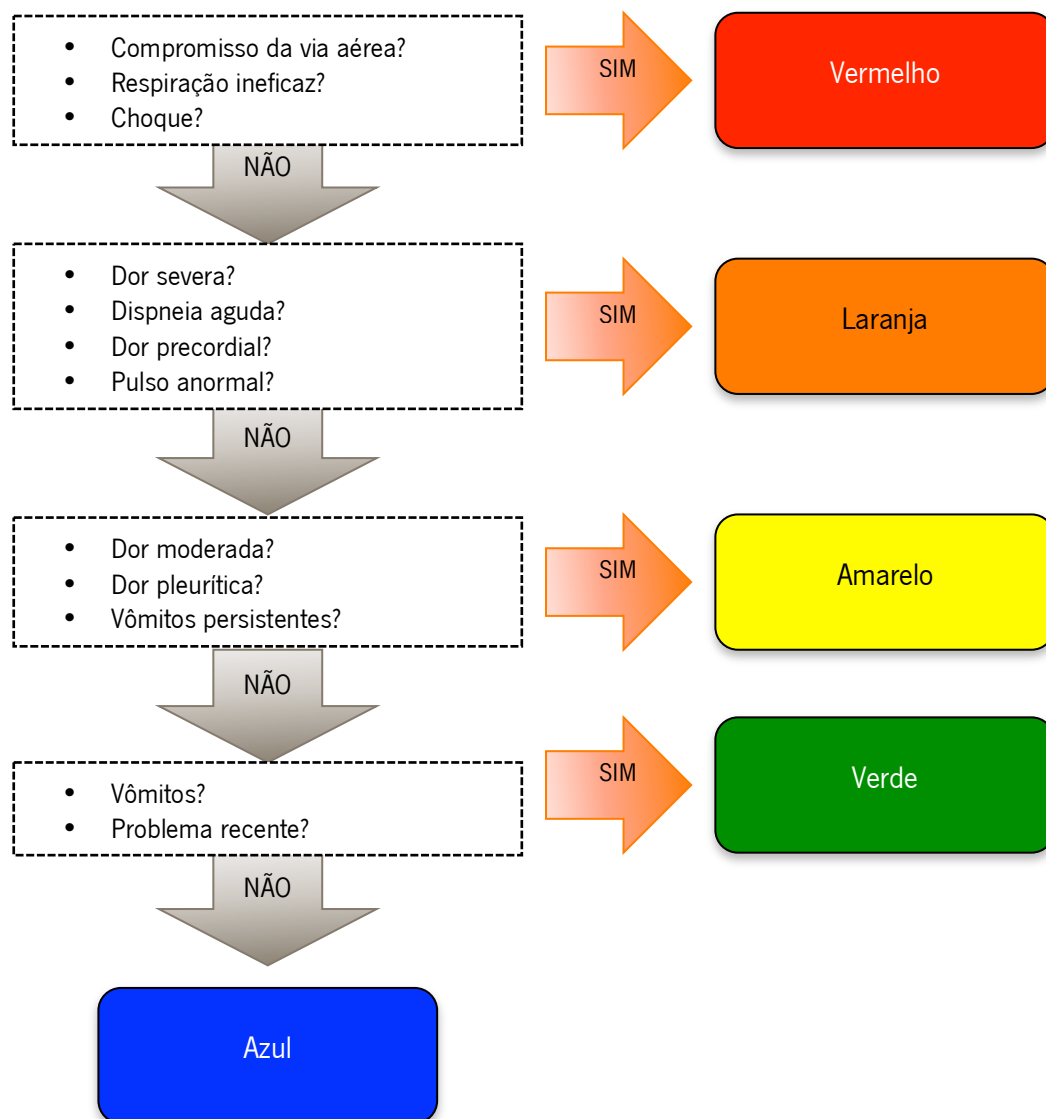


Figura 1 - Fluxograma para a queixa principal "dor torácica" do STM. Fonte [10]

### 2.3. Tempo de Espera

A espera por um serviço ocorre sempre que o número de pessoas que pretendem um serviço excede o número de pessoas que este consegue atender de imediato. Organizações prestadoras de serviços lidam constantemente com filas de espera portanto, esta é uma fonte de preocupação importante na qualidade do serviço que prestam. Os Serviços de Saúde têm de prestar especial atenção às filas de espera, visto que lidar com filas de espera grandes faz parte do seu dia-a-dia. [2]

Ninguém gosta de esperar por um serviço, esta espera nos Serviços de Urgência torna-se especialmente importante pelo facto de este em muitos casos ser um serviço público e como tal os utentes sentem-se no direito de reclamar sempre que consideram a espera exagerada ou injusta.

Como se pode observar na Tabela 3 o tempo de espera no serviço de urgência é a causa mais mencionada nas reclamações no SNS.

**Tabela 3 - As 10 causas mais mencionadas nas reclamações no SNS. (retirado de [3])**

Causas	Nº de reclamações 2009	Nº de reclamações 2010
Tempo de espera no serviço de urgência	7388	6828
Tempo de espera para atendimento	5745	4627
Falta de cortesia	3095	3362
Recusa de consulta	2015	2286
Dificuldade de marcação de consulta	2253	2053
Regras inadequadas/inaplicáveis	1921	1818
Má prática	1836	1805
Perfil desadequado	1600	1513
Desrespeito no trato interpessoal	1610	1397
Falta de Informação aos utentes	1375	1332
Total	28973	27021

A espera provoca nos utentes um estado de ansiedade, tensão e aborrecimento, e enquanto esperam podem estar sujeitos a contágios, pois nas salas de espera encontram-se outros utentes que podem conter doenças contagiosas. Não é nada benéfico para o Serviço de Urgência que um utente que tenha ido para ser tratado por uma determinada complicação acabe por contagiar outros utentes ou ele próprio seja contagiado.

As causas de espera no Serviço de Urgência devido à sua importância foram alvo de vários estudos, no entanto a variedade de respostas indica que as causas podem variar entre hospitais e podem depender de vários factores.

Algumas causas de espera publicadas são: [11]

- Níveis opcionais de tratamento na comunidade não se encontram disponíveis;
- Atrasos nos diagnósticos;
- Acesso aos diagnósticos;
- Indisponibilidade de camas;
- Inexperiência do staff médico;
- Características dos enfermeiros e médicos;
- Número de admissões;
- Sobrecarga de casos não urgentes;
- Falta de protocolos.

Os custos em espera de clientes e instalações desocupadas podem ser reduzidos através da racionalização dos métodos de trabalho, programação dos funcionários envolvidos nos serviços, atendimento rápido da demanda espontânea e antecipação da demanda programada. Uma gestão de filas eficiente tem como um dos seus objectivos principais a redução do congestionamento, esta pode ser obtida por exemplo através da utilização de um sistema de triagem efetivo, funcionários treinados, uso apropriado das TI, aumento da velocidade dos serviços, deslocamento de funcionários em horas de ponta, etc. Um utente que seja respeitado e informado sobre o possível tempo de espera associado a um atendimento rápido e com qualidade vai ser um utente satisfeito.

*David H. Maister* oferece algumas perspectivas interessante sobre o tempo de espera: [12]

- Tempo ocupado parece menor do que tempo desocupado;
- A espera durante o processo é menos longa do que a espera antes do início do processo;
- Ansiedade torna a espera mais longa;
- A espera incerta é mais longa do que a espera prevista;
- Espera injusta é mais longa do que a espera justificada;
- Quanto mais valioso o serviço, mais tempo o cliente tem de esperar;
- A espera em solitário parece mais longa do que a espera em companhia.

Esta dissertação tem em conta a problemática associada aos tempos de espera. Nesta medida a apresentação dos tempos de espera torna-se benéfica tanto para os utentes como para o Serviço de Urgência. Um utente tem o direito de ser informado e essa informação pode ajudar no seu processo de espera, pois como referido anteriormente a espera incerta é mais longa do que a espera prevista. Por outro lado um Serviço de Urgência que se preocupe com o tempo de espera a que os utentes estão sujeitos estará dando um passo em frente na melhoria da qualidade dos serviços que presta. Com base nos tempos de espera o Serviço de Urgência pode saber se os pacientes estão a esperar demais pelos serviços, e caso isso aconteça tomar medidas que possam diminuir o tempo de espera.

# Capítulo 3

---

## 3. Tecnologia

### 3.1. .NET *framework*

A plataforma .NET *framework* é um junção de tecnologias, que ajudam os desenvolvedores a produzirem uma variedade de aplicações. Estas tecnologias são: [13] [14]

- **As linguagens .NET** : estas incluem Visual Basic, C#, F# e C++.
- **Common Language Runtime (CLR)** : é uma camada de software que gere código em tempo de execução, proporcionando serviços automáticos como gestão de memória e gestão de *threads* enquanto proporciona tipos estritos de segurança e verifica se as aplicações podem ser executadas sem erros o que oferece segurança e robustez.
- **Biblioteca de Classes** : fornece aos desenvolvedores componentes que podem ser utilizados nas suas aplicações. Estas funcionalidades são normalmente organizadas em conjuntos de tecnologia, tais como ADO.NET (a tecnologia utilizada em aplicações com ligações a base dados) e *Windows Presentation Foundation* (WPF, a tecnologia utilizada para criar aplicações desktop).
- **ASP.NET** : é o motor que alberga as aplicações Web que são criadas pelo .NET e suporta quase todas as características da biblioteca de classes *.NET framework*. ASP.NET também possui um conjunto específico de serviços Web, tais como autenticação segura e armazenamento de dados.
- **Visual Studio (VS)** : Uma ferramenta opcional de desenvolvimento que contém um abundante conjunto de características de produção e *debugging* (processo de encontrar e reduzir erros). O VS inclui a .NET framework completa.

Por vezes a divisão entre estes componentes não é clara. Por exemplo, o termo ASP.NET é por ocasiões utilizado num sentido limitado referindo-se à porção da biblioteca de classes utilizadas no desenvolvimento de páginas Web. Por outro lado, ASP.NET refere-se a todo o tópico de aplicações Web

.NET, o que inclui linguagens .NET e muitas peças fundamentais da biblioteca de classes que não são específicas do desenvolvimento Web. [14]

### 3.1.1. VB, C# e as Linguagens .NET

A linguagem utilizada no desenvolvimento da aplicação foi *Visual Basic.NET* ou simplesmente chamada de VB.NET. É uma linguagem de programação orientada aos objetos, e é implementada na plataforma .NET framework. O VB.NET é um produto diferente do antigo *Visual Basic 6* (VB 6), diferindo deste na maneira de programar e no conceito da orientação aos objetos. Existem 5 versões do VB.NET implementadas, estas são *Visual Basic .NET* (VB 7) , *Visual Basic .NET* (VB 7.1) , *Visual Basic 2005* (VB 8.0) , *Visual Basic 2008* (VB 9.0) e a última versão o *Visual Basic 2010* (VB 10.0). [15] [14]

Ambos os exemplos mostrados em baixo têm como resultado a apresentação de uma caixa de mensagem com o texto “*Hello World*” com um botão OK.

Código 1 – Porção de código em VB 6.

```
Private Sub Command1_Click ( ) MsgBox(“Hello World”) End Sub
```

Código 2 – Porção de código em VB.NET.

