

# SISTEMA DE DETEÇÃO DE ARRITMIAS

Pedro Miguel Gomes Cerqueira



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2013



Este relatório satisfaz, parcialmente, os requisitos que constam na Ficha da Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Pedro Miguel Gomes Cerqueira, N° 1080518, 1080518@isep.ipp.pt

Orientação científica: Ramiro de Sousa Barbosa, RSB@isep.ipp.pt

Coorientação científica: Lino Manuel Baptista Figueiredo, LBF@isep.ipp.pt



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

7 de novembro de 2013



*Aos meus Pais*



## *Agradecimentos*

Ao longo deste último semestre em que estive envolvido na elaboração da dissertação de mestrado tive o prazer de contar com diversas pessoas que, direta ou indiretamente, contribuíram para a elaboração do projeto a que este documento se refere.

Em primeiro lugar, quero agradecer ao meu orientador, Eng.º Ramiro Barbosa, por todo o empenho, disponibilidade, valiosa ajuda e orientação, essencial à obtenção deste relatório. Sinto-me grato também pelas críticas, correções e sugestões relevantes feitas durante a orientação.

Ao Eng.º Lino Figueiredo pela disponibilidade e pela valiosa ajuda em apoio bibliográfico que contribuiu para a realização desta dissertação.

Aos meus pais por me proporcionarem a possibilidade de obter formação a nível superior, além de toda a compreensão demonstrada ao longo destes anos em que estive numa cidade distante.

À Joana que, sem dúvida nenhuma, foi a pessoa que mais me acompanhou nas várias etapas que contemplam a realização desta dissertação tendo criado todas as condições necessárias para uma boa predisposição psicológica.

Por fim quero agradecer a todos os meus amigos tanto do Porto como de Viana do Castelo, que estiveram sempre presentes em momentos onde não faltou a boa disposição.



## *Resumo*

Esta dissertação apresenta o trabalho realizado no âmbito da unidade curricular de Tese/Dissertação (TEDI), do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores no ramo de Automação e Sistemas. O principal objetivo desta dissertação consiste no desenvolvimento de um sistema que permita efetuar a deteção de um determinado número de anomalias num sinal eletrocardiográfico.

O coração é um dos órgãos mais importantes do corpo humano. É ele que recebe e bombeia o sangue pelo organismo. Isto é, recebe sangue pobre em oxigénio, encaminha-o para os pulmões onde será enriquecido em oxigénio. O sangue enriquecido em oxigénio é então encaminhado novamente para o coração que será enviado para todas as partes do corpo humano.

O eletrocardiograma desempenha um papel fundamental de modo a diagnosticar eventuais anomalias no correto funcionamento do coração. Estas anomalias podem dever-se a diversos fatores como tabaco, colesterol, pressão sanguínea alta ou diabetes entre outros. As anomalias associadas ao ritmo cardíaco são denominadas de arritmias. As arritmias são fundamentalmente originadas pela alteração da frequência ou do ritmo cardíaco.

Utilizando a lógica difusa, pretendeu-se desenvolver um sistema que fizesse a identificação de um determinado número de tipos de batimentos entre os quais: o bloqueio do ramo esquerdo (LBBB), bloqueio do ramo direito (RBBB), contração prematura ventricular (VPC) e contração prematura auricular (APC).

Todos os desenvolvimentos efetuados, a nível de programação, são neste documento relatados de forma a constituírem um possível guia para a utilização deste tipo de sistemas. Mais ainda, descrevem-se nele toda a pesquisa efetuada e as alternativas de desenvolvimento selecionadas.

O Sistema de Deteção de Arritmias (SDA) desenvolvido mostrou-se eficaz desde que o utilizador consiga identificar corretamente os parâmetros que lhe são pedidos. A interface gráfica desenvolvida permitiu também uma maior facilidade durante a análise do sinal eletrocardiográfico.

### ***Palavras-Chave***

Coração, Arritmias, ECG, Derivações eletrocardiográficas, Lógica difusa, Sistemas difusos, MATLAB, LBBB, RBBB, APV, VPC.

## *Abstract*

This thesis presents the work carried out during the course of Thesis/Dissertation (TEDI) of the 2nd year of the Master in Electrical and Computer Engineering in the field of Automation and Systems. The main objective of this dissertation is to develop a system that can be able to detect a certain number of anomalies in an ECG signal.

The heart is one of the most important organs of the human body. It's who receives and pumps blood through the body. First the heart receives the blood poor in oxygen and then sent it to the lungs. After the blood enriched in oxygen goes again to the heart and then it's sent to all parts of the human body.

The ECG plays a key role in order to diagnose any abnormalities in the correct functioning of the heart. These abnormalities may be due to several factors such as smoking, cholesterol, high blood pressure and diabetes among others. The abnormalities associated with cardiac rhythm are called arrhythmias. Arrhythmias are mainly caused by the change in the frequency or heart rate.

Using fuzzy logic, it was intended to develop a system that would make the identification of different heartbeat types including the left bundle branch block (LBBB), right bundle branch block (RBBB), ventricular premature contraction (VPC) and atrial premature contraction (APC) .

All developments made at the level of programming are reported here in order to constitute a possible guide for the use of such systems. Moreover, it describes all the research done and development alternatives selected.

The Arrhythmias Detection System (SDA) developed is effective in a way that the user can correctly identify the requested parameters. The GUI developed also allows greater insight in the analysis of the electrocardiographic signal.

***Keywords***

Heart, Arrhythmias, ECG, ECG leads, Fuzzy Logic, Fuzzy Systems, MATLAB, LBBB, RBBB, APC, VPC.

# Índice

<b>AGRADECIMENTOS</b> .....	<b>I</b>
<b>RESUMO</b> .....	<b>III</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>ÍNDICE</b> .....	<b>VII</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>IX</b>
<b>ÍNDICE DE TABELAS</b> .....	<b>XIII</b>
<b>ACRÓNIMOS</b> .....	<b>XV</b>
<b>1. INTRODUÇÃO</b> .....	<b>1</b>
1.1. OBJETIVOS .....	2
1.2. CALENDARIZAÇÃO .....	3
1.3. ORGANIZAÇÃO DO RELATÓRIO .....	3
<b>2. O CORAÇÃO E O ELETROCARDIOGRAMA</b> .....	<b>5</b>
2.1. CORAÇÃO .....	6
2.2. ELETROCARDIOGRAMA.....	17
<b>3. SISTEMAS DIFUSOS</b> .....	<b>31</b>
3.1. PERSPETIVA HISTÓRICA.....	31
3.2. CONCEITOS BÁSICOS.....	32
3.3. VARIÁVEIS LINGUÍSTICAS .....	33
3.4. CONJUNTOS DIFUSOS .....	35
3.5. PROPRIEDADES DOS CONJUNTOS DIFUSOS .....	40
3.6. OPERAÇÕES BÁSICAS SOBRE CONJUNTOS DIFUSOS.....	41
3.7. SISTEMAS DE INFERÊNCIA DIFUSA .....	44
3.8. APLICAÇÕES .....	51
<b>4. ESTUDO E ANÁLISE DOS COMPONENTES DO SDA</b> .....	<b>53</b>
4.1. FUNCIONAMENTO PRETENDIDO .....	53
4.2. MIT-BIH <i>ARRHYTHMIA DATABASE</i> .....	54
4.3. <i>PHYSIOBANK ATM</i> .....	60
4.4. <i>PHYSIO TOOLKIT</i> .....	61
4.5. REMOÇÃO DO RUÍDO DO SINAL ELETROCARDIOGRÁFICO .....	62
4.6. ALGORITMO DE IDENTIFICAÇÃO DO COMPLEXO QRS .....	63
4.7. SISTEMA DIFUSO.....	64
4.8. <i>TOOLBOX</i> “FUZZY LOGIC” DO MATLAB .....	71

<b>5. DESENVOLVIMENTO DOS COMPONENTES DO SDA.....</b>	<b>75</b>
5.1. OBTENÇÃO DOS REGISTOS .....	76
5.2. REPRESENTAÇÃO GRÁFICA E GRAVAÇÃO DOS DADOS .....	77
5.3. REMOÇÃO DO RÚIDO DO SINAL .....	80
5.4. IDENTIFICAÇÃO DO COMPLEXO QRS E OBTENÇÃO DAS VARIÁVEIS PARA O SISTEMA DIFUSO .....	82
5.5. DETERMINAÇÃO DO TIPO DE BATIMENTO .....	85
5.6. INTERFACE GRÁFICA .....	91
5.7. TESTES EFETUADOS AO SISTEMA.....	94
<b>6. CONCLUSÕES .....</b>	<b>101</b>
6.1. OBJETIVOS REALIZADOS.....	101
6.2. LIMITAÇÕES DO SISTEMA E PERSPETIVAS DE DESENVOLVIMENTO.....	102
6.3. APRECIÇÃO FINAL.....	103
<b>REFERÊNCIAS DOCUMENTAIS.....</b>	<b>105</b>
<b>ANEXO A. TOOLBOX “FUZZY LOGIC” DO MATLAB .....</b>	<b>109</b>
<b>ANEXO B. INFORMAÇÕES ÚTEIS SOBRE AS FUNÇÕES E A INTERFACE GRÁFICA DESENVOLVIDA .....</b>	<b>127</b>

## Índice de Figuras

Figura 1	Estrutura do coração e fluxo sanguíneo pelas diferentes câmaras [4].....	6
Figura 2	Aspeto do músculo cardíaco [4] .....	8
Figura 3	Membrana Celular [6].....	10
Figura 4	Potenciais de ação de uma célula nervosa e de uma fibra do músculo cardíaco – A. Fases do potencial de ação sem e com marca-passo – B e C respetivamente [5] .....	11
Figura 5	Diferentes formas de onda para as várias células especializadas do coração [9] .....	12
Figura 6	Sistema especializado de excitação e condução do coração [4] .....	13
Figura 7	Nódulo AV e intervalos de tempo a partir do nódulo sino-auricular [4].....	14
Figura 8	Transmissão do impulso cardíaco pelo coração em frações de segundo [4] .....	15
Figura 9	Eventos do ciclo cardíaco – pressão auricular esquerda, pressão ventricular esquerda, pressão aórtica, volume ventricular, eletrocardiograma e fonocardiograma [4] .....	16
Figura 10	Eletrocardiograma [4] .....	17
Figura 11	Despolarização (A e B) e repolarização (C e D) de uma fibra do músculo cardíaco e registo da respetiva onda gerada e captada através dos elétrodos [4] .....	20
Figura 12	Relação entre o potencial de ação de uma fibra muscular ventricular e o ECG registado simultaneamente [4] .....	21
Figura 13	Grelha eletrocardiográfica [11] .....	22
Figura 14	Derivações bipolares dos membros e respetivos registos eletrocardiográficos [4].....	24
Figura 15	Derivações pré-cordiais e respetivos registos eletrocardiográficos [4].....	26
Figura 16	Registos eletrocardiográficos das derivações unipolares aumentadas dos membros aVR, aVL e aVF [4] .....	27
Figura 17	Exemplos de ritmos cardíacos anormais [5].....	29
Figura 18	Da esquerda para a direita: Aristóteles (A), George Boole (B), Bertrand Russell (C), Jan Lukasiewicz (D) e Lofti Zadeh (E). .....	32
Figura 19	Variável linguística <i>ritmo cardíaco</i> e respetivos conjuntos difusos .....	34
Figura 20	Conjuntos clássicos (esquerda) e conjuntos difusos (direita) que caracterizam por exemplo a temperatura de uma sala [20] .....	36
Figura 21	Tipos de funções de pertença [21] .....	38
Figura 22	Tipos de funções de pertença [21] .....	40
Figura 23	Operações básicas sobre conjuntos difusos. A – União, B – Intersecção e C – Complemento [24] .....	44
Figura 24	Diagrama de um sistema de inferência difusa [23].....	44
Figura 25	Método do centro de massa .....	49
Figura 26	Método do centro da soma das áreas .....	50
Figura 27	Método da média dos máximos .....	51

Figura 28	Excerto 100 .....	55
Figura 29	Excerto 106 .....	56
Figura 30	PhysioBank ATM .....	60
Figura 31	<i>Output</i> gerado através dos parâmetros da Figura 30.....	60
Figura 32	Espetro de potência relativa do ECG, do complexo QRS, das ondas P e T, do ruído muscular e dos artefactos em movimento [36] .....	63
Figura 33	Diagrama representativo do algoritmo de deteção do complexo QRS .....	64
Figura 34	Parâmetros do eletrocardiograma [39].....	64
Figura 35	Gráficos dos intervalos de valores das características para cada tipo de batimento [39] 67	
Figura 36	Integração da <i>toolbox fuzzy logic</i> com outras ferramentas [41] .....	72
Figura 37	<i>FIS Editor</i> .....	72
Figura 38	Editor de funções de pertinência .....	73
Figura 39	Editor de regras .....	74
Figura 40	Diagrama representativo do sistema .....	76
Figura 41	Exemplo da obtenção do registo 100 .....	77
Figura 42	<i>Output</i> da execução da função <i>PC41_obtencao</i> .....	77
Figura 43	Exemplo da representação do sinal 100 .....	78
Figura 44	<i>Output</i> da execução da função <i>PC42_representacao</i> .....	78
Figura 45	Exemplo da representação do sinal 100 com anotações .....	79
Figura 46	<i>Output</i> da execução da função <i>PC43_rep_anot</i> .....	79
Figura 47	Exemplo da gravação do registo 123 .....	80
Figura 48	<i>Output</i> da execução da função <i>PC44_dat2mat</i> .....	80
Figura 49	Exemplo da remoção de ruído de um sinal .....	81
Figura 50	<i>Output</i> da execução da função <i>PC45_rem_ruído</i> .....	81
Figura 51	Exemplo da identificação do complexo QRS e obtenção das variáveis de entrada para o sistema difuso .....	83
Figura 52	Identificação do complexo QRS.....	84
Figura 53	Implementação do sistema difuso.....	86
Figura 54	Conjuntos difusos da variável de entrada <i>QRSdur</i> .....	86
Figura 55	Conjuntos difusos da variável de entrada <i>QTPint</i> .....	87
Figura 56	Conjuntos difusos da variável de entrada <i>ratioRR</i> .....	87
Figura 57	Conjuntos difusos da variável de entrada <i>areaRST</i> .....	88
Figura 58	Conjuntos difusos da variável de saída <i>TipoBatimento</i> .....	88
Figura 59	Regras do sistema difuso.....	89
Figura 60	Exemplo da classificação do tipo de batimento.....	91
Figura 61	Interface gráfica desenvolvida.....	91
Figura 62	Obter ou abrir ficheiros .....	92
Figura 63	Mostrar sinal.....	92

Figura 64	Identificar parâmetros e calcular tipo de batimento.....	93
Figura 65	Tipo de batimento e batimentos/minuto.....	93
Figura 66	Representação gráfica .....	94
Figura 67	Barra de estado .....	94
Figura 68	Identificação dos parâmetros para o caso NORM .....	95
Figura 69	Interface gráfica no caso NORM.....	96
Figura 70	Identificação dos parâmetros para o caso RBBB.....	97
Figura 71	Interface gráfica no caso RBBB .....	97
Figura 72	Identificação dos parâmetros para o caso VPC .....	98
Figura 73	Interface gráfica no caso VPC .....	99
Figura 74	Identificação dos parâmetros para o caso APC .....	100
Figura 75	Interface gráfica no caso APC .....	100
Figura 76	Janela principal da <i>toolbox fuzzy logic</i> .....	110
Figura 77	Escolha do tipo de controlador .....	110
Figura 78	Importar e exportar controladores difusos.....	111
Figura 79	Menu <i>Edit</i> do editor FIS.....	111
Figura 80	Adição de uma variável linguística de entrada .....	112
Figura 81	Editor das funções de pertença .....	112
Figura 82	Menu <i>Edit</i> do editor de funções de pertença .....	113
Figura 83	Editor de regras.....	114
Figura 84	<i>Rule Viewer</i> .....	115
Figura 85	<i>Surface Viewer</i> .....	116
Figura 86	Editor ANFIS .....	117
Figura 87	Interface gráfica para efectuar o <i>Clustering</i> .....	119
Figura 88	<i>Fuzzy Logic Library</i> .....	122
Figura 89	Parâmetros do <i>Fuzzy Logic Controller (A)</i> e do <i>Fuzzy Logic Controller with Ruleviewer (B)</i> .....	123
Figura 90	Sub-biblioteca para as funções de pertença.....	124
Figura 91	Exemplo prático da implementação de blocos de funções de pertença .....	125



## *Índice de Tabelas*

Tabela 1	Calendarização do projeto.....	3
Tabela 2	Quadro resumo do sinal ECG [5] .....	19
Tabela 3	Exemplos de variáveis linguísticas com valores típicos .....	35
Tabela 4	Anotações e respetivos significados .....	57
Tabela 5	Sumário do número de batimentos e o respetivo tipo de batimento .....	58
Tabela 6	Descrição das características do eletrocardiograma.....	65
Tabela 7	Intervalo de valores das características para o registo 103.....	66
Tabela 8	Intervalo de valores das características para o caso NORM .....	66
Tabela 9	Intervalo de valores das características para cada tipo de batimento .....	66
Tabela 10	Tabela de regras.....	70
Tabela 11	Regras do sistema difuso.....	70
Tabela 12	Quadro resumo de todas as funções desenvolvidas .....	127



## *Acrónimos*

ADC	–	<i>Analog-to-Digital Converter</i>
ANFIS	–	<i>Adaptive Neuro-Fuzzy Inference System</i>
APC	–	<i>Premature Atrial Contractions</i>
ATM	–	<i>Automated Teller Machine</i>
AV	–	<i>Auriculoventricular</i>
BIH	–	<i>Boston's Beth Israel Hospital</i>
BPM	–	<i>Batimento Por Minuto</i>
CD-R	–	<i>Compact Disc - Recordable</i>
ECG	–	<i>Eletrocardiograma</i>
EMG	–	<i>Eletromiograma</i>
FIS	–	<i>Fuzzy Inference System</i>
GPL	–	<i>General Public License</i>
GUIDE	–	<i>Graphical User Interface Development Environment</i>
LBBB	–	<i>Left Bundle Branch Block</i>
LD	–	<i>Lógica Difusa</i>
MATLAB	–	<i>MATrix LABoratory</i>
MIT	–	<i>Massachusetts Institute of Technology</i>
ms	–	<i>Milissegundos</i>

mV	– Milivolts
RBBB	– <i>Right Bundle Branch Block</i>
SA	– Sino-auricular
SDA	– Sistema de Detecção de Arritmias
TEDI	– Tese/Dissertação
VPC	– <i>Premature Ventricular Contraction</i>

# 1. INTRODUÇÃO

O coração é um dos órgãos mais importantes do corpo humano. É através dele que o resto do organismo recebe o sangue necessário ao seu correto funcionamento. De uma forma sucinta, o lado direito do coração recebe o sangue pobre em oxigênio que irá ser bombeado para os pulmões. Nos pulmões o sangue é oxigenado e devolvido ao lado esquerdo do coração que bombeará o sangue rico em oxigênio para todas as partes do organismo. Como qualquer organismo do corpo humano, o coração encontra-se sujeito ao aparecimento de anomalias que irão alterar o seu funcionamento. Estas anomalias podem ser tão significativas que podem levar o paciente à morte. De modo a prevenir tais anomalias, existem diversos métodos de diagnóstico tanto invasivos como não invasivos [1]. Entre esses métodos encontram-se: o eletrocardiograma, a prova de esforço, o estudo eletrofisiológico e o ecocardiograma.

O eletrocardiograma fornece um sinal cardíaco que é de grande utilidade para os cardiologistas uma vez que, dado o seu conhecimento nessa área, conseguem determinar diversas condições do doente analisando visualmente o sinal. Neste caso, é interessante o desenvolvimento de um sistema que permita analisar um batimento presente no eletrocardiograma de modo a verificar se houve ocorrência de alguma anomalia. Este sistema serviria de apoio ao cardiologista ou até poderia ser usado pelo paciente em casa permitindo assim efetuar uma maior prevenção à saúde do mesmo. Relativamente ao

médico, pode utilizar o sistema como uma ferramenta de ajuda para o reconhecimento de anomalias. Por sua vez, o paciente pode também possuir um sistema móvel que lhe indique o tipo de batimento que visualiza. Claro está que este sistema não dispensaria a consulta de um especialista.

Na tomada de decisão do sistema, isto é, o momento em que são ou não identificadas anomalias num batimento do sinal eletrocardiográfico, a ferramenta de controlo a usar pode ser a lógica difusa. Este tipo de lógica é bastante usado em situações onde é necessário quantificar expressões como muito grande, pequeno e médio entre muitas outras. Para isso é necessário alguém que conheça muito bem o sistema e seja capaz de dizer, por exemplo, o que acontece caso um batimento possua esta forma ou o que significam estes dois ciclos de ondas com espaçamentos diferentes. Facilmente se deduz que a criação de um sistema que use a lógica difusa pode ser de fácil desenvolvimento desde que o projetista do sistema conheça muito bem o mesmo. Desta forma a lógica difusa permite traduzir as expressões usadas pelos seres humanos para um formato numérico de fácil manipulação pelos computadores.

## **1.1. OBJETIVOS**

É pretensa do corrente do trabalho o desenvolvimento de um Sistema de Detecção de Arritmias (SDA) que permitirá, usando a lógica difusa, efetuar a deteção de anomalias num dado sinal eletrocardiográfico. Dada a complexidade adjacente a este objetivo, optou-se por efetuar a sua divisão em etapas de menor complexidade:

- Obtenção e manuseamento dos sinais eletrocardiográficos;
- Desenvolvimento de um sistema difuso capaz de detetar um conjunto de anomalias eletrocardiográficas;
- Desenvolvimento de uma interface gráfica que implemente o sistema difuso.

Tendo em conta a realização destas etapas, delineou-se um conjunto de passos a efetuar. Inicialmente o estudo do coração assim como o estudo de sinais eletrocardiográficos para que, numa fase seguinte se estudem os sistemas difusos. Seguidamente espera-se que já existam condições para uma escolha consciente das melhores formas de implementar o projeto pretendido, sobre as quais se deverá conduzir uma análise detalhada. Segue-se a aplicação das opções definidas, construindo um protótipo do sistema real onde se espera que incidam os testes necessários de validação. Concluídas estas tarefas será finalmente

possível a determinação da versão definitiva do projeto e o registo das considerações finais sobre este.

## 1.2. CALENDARIZAÇÃO

Na Tabela 1 é apresentada a calendarização referente ao trabalho desenvolvido durante a elaboração deste projeto.

**Tabela 1 Calendarização do projeto**

Tarefa	Duração	Fev 2013		Mar 2013			Abr 2013			Mai 2013			Jun 2013			Jul 2013			Ago 2013			Set 2013										
		17-2	24-2	3-3	10-3	17-3	24-3	31-3	7-4	14-4	21-4	28-4	5-5	12-5	19-5	26-5	2-6	9-6	16-6	23-6	30-6	7-7	14-7	21-7	28-7	4-8	11-8	18-8	25-8	1-9	8-9	15-9
1 Estudo do coração e do eletrocardiograma	34d	■																														
2 Estudo de sistemas difusos	21d							■																								
3 Obtenção e tratamento dos sinais da base de dados	19d										■																					
4 Desenvolvimento do Sistema de Detecção de Arritmias	92d													■																		
5 Intervalo para férias	13d																			■												
6 Fase de testes	6d																			■												
7 Conclusões e futuros desenvolvimentos	2d																			■												
8 Elaboração do relatório final	50d													■																		

## 1.3. ORGANIZAÇÃO DO RELATÓRIO

Este relatório é composto por 6 capítulos: Introdução, O coração e o eletrocardiograma, Sistemas difusos, Estudo e análise dos componentes do SDA, Desenvolvimento do SDA e Conclusões.

Neste Capítulo que se encerra foi apresentada uma breve contextualização sobre o coração e os sistemas difusos, os objetivos do presente projeto, a calendarização do trabalho desenvolvido e a organização do relatório.

No Capítulo 2, “O coração e o eletrocardiograma”, numa primeira fase é apresentado o funcionamento do coração e as características do sistema excitatório para que, numa segunda fase, seja exposto o conceito de eletrocardiograma e as suas particularidades.

O Capítulo seguinte, “Sistemas difusos”, tem como objetivo apresentar os conceitos básicos da lógica inerente a estes sistemas, a lógica difusa. As variáveis linguísticas, os conjuntos difusos e as suas operações básicas são alguns dos temas que são tratados neste

capítulo. Este capítulo é terminado com a apresentação de algumas aplicações onde são utilizados os sistemas difusos.

Por sua vez, o Capítulo 4, aborda a fase preparatória para o desenvolvimento do SDA. É neste capítulo que se apresenta o estudo e a análise dos componentes usados no sistema. Nele é apresentada a base de dados utilizada assim como ferramentas que permitam a obtenção, manipulação e processamento dos sinais eletrocardiográficos. É ainda apresentado o sistema difuso usado neste projeto assim como a ferramenta de software usada para a criação do mesmo.

O penúltimo capítulo, Capítulo 5, contém o desenvolvimento do SDA, apresentando os vários componentes produzidos de modo a atingir os objetivos traçados. Nele encontra-se presente as várias funções criadas, assim como a interface gráfica desenvolvida. Por fim são também apresentados alguns testes realizados ao sistema.

Por fim, o 6º e último capítulo reúne as principais conclusões retiradas ao longo do desenvolvimento deste projeto e dos resultados obtidos; são ainda perspetivados futuros desenvolvimentos.

## 2. O CORAÇÃO E O ELETROCARDIOGRAMA

O coração humano possui o tamanho de um punho mas é o músculo mais forte do corpo humano. Os batimentos do coração aparecem no útero muito antes do nascimento, tipicamente entre 21 a 28 dias após a concepção. Ao longo de uma vida (em média 70 anos) o coração efetua mais ou menos 2,5 bilhões de batimentos o que equivale em média a uns 100.000 batimentos por dia. Em cada batimento, o coração bombeia o sangue para todo o corpo do ser humano. O número de batimentos por minuto é aproximadamente 70 e este número pode dobrar durante o exercício ou em momentos de extrema emoção [2].

As doenças associadas ao coração são denominadas de doenças cardiovasculares. Este tipo de doenças é a causa de morte número um no mundo. Estima-se que, em 2008, 17,3 milhões de pessoas morreram de doenças cardiovasculares, o que representa 30% do número total de mortes. Esta doença tem uma maior incidência nos países de baixo e médio rendimento: 80% das mortes totais ocorrem nestes países e acontecem com a mesma probabilidade tanto em mulheres como em homens. Em 2030, vão morrer mais de 23 milhões de pessoas devido a doenças cardiovasculares [2]. A maioria destas doenças pode ser prevenida ao diminuir os fatores de risco como o tabaco, alimentações não saudáveis, obesidade, inatividade física, pressão arterial elevada, diabetes e lípidos saturados [3].

Uma das ferramentas de diagnóstico para as doenças do coração é o eletrocardiograma (ECG). O ECG permite registar a atividade elétrica do coração e efetuar a sua análise de modo a detetar anomalias cardíacas. Neste capítulo são apresentados os principais conceitos sobre o sistema cardiovascular e o sinal ECG. A grande maioria dos dados apresentados na subsecção 2.1 são retirados do livro Textbook of Medical Physiology [4].

## 2.1. CORAÇÃO

O coração é um dos órgãos mais importantes do corpo humano. Como já fora referido anteriormente, a sua principal função consiste em bombear o sangue para todos os órgãos do corpo humano. Tendo em conta o seu estudo, o coração pode dividir-se em duas partes distintas e identificadas da seguinte forma: *coração direito* e *coração esquerdo*. Na Figura 1 podem identificar-se estas duas partes onde do lado esquerdo do leitor se encontra representado o *coração direito* e do lado direito o *coração esquerdo*. O *coração direito* recebe o sangue pobre em oxigénio e este é bombeado para os pulmões (circulação pulmonar) de forma a ser enriquecido em oxigénio. Por sua vez, o *coração esquerdo* recebe o sangue rico em oxigénio vindo dos pulmões e vai bombeá-lo pelos órgãos periféricos do corpo (circulação sistémica).