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles  
Button1.Click  
MsgBox(“Hello World”) End Sub
```

Os motivos da utilização desta linguagem devem-se aos conhecimentos prévios nesta linguagem e o facto de ser uma linguagem muito intuitiva, optando então por esta em detrimento do C#.

### 3.1.2. Linguagem Intermédia

Todas as linguagens .NET são compiladas em outra linguagem de baixo nível antes de serem executadas. Esta linguagem de baixo nível é a Linguagem Intermédia Comum ou *Common Intermediate Language* (CIL). O CLR utiliza apenas código CIL, isto porque todas as linguagens .NET são baseadas nele, fazendo com que sejam bastante parecidas entre si. Esta é a razão pelo qual VB e C# proporcionam essencialmente as mesmas características e desempenho. Aliás, são tão compatíveis que

uma página escrita em VB pode utilizar um componente escrito em C# da mesma forma que usaria um componente em VB, e vice-versa. [14]

A Figura 2 mostra como as linguagens .NET são compiladas para CIL. O compilador de cada linguagem segue um conjunto de regras (*Common Language Specification*) para compilar os códigos de modo a que componentes escritos numa linguagem .NET consigam ser utilizados por outras. Todos os ficheiros EXE ou DLL que são construídos com linguagens .NET contêm código CIL. Este é o ficheiro que é enviado para outros computadores. No caso de uma aplicação Web, é enviado o código compilado para um servidor Web.

O CLR não tem ideia em que linguagem foi construída a aplicação pois apenas utiliza código CIL. Quando a aplicação é lançada e antes do código ser executado o código CIL é transformado em linguagem de máquina nativa apropriado a plataforma corrente.

### ***3.1.3. Common Language Runtime***

O *Common Language Runtime* (CLR) é o coração que suporta todas as linguagens .NET. Muitas linguagens modernas utilizam *runtimes*. Por exemplo em VB 6, o *runtime* está contido num ficheiro DLL chamado *msvbvm60.dll* e muitas aplicações em C++ estão ligadas a um ficheiro chamado *mscrt40.dll* para ganharem funcionalidade comum. Estes *runtimes* podem fornecer bibliotecas utilizadas pela linguagem ou podem ser responsáveis pela execução do código (como o Java).

O CLR não executa somente código, mas também proporciona todo um conjunto de serviços relacionados como verificação de erros no código, optimização e gestão de objetos.

Algumas das funções do CLR são: [15]

- Compilação e execução do código;
- Verificação e optimização de código;
- Gestão de memória e recolha de lixo;
- Permitir interoperabilidade entre linguagens .NET.

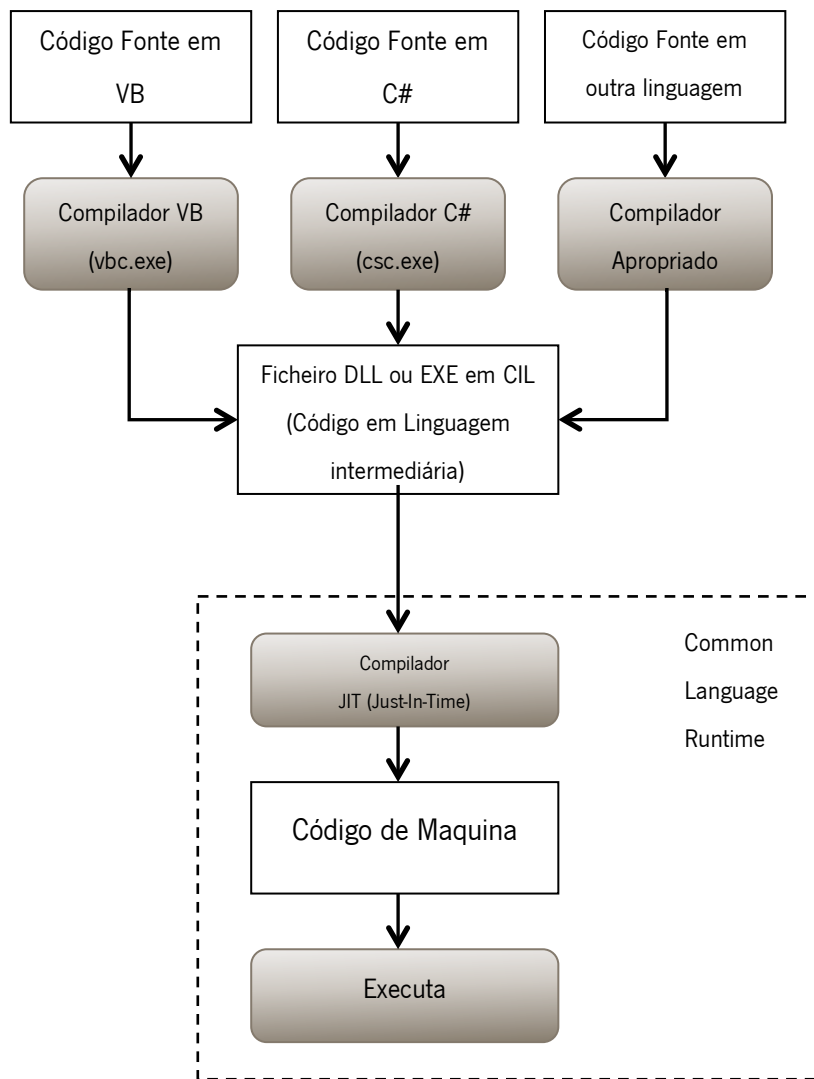


Figura 2 - Compilação de Linguagem em .NET. imagem adaptada de [14]

### 3.1.4. Biblioteca de Classes

A biblioteca de classes .NET é um repositório gigante de classes que fornecem serviços e objetos, facilitando o processo de desenvolvimento e proporcionando funcionalidades predefinidas para tudo, desde a leitura de um ficheiro XML ao envio de um email. A biblioteca é organizada em *namespaces* que representam o tipo de funções que proporcionam. Os *namespaces* são organizados hierarquicamente sendo o **System** a raiz de todos os *namespaces* do .NET framework. Alguns exemplos encontram-se na Tabela 4. [15] [14]



Tabela 4 - Algumas classes e *namespaces* importantes em .NET.

<b>Namespace</b>	<b>Descrição</b>
<b>System</b>	Contém as classes básicas que quase todos os programas utilizam. Exemplo de algumas classes simples pertencentes a este <i>namespace</i> são <i>Object</i> , <i>Char</i> , <i>String</i> , <i>Array</i> e <i>Exception</i>
<b>System.IO</b>	Fornece um conjunto de classes que suportam manipulação IO (Input/output) síncrona e assíncrona para fluxo de dados. Também proporciona classes que permitem manipular ficheiros de sistema, como por exemplo pela criação, gestão e eliminação de ficheiros e pastas.
<b>System.Threading</b>	Inclui um conjunto de classes que suportam programação <i>multithreaded</i> . Alguns dessas classes são <i>Thread</i> , <i>ThreadPool</i> , <i>Mutex</i> e <i>AutoResetEvent</i> .
<b>System.Security</b>	Inclui um conjunto de classes e <i>namespaces</i> descendentes que proporcionam suporte à segurança. Os <i>namespaces</i> descendentes importantes são <i>Cryptography</i> , <i>Permissions</i> , <i>Policy</i> , and <i>Principal</i> .
<b>System.Data</b>	Contém classes necessárias para aceder as bases de dados e manipular os seus dados assim como <i>namespaces</i> adicionais tais como <i>System.Data.OracleClient</i> que permite gerir e aceder aos dados dos provedores Oracle.

### 3.1.5. ASP.NET

ASP.NET é a parte do .NET *framework* que permite desenvolver dinamicamente páginas Web, aplicações Web e serviços Web. Em Janeiro de 2002 foi introduzida a primeira versão, com o .NET *framework* 1.0, passando pelas versões ASP.NET 2.0 e 3.5, atualmente encontra-se na versão 4.0.

No processamento de uma página existem 3 pontos principais que influenciam o modo como a página vai ser apresentada e o seu comportamento: [14]

- **Texto estático.** Como por exemplo HTML, CSS ou JavaScript que é colocado na página e é enviado para o browser diretamente.
- **ASP.NET Server Controls.** São colocados nas páginas aspx e quando são processados emitem HTML que é inserido na página. Estes vão desde os simples *Button* e *Label* aos *TreeView* e *GridView* que são capazes de apresentar o conteúdo de uma fonte de dados (Como uma base de dados ou um ficheiro XML).

- **Código de programação.** Código como VB.NET ou C# pode ser colocado num ficheiro separado, este pode ser executado automaticamente ou pode ser acionado com base numa ação do utilizador. A execução do código pode influenciar o modo como a página é apresentada, isto pela realização de cálculos, acesso a base de dados, controlo de informações que são mostrados na página, etc.

A aplicação Web, necessária para o controlo da aplicação Desktop SUTE, desenvolvida neste projeto podia ser construída utilizando Web Forms ou MVC.

#### 3.1.5.1. Web Forms

São uma parte vital das aplicações ASP.NET e proporcionam o *output* de uma aplicação Web. São páginas Web que os utilizadores de um browser visualizam. Basicamente através do web forms é possível criar uma aplicação web utilizando a mesma interface baseada em controlos que uma aplicação Windows. De modo a correr o ASP.NET web forms, é lido o ficheiro .aspx inteiro, o que gera os objetos correspondentes, e inicia uma série de eventos. O utilizador reage a estes eventos seguindo meticulosamente código orientado a objetos. [16]

#### 3.1.5.2. MVC

*Model-View-Controller* (MVC) oferece uma forma bastante diferente de desenvolver páginas Web quando comparado com o modelo web forms. A ideia chave do modelo MVC consiste em que a aplicação é separada em 3 partes lógicas. [16]

- *O código lógico*, por exemplo lógica de acesso a dados e regras de validação.
- *A visualização*, cria uma representação apropriada do modelo, tipicamente uma interface utilizador. Tipicamente a visualização corresponde ao HTML ou XHTML gerado pela aplicação.
- *O controlador*, coordena todo o conjunto, lidando com interações dos utilizadores, atualizando o modelo e enviando informação para a visualização.

#### 3.1.5.3. Comparação entre Web Forms e MVC

O modelo Web Forms foi o utilizado no desenvolvimento da aplicação Web que permite gerir certas funcionalidades da aplicação Desktop SUTE pelas seguintes razões:

- Apesar das vantagens do modelo MVC, este requer um desenvolvimento mais complexo e estruturado com vista em posteriores desenvolvimentos nas diferentes camadas sem que estes exijam um conhecimento aprofundado do modelo lógico das outras camadas, o que não se justifica neste projeto.
- O modelo Web Forms oferece rapidez de desenvolvimento de uma aplicação, possui um conjunto abundante de controlos web e é de fácil aprendizagem.
- O prévio conhecimento no modelo web forms e o facto de nunca ter trabalhado com MVC também influenciaram a escolha do web forms.

#### 3.1.5.4. ISS

De modo a colocar uma página Web, aplicação Web ou serviços Web em funcionamento é necessário que ocorra a sua publicação num servidor ISS (*Internet Information Services*), que é o servidor Web oficial da Microsoft.

Embora o ISS seja incorporado na maioria das versões Windows, este não se encontra instalado por defeito, necessita ser instalado. As versões *Starter* e *Home Basic* do Windows Vista e Windows 7 só possuem algumas partes do ISS, de modo a processar uma página ASP.NET é necessário pelo menos a edição *Home Premium*. Nas versões Windows Server o ISS é incorporado na sua totalidade. [17]

#### 3.1.6. WPF (*Windows Presentation Foundation*)

Através do WPF é possível criar aplicações com um grau elevado de personalização. O WPF (*Windows Presentation Foundation*) é um componente do *Microsoft .NET framework* 3.0 ou mais. Atualmente a Microsoft lançou 4 versões importantes: WPF 3.0 em Novembro de 2006, WPF 3.5 em Novembro 2007, WPF 3.5 sp1 em Agosto de 2008 e por último em Abril de 2010 o WPF 4.

WPF separa a aparência da interface utilizador do seu comportamento, o que faz com que seja composto por duas partes: um arquivo XML com características especiais denominado XAML (*eXtensible Application Markup Language*, pronuncia-se “zéme”), e um código para .NET, onde é especificado o comportamento que pode ser escrito em qualquer linguagem compatível (VB.NET, C#, etc). O arquivo XAML possui diretrizes que especificam a aparência da interface, podendo ser comparado ao XHTML em relação ao ASP.NET.

Em baixo pode-se ver como é declarada um elemento “*label*” com o texto “*Hello World*”

Código 3 - Declaração de um elemento label com o nome Label1 e conteúdo "Hello World".

```
<Label x:Name="Label1" Content="Hello World">
```

No ficheiro do código VB.NET pode-se alterar os atributos dos elementos, por exemplo o Código 4 altera o conteúdo da *label* que foi criada na interface para "Label alterada".

Código 4 – Código em VB.NET que altera o conteúdo da Label1 para "Label Alterada".

```
Label1.Content = "Label Alterada"
```

O arquivo XAML e a classe do código são ligadas por *databinding*, eventos ou comandos. [18].

### 3.1.6.1. Windows Forms vs Windows Presentation Foundation

Inicialmente, foi considerada a utilização do *Windows Forms (WF)* como ferramenta de desenvolvimento da aplicação SUTE. O WF é um componente da *.NET framework* que permite criar aplicações desktop e é baseado nas classes do *namespace System.Windows.Forms*. No entanto o WPF apresenta mais vantagens sobre o WF. As vantagens do WPF em relação ao WF são: [19]

- **Melhor utilização do hardware gráfico**, o WPF é baseado na mais recente biblioteca DirectX em vez do GDI ou GDI+ , que é utilizado pelo WF. O DirectX é uma biblioteca de ferramentas que proporciona acesso de alta performance a gráficos e hardware multimédia.
- **Propriedades *binding* que proporcionam animação**, WFP utiliza propriedades de dependência, estas suportam características como: valores por defeito, herança de propriedades, *data binding* e notificação de alteração de valores. Esta última possibilita saber quando uma propriedade altera o seu valor e realizar tarefas se necessário.
- **Herança de propriedades**, um controlo WPF herda valores de propriedades dos controlos que o contém. Por exemplo se um botão for colocado dentro de uma *Grid* que tem definidas as propriedades do tipo de letra, então o botão herda essas propriedades.
- **Estilos**, é possível definir um conjunto de propriedades. Por exemplo um estilo pode definir valores como altura, largura e cor para um botão. Posteriormente, pode aplicar-se esse estilo a um botão colocando as propriedades do botão de acordo com o estilo definido.
- **Templates**, enquanto os estilos permitem modificar a aparência do controlo, os *templates* permitem modificar a estrutura e comportamento do controlo.
- **Contenção de controlos consistente**, muitos tipos de controlos do WF somente podem conter um tipo de conteúdo. Por exemplo, em WF um Botão tem a propriedade *Text*

que determina o que o controlo apresenta, enquanto que em WPF um botão pode conter por exemplo, um controlo *StackPanel* e este por sua vez, pode conter uma imagem e uma *label*.

- **Separação da interface utilizador e *Code-Behind***, utilizando XAML para definir a interface utilizador e código C# ou VB para fornecer funções aplicacionais permite separar as tarefas de construção de interface e programação.
- **Novos Controlos**, WPF oferece novos controlos que não se encontram disponíveis no WF.

De acordo com as vantagens apresentadas do WPF sobre o WF, optou-se foi optado pela utilização do WPF. Esta tecnologia além de ser a mais atualizada fornece a possibilidade de construir interfaces atrativas, interativas e esteticamente agradáveis, daí ter-se optado por esta ferramenta de desenvolvimento. Interfaces criadas em WPF podem incluir documentos que rearranjam o seu conteúdo como necessário, gráficos 2D e 3D, gráficos vectoriais de alta resolução que desenharam linhas e curvas em vez de utilizar *bitmaps*, animações, áudio e vídeo.

### 3.1.7. ADO.NET

ADO.NET é a tecnologia que permite a aplicações .NET estabelecer interligação com as bases de dados. ADO.NET baseia a sua funcionalidade num conjunto de classes, as quais podem ser divididas em 2 grupos: os que são utilizados para armazenar e gerir dados (por exemplo, *DataSet*, *DataTable* e *DataRow*) e os que são utilizados para estabelecer a conexão a uma base de dados específica (*Connection*, *Command* e *DataReader*). [15]

O grupo de classes que permite armazenar e gerir dados é genérico a todo o tipo de aplicações desenvolvidas. Independentemente do tipo de base de dados, uma vez que são extraídos os dados, estes podem ser armazenados utilizando a classe *DataSet*. O *DataSet* pode ser entendido como uma camada de dados, desempenhando o mesmo papel que um *array*. A diferença é que o *DataSet* é personalizado para dados relacionais, o que significa que compreende conceitos como linhas, colunas e relações entre tabelas.

As classes responsáveis por estabelecer a ligação não são genéricas, cada conjunto de classes utilizadas para interagir com dados é chamada ADO.NET *data provider*. Os *data providers* são específicos para as suas bases de dados. Por exemplo o *SQL Server data provider* é concebido para trabalhar com *SQL Server* e inclui classes como *SqlConnection* e *SqlCommand* enquanto o *Oracle data provider (ODP.NET)* é utilizado com bases de dados Oracle e inclui classes como *OracleConnection* e *Oracle Command*. [14]

### 3.1.8. Visual Studio

É possível criar uma aplicação ASP.NET escrevendo todo o código fonte num editor de texto e compilando com os compiladores .NET na linha de comandos. No entanto esta tarefa seria árdua, fatigante e propicia a erro. Por isso a ferramenta de desenvolvimento Visual Studio *Integrated Development Environment (IDE)* proporciona um ambiente produtivo onde se podem criar rapidamente aplicações avançadas.

O VS IDE é uma ferramenta de ambiente de desenvolvimento integrado (em inglês an *Integrated Development Environment - IDE* ) utilizada para desenvolver aplicações utilizando o novo .NET *framework* 4.0. O VS IDE inclui editor, compilador, *debugger* e um sistema de execução. É compatível com as *framework* anteriores tal como .NET *framework* 3.5, 3.0 e 2.0. Utilizando .NET *framework* 4.0 é possível por exemplo desenvolver as seguintes aplicações :

- Aplicações de consolas;
- Aplicações *Windows Forms*;
- Serviços Windows;
- Aplicações WPF *Browser*;
- Aplicações WPF;
- Aplicações Web;
- Serviços Web, Serviços Web capacitados com Ajax;
- Aplicações Office;
- Projetos SharePoint, modelos de conectividade de dados de negócio e aplicações de fluxo de trabalho;
- Aplicações *Silverlight*;
- Aplicações Windows Mobile;
- *Add-ins* Visual Studio;

A vantagem em utilizar o VS 2010 IDE é que se pode utilizar o VB 10.0 ou C# e as novas características do IDE. A IDE do Visual Studio foi construída utilizando WPF.

## 3.2. Outras Linguagens utilizadas

### 3.2.1.1. HTML

Desde o início dos anos 90 que tem sido um motor na evolução da *World Wide Web* (WWW), a parte da internet referente às páginas Web. HTML (HyperText Markup Language) é uma linguagem vital para a criação de páginas Web e é compreendido por todos os browser existentes. Os documentos HTML são ficheiros simples que contêm um conjunto de *etiquetas*(tags) e elementos, texto e dados adicionais que influenciam o texto. [20]

HTML utiliza texto entre parêntesis angulares (“<” e “>”) de modo a indicar como o conteúdo deve ser mostrado no browser. O texto dentro dos parêntesis angulares tem o nome de *etiqueta*. Um par de *etiquetas* rodeando algum texto ou conteúdo é denominado elemento.

No exemplo em cima de um elemento com a *etiqueta* de abertura <h1> é a *etiqueta* de encerramento </h1>. O elemento h1 representa um título de primeiro nível, qualquer texto dentro das *etiquetas* <h1> e </h1> é considerado pelo browser como texto pertencente ao elemento e, como tal, parte do título.

**Código 5 - Etiqueta h1 com o texto "Olá Mundo".**

```
<h1>Olá Mundo</h1>
```

Um elemento pode conter atributos, estes contêm informação adicional de como alterar o comportamento do elemento.

Por exemplo, a *etiqueta* <img> é utilizada para mostrar uma imagem e o atributo src indica a localização da imagem.(Código 6)

**Código 6 - Exemplo de Etiqueta imagem.**

```

```

Em adição ao HTML, existe o XHTML. Embora possuam nomes muito semelhantes, existem diferenças entre eles: o XHTML é uma reformulação do HTML em XML – *eXtensible Markup Language*. XML de um modo genérico, é uma linguagem baseada em *etiquetas* e texto, utilizada para descrever dados e como linguagem base para outras linguagens.

XHTML é basicamente HTML reescrito utilizando regras XML. Algumas regras do XHTML são: fechar sempre os elementos, utilizar minúsculas para os nomes dos atributos e etiquetas, rodear os valores dos atributos com aspas. [14]

### 3.2.1.2. CSS

A linguagem *Cascading Style Sheets* (CSS) permite formatar e construir páginas *Web* de um modo mais simples e muito mais eficaz.

O CSS oferece um conjunto de opções que podem modificar o aspeto de uma página *Web*, incluindo: fontes (tamanho, cor, tipo, etc), cores e cores de fundo, margens de elementos HTML, posicionar elementos na página e muito mais. Com a utilização de CSS separam-se os dados da apresentação: o documento HTML contém o que se pretende apresentar enquanto o ficheiro CSS define como é apresentado o que se pretende mostrar. [21]

As páginas ASPX ou HTML podem referenciar a utilização de ficheiros CSS ou o código CSS pode ser colocado diretamente nas páginas. No entanto, a última opção tem a desvantagem de a informação sobre estilos não se encontrar num ficheiro separado.

Utilizando estilos é possível criar classes de elementos, na qual são aplicadas as mesmas regras para os elementos que possuem o atributo *class* com o nome atribuído à regra. Por exemplo, no Código 7 é possível ver uma regra que permite centrar o conteúdo de um elemento, para tal o elemento só tem de possuir o atributo *class* configurado da seguinte forma: *class="imgcentered"*.

Código 7 - Parte de um ficheiro CSS contendo uma regra.

```
.imgcentered{ text-align:center;}
```

### 3.2.1.3. JavaScript

JavaScript é uma linguagem de script orientada a objetos e não uma linguagem de programação. A diferença entre uma linguagem de script e de programação é subtil mas necessita ser entendida. [20]

As linguagens de programação necessitam ser compiladas antes de o programa ser corrido. No entanto, a linguagem de script não necessita que o programador inicie o processo de conversão, ele ocorre automaticamente quando o código é processado pelo browser *Web*.

JavaScript é a linguagem de script mais popular na internet e funciona em todos os principais browsers , tal como *Internet Explorer, Firefox, Chrome, Opera* e *Safari*. [22]

Com a utilização do JavaScript pode-se personalizar o conteúdo das páginas de modo a que esta apresente por exemplo, informações diferentes a diferentes utilizadores.

O JavaScript é constituído por 3 partes: [23]



- ECMAScript padronizada pela Ecma internacional. ECMA-263 é o standard oficial do JavaScript.
- O Módulo do Objeto do Documento, que proporciona métodos e interfaces para trabalhar com o conteúdo da página Web.
- O Módulo do Objecto do Browser, que proporciona métodos e interfaces para interagir com o browser.

O código JavaScript encontra-se rodeado no documento pelas etiquetas `<script>` e `</script>`. O browser interpreta o conteúdo entre essas etiquetas como sendo código JavaScript em vez de texto HTML. No Código 8 é possível observar uma função JavaScript com o nome *displaymessage*. A função tem como objetivo mostrar uma mensagem com o texto *"Hello World!"*.

O JavaScript é frequentemente utilizado de modo a ser associado a eventos. No Código 9 é possível verificar a associação da função *displaymessage* ao clique do botão, a função é executada quando o utilizador clica no botão.

Código 8 - Exemplo de uma função JavaScript. [24]

```
<script type="text/javascript">
  function displaymessage()
  {
    alert("Hello World!");
  }
</script>
```

Código 9 – Exemplo da utilização de uma função JavaScript associada ao evento de clicar no botão. [24]