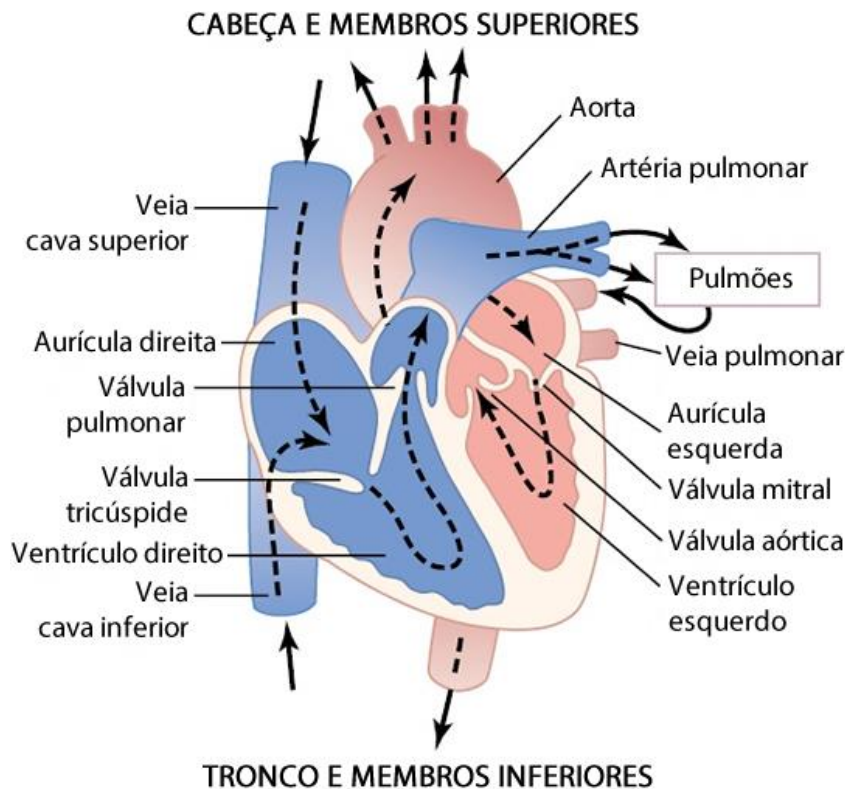


Figura 1 Estrutura do coração e fluxo sanguíneo pelas diferentes câmaras [4]

Tanto o *coração direito* como o *coração esquerdo* são compostos por duas câmaras: aurícula e ventrículo. A aurícula tem como principal objetivo armazenar sangue e consiste também numa via de entrada para o ventrículo. Além disto, também possui a capacidade de bombear fracamente o sangue, de modo a ajudá-lo a circular até ao ventrículo. A aurícula direita e a aurícula esquerda são separadas pelo septo interauricular. Por sua vez, o ventrículo é a bomba principal que impulsiona o sangue para os pulmões no caso do *coração direito* e para os órgãos periféricos no caso do *coração esquerdo*. O ventrículo direito e o ventrículo esquerdo são separados pelo septo interventricular.

Além das câmaras atrás referidas, tanto o *coração direito* como o *esquerdo* possuem válvulas cardíacas. Estas válvulas podem ser divididas em dois tipos: válvulas auriculoventriculares e válvulas semilunares. As válvulas auriculoventriculares (AV) são as válvulas tricúspide e mitral; têm como função impedir o refluxo de sangue dos ventrículos para as aurículas durante a sístole<sup>1</sup>. Por sua vez, as válvulas semilunares são constituídas pela válvula aórtica e pela válvula pulmonar cujo objetivo é impedir o refluxo de sangue das artérias aorta e pulmonar, respetivamente, para os ventrículos durante a diástole<sup>2</sup>. O conjunto sístole mais diástole forma o ciclo cardíaco.

Ainda na Figura 1, as setas a tracejado indicam o sentido do fluxo sanguíneo. A veia cava superior e inferior recebe o sangue pobre em oxigénio das extremidades superiores e inferiores respetivamente. O sangue é encaminhado para os pulmões através da artéria pulmonar e, depois de enriquecido, volta a entrar pelas veias pulmonares e será encaminhado para os órgãos superiores e inferiores pela aorta.

O coração como órgão que é, também necessita de ser nutrido. Essa função é assegurada pelas artérias coronárias direita e esquerda. O sangue passa para estas artérias através das ósteas coronárias (dois orifícios que se encontram no início da aorta) e isso acontece durante a diástole. A não oxigenação dos tecidos do coração por falta de irrigação de sangue é denominada de isquemia e é uma das principais causas de morte.

---

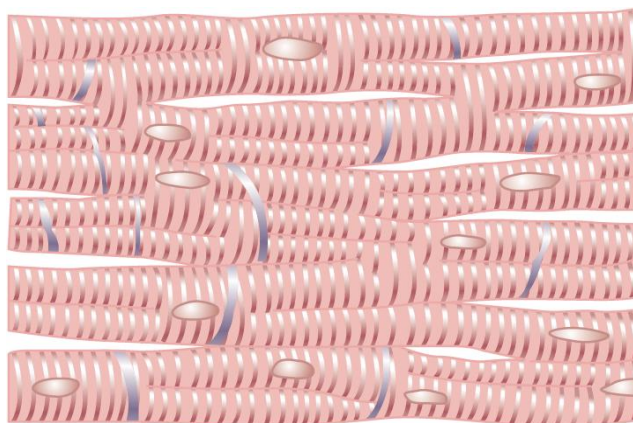
<sup>1</sup> A sístole é um período de contração do coração onde o sangue é ejetado para as artérias.

<sup>2</sup> A diástole é um período de relaxamento onde o coração se enche de sangue proveniente das veias.

### 2.1.1. MÚSCULO CARDÍACO

O coração é constituído por várias camadas. A camada média e mais espessa é denominada de miocárdio. Por sua vez, a camada que reveste e protege o coração exteriormente é o pericárdio. Do lado oposto situa-se o endocárdio que é a camada mais interna do coração. Relativamente à sua constituição, o coração apresenta três tipos de músculo: fibras musculares auriculares, fibras musculares ventriculares e fibras musculares condutoras e excitatórias especializadas. As fibras musculares auriculares e ventriculares contraem-se da mesma maneira que o músculo esquelético mas com uma duração de contração muito maior. Por sua vez, as fibras musculares condutoras e excitatórias especializadas contraem-se fracamente pois possuem poucas fibras contráteis; em vez disso, apresentam descargas elétricas ritmadas e automáticas na forma de potenciais de ação ou a condução do potencial de ação através do coração, fornecendo um sistema de excitação que controla o batimento rítmico do coração.

A Figura 2 apresenta uma imagem histológica do músculo cardíaco. Nesta figura podem ver-se as fibras musculares cardíacas com uma estrutura entrelaçada com as fibras dividindo-se, juntando-se e dividindo-se novamente. Ainda na análise desta figura pode ver-se que o músculo cardíaco é estriado assim como o músculo esquelético. Além disto, o músculo cardíaco possui miofibrilas que contém filamentos de actina e miosina quase idênticos aos encontrados no músculo esquelético. Por fim, as áreas escuras anguladas que cruzam as fibras musculares cardíacas são discos intercalares. Estes discos separam células musculares cardíacas individuais umas das outras. Constata-se então que as fibras musculares cardíacas são constituídas por muitas células individuais ligadas.



**Figura 2** Aspeto do músculo cardíaco [4]

Em cada disco intercalado, as membranas celulares fundem-se umas com as outras de tal forma que criam uma junção de comunicação permeável denominada de *gap junction* e permitem a quase total difusão livre dos íons. Isto possibilita o movimento dos íons com facilidade ao longo das fibras musculares cardíacas de modo a que os potenciais de ação vão de uma célula muscular cardíaca para outra. Posto isto, o músculo cardíaco pode ser visto como um sincício<sup>3</sup> devido à numerosa quantidade de células musculares cardíacas presentes e interligadas por discos intercalados. Quando uma célula é excitada, o potencial de ação espalha-se para as outras células passando pelas interligações disponíveis para o efeito.

Este sincício pode ser dividido em dois: sincício auricular e sincício ventricular. O primeiro constitui as paredes das duas aurículas e o segundo as paredes dos dois ventrículos. As aurículas encontram-se separadas dos ventrículos através de um tecido fibroso. Normalmente, os potenciais de ação só podem passar do sincício auricular para o ventricular através do feixe de His [4]. Esta divisão permite a contração da aurícula antecipadamente em relação ao ventrículo.

### **2.1.2. POTENCIAIS DE AÇÃO NO MÚSCULO CARDÍACO**

O músculo esquelético só se contrai em resposta à atividade do nervo motor apropriado. Quando não há existência de nervo, o músculo esquelético não se contrai. O mesmo não acontece com o músculo cardíaco que na ausência de nervos apresenta contrações rítmicas. Esta propriedade do músculo cardíaco é a responsável pelos batimentos contínuos do coração. Trata-se de uma ritmicidade intrínseca cuja origem se encontra nas fibras musculares cardíacas encontradas na junção entre a veia cava superior e a aurícula direita. A região responsável por esta ritmicidade é denominada de nódulo sino-auricular (SA) cujas células são conhecidas como células marca-passo (a sua atividade estabelece a frequência cardíaca básica). Os potenciais de ação, com origem no nódulo SA, propagam-se por todo o coração de modo a que o músculo cardíaco se comporte como um sincício.

As células cardíacas assim como todas as células vivas do corpo possuem potencial elétrico através da membrana celular. Este potencial pode ser medido colocando um microeléctrodo no interior da célula e medindo o potencial elétrico (em milivolts – mV) no

---

<sup>3</sup> O sincício é um conjunto de várias células mononucleadas e justapostas que dão origem a uma célula multinucleada.

interior da célula em relação ao exterior da mesma. Por convenção é assumido que o exterior da célula é 0 mV [5].

Dos vários íons presentes tanto no interior como no exterior das células, os mais importantes para determinar o potencial através da membrana celular são os íons de sódio ( $\text{Na}^+$ ), de potássio ( $\text{K}^+$ ) e de cálcio ( $\text{Ca}^{2+}$ ). Apesar de os íons de cloreto ( $\text{Cl}^-$ ) também se encontrarem tanto no interior como no exterior da célula, apresentam um pequeno contributo para o potencial de repouso da membrana. A Figura 3 apresenta uma membrana celular onde se podem ver íons de sódio, de potássio e de cloreto. Destes íons, o mais importante para determinar o potencial de repouso da membrana é o íon de potássio. Nas células cardíacas, este íon encontra-se em grandes concentrações no meio intracelular e em pequenas concentrações no meio extracelular. Consequentemente existe um gradiente de concentração para que os íons de potássio se difundam para o exterior da célula. Por sua vez, os íons de cloreto, de sódio e de cálcio encontram-se em grandes concentrações no meio extracelular e em pequenas concentrações no meio intracelular. No caso destes íons, o gradiente de concentração surge de modo a que estes se difundam para o meio intracelular. Ainda no exemplo da Figura 3, a membrana celular é permeável ao potássio. Esta permeabilidade permite que os íons de potássio carregados positivamente sejam difundidos para o exterior da célula deixando para trás cargas negativas não compensadas. Esta separação de cargas causa o potencial de membrana [5] [6].

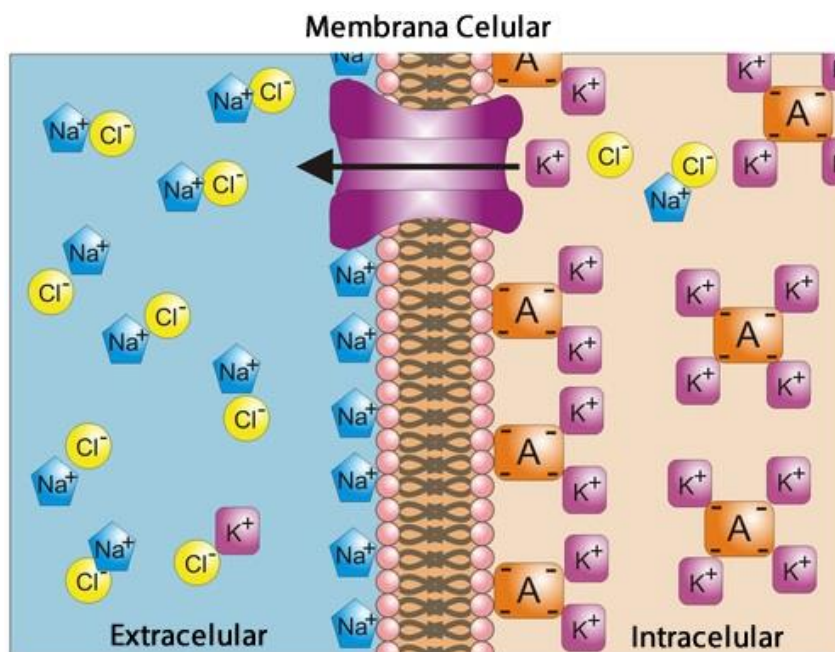
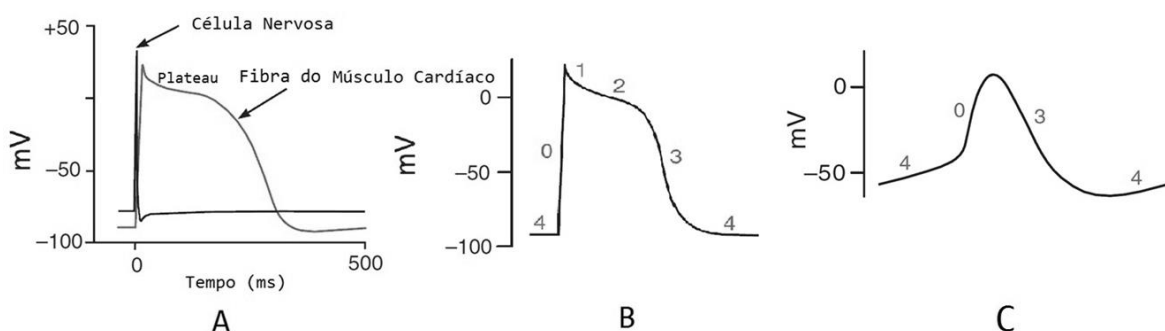


Figura 3 Membrana Celular [6]

Os iões movem-se do meio intracelular para o meio extracelular e vice-versa através de canais iónicos especializados. Estes canais são especializados devido à sua seletividade em permitir a passagem a diferentes catiões e aniões. No músculo cardíaco temos canais que permitem a passagem de iões de sódio, potássio e cálcio. Estes canais apenas deixam passar um tipo de iões bloqueando a passagem aos restantes [7].