```
<input type="button" value="Click me!" onclick="displaymessage()" />
```

#### 3.2.1.4. SQL

Quando se interage com uma base de dados através de ADO.NET é utilizado SQL (*Structured Query Language*) para extrair, modificar e atualizar informação. Para desenhar uma aplicação que acede a uma base de dados é necessário compreender os conceitos básicos do SQL. [25]

SQL é uma linguagem standard de acesso a dados utilizada para interagir com base de dados relacionais. Dependendo das bases de dados a linguagem SQL pode apresentar diferentes características, no entanto os comandos comuns principais são:

- **Select**, expressão que permite extrair registos.

- **Update**, expressão que permite alterar registos.
- **Insert**, expressão que permite inserir novos registos.
- **Delete**, expressão que permite eliminar registos existentes.

### 3.3. Base de Dados

Uma base de dados é um coleção estruturada de dados armazenados, que normalmente é organizada de acordo com aspectos relevantes permitindo aceder, gerir e atualizar informação. Para o propósito desta dissertação é assumido que os dados na base de dados são armazenados em formato electrónico. O termo base de dados implica que os dados sejam geridos com algum nível de qualidade, em termos de precisão, disponibilidade, utilidade e consistência, e isto é feito pelo Sistema de Gestão de Base de Dados (SGBD). [17]

O tipo mais popular de base de dados é a base de dados relacional. Além do tipo relacional existe também: *flat-file*, objecto relacional e orientada a objetos. A base de dados utilizada nesta dissertação é Oracle, esta possui um modelo relacional.

O modelo relacional tem três principais aspectos: [26]

#### Estruturas

As estruturas são objetos bem definidos, como tabelas, visões, índices e assim por diante, que armazenam ou acedem aos dados de uma base de dados. As estruturas e os dados contidos dentro delas podem ser manipulados pelas operações.

#### Operações

As operações são ações definidas claramente que permitem aos usuários manipular os dados e as estruturas de uma base de dados, estas devem obedecer a um conjunto predefinido de regras de integridade.

#### Regras de Integridade

As regras de integridade são as leis que governam as operações permitidas nos dados e nas estruturas de uma base de dados. São estas que proporcionam proteção aos dados e estruturas na base de dados.

Figura 3 - Os 3 principais aspectos de uma base de dados com modelo relacional.

De modo a introduzir e retirar dados de uma base de dados é necessário a utilização de *Structured Query Language (SQL)*.

### 3.3.1. Oracle

A companhia Oracle tem a sua sede na Califórnia e foi fundada em 1977. Em 1979 foi o primeiro provedor a oferecer um SGBDR(Sistema de Gestão de Base de Dados Relacional) comercial.

O software Oracle encontra-se disponível para muitas plataformas, desde computadores pessoais até grandes *mainframes* e sistemas de processamento paralelo massivo. Este é um dos pontos fortes do Oracle, pois garante um alto grau de independência dos fornecedores de hardware, bem como vários cenários de crescimento, sem perda de benefícios de investimentos anteriores, e oferece amplas possibilidades de transporte e comunicação em ambientes heterogêneos. [25]

Alguns exemplos de componentes do software Oracle são:

- **SQL Plus e SQL Developer:** são duas ferramentas que permitem executar SQL e aceder à base de dados. SQL Plus é uma ferramenta de linha de comandos enquanto o SQL Developer é uma ferramenta gráfica de desenvolvimento e administração de base de dados.
- **Oracle Developer Suite :** É um conjunto integrado de ferramentas de desenvolvimento, com os seguintes componentes principais : Oracle JDeveloper, Oracle Forms E Oracle Reports.
- **Oracle Enterprise Manager:** É uma interface de utilizador gráfica, que é executada num ambiente browser. Proporciona ajuda aos administradores da base de dados Oracle nas tarefas diárias. Tarefas como iniciar, desligar, backup, recuperação, manutenção e gestão de performance podem ser realizadas pelo Enterprise Manager.

### 3.4. Síntese de Voz

Síntese de voz é o processo de produção de sons naturais, voz sintetizada altamente perceptível simulada por uma máquina de certa maneira que soa como se fosse produzida por uma pessoa. Um sistema de síntese texto-voz ou *TTS (Text-To-Speech)* é um sistema computadorizado em que texto é transformado em voz. Um sistema TTS é composto por dois processos (Figura 4) no qual o texto é transformado no seu equivalente em voz sintetizada. O módulo de análise de texto e linguística processa o texto gerando a sua fonética equivalente e analisando a linguística de modo a determinar as características prosódicas<sup>1</sup> do texto. Um gerador de formas de onda produz a voz sintetizada. [27]

---

<sup>1</sup> Características relacionadas com o ritmo, entoação e demais atributos da fala.

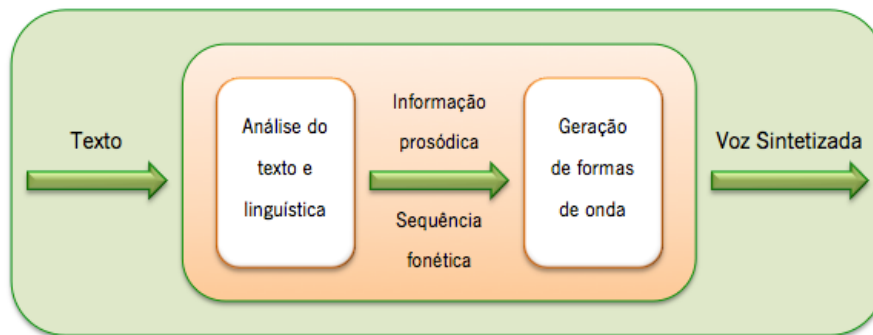


Figura 4 - Visão geral de um sistema TSS típico. (adaptado de [27])

Os sistemas TTS têm inúmeras utilidades, podem ser utilizados por exemplo em campos como a educação, telecomunicações, monitorização vocal, dispositivos com capacidades TTS. Os sintetizadores TSS são particularmente úteis para pessoas incapacitadas: ler *e-books* (livros digitais) e emails para pessoas com problemas de visão ou ler texto introduzido por pessoas que possuam problemas vocais são algumas das possíveis utilizações. Estes sistemas podem ainda evitar armazenamento de ficheiros de áudio, pois algumas informações de áudio podem ser muito grandes para serem guardadas, a sua pré-gravação pode ser muito dispendiosa ou simplesmente não se pode prever que tipo de resposta sonora seja necessária numa determinada situação. Vantagens da utilização de TTS incluem ler texto dinâmico, redução de espaço de armazenamento necessário, proporcionar feedback audível, avisar utilizadores de possíveis eventos, ler documentos, etc. [27] [28]

É com base nas vantagens da tecnologia TTS que esta é utilizada nesta dissertação. Com a utilização de sistemas TTS na chamada de paciente e leituras de mensagens eliminam-se variáveis que podem influenciar a perceptibilidade da mensagem lida. Muito barulho na sala do funcionário no momento da chamada, pronúncia e estado da voz do funcionário são exemplos de alguns problemas que são evitados com a utilização destes sistemas.

Na elaboração deste projeto foi utilizado um demo disponibilizado pela *NextUp.com*. Esta companhia é especializada em TTS. A voz utilizada chama-se *Nuance Madalena European Portuguese Female 22khz*, e é facilmente instalada no Sistema Operativo, para tal só necessário instalar o executável descarregado do site oficial.

A escolha do demo TTS da *NextUp.com* teve em consideração:

- Impossibilidade da criação de um TTS, este é um sistema complexo e de difícil construção.
- Os TTS disponíveis na internet são sempre pagos, pelo menos os que efetuam uma leitura razoável do texto introduzido.

- Os demos obtidos de outras companhias não permitiam utilizar o TTS de modo a testar as funcionalidades da aplicação.

# Capítulo 4

---

## 4. Trabalho Elaborado

### 4.1. Aplicação Desenvolvida

Como referido antes no capítulo 2.3 o tempo de espera no Serviço de Urgência desempenha um papel importante na qualidade dos serviços prestados e, por consequência, na satisfação dos utentes. A espera dos pacientes pode tornar-se menos exaustiva e cansativa com a disponibilização de informação relevante, por exemplo número de pessoas em espera, tempo de espera médio por resultado de triagem, tempo que o último paciente esperou. Através destas informações o Serviço de Urgência pode verificar se os tempos de atendimento estão dentro do normal, e, caso não estejam, poderá adotar medidas para melhorar os serviços prestados.

Outro factor a considerar no Serviço de Urgência é a chamada do paciente realizada pelo funcionário. Esta pode acarretar alguns problemas quanto à compreensão da mensagem, isto pode ocorrer por pronúncia incorreta ou estado da voz do funcionário, demasiado barulho na sala do funcionário, etc.

Considerando a problemática anterior desenvolveu-se uma aplicação que pretende melhorar a forma como os serviços são prestados no serviço de urgência. Esta aplicação fornece informação relevante sobre tempos de espera aos pacientes, realiza a chamada de pacientes um sistema TTS ,permite mostrar mensagens personalizadas aos utentes, ler mensagens e ainda escolher os vídeos que passam na aplicação. Como referido anteriormente na Estrutura da dissertação a aplicação é aludida pelo nome **SUTE** (Serviço de Urgência e Tempos de Espera), de modo a facilitar a sua referência.

#### 4.1.1. Funcionalidades da Aplicação desenvolvida

De modo a entender melhor os subcapítulos seguintes torna-se necessário explicar de uma forma geral as funcionalidades da aplicação SUTE, estas são:

- Apresentar o número de pessoas em espera por resultado de triagem e as pessoas que ainda não foram triadas.
- Apresentar os tempos de espera padrão para cada resultado do sistema de triagem de Manchester.
- Apresentar os tempos médios de espera para cada resultado de triagem do sistema de triagem de Manchester.
- Apresentar o tempo médio de espera para um paciente ser triado.
- Apresentar o tempo que o último paciente de cada resultado do sistema de triagem de Manchester esperou até receber tratamento.
- Apresentar o tempo que o último paciente esperou até ser triado.
- Mostrar mensagens personalizáveis. Estas são apresentadas em movimento e possuem hora de início e hora final de apresentação. As mensagens introduzidas podem ser utilizadas para passar informação relevante para os pacientes.
- Reproduzir vídeos na aplicação. Estes podem ter a função de publicitar algum tipo de produto/evento ou podem promover hábitos, por exemplo, a maneira correta de lavar as mãos.
- Mostrar hora e data.
- Realizar a chamada de pacientes para as respectivas salas utilizando voz sintetizada.
- Ler mensagens personalizáveis utilizando voz sintetizada por computador. As mensagens são ativadas através da hora de início indicada, podendo ser repetidas se um intervalo de tempo for indicado. Estas mensagens podem ser utilizadas quando existe a necessidade de informar os pacientes sobre alguma informação, a qual é proporcionada utilizando recursos visuais e sonoros.

#### 4.1.2. Tabelas Criadas e Modelo Relacional

Toda a informação necessária ao funcionamento da aplicação encontra-se numa base de dados armazenada em tabelas. Algumas das tabelas necessárias para o desenvolvimento da aplicação já se encontravam criadas no início do desenvolvimento deste projeto e a sua relação está representada na Figura 5, no entanto somente as tabelas **ED\_ATENDIMENTO**, **ED\_CHAMADA** e **ED\_SALAS** contêm dados utilizados no funcionamento da aplicação desenvolvida.

A tabela **ED\_ATENDIMENTO** contém informações pessoais sobre o paciente (sexo, contacto, data de nascimento, etc.) e sobre o episódio de atendimento do paciente (data de admissão no hospital, resultado da triagem, etc.). Nesta tabela os dados que são necessários ao funcionamento da aplicação

são: a data de admissão no hospital (*DTA\_AMI*), data da realização da triagem (*DTA\_TRIA*), data de chamada do paciente depois de já ter sido triado (*DTA\_CHAMADA*), e o resultado da triagem de Manchester (*RES\_TRIA*).

A tabela **ED\_CHAMADA** é necessária no processo de chamada dos pacientes para as salas de atendimento, e contém informação sobre o código da sala onde se encontra o paciente (*COD\_SALA*), nome do paciente (*NOME*), código da sala para a qual o paciente foi chamado (*SALA*), data da realização do pedido da chamada do paciente (*DTA\_PED*), data na qual o paciente foi avisado para se dirigir à respectiva sala (*DTA\_EXE*) e o estado da chamada, ou seja se já foi efectuada ou não (*ESTADO*).

A tabela **ED\_SALAS** é necessária de modo a saber o nome da sala (*DES\_SALA*) para qual o utente é chamado. O nome da sala é encontrado fazendo a correspondência entre o *COD\_SALA* da tabela **ED\_CHAMADA** com o *COD\_SALA* da tabela **ED\_SALAS**.

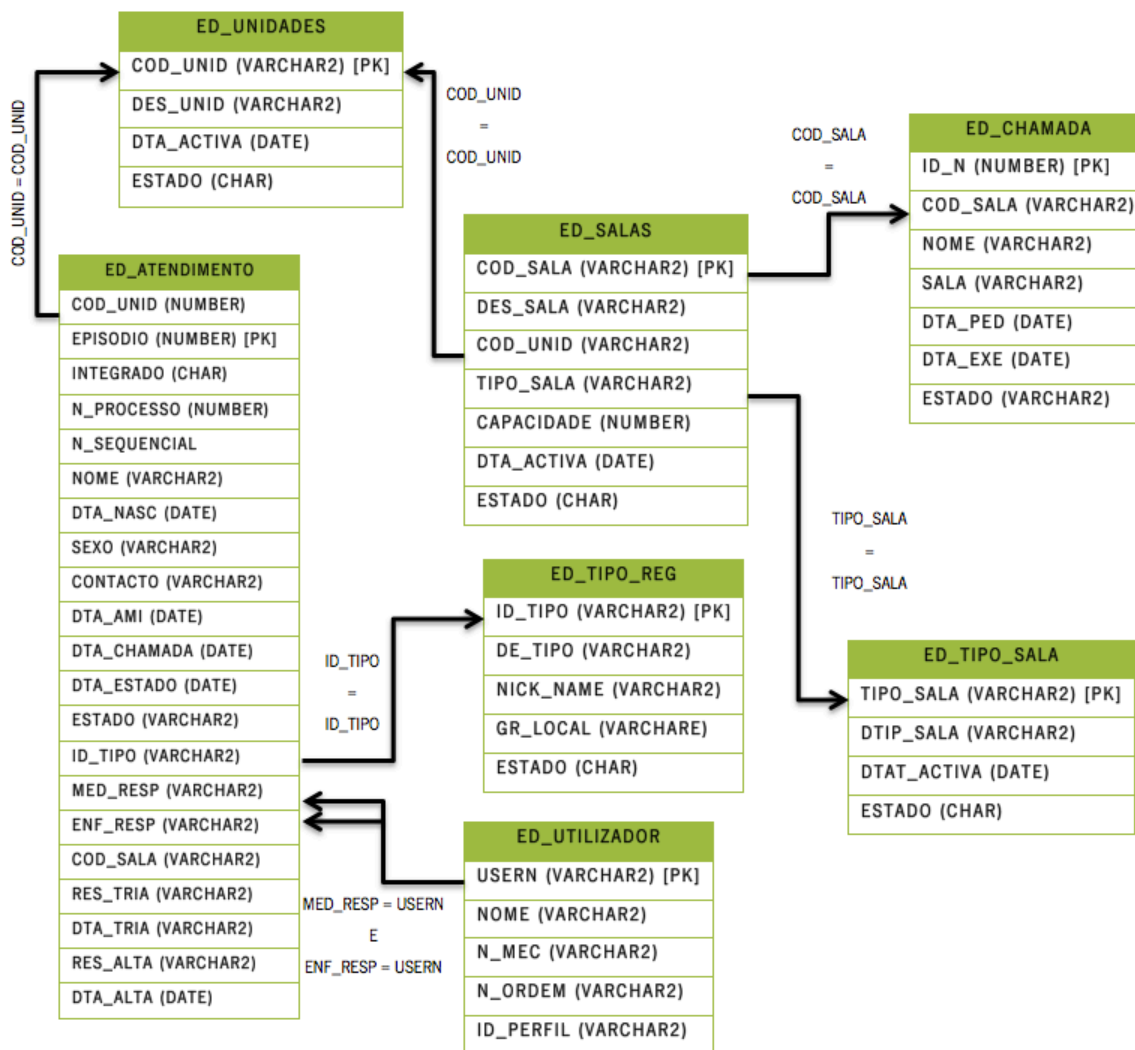


Figura 5 - Modelo Relacional das tabelas utilizadas no desenvolvimento da aplicação.



De forma a implementar as funcionalidades pretendidas na aplicação houve necessidade da criação de novas tabelas, são elas a **ED\_MENSAGEM\_VOZ**, **ED\_MENSAGEM\_RODAPE**, **ED\_APELIDOS**, **ED\_NOMES**.(Figura 6)(Anexos)

A tabela **ED\_MENSAGEM\_VOZ** é utilizada na leitura de mensagens personalizáveis e possui um *ID* auto-incrementado que identifica unicamente uma mensagem, o conteúdo da mensagem a ser lido é indicado na *MENSAGEM*, o *ESTADO* indica quantas mensagens restam por ler, *HORA\_INI* é a hora inicial a que a mensagem é lida, *ULT\_HORA\_ACT* é a última vez que a mensagem foi lida e *NREP* são o número de vezes que a mensagem deve ser lida.

ED_MENSAGEM_VOZ	ED_MENSAGEM_RODAPE	ED_APELIDOS
ID (NUMBER)	ID (NUMBER)	APELIDO (VARCHAR2)
MENSAGEM (VARCHAR2)	MENSAGEM (VARCHAR2)	APELIDOACENT (VARCHAR2)
ESTADO (NUMBER)	HORA_INI (VARCHAR2)	
HORA_INI (VARCHAR2)	HORA_FIM (VARCHAR2)	
ULT_HORA_ACT (VARCHAR2)	ESTADO (CHAR)	
NREP (NUMBER)		
		ED_NOMES
		NOME (VARCHAR2)
		NOMEACENT (VARCHAR2)

Figura 6 - Estrutura das Tabela Criadas (ED\_MENSAGEM\_VOZ,ED\_MENSAGEM\_RODAPE, ED\_APELIDOS e ED\_NOMES).

As mensagens escritas a ser mostradas na aplicação são armazenadas na tabela **ED\_MENSAGEM\_RODAPE** que tem um *ID* auto-incrementado que identifica unicamente uma mensagem, o conteúdo da mensagem é especificado no campo *MENSAGEM*, o momento em que a mensagem é mostrada é definido pela hora inicial (*HORA\_INI*) e hora final (*HORA\_FIM*), o *ESTADO* indica se a mensagem deve ser mostrada (*ESTADO* = 1) ou não (*ESTADO* = 0).

Na chamada do paciente é utilizado o nome próprio e o apelido. No entanto o nome completo do paciente é introduzido sem acentuação na base de dados, por isso torna-se necessário obter o respectivo nome acentuado de modo a que o sistema TTS consiga ler o nome corretamente. Com o recurso a variados sites e em particular ao site do Instituto dos Registos e do Notariado foram construídas duas tabelas **ED\_NOMES** e **ED\_APELIDOS**. A primeira realiza a correspondência entre um nome sem acento (*NOME*) e o respectivo nome acentuado (*NOME\_ACENT*), por exemplo Joao para João, enquanto a tabela **ED\_APELIDOS** realiza a correspondência entre um apelido sem acento (*APELIDO*) e o respectivo apelido acentuado (*APELIDOACENT*), por exemplo Napoleao para Napoleão.  
[29] [30] [31] [32] [33]

### 4.1.3. Arquitetura Da Aplicação

A arquitetura da aplicação **SUTE** é constituída por quatro camadas: camada de apresentação (Interface Utilizador), camada lógica (Classes vb.net), camada de acesso a dados(ADO.NET) e a camada de dados (servidor de base de dados Oracle).

Este modelo exige os seguintes componentes físicos, essenciais para o seu funcionamento, um computador, o qual possui ligação a um monitor que disponibiliza a informação aos pacientes do serviço de urgência, e um servidor para alojar o gestor de base de dados Oracle.

- **Camada de Apresentação**, é a camada responsável pela apresentação da informação ao utilizador. O aspecto da aplicação e a forma como a informação é disponibilizada encontra-se especificada em linguagem XAML.
- **Camada Lógica**, é constituída por classes VB.NET responsáveis por todos os processos da aplicação que tratam os dados provenientes da camada de dados antes de serem apresentados aos utilizadores.
- **Camada Acesso a Dados**, é o elo de ligação entre a camada lógica e a camada de dados, assente sob a tecnologia ADO.NET.
- **Camada de Dados**, é a camada onde reside, de forma estruturada, a informação inerente á aplicação. Informação, esta, gerida e armazenada num servidor de dados Oracle.



Figura 7 - Arquitetura utilizada no desenvolvimento da aplicação SUTE.

#### 4.1.4. Interface

A Interface da aplicação **SUTE** é dividida em 3 interfaces diferentes que resultam da necessidade da apresentação de informações diferentes, são elas:

- Interface principal.
- Interface de chamada.
- Interface de mensagens de voz.

##### 4.1.4.1. Interface Principal

A interface principal da aplicação **SUTE** é a interface que se encontra ativa maior parte do tempo. Foi desenvolvida de modo a permitir apresentar informação ao utilizador de forma simples e intuitiva. O tipo de informação e o modo como é distribuída encontra-se representado na Figura 8.

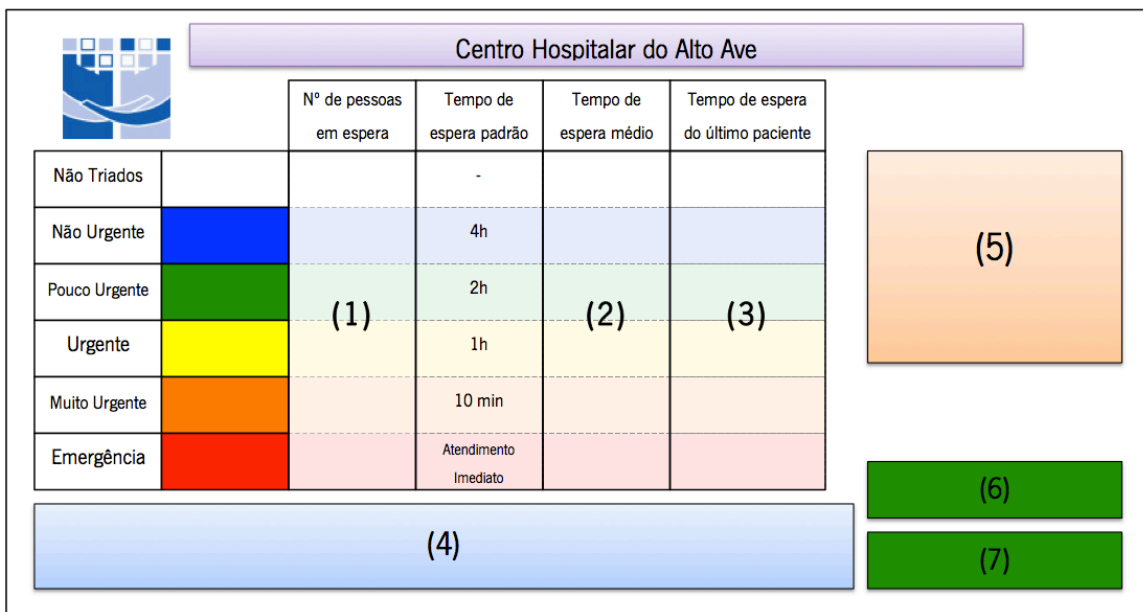


Figura 8 - Esquema representativo da interface principal.

Os campos numerados de 1 a 7 são os campos dinâmicos da aplicação, ou seja são os únicos que são alterados durante o funcionamento da aplicação, a restante informação mantém-se sempre igual. Os campos dinâmicos são:

- No campo 1 é apresentado o nº de pessoas em espera para ser triadas e por resultado de triagem;

- No campo 2 é apresentado o tempo de espera médio para ser triado e por resultado de triagem;
- No campo 3 é apresentado o tempo que o último paciente triado esperou e o tempo que último paciente de cada resultado de triagem esperou até receber tratamento;
- No campo 4 são apresentadas mensagens escritas;
- No campo 5 são reproduzidos vídeos;
- No campo 6 é indicada a data atual;
- No campo 7 é indicada a hora atual.

A informação dos campos 1, 2 e 3 e tempo de espera padrão é apresentada em forma de tabela. Esta é a única forma de apresentar essa informação de maneira estruturada e simples pois existem 6 resultados de triagem (Não triado, Não Urgente, Pouco Urgente, Urgente, Muito Urgente e Emergência) e 4 valores para cada (Nº de pessoas em espera, Tempo de espera padrão, Tempo de espera médio e Tempo de espera do último paciente).

#### 4.1.4.2. Interface de Chamada

Esta interface é apresentada aos utentes quando ocorre a chamada de um paciente para uma determinada sala. A sua interface possui uma estrutura muito simples, possuindo somente dois campos dinâmicos(Figura 9).

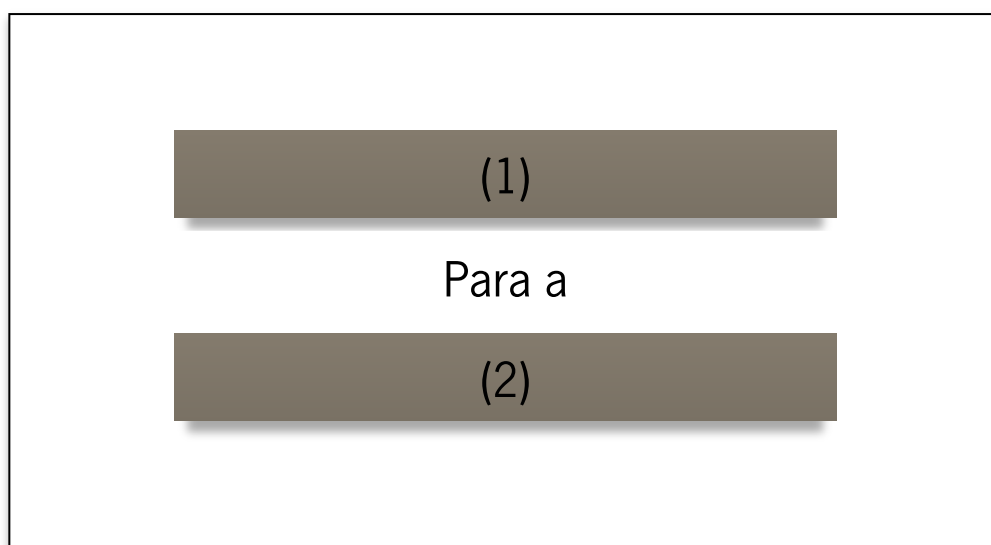


Figura 9 - Esquema representativo da interface de chamada.

Os campos que variam consoante a necessidade da aplicação são: campo 1 corresponde ao nome do utente que é chamado e o campo 2 é a sala para a qual o utente tem de se dirigir.

#### 4.1.4.3. Interface de Mensagem de Voz

Esta é a interface mais simples de todas e é apresentada aos utentes do Serviço de Urgência quando a aplicação apresenta e realiza a leitura de uma mensagem. Como se pode ver na Figura 10, o único campo que é modificado segundo o funcionamento da aplicação é o campo 1. O campo 1 representa conteúdo da mensagem a ser mostrada.

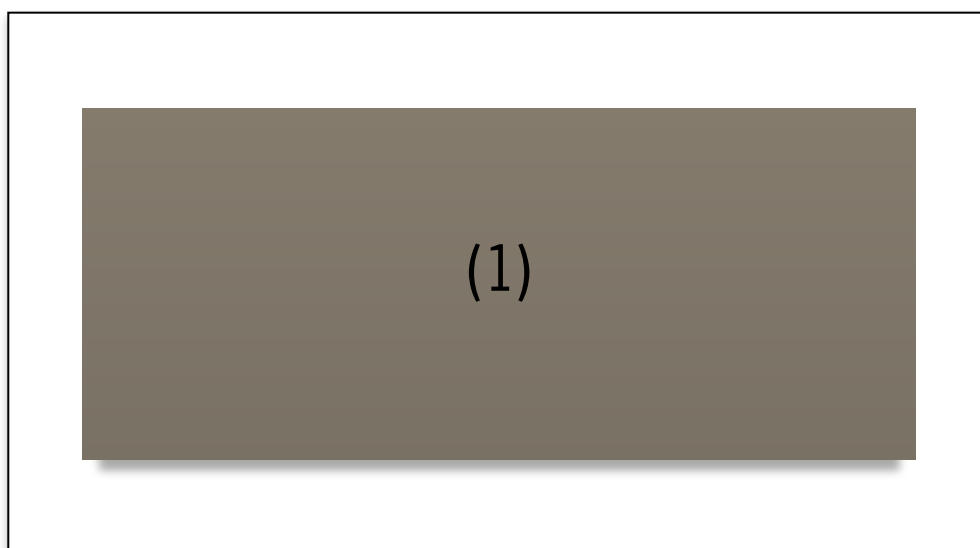


Figura 10 - Esquema representativo da Interface de Mensagem de Voz.

#### 4.1.5. Funcionalidades da Aplicação

As funcionalidades da aplicação vão ser explicadas separadamente de acordo com a interface na qual é manifestada a sua ação. Isto porque, apesar de os processos não dependerem somente da interface em que se manifestam, torna-se mais simples a sua explicação quando acompanhado do resultado da funcionalidade. Assim as funcionalidades da aplicação estão divididas pela interface principal, interface de chamada e interface de leitura de mensagens de voz.

#### 4.1.5.1. Funcionalidades Apresentadas na Interface Principal

A informação que é apresentada nesta interface é retirada de 2 tabelas, estas são **ED\_ATENDIMENTO**, **ED\_MENSAGEM\_RODAPE**.

Informações como o número de pessoas em espera, tempos de espera médios, tempo de espera do último paciente são retiradas da tabela **ED\_ATENDIMENTO**. Todos os dados exibidos na Figura 11 são calculados com base nas últimas 24 horas tendo em conta a hora atual da apresentação da informação, isto deve-se ao facto de nenhum paciente permanecer mais de 24 horas no serviço de urgência. Esta consideração permite tornar a aplicação mais eficiente visto que são utilizados menos dados e permite obter tempos de espera médios correspondentes a situação atual do serviço de urgência, pois, dependendo do dia, os tempos de espera podem variar.

A informação de toda a aplicação é atualizada a cada 3 segundos. Esta atualização frequente é necessária devido à mudança no fluxo de pacientes no Serviço de Urgência, a qualquer momento pode acontecer: entrada de novos pacientes no serviço de urgência, triagem, chamada ou alta de pacientes.

Centro Hospitalar Do Alto Ave					
		Nº de pessoas em espera	Tempo de espera padrão	Tempo de espera médio	Tempo de espera do último paciente
Não Triados		0	-	12 min	5 min
Não Urgente	8	6	4 h	2 h e 55 min	3 h
Pouco Urgente		4	2 h	1 h e 23 min	1 h e 6 min
Urgente		3	1 h	1 h e 1 min	1 h e 20 min
Muito Urgente		4	10 min	14 min	15 min
Emergência		2	Atendimento Imediato	3 min	4 min

Atenção: O acesso às casas de banho se é seu dirija-se a recepção

14-11-2011

**22:37:40**

Figura 11 - Interface Principal da aplicação SUTE em funcionamento.

Como foi explicado no capítulo 2.2.1, a cada resultado de triagem é associado um valor, este valor é armazenado na tabela **ED\_ATENDIMENTO** na coluna **RES\_TRIA**. O número de pessoas em espera por situação clínica é facilmente retirado contando o número de utentes em espera por resultado de triagem, se o resultado de triagem não se encontrar definido este encontra-se não triado.

O tempo médio de espera para ser triado foi calculado de acordo com:

$$T_{\text{Médio de espera triagem}} = \frac{1}{n} \sum_{i=1}^n (DTA_{TRIA} - DTA_{AMI})$$

em que  $n$  é o número de pessoas em espera para ser triadas,  $DTA_{TRIA}$  é a data em que a triagem foi efectuada e  $DTA_{AMI}$  é a data em que o paciente realizou a admissão no serviço de urgência.

O tempo médio de espera para as diferentes resultados de triagem resulta do seguinte cálculo:

$$T_{\text{médio de espera}} = \frac{1}{n} \sum_{i=1}^n (DTA_{CHAMADA} - DTA_{TRIA})$$

em que  $n$  é o número de pessoas em espera para um determinado resultado de triagem,  $DTA_{CHAMADA}$  é a data em que o utente é chamado após ser triado e  $DTA_{TRIA}$  é a data em que a triagem é realizada.

Para obter o tempo do último utente a ser triado é utilizado o cálculo:

$$T_{\text{último triado}} = (DTA_{TRIA} - DATA_{AMI})$$

em que  $DTA_{TRIA}$  é a data em que a triagem foi realizada e  $DATA_{AMI}$  é a data em que se realizou o registo de entrada no hospital. Ambas as datas são referentes ao último paciente triado.

O tempo do último paciente a ser atendido por cada resultado de triagem é obtido segundo:

$$T_{\text{último atendido}} = (DTA_{CHAMADA} - DATA_{TRIA})$$

em que  $DTA_{CHAMADA}$  é a data de chamada após triagem e  $DATA_{TRIA}$  é a data em que a triagem foi efectuada. Ambas as datas são relativas ao último paciente chamado após triagem.

O formato dos tempos de espera foi modificado de forma a facilitar a leitura aos pacientes. Para tal foi desenvolvida a função *formathoras*(Código 10). A função recebe um tempo no formato

“horas:minutos” e transforma em “horas e minutos”. Na

Tabela 5 são mostrados 3 exemplos da conversão de tempos de espera pela função *formathoras*.

### Código 10 – Parâmetros da função *formathoras*.

```
Private Function formathoras(ByVal hora As String) As String ...
```

Tabela 5 – Exemplo de conversão de tempos de espera realizado pela função *formathoras*.

Tempo de espera devolvido pela base de dados	Tempo de espera formatado
00:03	3 min
01:05	1 h e 5 min
02:00	2 h

### *Mensagens de Rodapé*

As mensagens a ser exibidas na aplicação são armazenadas na tabela **ED\_MENSAGEM\_RODAPE**. De modo a explicar o funcionamento das mensagens de rodapé é necessário fornecer um exemplo. Imaginando que o conteúdo da tabela é o representado na Tabela 6, é possível observar que todas as mensagens se encontram ativas, pois o estado é 1, a mensagem com o *ID* igual a 62, deve ser mostrada entre as 14:00 e as 20:00, a mensagem com *ID* igual a 63 entre as 15:30 e as 01:00 e a mensagem com *ID* igual a 64 entre as horas 09:30 e 14:00

Tabela 6 - Conteúdo da tabela **ED\_MENSAGEM\_RODAPE** contendo 2 mensagens.

ID	Mensagem	HORA_INI	HORA_FIM	ESTADO
62	Informam-se todos os utentes que amanhã as urgências funcionarão somente na parte de tarde	14:00	20:00	1
63	Encontrou-se um telemóvel perto das casas de banho. Se é seu dirija-se à recepção	15:30	01:00	1
64	O bar do hospital disponibiliza almoços e pequenos almoços económicos a todos os utentes.	09:30	14:00	1

A função *addmsg* (Código 11) é responsável por verificar se uma mensagem ativa deve ser mostrada ou não. A função *addmsg* é necessária pois, uma vez que a hora de início e fim não tem em conta datas, não se pode simplesmente verificar se a hora atual é maior que a hora inicial e menor que a hora final. Por exemplo, a mensagem com *ID*=63 da Tabela 6 deve ser mostrada caso a hora atual



seja, por exemplo, 19:07. No entanto, se compararmos a hora atual, esta é superior à inicial e superior à final o que torna a simples comparação hora inicial < hora atual < hora final impraticável em determinadas situações.

**Código 11 - Parâmetros da função *addmsg*.**

```
Private Function addmsg(ByVal Horaini As Integer, ByVal Horafim As Integer) As Boolean
```

A função *addmsg* recebe a hora inicial e hora final como argumentos e devolve um valor booleano, ou seja a função devolve um valor verdadeiro caso a mensagem deva ser apresentada ou um valor falso caso se verifique o contrário. A hora inicial, atual e final são transformadas do formato hh:mm para o formato hhmm, por exemplo 13:34 para 1334. Esta transformação é necessária de modo a comparar as horas sem ter em conta as datas. Depois desta transformação podem ocorrer duas situações:

- **Hora inicial < Hora final**, por exemplo hora inicial é 09:15 e hora final é 13:00. Estes casos são simples e são tratados de acordo com a condição em baixo. A mensagem só é apresentada caso a hora atual esteja no intervalo : hora inicial ≤ hora atual < hora final.

*Valor Booleano = (Hora inicial ≤ hora atual) E (Hora final > Hora atual)*

- **Hora inicial > Hora final**, por exemplo hora inicial é 20:35 e a hora final é 01:45 . Estes casos em que a hora final é maior que a hora inicial devem ser tratados de forma especial. A mensagem deve ser apresentada caso a hora atual seja maior ou igual que a hora inicial (hora inicial ≤ hora atual) ou hora atual for menor que a hora final (hora final > hora atual). Por exemplo se a hora atual é 00:15 a mensagem é apresentada pois verifica pelo menos uma das condições indicadas na condição em abaixo:

*Valor Booleano = (Hora inicial ≤ hora atual) OU (Hora final > Hora atual)*

Na Figura 12 podem-se ver que mensagens presentes na Tabela 6 são apresentadas na aplicação. As mensagens que são mostradas entre a hora 19:06:44 e 19:07:11 são as mensagens com *ID* igual a 62 e 63, pois a hora atual da aplicação encontra-se dentro do intervalo de início e fim da apresentação da mensagem, 14:00 < 19:06 < 20:00 e 15:30 < 19:06 < 01:00, respectivamente. A mensagem com *ID*=64, como era de esperar não é apresentada pois a hora atual da aplicação não se enquadra no intervalo, 09:30 < 19:06 > 14:00.

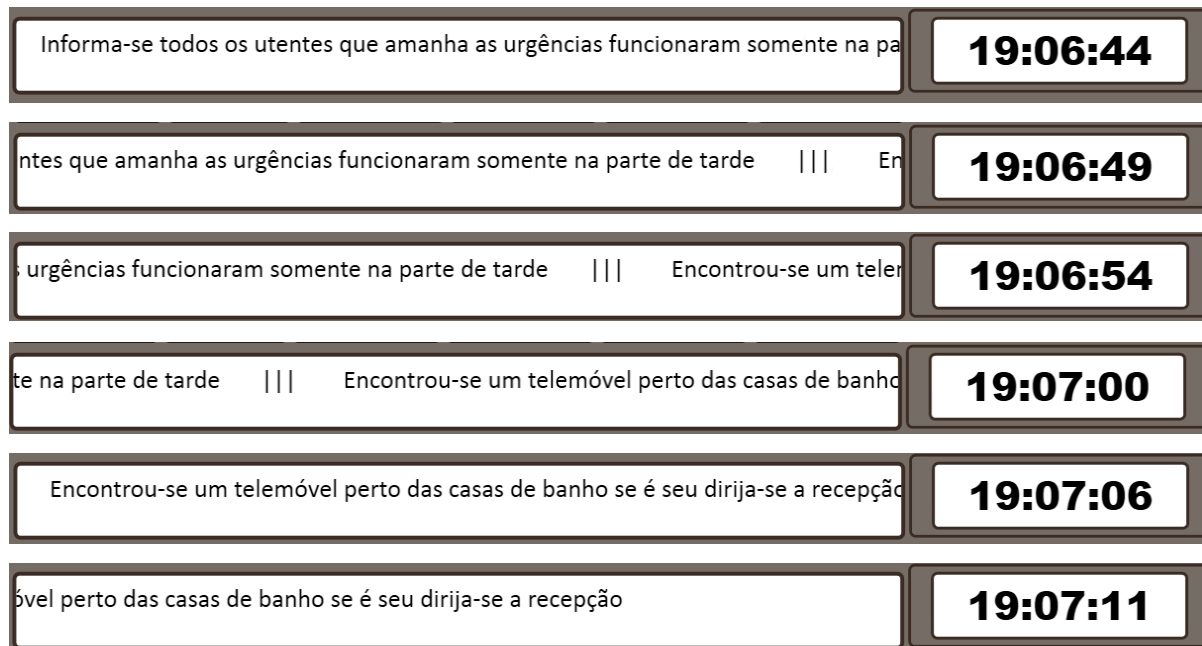


Figura 12 - Mensagens apresentadas na aplicação em diferentes tempos.

As mensagens de rodapé são mostradas em movimento da direita para esquerda como é possível observar na Figura 12. A função responsável pelo movimento das mensagens é a *Bannernews*. Esta função recebe as mensagens separadas por “|||” que é necessário apresentar e tem em conta o tamanho das mensagens, tipo de letra e propriedades dos controlos WPF (*Canvas*, *TextBlock*) responsáveis por apresentar as mensagens.

Código 12 – Parâmetros da função *Bannernews*.