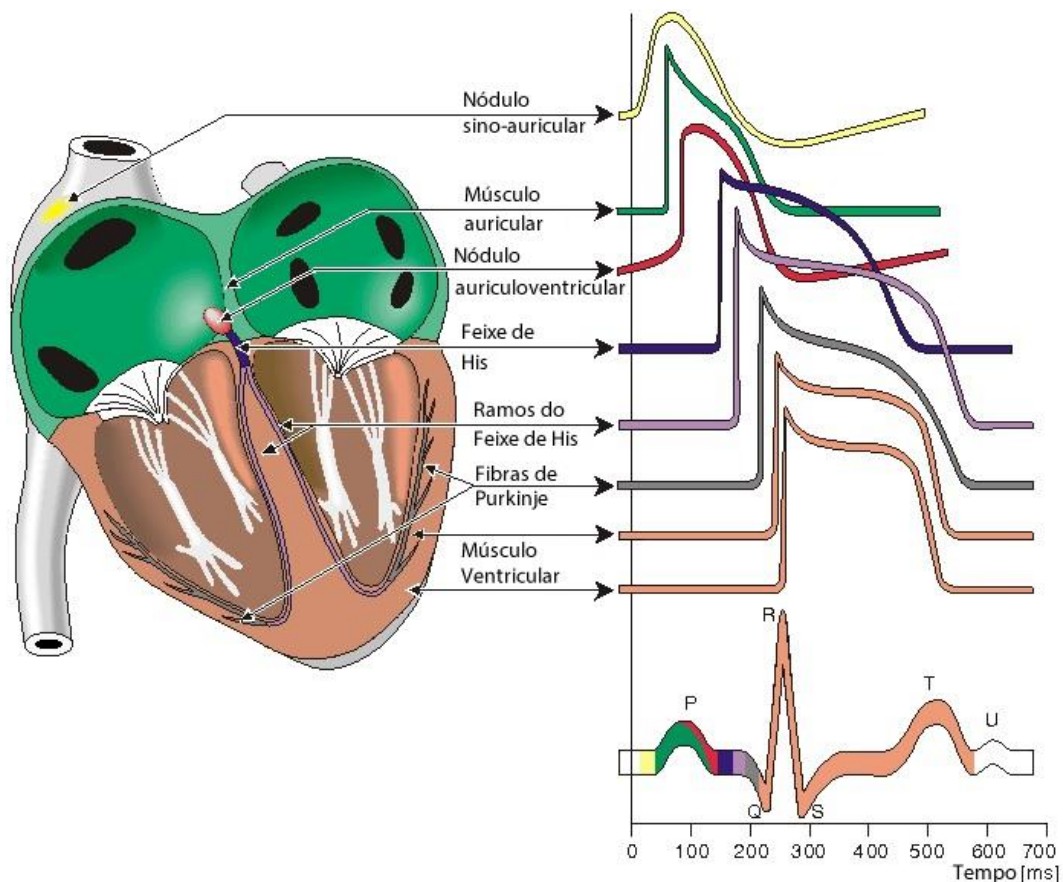
Os potenciais de ação do músculo cardíaco podem ser divididos em dois tipos: potenciais de ação sem propriedade marca-passo e com propriedade marca-passo. O potencial de ação do músculo cardíaco sem propriedade marca-passo é diferente daquele que se observa nas células dos nervos (Figura 4A). Este potencial de ação possui uma fase denominada de *plateau* que dura cerca de 300 milissegundos (ms) em comparação aos potenciais das células nervosas que duram 1 ms. O potencial de ação cardíaco tem cinco fases como se pode ver na Figura 4B. Na primeira fase, fase 0, a permeabilidade da membrana ao potássio diminui (fluxo de potássio diminui) e são abertos canais rápidos de sódio que produzem uma rápida despolarização (de aproximadamente -90 mV até +10 mV). Durante a fase 1, existe uma repolarização parcial devido à diminuição da permeabilidade da membrana ao sódio. O que faz que a passagem de iões de sódio através dos canais rápidos de sódio diminua devido ao fecho destes. Na fase 2 pode ver-se o já referido *plateau* dos potenciais de ação cardíacos sem propriedade marca-passo. Nesta fase, a permeabilidade da membrana ao cálcio aumenta (abertura dos canais de cálcio-sódio) mantendo a despolarização e o potencial de ação. No final da fase 2, a permeabilidade ao cálcio diminui (fecho dos canais de cálcio-sódio) e o *plateau* é parcialmente mantido por uma corrente inversa de sódio. Na fase 3 ocorre a repolarização através de uma corrente de potássio para o meio intracelular. Finalmente, na fase 4 ocorre o potencial de repouso da membrana que retorna ao seu valor inicial de aproximadamente -90 mV [8].



**Figura 4** Potenciais de ação de uma célula nervosa e de uma fibra do músculo cardíaco – A. Fases do potencial de ação sem e com marca-passo – B e C respetivamente [5]

O potencial de ação do músculo cardíaco com propriedade marca-passo é visto nas células dos nódulos SA e AV. As células do feixe de His e algumas fibras de *Purkinje* também são capazes de apresentar um potencial de ação com propriedade marca-passo. Na Figura 4C pode ver-se o potencial de ação com propriedade marca-passo. Como se pode comprovar, a diferença relativamente ao potencial de ação do músculo com propriedade sem marca-passo reside na ausência das fases 1 e 2. A fase 0 diz respeito à despolarização das células devido ao aumento da condutância do cálcio e uma diminuição da corrente de sódio. Por sua vez, a fase 3 resulta de um incremento da permeabilidade ao potássio e de uma diminuição da permeabilidade do cálcio. Por fim a fase 4 sofre uma despolarização lenta devido à diminuição da permeabilidade do potássio [8].

Na Figura 5 é possível visualizar as diferentes formas de ondas para as várias células especializadas existentes no coração. A soma de todas estas formas de onda resulta no sinal ECG evidenciado na parte inferior da figura. Ainda na mesma figura pode ver-se que o sinal ECG é caracterizado pelas ondas P, Q, R, S, T e U; estas ondas serão mais à frente explanadas.



**Figura 5** Diferentes formas de onda para as várias células especializadas do coração [9]

### 2.1.3. EXCITAÇÃO RÍTMICA DO CORAÇÃO

O coração possui um sistema especializado para gerar os impulsos rítmicos, para causar a contração do músculo cardíaco e para a condução rápida desses impulsos por todo o coração.

A Figura 6 apresenta o sistema especializado de excitação e condução do coração que controla as contrações cardíacas. Nela destacam-se os seguintes elementos cujas funções são:

- Nódulo sino-auricular (SA) – é aqui que é gerado o impulso rítmico normal;
- Vias internodais – vias que conduzem o impulso do nódulo sino-auricular para o nódulo auriculoventricular;
- Nódulo auriculoventricular (AV) – neste nódulo o impulso sofre um atraso antes de entrar no ventrículo;
- Feixe de His ou feixe auriculoventricular (AV) – conduz o impulso das aurículas para os ventrículos;
- Feixe esquerdo e direito das fibras de *Purkinje* – conduzem o impulso cardíaco para todas as partes dos ventrículos.

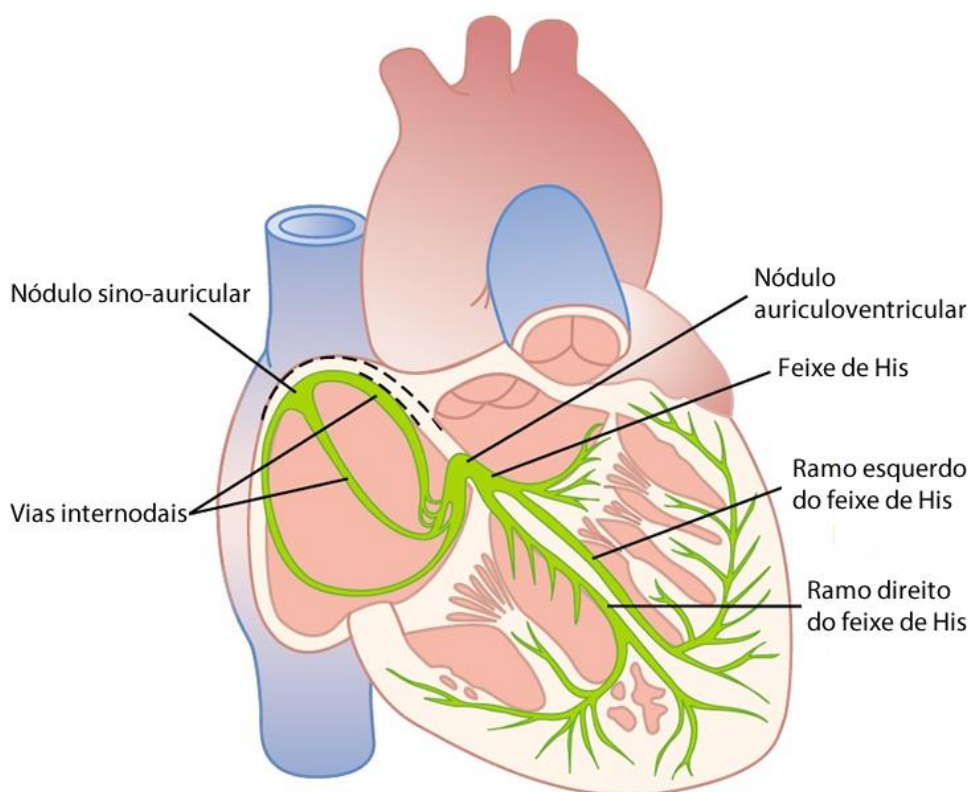
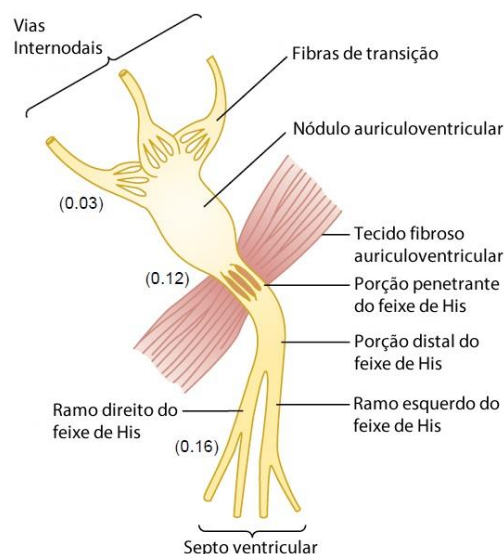


Figura 6 Sistema especializado de excitação e condução do coração [4]

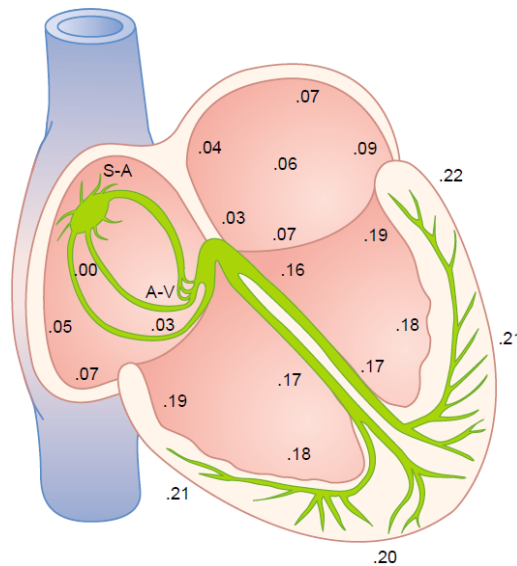
O nódulo SA encontra-se localizado na parede superior lateral da aurícula direita, imediatamente abaixo e lateral da abertura da veia cava superior. Existem fibras cardíacas que possuem a capacidade de autoexcitação, um processo que pode originar contrações rítmicas automáticas. É esse o caso das fibras do nódulo sino-auricular que, em condições normais, possuem a capacidade de controlar a frequência do batimento cardíaco.

Como já foi referido atrás, o nódulo SA gera um impulso elétrico que percorre a aurícula direita e a esquerda, levando-as à sua contração. O facto das extremidades das fibras do nódulo sino-auricular encontrarem-se ligadas às fibras auriculares circundantes permite que os potenciais de ação originados no nódulo SA se propaguem para toda a massa muscular auricular acabando por chegar ao nódulo AV. A velocidade de condução no músculo auricular é de cerca de 0,3 m/s. O nódulo AV não encaminha diretamente o impulso; é inserido um atraso na passagem das aurículas para os ventrículos. Isto acontece para que as aurículas possuam tempo suficiente para enviar o seu conteúdo para os ventrículos antes do início da contração ventricular. Na Figura 7 pode ver-se a representação do nódulo AV e como se encontra ligado às vias internodais e ao feixe de His. Nela são também visíveis os intervalos em frações de segundos desde a geração do impulso no nódulo SA até ao seu aparecimento no feixe de His. Como se pode ver, o impulso chega ao nódulo AV em 0,03 s após a sua origem. No nódulo AV existe um atraso de 0,09 s antes de ser propagado para a parte penetrante do feixe de His. Neste feixe surge mais um atraso de 0,04 s. O atraso total no nódulo AV e no feixe de His é de aproximadamente 0,13 s.