```
Private Sub Bannernews(ByVal theText As String)
```

### ***Reprodução de Vídeos***

Para reproduzir vídeos na aplicação basta somente colocar os vídeos numa pasta predefinida para a aplicação. A aplicação quando é iniciada procura ficheiros com extensão que o *Windows Media Player* (WMP) consiga reproduzir e reproduz a lista de vídeos presentes na pasta. Por definição o WMP consegue reproduzir poucos formatos de vídeo (ASF, WMA, WMV e WM), no entanto, com a instalação de determinados pacotes de *codecs*, como por exemplo o K-Lite Codec Pack, é possível reproduzir outros ficheiros de vídeo AVI, MKV, MP4, FLV, OGM, MPEG, MOV, HDMOV, TS, M2TS, e ficheiros OGG.

A função *getlistvideo* é a função responsável por obter a lista de vídeos que devem ser reproduzidos na aplicação, enquanto a função *video\_MediaEnded* está associada ao evento de final de

reprodução de um vídeo. Quando um vídeo acaba de ser reproduzido a função `video_MediaEnded` é chamada e esta seleciona o próximo vídeo da lista.

**Código 13 – Declaração da função *getlistvideo*.**

```
Private Sub getlistvideo() ...
```

**Código 14 – Declaração da função *video\_MediaEnded*.**

```
Private Sub video_MediaEnded(ByVal sender As System.Object, ByVal e As  
System.Windows.RoutedEventArgs) Handles video.MediaEnded ...
```

#### 4.1.5.2. Hora e Data

A hora e data atual são mostradas na parte inferior da aplicação, as funções *GetCurrentDate* e *GetCurrentTime* são responsáveis por obter a data e a hora (Código 15).

**Código 15 – Parâmetros das funções *GetCurrentTime* e *GetCurrentDate*.**

```
Private Function GetCurrentTime() As String ...  
Private Function GetCurrentDate() As String ...
```

#### 4.1.5.3. Funcionalidades Relacionadas com a Interface de Chamada

Só a funcionalidade de chamada de pacientes se encontra relacionada com esta interface, no entanto esta funcionalidade é umas das mais importantes e, de modo a poder ser executada, necessitava de uma interface única para o fazer.

Na chamada de um paciente para uma sala intervêm as tabelas **ED\_CHAMADA**, **ED\_SALAS**, **ED\_NOMES** e **ED\_APELIDOS**. Na Figura 13 é possível observar um exemplo da chamada de um paciente para uma sala, em que o paciente é Mário Veloso e a sala para a qual se deve dirigir é a sala de espera imagem.

A chamada de um paciente é desencadeada pela introdução de uma nova linha na tabela **ED\_CHAMADA**. Este processo é iniciado por um médico ou enfermeiro e ocorre sempre que um paciente é chamado para uma sala. Na chamada do paciente é utilizado o nome próprio e apelido do paciente e a sala para onde o utente se deve dirigir.

Como referido anteriormente, o nome completo dos pacientes não é introduzido com acentuação, então torna-se necessário retirar o nome próprio e apelido do nome completo do paciente e verificar se estes existem nas tabelas **ED\_NOMES** e **ED\_APELIDOS**. As funções responsáveis pela formação do nome próprio e apelido que são utilizados na chamada são respectivamente a *getnome* (Código 16) e *getapelido* (Código 17).

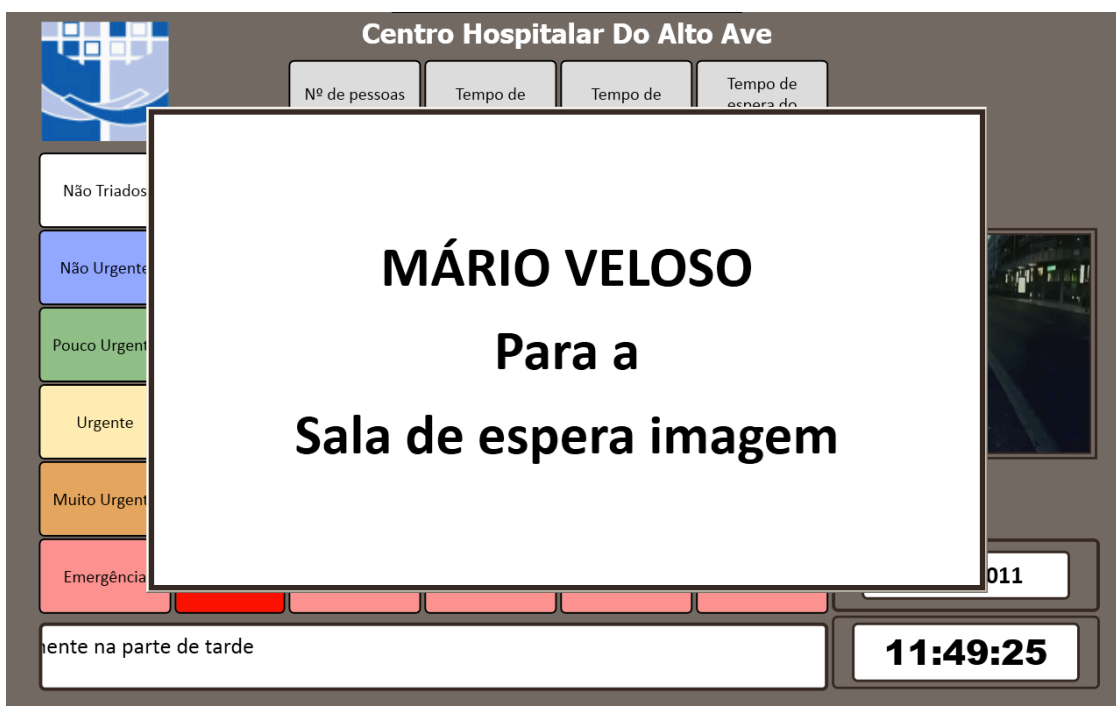


Figura 13 - Exemplo de uma chamada de um paciente, apresentando a interface de chamada.

A função *getnome* recebe como argumento o nome próprio retirado da tabela **ED\_CHAMADA** e verifica se este possui uma correspondência com um nome acentuado na tabela **ED\_NOMES**. Se existe, o nome devolvido pela função é o nome acentuado. Se não existe, o nome devolvido pela função é o nome que lhe foi dado como argumento inicialmente. Para o apelido do paciente ocorre o mesmo processo que o nome próprio mas a função utilizada é a *getapelido* e a tabela utilizada é a **ED\_APELIDOS**.

Código 16 - Parâmetros da função *getnome*.

```
Private Function getnome(ByVal nomepessoa As String) As String
```

Código 17 - Parâmetros da função *getapelido*.

```
Private Function getapelido(ByVal nomepessoa As String) As String
```

Para obter o nome da sala é necessário recorrer à tabela **ED\_SALAS**. Através da correspondência do *COD\_SALA* da tabela **ED\_CHAMADA** com o *COD\_SALA* da tabela **ED\_SALAS**, é possível obter o nome da sala para a qual o paciente necessita se dirigir.

Obtendo o nome próprio e apelido corretos e o nome da sala, é então mostrada a interface presente na Figura 13. Acompanhado da apresentação da janela ocorre a chamada sonora do paciente. A função *ChamarVoz* é a responsável pela chamada sonora. Esta função recebe como argumentos o

nome próprio e apelido do paciente a chamar e o nome da sala para a qual o paciente se tem de dirigir. Através de funções proporcionadas pela biblioteca *SpeechLib* e da utilização da voz *NextUp.com* instalada no Sistema Operativo, é efectuada a chamada sonora do paciente.

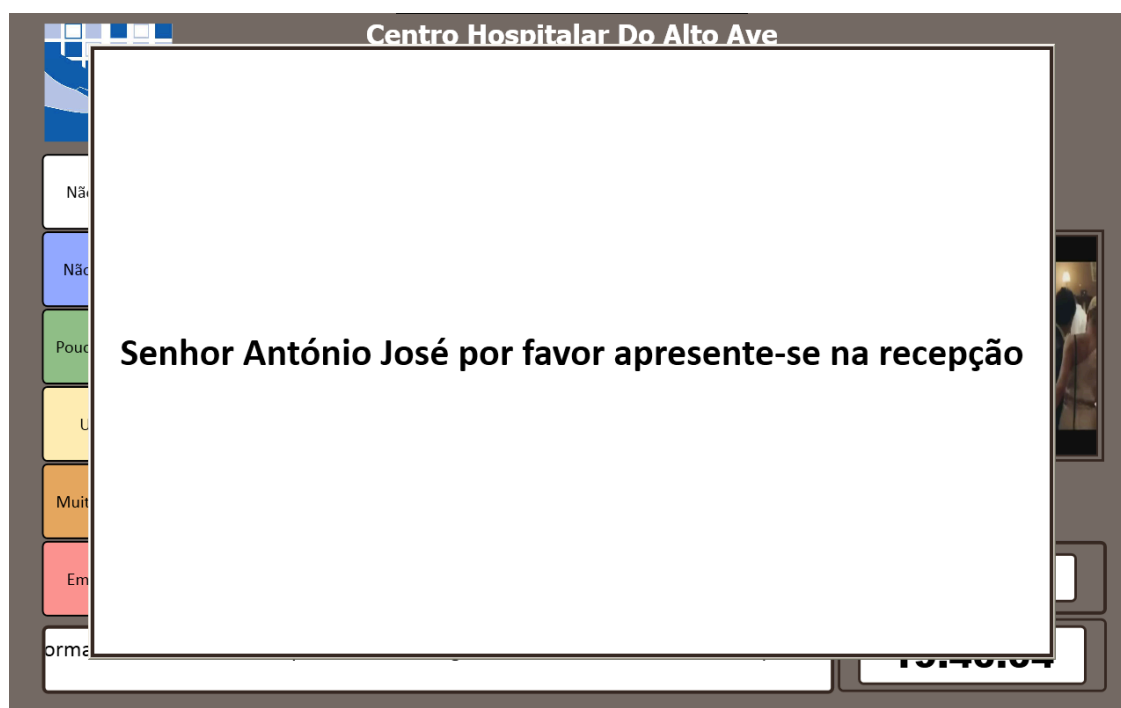
**Código 18 - Parâmetros da função *ChamarVoz*.**

```
Public Shared Sub ChamarVoz(ByVal nome As String, ByVal sala As String)
```

#### 4.1.5.4. Funcionalidades Apresentadas na Interface de Leitura de Mensagens de Voz

Na leitura de mensagens de voz, a tabela **ED\_MENSAGEM\_VOZ** é a única tabela da base de dados utilizada. As mensagens podem ser enunciadas uma só vez ou pode ser especificado um determinado intervalo e número de repetições de modo a ser apresentada o número de vezes indicado.

Na Figura 14 é apresentado um exemplo de como as mensagens personalizáveis são mostradas aos utentes. Além de apresentar a mensagem aos utentes, esta é também é anunciada através da utilização da voz sintetizada.



**Figura 14 - Exemplo da apresentação de mensagem personalizável, apresentando a interface de leitura de mensagem de voz.**

Na Tabela 7 pode-se observar o conteúdo da tabela **ED\_MENSAGEM\_VOZ**. A mensagem com *ID=82* possui o campo *ULT\_HORA\_ACT* vazio o que nos indica que ainda não foi lida nenhuma vez,

esta mensagem deve ser lida 2 vezes. A mensagem com o *ID=83* já foi lida o número de vezes pretendido, pois o *ESTADO= 0*.

Tabela 7 - Conteúdo da tabela ED\_MENSAGEM\_VOZ.

ID	MENSAGEM	ESTADO	HORA_INI	ULT_HORA_ACT	INTERVAL	NREP
82	Senhor António José por favor apresente-se na recepção	2	16:40		00:05	2
83	O Doutor Tavares hoje não poderá realizar consultas na parte da manha	0	12:30	12:30	00:00	1

Antes da leitura das mensagens de voz é necessário saber quais as mensagens que ainda se encontram ativas e em que instante de tempo devem ser lidas. As mensagens com *ESTADO > 0* são as mensagens ativas, e portanto somente estas necessitam ser consideradas no processo de verificação de mensagens a ler. Uma mensagem de voz ativa é lida quando se verifica uma destas situações (ter em consideração que a Hora atual é a hora em que ocorre a verificação destas situações):

1. ***ULT\_HORA\_ACT = NULL E HORA\_INI = Hora atual***, *ULT\_HORA\_ACT = NULL* é indicativo de que a mensagem não foi lida nenhuma vez, pois sempre que uma mensagem é lida este toma o valor da hora em que ocorreu a leitura. A leitura da mensagem neste caso deve ocorrer quando a Hora atual for igual a *HORA\_INI*.
2. ***(ULT\_HORA\_ACT + INTERVAL) = Hora atual E ULT\_HORA\_ACT ≠ NULL***, *ULT\_HORA\_ACT ≠ NULL* indica que a mensagem já foi lida e como tal agora deve ser considerado a *ULT\_HORA\_ACT + INTERVAL* como sendo a hora que deve ocorrer a próxima leitura da mensagem, ou seja a mensagem é lida quando se verifica ***(ULT\_HORA + INTERVAL)= Hora atual***.

Sempre que uma mensagem é lida, o seu *ESTADO* e *ULT\_HORA\_ACT* são alterados. Ao valor do *ESTADO* é subtraída uma unidade (*ESTADO = ESTADO -1*) indicando que a mensagem foi lida uma vez. Enquanto a *ULT\_HORA\_ACT* toma o valor da hora em que foi realizada a leitura (*ULT\_HORA\_ACT = Hora atual*).

A função responsável pela leitura da mensagem é a *LerMensagemVoz* (Código 19) que realiza as mesmas operações que a função *ChamarVoz* (Código 18), mas neste caso o único valor que é dado à função é o conteúdo da mensagem a ser lida.

```
Public Shared Sub LerMensagemVoz(ByVal mensagem As String)
```

## 4.2. Aplicação de Configuração

De modo a poder gerir certas funcionalidades da aplicação, era necessária a criação de uma aplicação Web que permitisse:

- Gerir os vídeos que são apresentados.
- Inserir novas mensagens de rodapé e alterar o estado de mensagens existentes.
- Apagar mensagens de rodapé.
- Inserir novas mensagens de voz.
- Apagar mensagens de voz.

A aplicação de configuração foi desenvolvida no Visual Studio 2010 utilizando ASP.NET e as linguagens HTML, CSS, JavaScript e VB.NET.

### 4.2.1. Interface

No desenvolvimento da aplicação Web de configuração da aplicação **SUTE** teve-se em conta a necessidade de: possuir uma interface simples, facilidade de utilização e prevenção de possíveis erros dos utilizadores.

A aplicação de configuração foi dividida em 5 partes de modo a melhorar a sua estrutura e organização. A primeira é uma simples página de apresentação onde é apresentada uma imagem da aplicação e onde é referido o que é possível realizar na aplicação de configuração (Figura 15).

Para aceder às 4 principais páginas que permitem gerir e modificar o comportamento da aplicação basta clicar num dos 4 botões situados na parte superior da página (Figura 16). Esta parte superior é comum a todas as páginas. O tipo de tarefas e o que é apresentado nestas 4 páginas é explicado em pormenor mais adiante.

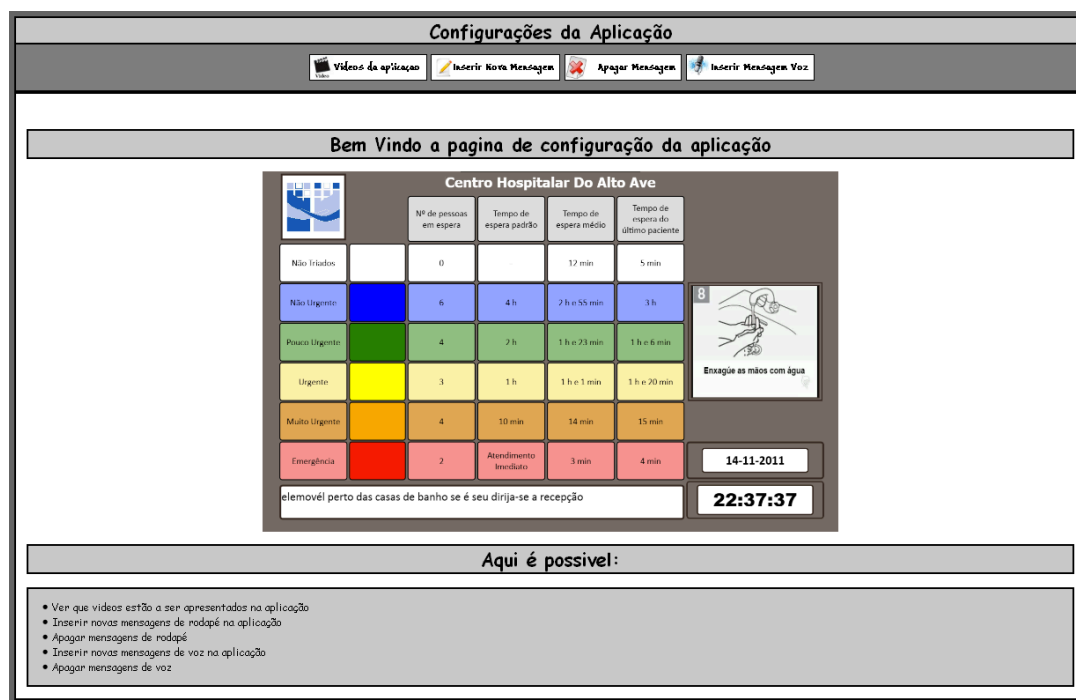


Figura 15 - Página Inicial de Configuração da Aplicação.



Figura 16 – Parte comum em todas as páginas da aplicação de configuração.

#### 4.2.2. Arquitetura

A aplicação Web desenvolvida apresenta uma arquitetura distribuída de três camadas: camada de apresentação, camada de aplicação e camada de dados.

O acesso a esta aplicação é fornecido por intermédio de um *web browser*.

Na Figura 17 é possível compreender a relação entre as diferentes camadas da arquitetura:

- **Camada de apresentação**, é o componente da aplicação responsável pela interface que é apresentada ao utilizador. É responsável por exemplo, pelo modo como os dados são apresentados e como o utilizador interage com eles. Esta camada é baseada na tecnologia ASP.NET Web Forms e estruturada em ficheiros de extensão *aspx* sob linguagem HTML.



- **Camada lógica**, assume a computação lógica dos processos gerados por esta aplicação Web assim como o acesso à camada de dados. Esta camada é alojada no servidor Web e definida na linguagem de programação VB.NET.
- **Camada de dados**, é a camada onde reside toda a informação utilizada inerente à aplicação. É nesta camada que se encontra o servidor de gestão de base de dados Oracle.



Figura 17 - Arquitetura de 3 camadas utilizada na Aplicação Web

#### 4.2.3. Gestão de Vídeos

Ao clicar no botão “Vídeos da aplicação” o utilizador é direcionado para uma página semelhante à Figura 18. Nesta página pode-se observar quais os vídeos que estão a ser utilizados pela aplicação, é apresentada a pasta onde se encontram, nome e tamanho. Como referido anteriormente em 4.1.5.1, os vídeos que são reproduzidos na aplicação encontram-se numa pasta. E para os vídeos serem reproduzidos basta colocá-los na pasta, ou se o utilizador pretender pode retirar o vídeo da pasta fazendo assim com que o vídeo deixe de ser reproduzido pela aplicação.



Figura 18 - Página que permite ver os vídeos que são reproduzidos na aplicação

#### 4.2.4. Inserir Novas Mensagens De Rodapé

Clicando no botão “Inserir Nova Mensagem” na parte superior da aplicação de configuração é apresentada a página da Figura 19 que permite inserir novas mensagens de rodapé na aplicação. Nesta página é possível ver quais as mensagens de rodapé que se encontram na base de dados, alterar o seu estado e inserir novas mensagens na base de dados.

As mensagens existentes na tabela **ED\_MENSAGEM\_RODAPE** são apresentadas numa tabela, na qual é possível observar o ID, Mensagem, Hora Inicial, Hora Final e Estado. O estado como foi explicado anteriormente em 4.1.5.1 indica se a mensagem deve ser apresentada (estado 1) ou não (estado 0) na aplicação. Para alterar o estado de uma mensagem basta simplesmente clicar no botão “Alterar Estado” correspondente à mensagem pretendida. Se o estado for 1 este é alterado para 0, se for 0 é alterado para 1.

Na introdução de uma mensagem de rodapé é necessário indicar: Hora de início, Hora final, estado e o corpo da mensagem. O *ID* não é introduzido pois, como foi explicado anteriormente, é auto incrementado.

**Configurações da Aplicação**

Vídeos da aplicação
 Inserir Nova Mensagem
 Apagar Mensagem
 Inserir Mensagem Voz

**Inserir Nova Mensagem De Rodapé**

Mensagens existentes na base de dados:

ID	Mensagem	Hora Inicial	Hora Final	Estado	Alterar Estado
62	Informa-se todos os utentes que amanha as urgências funcionaram somente na parte de tarde	14:00	20:00	1	<input type="button" value="Alterar Estado"/>
63	Encontrou-se um telemovel perto das casas de banho se e seu dirija-se a recepção	14:00	01:00	1	<input type="button" value="Alterar Estado"/>

**Inserir Novas Mensagens**

Início: Hora  Minutos  Fim: Hora  Minutos

Estado (1 sendo activo ou 0 inactivo)

Mensagem

Figura 19 - Página que permite inserir novas mensagens de rodapé na base de dados

Quando é inserida uma mensagem tem de ser verificado se a informação introduzida se encontra correta. Os impedimentos na introdução das mensagens são: a hora inicial e final não pode ser igual, o estado só pode ser “1” ou “0” e a mensagem não pode ser nula. Nenhuma mensagem é inserida na base de dados se for verificado pelo menos um dos casos anteriormente mencionados. Além disso, o utilizador é informado com mensagens a vermelho no local onde o preenchimento foi incorreto. É possível observar na Figura 20 as mensagens que são obtidas por preenchimento incorreto de informação.

**Inserir Novas Mensagens**

Início: Hora  Minutos  Fim: Hora  Minutos

Estado (1 sendo activo ou 0 inactivo)  Tem de escrever 0 ou 1

Mensagem

A mensagem tem de conter informação

Figura 20 - Mensagens de erro obtidas pela introdução de informação incorreta.

#### 4.2.5. Inserir Novas Mensagens De Voz

Na Figura 21 é possível observar a estrutura da página que permite introduzir novas mensagens de voz na base de dados de modo a que a aplicação possa utilizá-las. Esta página é apresentada

clicando no botão “Inserir Mensagem Voz”, nesta é possível ver o conteúdo da tabela **ED\_MENSAGEM\_VOZ** e introduzir novas mensagens.

De modo a introduzir mensagens novas é necessário:

- Introduzir o texto a ler e apresentar pela aplicação.
- Especificar a hora inicial a que a mensagem vai ser enunciada.
- Introduzir o intervalo de tempo entre mensagens e, conseqüentemente, o número de repetições da mensagem, caso seja necessário apresentar a mensagem mais do que uma vez. Neste caso o número de repetições também tem de ser maior do que 1.

O *ID* tal como nas mensagens de rodapé, não é introduzido pois é auto-incrementado.

A informação sobre quantas vezes uma mensagem ainda tem de ser lida é dada na coluna “Número de Mensagens Por Ler”. Uma mensagem foi lida o número de vezes pretendido quando o valor apresentado nesta coluna é 0, ou seja a mensagem encontra-se inativa. Uma mensagem inativa pode voltar a ficar ativa se o utilizador clicar no botão “repor repetições”, ao repor as repetições a mensagem vai assumir os valores que tinha quando foi inserida na base de dados. A ativação da mensagem implica apagar o valor da coluna “Hora da Última Chamada Realizada” e colocar o valor da coluna “Numero de Mensagens Por Apresentar” igual ao da coluna “Número De repetições”.

**Configurações da Aplicação**

Videos da aplicacao Inserir Nova Mensagem Apagar Mensagem Inserir Mensagem Voz

**Inserir Novas Mensagens de Voz**

Mensagens de Voz existentes na base de dados:

ID	Mensagem	Hora Inicial	Intervalo Entre Mensagens	Número De Repetições	Hora da Última Chamada Realizada	Número De Mensagens Por Apresentar	Repor Repetições
82	Lamentamos informar mas o hospital vai encerrar dentro de uma hora	00:16	00:03	2	00:19	0	<input type="button" value="Repor Repetições"/>

Preencha os campos em baixo de modo a inserir uma nova mensagem

Mensagem :

Hora Inicial:  
Hora  Minutos

Intervalo Entre Mensagens:  
Hora  Minutos

Número de Repetições:

Figura 21 - Página que permite inserir novas mensagens de voz na base de dados

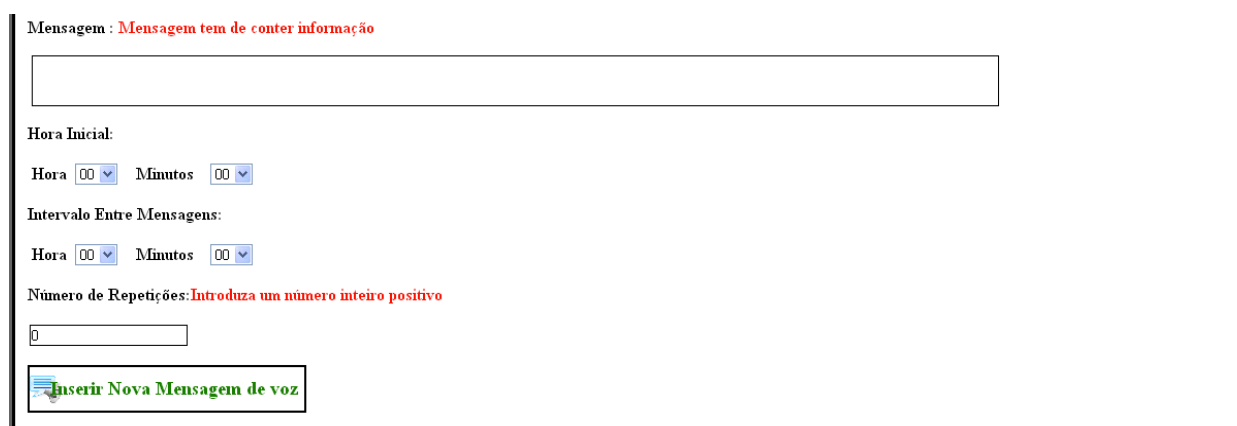
Todos os campos no preenchimento de uma mensagem de voz são de preenchimento obrigatório. Os impedimentos na introdução de uma nova mensagem de voz são:

- A mensagem não pode ser nula.
- O número de repetições tem de ser um número inteiro positivo (1,2,3,4...).

Caso o preenchimento seja incorreto são apresentadas as mensagens a vermelho presentes na Figura 22 e a mensagem não é inserida. É ainda considerado o seguinte caso especial:

- O intervalo entre mensagens é “00:00” e o número de repetições é maior do que 1 .

Neste caso a mensagem é inserida na base de dados com o intervalo entre mensagens “00:00” e número de repetições igual a 1.



Mensagem : Mensagem tem de conter informação

Hora Inicial:

Hora 00 Minutos 00

Intervalo Entre Mensagens:

Hora 00 Minutos 00

Número de Repetições: Introduza um número inteiro positivo

0

Inserir Nova Mensagem de voz

Figura 22 - Mensagens de erro obtidas pelo preenchimento incorreto dos campos

#### 4.2.6. Apagar Mensagens De Rodapé e De Voz

A necessidade de apagar mensagens pode ocorrer por exemplo pela introdução de mensagens com conteúdo que não é o pretendido ou simplesmente a mensagem já cumpriu o seu objectivo e não é mais necessária.

Na Figura 23 pode-se observar o aspecto da página que permite apagar mensagens de rodapé e voz. Para tal o utilizador tem de clicar no botão “Apagar Mensagem” que se encontra no topo da página. Na página são apresentadas duas tabelas: uma tabela corresponde às mensagens de rodapé cuja informação é retirada da tabela **ED\_MENSAGEM\_RODAPE** e outra tabela representa o conteúdo da tabela **ED\_MENSAGEM\_VOZ** que contém as mensagens de voz.

Nas tabelas cada linha corresponde a uma mensagem. Na apresentação das mensagens foi adicionado a cada linha um botão com o nome “Apagar Mensagem”. Este botão permite facilitar o

processo de eliminação de uma mensagem da base de dados, para tal basta clicar no botão correspondente à mensagem que se pretende eliminar. De modo a prevenir que se apaguem mensagens por um clique involuntário, foi introduzida uma caixa de mensagem de confirmação que permite confirmar se o utilizador pretende de facto eliminar a mensagem seleccionada(Figura 24).

### Configurações da Aplicação

Vídeos da aplicação
 Inserir Nova Mensagem
 Apagar Mensagem
 Inserir Mensagem Voz

#### Apagar Mensagem De Rodapé

Mensagens de rodapé existentes na base de dados:

ID	Mensagem	Hora Inicial	Hora Final	Estado	
62	Informa-se todos os utentes que amanhã as urgências funcionaram somente na parte de tarde	14:00	20:00	1	<input type="button" value="Apagar Mensagem"/>
63	Encontrou-se um telemóvel perto das casas de banho se é seu dirija-se a recepção	14:00	01:00	1	<input type="button" value="Apagar Mensagem"/>

#### Apagar Mensagem De Voz

Mensagens de Voz existentes na base de dados:

ID	Mensagem	Hora Inicial	Intervalo Entre Mensagens	Número De Repetições	Hora da Última Chamada Realizada	Número De Mensagens Por Apresentar	
123	Lamentamos informar mas o Serviço de Urgência vai encerrar dentro de uma hora	23:40	00:10	2		2	<input type="button" value="Apagar Mensagem"/>

Figura 23 - Página que permite apagar mensagens de rodapé e mensagens de voz

### Apagar Mensagem De Rodapé

### Apagar Mensagem De Voz

Mensagens de rodapé existentes na base de dados:

ID	Mensagem	Hora Inicial	Hora Final	Estado	
62	Informa-se todos os utentes que amanhã as urgências funcionaram somente na parte de tarde	14:00	20:00	1	<input type="button" value="Apagar Mensagem"/>
63	Encontrou-se um telemóvel perto das casas de banho se é seu dirija-se a recepção	14:00	01:00	1	<input type="button" value="Apagar Mensagem"/>

Tem a certeza que pretende apagar esta mensagem ?

Figura 24 - Mensagem de confirmação de eliminação de mensagem

# Capítulo 5

---

## 5. Conclusão

Os Serviços de Urgência representam cada vez mais um papel fulcral no Serviço Nacional de Saúde. Estes serviços são destinados a emergências e pacientes em situações urgentes, no entanto as pessoas por vezes recorrem ao serviço de urgência com problemas que não são urgentes e estes podem ser resolvidos por exemplo pela marcação duma consulta com o médico de família. Isto provoca uma grande afluência de pacientes no serviço e, conseqüentemente, problemas como longos tempos de espera e descontentamento dos pacientes.

A disponibilização de informações sobre os tempos de espera ajuda os pacientes no processo de espera e o próprio serviço de urgência a ter noção da qualidade dos seus serviços. Um serviço de urgência que se preocupe com os tempos de espera dos pacientes pode adotar medidas para os melhorar caso estes não sejam normais. É esta ideia que esta dissertação tem por base, e é a pensar no benefício dos pacientes e do serviço de urgência que foi desenvolvida a aplicação SUTE.

A aplicação SUTE possui um aspecto simples e apelativo o que favorece a compreensão da informação por parte dos pacientes. Apresenta informações sobre os tempos de espera médios e tempo que o último paciente esperou, ambos relativos à espera necessária para se realizar a triagem e a espera resultante para cada uma das 5 categorias diferentes do Sistema de Triagem de Manchester. As funcionalidades de apresentação de vídeos e mensagens de rodapé foram introduzidas com sucesso na aplicação.

A voz sintetizada por computador ou TTS utilizada realiza a chamada e leitura de mensagens de um modo bastante claro e perceptível. A sua utilização na aplicação SUTE representa uma boa opção.

A aplicação Web desenvolvida para gerir algumas funcionalidades da aplicação SUTE, apresenta uma interface de fácil utilização onde são apresentados os vídeos que são reproduzidos e podem ser inseridas, alteradas ou eliminadas mensagens de rodapé e de voz.

Neste sentido, os objectivos principais propostos foram cumpridos com sucesso e a aplicação encontra-se funcional e pronta a ser testada no serviço de urgência do centro hospitalar do Alto Ave.

## 5.1. Trabalho Futuro

As tecnologias de Informação estão sempre a avançar, portanto é sempre possível melhorar o trabalho realizado na medida em que surgem sempre novas tecnologias, melhoram-se tecnologias existentes ou até algumas acabam por cair em desuso.

A principal melhoria que a aplicação SUTE pode sofrer é ao nível do sistema TTS utilizado. Com a obtenção de um TTS que permita realizar a chamada e leitura de mensagens de um modo mais realista é possível melhorar muito a vantagem da utilização desta aplicação.

A introdução dos nomes dos pacientes sem acentuação na base de dados é outro aspecto que pode sofrer alterações. A possibilidade da introdução do nome acentuado na base de dados evita a utilização das tabelas **ED\_NOMES** e **ED\_APELIDOS** fazendo com que a aplicação se torne mais eficiente.

Poderia também ser adicionada à aplicação a capacidade de visualização de canais de televisão. Com isto os pacientes de um serviço de urgência poderiam simultaneamente ver televisão e estar atentos às informações disponibilizadas pela aplicação SUTE. Contudo de modo a tornar a visualização de canais de televisão possível, a informação que é apresentada atualmente na SUTE teria de ser readaptada.



# CAPÍTULO 6

---

## 6. Bibliografia

1. **Blobel, Bernd.** *eHealth: Combining Health Telematics, Telemedicine Biomedical Engineering and Bioinformatics to the Edge*. Berlin : Akademische Verlagsgesellschaft Aka GmbH, 2008. p. 235.
2. **Grupo de Trabalho das Urgências.** Reorganização das Urgências Hospitalares. *Hospitais EPE*. [Online] [Citação: 15 de Junho de 2011.] [http://www.hospitalsepe.min-saude.pt/Downloads\\_HEPE/producao\\_qualidade/livro%20urgencias%202006.pdf](http://www.hospitalsepe.min-saude.pt/Downloads_HEPE/producao_qualidade/livro%20urgencias%202006.pdf).
3. **Direcção-Geral da Saúde.** Direcção-Geral da Saúde. *Departamento da Qualidade na Saúde - Relatório do Sistema SIM-Cidadão 2010*. [Online] 2011. [Citação: 13 de Junho de 2011.] <http://www.dgs.pt/ms/8/default.aspx?pl=&id=5521&access=%200&codigono=001100150043AAAAAAA> AAAAA.
4. *A IMPLANTAÇÃO DE NOVA TECNOLOGIA: IMPLICAÇÃO NA EFICIÊNCIA DO TRABALHO NA UNIDADE DE PRONTO ATENDIMENTO DE UM HOSPITAL PÚBLICO DE URGÊNCIA E EMERGÊNCIA*<sup>1</sup>. **Santos, Daniela Soares, et al.** 15 de Dezembro de 2009, Revista Gestão Organizacional, Vol. 3.
5. **Ministério da Saúde Direcção-Geral da Saúde.** *Plano Nacional de Saúde 2004/2010*. Ministério da Saúde Direcção-Geral da Saúde. Lisboa : s.n., 2004.
6. **SONHO.** *Portal de Codificação e Dos GDH*. [Online] 30 de Janeiro de 2010. [Citação: 12 de Julho de 2011.] <http://portalcodgdh.min-saude.pt/index.php/SONHO>.
7. *Concept Matching between OpenEHR and SONHO*. **Cruz-Correia, Ricardo, Ávila, Carla e Chen, Rong.** 2009. Proceedings of the 1st International Workshop on Open Source in European Health care - OSEHC 2009.
8. **Ponce, Pedro e Teixeira, Jorge.** *Manual de Urgências e Emergências*. s.l. : Lidel, 2006. 9789727573653.
9. *Five-Level Triage System More Effective Than Three in Teartary Emergency Department*. **Tavares, D.** s.l. : Journal Emergency Nursing, October de 2002.
10. **Freitas, Paulo.** *Triagem no serviço de urgência: Grupo de Triagem de Manchester*. 2º. s.l. : BMJ Publishing Group, 2002.

11. **Cooke, Matthew, et al.** *Reducing Attendances and Waits in Emergency Departments A systematic review of present innovations*. Warwick Medical School The University of Warwick . Coventry : s.n., 2004.
12. *The Psychology of Waiting Lines.* **Maister, David H.** 1985.
13. **Microsoft.** Overview of the .NET Framework. *MSDN - Explore Windows, Web, Cloud and Windows Phone Software Development*. [Online] 2011. [Citação: 25 de Agosto de 2011.] <http://msdn.microsoft.com/en-us/library/zw4w595w%28v=VS.110%29.aspx>.
14. **MacDonald, Matthew.** *Beginning ASP.NET 4 in VB 2010*. [ed.] Ewan Buckingham. s.l. : Apress, 2010. 978-1-4302-2612-3.
15. **Tbat, Tbulan e Lam, Hoang.** *.NET Framework Essencials*. [ed.] Valerie Quercia e John Osborn. 3°. s.l. : O'Reilly, 2003. pp. 38-39. 0-596-00505-9.
16. **MacDonald, Matthew, Freeman, Adam e Szpuszta, Mario.** *Pro ASP.NET 4 in C# 2010*. 4°. New York : Apress, 2010. pp. 17-18,71. 978-1-4302-2530-0.
17. **Spaanjaars, Imar.** *Beginning ASP.NET 4: in C# and VB*. Indianapolis : Wiley, 2010. 978-0-470-50221-1.
18. **Chris Sells, Ian Griffiths.** *Programming Wpf: Building Windows Ui with Windows Presentation Foundation*. s.l. : O'Reilly Media, 2007.
19. **Stephens, Rod.** *WPF Programmer's Reference - Window Presentation Foundation With C# 2010 And .NET4*. s.l. : Wiley, 2010.
20. **Turner, Shane e Barksdale, Karl.** *HTML and JavaScript Basics*. 4°. s.l. : Course Technolog, Cengage Learning, 2011.
21. **Meyer, Eric A.** *CSS: the definitive guide*. [ed.] Tatiana Apandi. 3°. s.l. : O'Reilly, 2007. pp. 1-11. 978-0-596-52733-4.
22. **w3schools.** JavaScript Introduction. *w3schools.com*. [Online] [Citação: 8 de Janeiro de 2011.] [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp).
23. **Zakas, Nicholas C.** *Professional JavaScript for Web Developers*. 2°. s.l. : wrox, 2011.
24. **w3schools.** JavaScript Functions. *w3schools*. [Online] [Citação: 8 de Janeiro de 2011.] [http://www.w3schools.com/js/js\\_functions.asp](http://www.w3schools.com/js/js_functions.asp).
25. **Haan, Lex de, et al.** *Beginning Oracle SQL*. New York : Apress, 2009. pp. 17-19 , 25-39. 978-1-4302-7196-3.
26. **Ramalho, Jose.** *Oracle 10G*. s.l. : THOMSON, 2005. pp. 14,15.

27. **Syed, Mahbubur R., Chakrobartty, Shuvro e Bignall, Robert J.** Text - to - Speech Syntesis. [autor do livro] Margherita Pagani. *Encyclopedia of Multimedia Technology and Networking*. s.l. : Idea Group Reference, 2005, Vol. 1, pp. 957-963.
28. **Thierry, Dutoit.** *An Introduction to Text-to-Speech Synthesis*. s.l. : Kluwer Academic, 1997. pp. 30-32.
29. **Instituto dos Registos e do Notariado.** Vocábulos Admitidos ou Não Admitidos como Nomes Próprios. *Instituto dos Registos e do notariado*. [Online] [Citação: 3 de Fevereiro de 2011.] [http://www.irn.mj.pt/sections/irn/a\\_registral/registos-centrais/docs-da-nacionalidade/vocabulos-admitidos-e/downloadFile/file/Lista\\_de\\_nomes30-06-2011.pdf?nocache=1311428953.52](http://www.irn.mj.pt/sections/irn/a_registral/registos-centrais/docs-da-nacionalidade/vocabulos-admitidos-e/downloadFile/file/Lista_de_nomes30-06-2011.pdf?nocache=1311428953.52).
30. Anexo:Lista de prenomes portugueses. *Wikipédia*. [Online] [Citação: 4 de Fevereiro de 2011.] [http://pt.wikipedia.org/wiki/Anexo:Lista\\_de\\_prenomes\\_portugueses](http://pt.wikipedia.org/wiki/Anexo:Lista_de_prenomes_portugueses).
31. **Ferrão, Lúcio** . Ferrão.org. *ferrao.org - onomástica portuguesa*. [Online] 4 de Fevereiro de 2011. <http://ferrao.org/onomastica/>.
32. wikipedia. *Lista de apelidos de família da língua portuguesa*. [Online] 09 de Dezembro de 2010. [Citação: 03 de Fevereiro de 2011.] [http://pt.wikipedia.org/wiki/Anexo:Lista\\_de\\_apelidos\\_de\\_fam%C3%ADlia\\_da\\_l%C3%ADngua\\_portuguesa](http://pt.wikipedia.org/wiki/Anexo:Lista_de_apelidos_de_fam%C3%ADlia_da_l%C3%ADngua_portuguesa).
33. Solução de Suporte para Marketing. *SPIE*. [Online] [http://www.bbg.pt/sites\\_off/SPIE\\_01/PDFs/100\\_apelidos.pdf](http://www.bbg.pt/sites_off/SPIE_01/PDFs/100_apelidos.pdf).

# Anexos

---

## Anexo A – Código SQL para a criação das tabelas ED\_MENSAGEM\_VOZ, ED\_MENSAGEM\_RODAPE, ED\_NOMES e ED\_APELIDOS