**Figura 7 Nódulo AV e intervalos de tempo a partir do nódulo sino-auricular [4]**

A partir do nódulo AV, o impulso segue para o feixe de His que se divide no ramo direito para o ventrículo direito e no ramo esquerdo para o ventrículo esquerdo. Aqui o impulso estende-se para todas as partes dos ventrículos através das fibras de *Purkinje* (velocidades de 1,5 até 4 m/s). Cada ramo mencionado desce até ao ápice do ventrículo dividindo-se em ramos menores que circundam as câmaras ventriculares e regressam em direção à base do coração, como se pode ver na Figura 6.



**Figura 8** Transmissão do impulso cardíaco pelo coração em frações de segundo [4]

A Figura 8 ilustra a transmissão do impulso cardíaco pelo coração humano. Os números presentes na figura representam os intervalos de tempo, em centésimos de segundo, do tempo decorrido desde a origem do impulso cardíaco por parte do nódulo SA até ao aparecimento em cada ponto respetivo do coração.

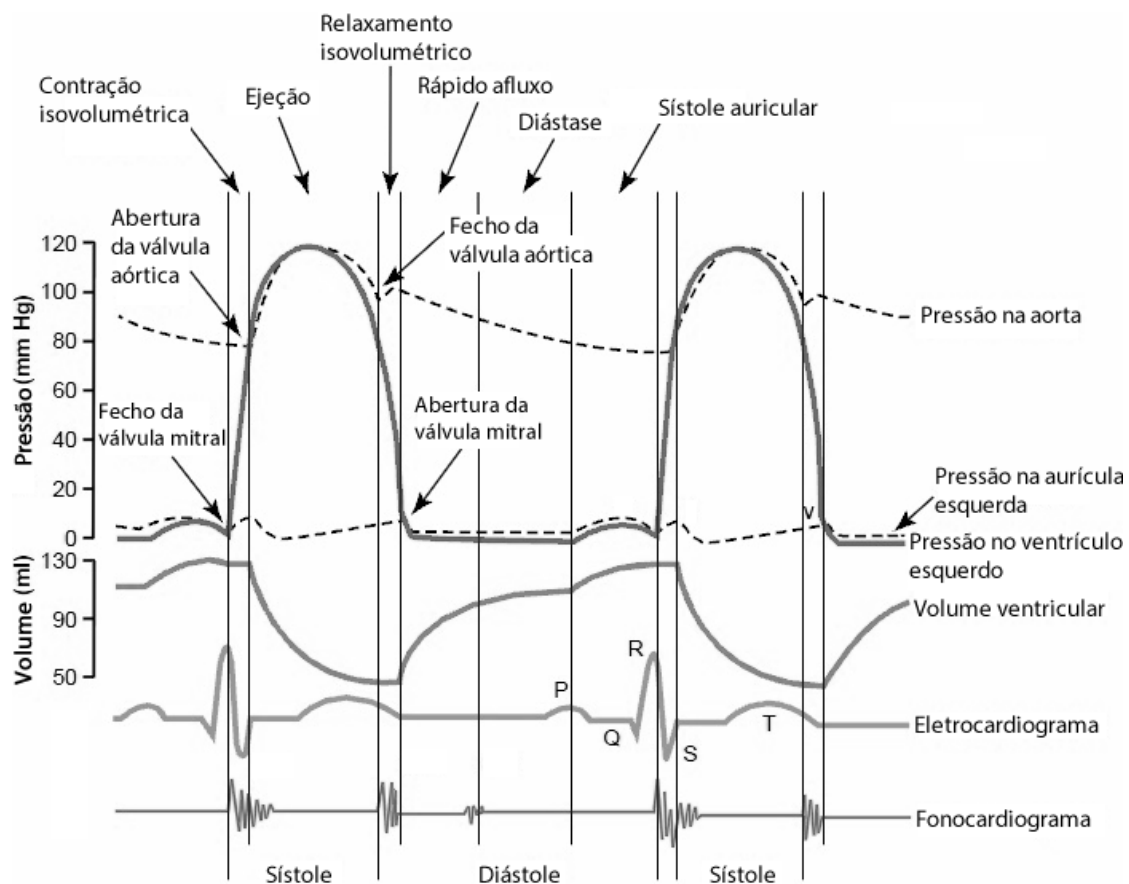
#### 2.1.4. CICLO CARDÍACO

A função do coração é bombear o sangue para os vários órgãos do corpo. O intervalo de tempo que decorre entre o início de um batimento cardíaco e o batimento seguinte é denominado de ciclo cardíaco. O início de um batimento cardíaco é marcado pela geração espontânea de um potencial de ação no nódulo sino-auricular.

Como já fora mencionado anteriormente, o ciclo cardíaco consiste em dois períodos: diástole e sístole. Na Figura 9 podem ver-se os vários eventos durante um ciclo cardíaco. As três curvas superiores mostram as alterações da pressão na aorta, na aurícula esquerda e no ventrículo esquerdo. A quarta curva a contar da parte superior mostra as alterações do

volume ventricular, a quinta o eletrocardiograma e a sexta um fonocardiograma (registro dos sons produzidos pelo coração).

Torna-se então importante analisar, no conceito deste projeto, o eletrocardiograma presente nesta figura que introduz as ondas P, Q, R, S e T. Estas ondas são registadas pelo eletrocardiógrafo a partir da superfície corporal. A onda P é causada pela despolarização das aurículas, cujo acontecimento seguinte é a contração auricular que gera uma ligeira subida na curva da pressão auricular imediatamente após a onda P. Num intervalo de tempo de aproximadamente 0,16 s, aparece o complexo QRS que é traduzido pela despolarização dos ventrículos. Através da figura pode ver-se que quando a onda R atinge o seu máximo, a pressão ventricular aumenta acontecendo a contração dos ventrículos. Constata-se então que o complexo QRS acontece pouco antes do início da sístole ventricular. Por fim, no eletrocardiograma, aparece a onda T. Esta onda é referente à repolarização dos ventrículos (fibras musculares dos ventrículos começam a relaxar) e acontece alguns instantes antes do fim da contração ventricular.



**Figura 9** Eventos do ciclo cardíaco – pressão auricular esquerda, pressão ventricular esquerda, pressão aórtica, volume ventricular, eletrocardiograma e fonocardiograma [4]

## 2.2. ELETROCARDIOGRAMA

O conhecimento em bioeletricidade começou em 1787 com Luigi Galvani a observar a contração do músculo de uma rã quando exposto a descargas elétricas. Galvani postulou a ideia de “eletricidade animal” que fora, por um longo período de tempo, fortemente rejeitada por Alessandro Volta. O desenvolvimento de galvanômetros permitiu comprovar a existência de cargas e correntes dentro da rã. Em 1843, Matteuci, demonstrou que a partir do coração em repouso também podiam ser medidas correntes elétricas. DuBois-Reymond investigou este fenômeno com um maior detalhe e introduziu o conceito de potenciais de ação para a mudança de corrente associada à contração muscular. Em 1878, Englemann apresentou graficamente as variações do potencial elétrico do coração de uma rã. Entre 1887 e 1888, Augustus Desiree Waller apresenta os primeiros registos dos potenciais elétricos associados ao batimento do coração. Alguns anos depois, em 1893, William Einthoven introduziu o conceito de eletrocardiograma. Dois anos depois, Einthoven distinguiu 5 deflexões no traçado do ECG que denominou P, Q, R, S e T e que ainda hoje são utilizadas [10].

Como já foi anteriormente referido, o eletrocardiograma permite registrar a atividade elétrica do coração. Isto é, o impulso elétrico gerado no nódulo SA irá propagar-se pelo coração e pelos tecidos que o circundam até que uma pequena porção destas correntes elétricas atinja a superfície do corpo humano. Estes sinais são detetados por elétrodos metálicos (ligados aos membros e na parede torácica) e registados pelo eletrocardiógrafo. O ECG é o exame auxiliar mais usado no diagnóstico de doenças cardíacas devido ao seu preço (barato), a sua versatilidade (extremamente versátil) e por se tratar de um exame não invasivo [11]. Na Figura 10 pode ver-se representado um eletrocardiograma normal num intervalo com dois batimentos cardíacos.

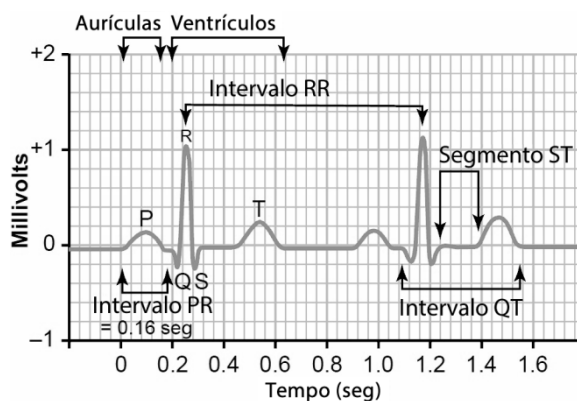


Figura 10 Eletrocardiograma [4]

A análise da Figura 10 permite identificar os três parâmetros principais do sinal ECG [4]:

- Onda P;
- Complexo QRS (constituído por: onda Q, onda R e onda S);
- Onda T.

A primeira onda que surge no ECG normal, onda P, representa a despolarização das aurículas antes da sua contração. Por sua vez, o complexo QRS resulta da despolarização dos ventrículos quando são gerados potenciais antes da sua contração. Tanto a onda P como as ondas que constituem o complexo QRS são ondas de despolarização. Por sua vez, a onda T, é uma onda de repolarização: isto deve-se ao facto da onda T ser gerada por potenciais quando os ventrículos regressam ao estado normal. Pode constatar-se então que o eletrocardiograma é constituído tanto por ondas de despolarização como de repolarização [4].

De notar que a repolarização das aurículas (onda  $T_a$ ) não costuma ser registada. O seu não registo deve-se ao facto da repolarização acontecer concomitantemente com a despolarização dos ventrículos e como esta última possui uma maior potência, ofusca a repolarização auricular. Além desta particularidade, em algumas situações aparece ainda uma onda U de baixa amplitude que surge depois da onda T e cuja origem não se encontra perfeitamente esclarecida [12] [13].

Além dos três parâmetros principais já referidos, a Figura 10 permite observar outros parâmetros importantes [5]:

- Intervalo RR – tempo de um ciclo cardíaco (sístole + diástole) que corresponde à distância entre duas ondas R consecutivas.
- Intervalo PR – tempo entre o início da despolarização auricular e da despolarização ventricular. É medido desde o início da onda P até ao início do complexo QRS.
- Intervalo QT – tempo entre o início da despolarização dos ventrículos e a sua repolarização. É medido desde o início da onda Q até ao final da onda T.
- Segmento ST – normalmente isoeletrico<sup>4</sup>, tem início no fim do complexo QRS (nomeadamente no ponto J – final do complexo QRS e início do segmento ST) e termina no início da onda T.

---

<sup>4</sup> Segmento isoeletrico quer dizer que se encontra ao mesmo nível elétrico que o sinal do ECG em repouso cardíaco.

Na Tabela 2 pode ver-se um quadro resumo com às características do sinal ECG.

**Tabela 2 Quadro resumo do sinal ECG [5]**

<b>Componente do ECG</b>	<b>Representa</b>	<b>Duração (segundos)</b>
<b>Onda P</b>	Despolarização auricular	0,08 – 0,10
<b>Complexo QRS</b>	Despolarização ventricular	0,06 – 0,10
<b>Onda T</b>	Repolarização ventricular	<sup>5</sup>
<b>Intervalo PR</b>	Despolarização auricular mais atraso do nóculo AV	0,12 – 0,20
<b>Intervalo QT</b>	Comprimento da despolarização mais a repolarização – corresponde à duração do potencial de ação	0,20 – 0,40 <sup>6</sup>
<b>Segmento ST</b>	Período isoeletrico depois da despolarização dos ventrículos	<sup>5</sup>

De modo a compreender os aspetos fundamentais inerentes ao eletrocardiograma vão ser introduzidas as seguintes seções: ondas de despolarização e repolarização, definições eletrocardiográficas, derivações eletrocardiográficas e ritmos cardíacos normais e anormais.

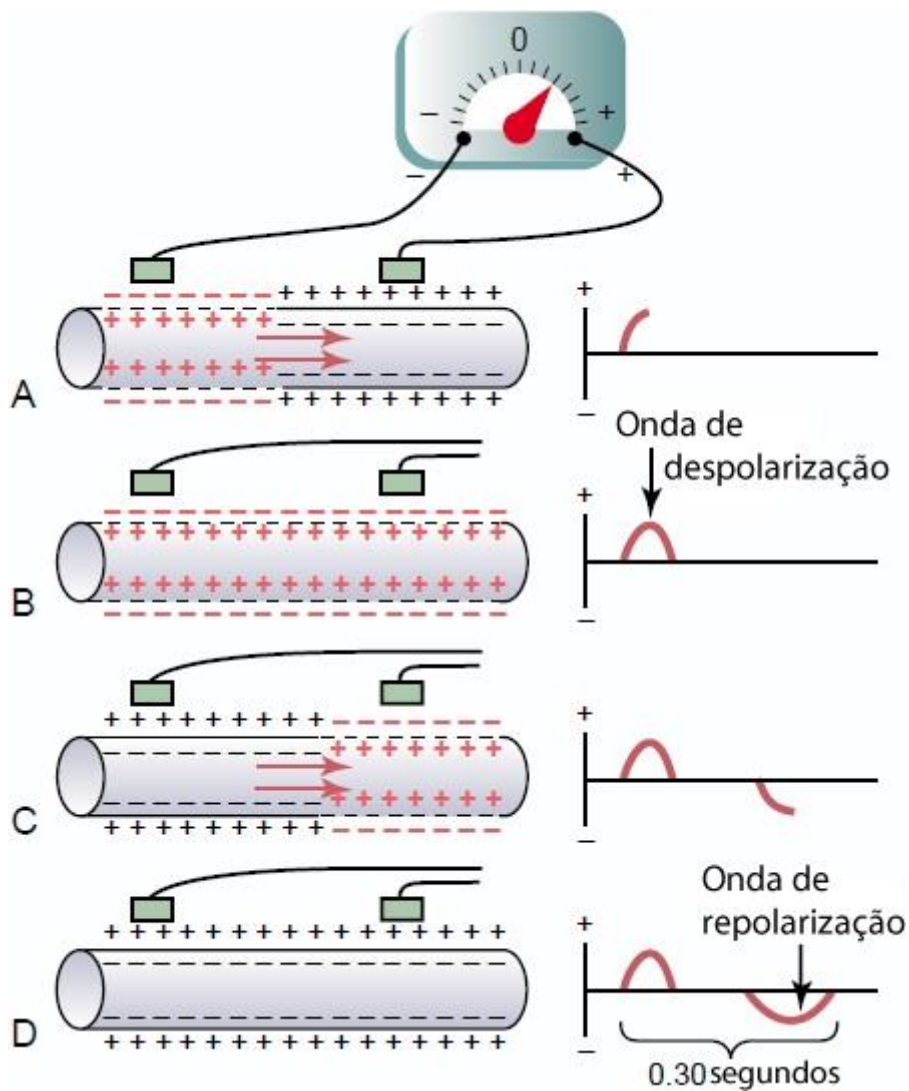
### **2.2.1. ONDAS DE DESPOLARIZAÇÃO E DE REPOLARIZAÇÃO**

De modo a perceber como é feito o registo das ondas de despolarização e de repolarização é apresentada a Figura 11. Nela encontra-se representada uma fibra muscular cardíaca em quatro estágios de despolarização e repolarização. No seu estado de repouso, a fibra encontra-se com um potencial negativo no interior e positivo no exterior. As Figuras 11A e 11B são relativas à onda de despolarização enquanto as Figuras 11C e 11D referem-se à onda de repolarização. Na Figura 11A o processo de despolarização é representado pelas cargas que se movimentam da esquerda para a direita. Verifica-se que metade da fibra já se despolarizou enquanto a outra metade ainda se encontra polarizada. Nesta fase o eléctrodo esquerdo encontra-se sobre cargas negativas e o eléctrodo direito sobre cargas positivas o que faz com que o registo da onda seja positivo como se pode ver no gráfico à direita da fibra muscular. É de referir que quando a despolarização se encontra na sua fase

<sup>5</sup> Duração normalmente não medida.

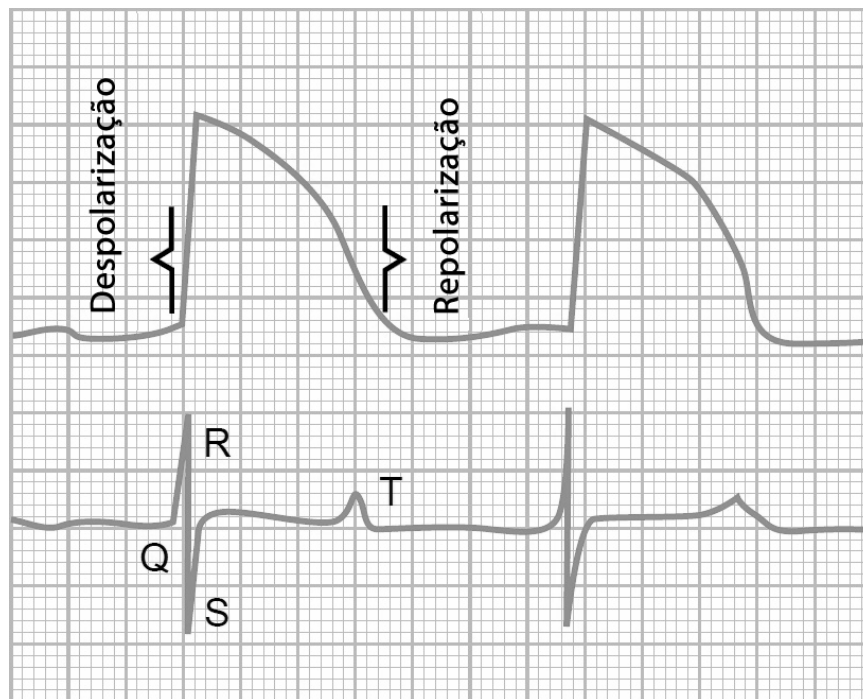
<sup>6</sup> Frequências cardíacas elevadas reduzem o potencial de ação, conseqüentemente também reduzem o intervalo QT.

intermédia, o valor registado é máximo. A Figura 11B apresenta a despolarização total da fibra muscular cardíaca. No gráfico à direita desta figura pode ver-se que o valor do registo retornou a zero pois ambos os eléctrodos encontram-se situados sobre cargas negativas logo não existe diferença de potencial entre eles. Na Figura 11C encontra-se representado o processo de repolarização da fibra. Assim como nas Figuras 11A e 11B, aqui este processo também é representado pelas cargas que se movimentam da esquerda para a direita. Neste caso, como o eléctrodo esquerdo se encontra sobre cargas positivas e o eléctrodo direito sobre cargas negativas, o registo da onda será negativo. Por fim, na Figura 11D, a fibra encontra-se totalmente repolarizada e ambos os eléctrodos encontram-se sobre cargas positivas o que resulta no regresso da onda a zero pois não é registado nenhuma diferença de potencial entre eles [4].



**Figura 11** Despolarização (A e B) e repolarização (C e D) de uma fibra do músculo cardíaco e registo da respetiva onda gerada e captada através dos eléctrodos [4]

Uma das conclusões retiradas aquando da análise da Figura 11 reside no facto do ECG só apresentar variações no sinal quando existem diferenças de potencial que chegam aos eléctrodos. Para exemplificar isto, é apresentada a Figura 12. Na parte superior desta figura é apresentado o potencial de ação registado com um microeléctrodo colocado dentro de uma fibra muscular ventricular. De realçar que a fase ascendente é originada pela despolarização e a fase descendente (retorno ao estado de repouso) é originada pela repolarização. Por sua vez, na parte inferior da Figura 12 é apresentado o registo simultâneo do ECG nesse mesmo ventrículo. Neste ECG surge inicialmente o complexo QRS que coincide com o início do potencial de ação registado e a onda T que coincide com o fim do potencial de ação registado.



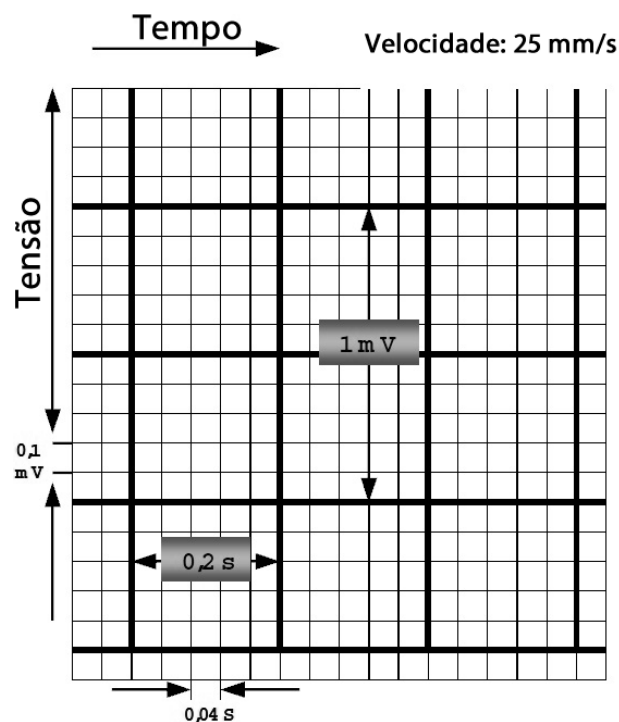
**Figura 12 Relação entre o potencial de ação de uma fibra muscular ventricular e o ECG registado simultaneamente [4]**

Como se pode observar na Figura 12 e como já foi anteriormente referido, nenhuma alteração é visualizada quando a fibra se encontra totalmente polarizada ou totalmente despolarizada.

### **2.2.2. DEFINIÇÕES ELETROCARDIOGRÁFICAS**

O eletrocardiograma é registado numa grelha eletrocardiográfica que pode já estar marcada no papel ou pode ser esboçada ao mesmo tempo que é registado o eletrocardiograma. Na

Figura 13 pode ver-se um exemplo de uma grelha eletrocardiográfica. As linhas horizontais representam a tensão enquanto as linhas verticais representam o espaço temporal. Cada linha horizontal está espaçada de 1 mm das linhas adjacentes e representa 0,1 mV. Um conjunto de 10 linhas horizontais representam 1 mV de amplitude sendo positivo no sentido ascendente e negativo no sentido descendente. Por sua vez, cada linha vertical também está espaçada de 1 mm das linhas verticais adjacentes e representa 0,04 s. Um conjunto de 10 linhas verticais corresponde a 0,4 s. Tendo em conta as linhas verticais pode então dizer-se que a velocidade é de 25 mm/s. Por fim, as linhas mais carregadas encontram-se presentes em intervalos de 5 mm [11].



**Figura 13 Grelha eletrocardiográfica [11]**

Como já se verificou, só há alteração do sinal do ECG quando há despolarizações ou repolarizações. Para calcular a tensão das várias deflexões registadas é necessário partir da linha de base (estado de repouso) até ao pico desejado; pico superior para a porção positiva e pico inferior para a porção negativa. Por sua vez, a frequência cardíaca pode ser calculada de duas formas diferentes caso o ritmo seja regular ou irregular. Se o ritmo é regular, a frequência cardíaca pode ser calculada dividindo 300 pelo número de divisões grandes (linhas mais carregadas) entre duas ondas R consecutivas. Também é possível dividir 1500 pelo número de divisões mais pequenas (divisões de 1 mm). No caso do ritmo

irregular, é necessário efetuar a contagem de ondas R durante um período de 6 segundos e multiplicar esse valor por dez [11].

De modo a evitar erros de interpretação do eletrocardiograma, é necessário adotar uma abordagem sistemática quando se observa um traçado. Para isso, os seguintes pontos devem ser analisados cuidadosamente em cada eletrocardiograma [11]:

1. Padronização (calibração) e características técnicas (incluindo a colocação dos elétrodos e artefactos);
2. Frequência cardíaca;
3. Ritmo;
4. Intervalo PR;
5. Intervalo QRS;
6. Onda P;
7. Tensão do complexo QRS;
8. Eixo elétrico médio do complexo QRS;
9. Progressão da onda R nas derivações pré-cordiais;
10. Onda T;
11. Onda U.

### **2.2.3. DERIVAÇÕES ELETROCARDIOGRÁFICAS**

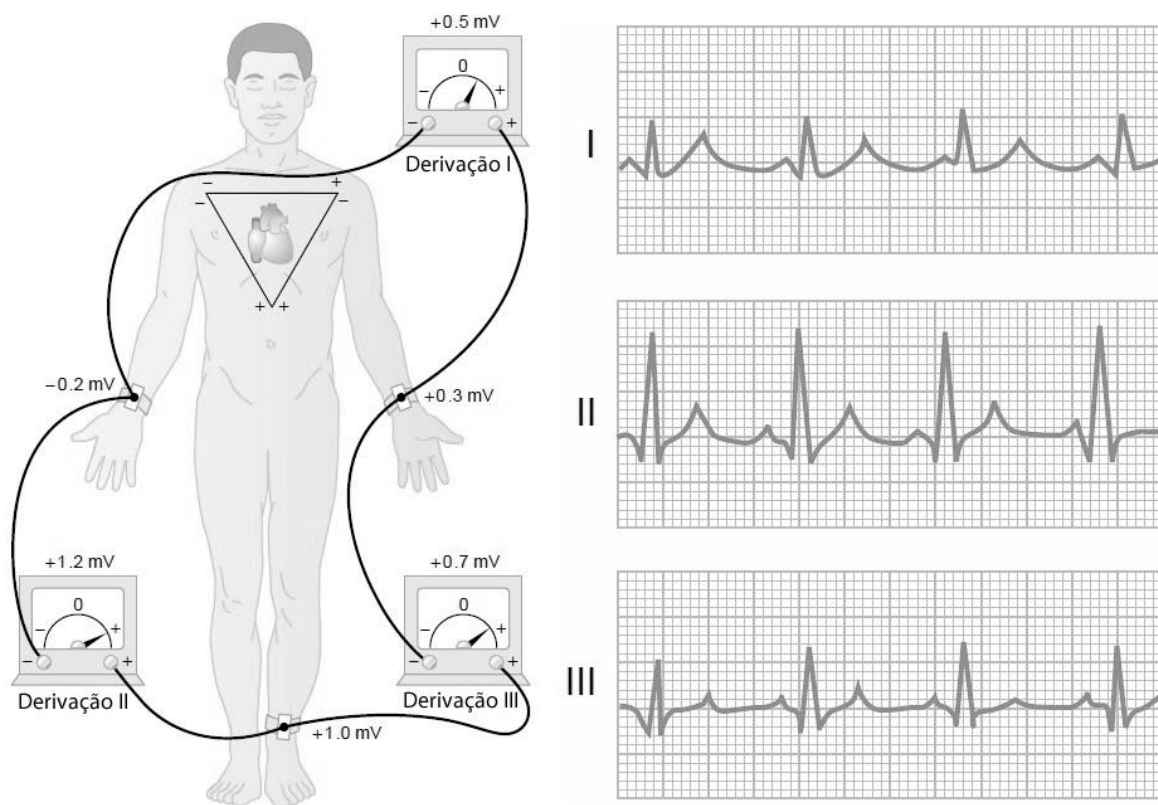
A maneira mais direta de fazer o registo do eletrocardiograma é colocando elétrodos diretamente no músculo cardíaco. No entanto, esta opção raramente é utilizada pois é necessário abrir o tórax. Em casos onde sejam necessários procedimentos cirúrgicos, nomeadamente na abertura do tórax, essa opção é utilizada. Em detrimento desta opção é usado um método não invasivo que consiste na colocação de elétrodos em diferentes pontos da superfície corporal e medir as respetivas diferenças de tensão. Quando um par de elétrodos é colocado numa dada posição, o fluxo de corrente detetado corresponde a uma parte específica que ocorre durante a despolarização e repolarização. De modo a obter o quadro completo das informações originadas pelo fluxo da corrente, é necessário colocar os elétrodos em posições diferentes sobre a superfície corporal. O conjunto das várias posições onde são colocados os elétrodos recebe o nome de derivações [12].