```
CREATE TABLE ED_MENSAGEM_RODAPE
  ("ID" NUMBER (9,0),
   "MENSAGEM" VARCHAR2 (500 BYTE),
   "HORA_INI" VARCHAR2 (5 CHAR),
   "HORA_FIM" VARCHAR2 (5 CHAR),
   "ESTADO" CHAR (1 BYTE)
);
```

Figura A.1 – Código SQL para a criação da tabela ED.MENSAGEM

```
CREATE SEQUENCE seq_ED_MENSAGEM_RODAPE_id START WITH 1
```

Figura A.2 – Código SQL de criação da sequência "seq\_ED\_MENSAGEM\_RODAPE\_id" para a tabela ED\_MENSAGEM\_RODAPE

```
CREATE TRIGGER ED_MENSAGEM_RODAPE_ID_TRIGGER
  Before insert on ED_MENSAGEM_RODAPE
  For each row
  Begin
    Select seq_ED_MENSAGEM_RODAPE_id.nextval into: new.id from
  dual;
```

Figura A.3 Código SQL de criação do trigger "ED\_MENSAGEM\_RODAPE\_ID\_TRIGGER" para a tabela ED\_MENSAGEM\_RODAPE

```

CREATE TABLE ED_MENSAGEM_VOZ
  ("ID" NUMBER (*, 0),
   "MENSAGEM" VARCHAR2 (500 BYTE),
   "ESTADO" NUMBER (*, 0),
   "HORA_INI" VARCHAR2 (5 CHAR),
   "ULT_HORA_ACT" VARCHAR2 (5 CHAR),
   "INTERVAL" VARCHAR2 (5 CHAR),
   "NREP" NUMBER (*, 0)
);

```

Figura A.4 - Código SQL para a criação da tabela ED.MENSAGEM

```

CREATE SEQUENCE seq_ED_MENSAGEM_VOZ_id START WITH 1 INCREMENT

```

Figura A.5 - Código SQL de criação da sequência "seq\_ED\_MENSAGEM\_VOZ\_id" para a tabela ED\_MENSAGEM\_RODAPE

```

CREATE TRIGGER ED_MENSAGEM_VOZ_ID_TRIGGER
Before insert on ED_MENSAGEM_VOZ
For each row
Begin
  Select seq_ED_MENSAGEM_VOZ_id.nextval into: new.id from
dual;

```

Figura A.6 - Código SQL de criação do trigger "ED\_MENSAGEM\_VOZ\_ID\_TRIGGER" para a tabela ED\_MENSAGEM\_RODAPE

```

CREATE TABLE ED_NOMES
  ("NOME" VARCHAR2 (100 BYTE),
   "NOMEACENT" VARCHAR2 (100 BYTE)
);

```

Figura A.7 - Código SQL para a criação da tabela ED.NOMES

```
CREATE TABLE ED_APELIDOS
  ("APELIDO" VARCHAR2 (100 BYTE),
  "APELIDOACENT" VARCHAR2 (100 BYTE)
);
```

Figura A.8 – Código SQL para a criação da tabela ED.APELIDOS