As derivações são constituídas por um par de elétrodos; um elétrodo positivo e um elétrodo negativo ligados à superfície corporal. Entre estes dois elétrodos existe um vetor

denominado de eixo de derivação. O eletrocardiograma normal é constituído por doze derivações [11]. Estas derivações podem ser divididas em dois grupos: derivações bipolares e unipolares [12]. Estas doze derivações permitem obter uma representação tridimensional da atividade elétrica cardíaca. Cada uma das doze derivações pode ser equiparada a uma câmara fotográfica que observa segundo ângulos diferentes os mesmos acontecimentos [11].

As derivações bipolares registam a tensão entre eléttodos colocados nos punhos e no tornozelo esquerdo (o tornozelo direito age como terra). Por sua vez, as derivações unipolares registam a tensão entre um eléttodo colocado na superfície corporal e outro eléttodo que é considerado neutro [12].

As três derivações bipolares dos membros são por convenção designadas de derivações bipolares dos membros I, II e III. Na Figura 14 é possível visualizar as três derivações bipolares dos membros [12].



**Figura 14** Derivações bipolares dos membros e respectivos registros eletrocardiográficos [4]

Na derivação I, o terminal positivo é ligado ao braço esquerdo e o terminal negativo ao braço direito. Assim, quando o ponto em que o braço direito se liga ao tórax for negativo

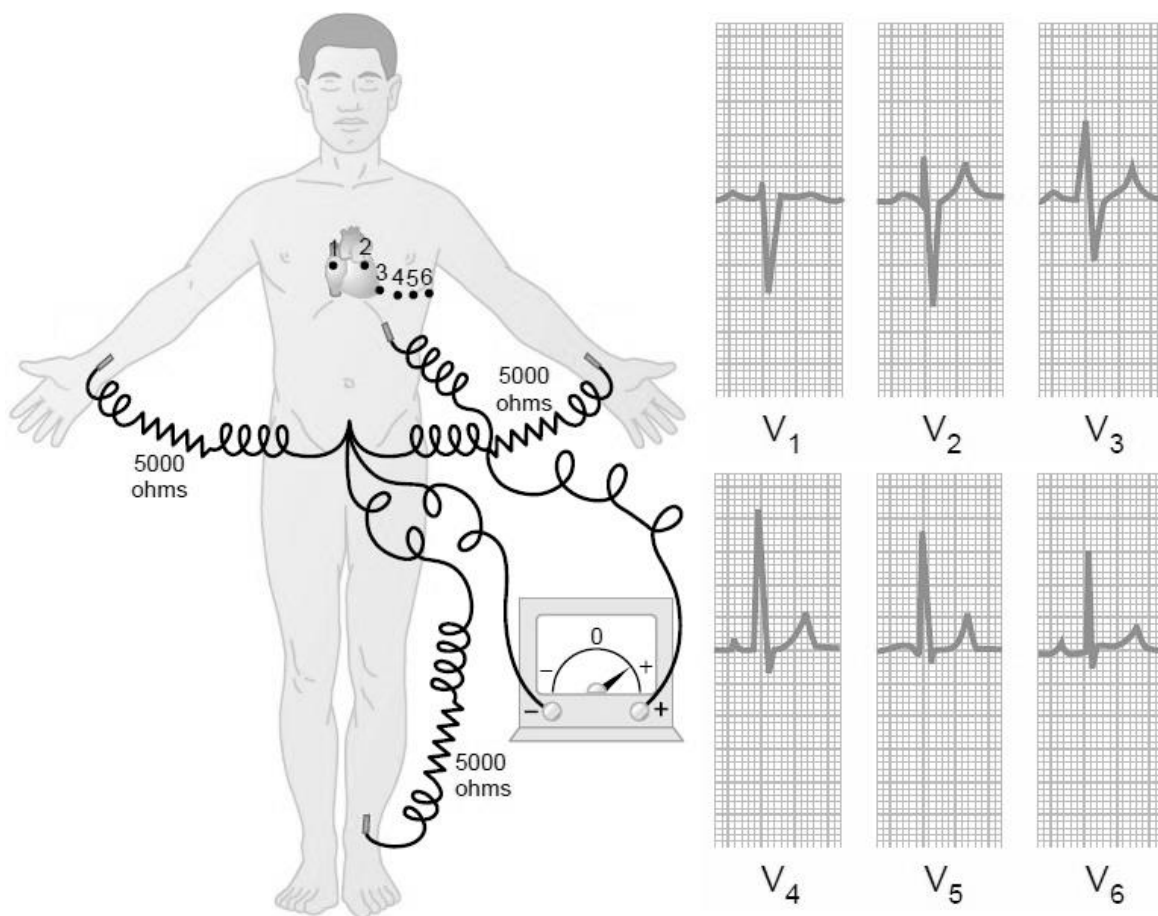
em relação ao ponto em que o braço esquerdo se liga ao tórax, o registo do eletrocardiograma é positivo (acima do valor de tensão zero). Caso contrário, o eletrocardiograma regista negativo (abaixo do valor de tensão zero). Na derivação II, o braço direito é o terminal negativo e a perna esquerda o terminal positivo. Quando o braço direito for negativo em relação à perna esquerda, o registo do eletrocardiograma é positivo. Caso contrário será negativo. Por fim, na derivação III, o terminal negativo é ligado ao braço esquerdo e o terminal positivo é ligado à perna esquerda. Nesta derivação, o eletrocardiograma regista positivamente quando o braço esquerdo for negativo em relação à perna esquerda [4].

Ainda na Figura 14, é possível visualizar o triângulo de Einthoven; método desenvolvido pelo próprio Einthoven que permite o estudo da atividade elétrica do coração cuja representação é feita graficamente através de um triângulo equilátero. Nesta representação os dois braços e a perna esquerda formam os vértices do triângulo cujo coração se encontra no centro. Os dois vértices superiores e o vértice inferior representam os pontos em que os dois braços e a perna esquerda se encontram eletricamente ligados [4].

Associada ao triângulo de Einthoven surge a Lei de Einthoven. Esta lei diz que se o potencial elétrico de duas quaisquer derivações bipolares for conhecido num dado instante, a terceira pode ser calculada pela fórmula  $D_{II} = D_I + D_{III}$ . Esta relação é baseada nas leis de Kirchoff. Por exemplo, no caso da Figura 14 sabemos que num dado instante o braço direito apresenta uma negatividade de 0,2 mV em relação ao resto do corpo, o braço esquerdo apresenta uma positividade de 0,3 mV e a perna esquerda apresenta uma positividade de 0,1 mV. Observando os medidores presentes nesta mesma Figura 14 pode ver-se que na derivação I a diferença de potencial é de 0,5 mV pois corresponde à diferença entre os -0,2 mV do braço direito e os 0,3 mV do braço esquerdo. O mesmo acontece para as derivações II e III onde a diferença de potencial é de 1,2 e 0,7 mV. Posto isto, pode verificar-se que a soma das tensões da derivação I e III é igual à derivação II [4].

De realçar também a presença dos registos eletrocardiográficos das derivações bipolares dos membros na Figura 14. Verifica-se que são bastante semelhantes pois todos eles apresentam ondas P e T positivas e a parte principal do complexo QRS também é positiva em cada eletrocardiograma. Também é possível verificar que a cada instante a soma dos potenciais nas derivações I e III é igual ao valor do potencial na derivação II [4].

Relativamente às derivações unipolares, são usadas dois tipos: derivações amplificadas dos membros e derivações pré-cordiais [12]. Por princípio, as derivações unipolares tentam medir potenciais locais e não diferenças de potencial [11]. As derivações pré-cordiais usam um elétrico indiferente, formado pela união das três derivações bipolares dos membros e de resistências em cada uma delas, e um elétrico explorador que é colocado em várias posições específicas no tórax (ao nível do coração). Cada uma dessas posições é denominada de  $V_1$ ,  $V_2$ ,  $V_3$ ,  $V_4$ ,  $V_5$  e  $V_6$ . Na Figura 15 é possível visualizar a colocação dos elétricos para as derivações pré-cordiais [12].

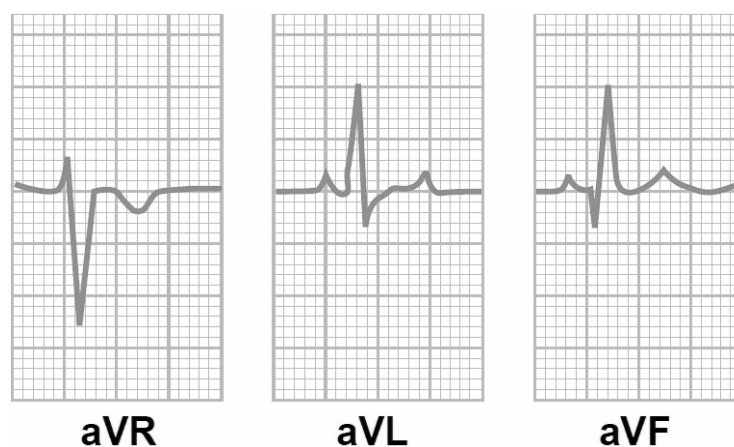


**Figura 15** Derivações pré-cordiais e respectivos registros eletrocardiográficos [4]

Nas derivações pré-cordiais, o elétrico indiferente é ligado a um terminal central que por sua vez é ligado ao polo negativo do eletrocardiógrafo; o elétrico explorador é ligado ao polo positivo. Relativamente ao elétrico indiferente, considera-se que a soma dos três potenciais (braços e perna esquerda) é igual a zero, isto é, o potencial do elétrico indiferente é zero [11].

Ainda na Figura 15, é também possível observar os registos eletrocardiográficos das derivações pré-cordiais. Dada a proximidade entre a superfície cardíaca e a parede torácica, cada elétrico regista o potencial elétrico da musculatura cardíaca imediatamente abaixo do elétrico. Consequentemente, algumas anormalidades relativamente pequenas nos ventrículos, por vezes causam alterações acentuadas nos eletrocardiogramas registados através das derivações pré-cordiais. Quanto à Figura 15, as derivações  $V_1$  e  $V_2$  apresentam um complexo QRS predominantemente negativo pois o elétrico encontra-se mais próximo da base do coração (direção da eletronegatividade durante maior parte do processo de despolarização) do que do ápice. O contrário se passa com as derivações  $V_4$ ,  $V_5$  e  $V_6$  onde o complexo QRS é predominantemente positivo pois os elétricos estão colocados perto do ápice (direção da eletropositividade durante maior parte do processo de despolarização) [4].

A tensão que pode ser registada a partir dos braços e da perna esquerda pode ser aumentada usando as derivações amplificadas dos membros. Nesta derivação, dois dos membros são ligados por resistências elétricas ao terminal negativo do eletrocardiógrafo enquanto o outro membro é ligado ao terminal positivo do eletrocardiógrafo. Quando o terminal positivo se encontra no braço direito a derivação tem o nome de aVR; quando se encontra no braço esquerdo é aVL; e por fim aVF quando se encontra na perna esquerda. Na Figura 16 é possível visualizar os registos das derivações unipolares aumentadas dos membros. Como se pode ver, estes registos são semelhantes aos registos das derivações bipolares dos membros à exceção da derivação aVR que se encontra invertida devido à polaridade das conexões com o eletrocardiógrafo [4].



**Figura 16** Registos eletrocardiográficos das derivações unipolares aumentadas dos membros aVR, aVL e aVF [4]

#### 2.2.4. RITMOS CARDÍACOS NORMAIS E ANORMAIS

Uma das funcionalidades do ECG consiste na análise dos ritmos cardíacos (lento, rápido ou irregular) por parte dos médicos. A taxa de despolarização auricular e ventricular pode ser determinada através das ondas P e do complexo QRS. Isto é feito através do registo dos eventos cardíacos do paciente onde normalmente é usada a derivação II. Num traçado normal do ECG, existe uma relação consistente entre a onda P e o complexo QRS. Isto é, normalmente a onda P antecede o complexo QRS. Esta característica dos traçados normais indica que a despolarização ventricular é desencadeada pela despolarização auricular. Em condições normais diz-se que o coração apresenta um ritmo sinusal devido ao controlo do ritmo cardíaco por parte do nódulo SA [5]. O ritmo sinusal implica três condições [11]:

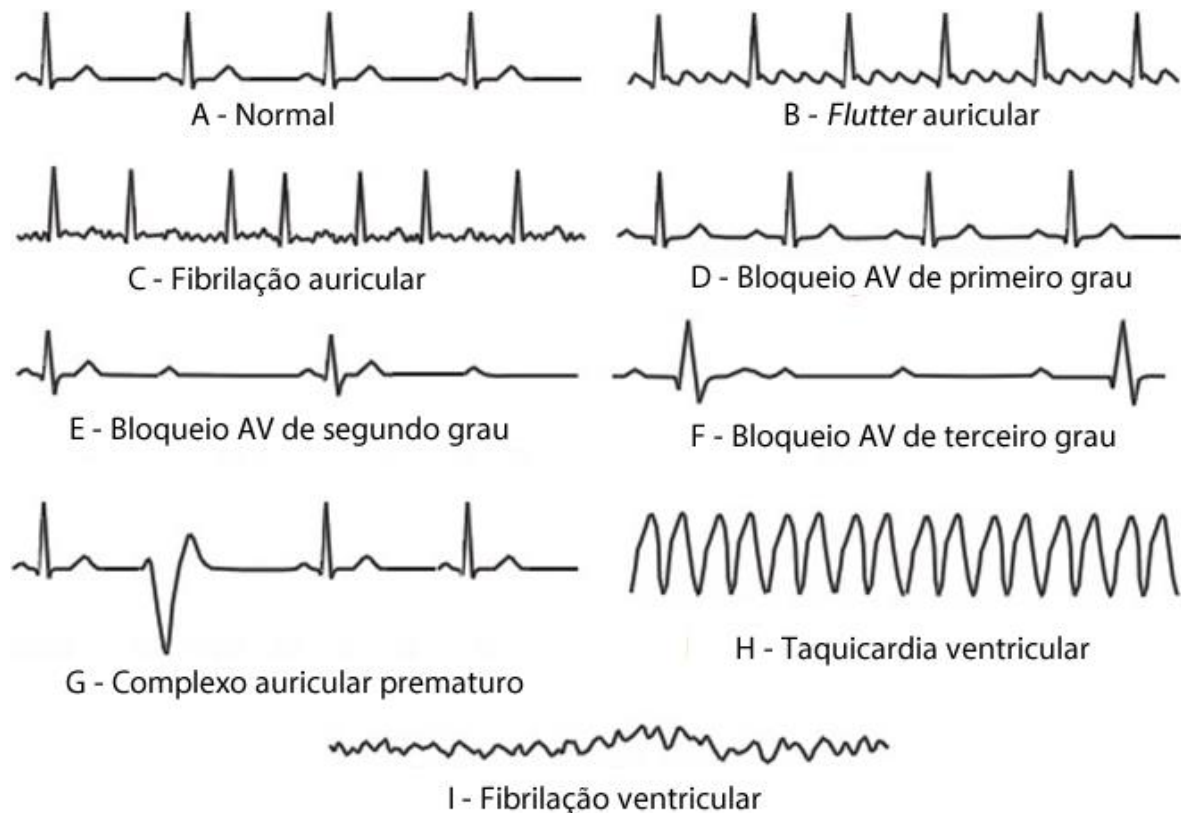
- Presença da onda P antes do complexo QRS;
- Onda P com posição espacial normal (positiva em I, II e aVF);
- Frequência adequada ao nódulo SA (entre 60 e 100 batimentos/minuto).

Ritmos anormais – arritmias – podem ser causados por formações anormais dos potenciais de ação. Quando a taxa de batimentos cardíacos é menor que 60 batimentos/minuto, diz-se que estamos perante uma – braquicardia sinusal. Em algumas pessoas, nomeadamente atletas, é possível encontrar taxas de batimentos cardíacos (no estado de repouso) menores que 60 batimentos/minuto. Quando a taxa de batimentos cardíacos estiver compreendida entre 100 e 180 batimentos/minuto, surge outro tipo de arritmia denominada de – taquicardia sinusal. Trata-se de um ritmo anormal para pessoas no estado de repouso, no entanto, é uma resposta normal durante a prática de exercício físico ou em situações emocionais intensas [5].

Como já tem vindo a ser referido, no traçado de um ECG normal o complexo QRS segue cada onda P. No entanto podem ocorrer situações onde a frequência das ondas P e dos complexos QRS é diferente criando ritmos cardíacos anormais. Na Figura 17 podem ver-se alguns exemplos de ritmos cardíacos anormais. É também visível um ritmo cardíaco normal de modo a poder verificar as diferenças existentes relativamente aos ritmos cardíacos anormais [5].

O *flutter* auricular, Figura 17, surge quando o ritmo auricular é bastante alto; na ordem dos 250 até aos 350 batimentos/minuto. Nesta situação, como nem todos os impulsos são encaminhados através do nódulo AV, pode então verificar-se uma frequência ventricular inferior à metade da frequência auricular. Por sua vez, na fibrilação auricular, o nódulo SA

não consegue ativar a despolarização auricular. Surgem então correntes de despolarização provenientes de várias zonas das aurículas que levam a despolarizações descoordenadas com ondas P de difícil distinção. Nesta situação (fibrilação auricular), o ritmo ventricular é irregular e normalmente rápido. Tanto o *flutter* como a fibrilação auricular ilustram a capacidade que o nódulo AV tem de limitar a frequência dos impulsos que conduz, limitando assim a frequência ventricular [5].



**Figura 17 Exemplos de ritmos cardíacos anormais [5]**

Ainda na Figura 17, podemos ver bloqueios AV de primeiro, segundo e terceiro grau. Os bloqueios AV são exemplos de arritmias causadas por condução de impulsos anormais. Nos bloqueios AV, o ritmo auricular é normal (Figura 17A), no entanto, cada despolarização auricular pode não ser seguida por uma despolarização ventricular. Por exemplo, o bloqueio AV de segundo grau (Figura 17E) pode ter duas ou três ondas P antes de cada complexo QRS; isto porque o nódulo AV não conduziu corretamente cada impulso. No caso do bloqueio AV de primeiro grau (Figura 17D), a condução do impulso através do nódulo AV é demorada no entanto o impulso é capaz de passar pelo nódulo AV e excitar os ventrículos. Nesta situação, o intervalo PR é maior que 0,2 segundos. Em casos extremos, bloqueio AV de terceiro grau (Figura 17F), as despolarizações auriculares não

são conduzidas através do nódulo AV para os ventrículos e as ondas P encontram-se completamente desassociadas dos complexos QRS [5].

Quando o ritmo ventricular é maior que o ritmo auricular, isto é, a frequência dos complexos QRS é maior que a frequência das ondas P, é observável taquicardia ventricular (Figura 17H) ou *flutter* ventricular. A taquicardia ventricular é caracterizada por possuir entre 100 a 200 batimentos/minuto. Por sua vez, o *flutter* ventricular possui mais de 200 batimentos/minuto. Tanto a taquicardia ventricular e o *flutter* ventricular são condições clínicas graves e podem levar à fibrilação ventricular (Figura 17I). Este ritmo cardíaco é identificado no ECG como despolarizações rápidas, de baixa tensão e descoordenadas onde não se consegue identificar o complexo QRS. Pode ser letal, no entanto, também pode ser revertida colocando elétrodos no peito e aplicando fortes mas breves correntes elétricas ao coração (desfibrilação) [5].

Por fim, existe ainda outro tipo de arritmia denominada de despolarização prematura. Esta despolarização pode ocorrer nas aurículas – complexo auricular prematuro – e nos ventrículos – complexo ventricular prematuro (Figura 17G). Estas arritmias devem-se ao aparecimento de locais com capacidade de *pacemaker* e cujo impacto no traçado do ECG reflete-se na adição de ondas P e complexos QRS com formas anormais [5].

# 3. SISTEMAS DIFUSOS

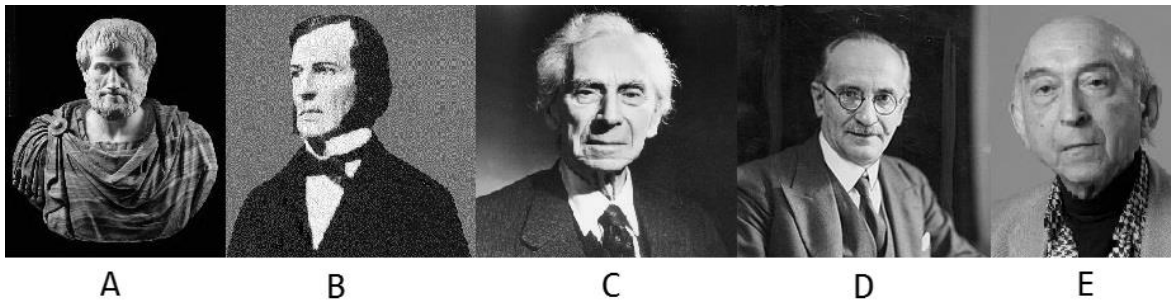
Neste capítulo é apresentado e explanado o conceito de lógica difusa. Isto terá como objetivo a preparação do leitor para uma correta compreensão dos sistemas difusos e conseqüentemente uma ajuda para o entendimento do capítulo seguinte. Este capítulo pretende introduzir, numa primeira fase, uma perspetiva histórica de como surgiu a lógica difusa para que, numa segunda fase, sejam apresentados os conceitos fundamentais inerentes a este tipo de lógica. Por fim, são também enunciadas as aplicações mais significativas onde esta lógica tem efeito no quotidiano.

## 3.1. PERSPETIVA HISTÓRICA

Segundo o dicionário, a lógica é uma disciplina normativa que se propõe determinar as condições de verdade nos diferentes domínios do saber [14]. A lógica tradicional, proposta por Aristóteles (384 - 322 a.C., Figura 18A), é caracterizada por dois princípios, o princípio da não contradição e o princípio do terceiro excluído [15]. O princípio da não contradição afirma que nenhuma proposição pode ser verdadeira e falsa ao mesmo tempo enquanto o princípio do terceiro excluído afirma que uma proposição tem de ser verdadeira ou falsa, não existindo uma terceira opção [16]. Em 1847, George Boole (Figura 18B) atribui valores numéricos às proposições falsas e verdadeiras, 0 para as proposições falsas

e 1 para as verdadeiras. Surgiu assim a lógica booleana que constituiu um avanço significativo na área da computação [15].

Em 1903, Bertrand Russell (Figura 18C) provou que nem todos os problemas podiam ser resolvidos através da lógica tradicional, dando origem ao problema conhecido como *paradoxo de Russell*. No início do século XX, Jan Lukasiewicz (Figura 18D) introduz a lógica multinível [15]. Multinível porque para além dos estados falso (0) e verdadeiro (1), podiam existir estados intermédios com diferentes graus de verdade (e.g. grau de verdade de 0,5). Algumas décadas depois, mais concretamente em 1965, surge o conceito de *conjunto difuso* publicado por Lofti Zadeh (Figura 18E). Esta publicação permitiu mostrar de forma matemática o tratamento dos aspetos imprecisos e ambíguos apresentados no princípio da contradição. Nasce então o conceito de lógica difusa [17].



**Figura 18 Da esquerda para a direita: Aristóteles (A), George Boole (B), Bertrand Russell (C), Jan Lukasiewicz (D) e Lofti Zadeh (E).**

### 3.2. CONCEITOS BÁSICOS

A lógica difusa permite captar e representar o conhecimento humano, designadamente, incertezas ou imprecisões subjacentes ao próprio conhecimento [18]:

- Hoje o dia está quente;
- O filme até nem é mau.

Para além de captar o conhecimento humano com a incerteza associada, a lógica difusa permite também gerar decisões baseadas nesse conhecimento devido ao mecanismo de inferência difusa [17]. Este tipo de lógica confere diferentes graus às proposições, entre a verdade e a falsidade absoluta, sendo absolutamente verdadeiro e falso as classificações extremas possíveis.

Na teoria clássica dos conjuntos, o conceito de pertença de um elemento a um conjunto fica bem definido [19]. Isto é, um dado elemento do universo pertence ou não pertence a um referido conjunto não existindo outras hipóteses. Por exemplo, dado um conjunto  $A$  num universo  $X$ , os elementos do universo apenas pertencem ou não pertencem ao conjunto  $A$ . Isto pode ser expresso pela seguinte função característica:

$$f_A(x) = \begin{cases} 1, & \text{se e somente se } x \in A \\ 0, & \text{se e somente se } x \notin A \end{cases} \quad (1)$$

A teoria dos conjuntos difusos, proposta de L. Zadeh, generalizou a função característica (1) de modo que tivesse a capacidade de poder assumir um número infinito de valores no intervalo  $[0, 1]$ . Esta capacidade de poder assumir diferentes valores é designada de grau de pertença e é determinada através de uma função característica real, denominada de função de pertença. Uma função de pertença permite dar, para cada elemento de um determinado conjunto, valores entre  $[0, 1]$  mediante o peso de cada elemento. Por exemplo, um conjunto difuso  $A$  presente num universo  $X$  é definido por uma função de pertença  $\mu_A$  que mapeia os elementos de  $X$  no intervalo  $[0, 1]$  de acordo com a equação (2):

$$\mu_A: U \rightarrow [0, 1] \quad (2)$$

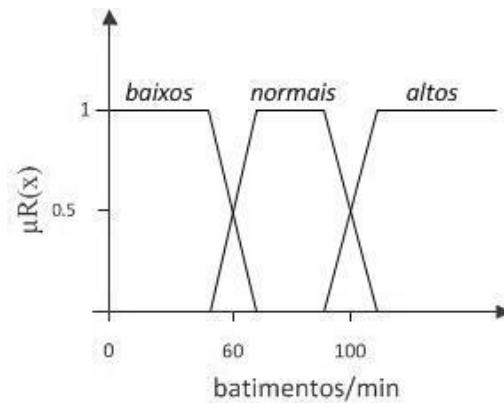
e representado por um conjunto de pares ordenados (equação (3)):

$$A = \{\mu_A(x)/x\}, x \in X \quad (3)$$

onde  $\mu_A(x)$  representa o grau de compatibilidade com o conjunto  $A$ . Um dado elemento pode pertencer a mais do que um conjunto difuso com diferentes graus de pertença.

### 3.3. VARIÁVEIS LINGUÍSTICAS

As variáveis linguísticas são identificadores que podem assumir um de entre vários valores [17]. A título de exemplo pode considerar-se a variável linguística denominada de *ritmo cardíaco*; esta variável pode ser definida por três conjuntos difusos: *baixos*, *normais* ou *altos*. Os conjuntos difusos referidos são definidos pelas suas funções de pertença. Na Figura 19 pode visualizar-se uma possível representação da variável *ritmo cardíaco* e dos conjuntos difusos a ela associados.



**Figura 19 Variável linguística *ritmo cardíaco* e respectivos conjuntos difusos**

Através da Figura 19 verifica-se que cada uma das funções de pertinência dos conjuntos difusos define um intervalo de valores para o *ritmo cardíaco*. Nestes intervalos, os graus de verdade ou de pertinência (observáveis no eixo das ordenadas) variam entre 0 e 1.

De uma forma geral, os valores de uma variável linguística podem ser sentenças numa linguagem especificada [19]. Estas sentenças são construídas a partir de conjuntos difusos (e.g. alto, baixo, quente, frio, pequeno, grande), de conectivos lógicos (*e*, *ou* e negação *não*) e de modificadores (e.g. muito, pouco, lentamente).

As variáveis linguísticas são caracterizadas por uma quintupla (N, T(N), X, G e M), onde:

- N – nome simbólico do conjunto de termos (nome da variável linguística);
- T(N) – conjunto de termos da variável N;
- X – universo de discurso;
- G – regra sintática para gerar os valores de N como uma composição de termos de T(N), conectivos lógicos e modificadores;
- M – regra semântica que fornece o significado dos termos linguísticos em termos dos valores quantitativos.

Retomando o exemplo da Figura 19, *ritmo cardíaco*, a variável linguística seria caracterizada da seguinte forma:

- N – *ritmo cardíaco*;
- T(N) – {*baixos*, *normais*, *altos*};
- X – 0 a 180 batimentos/minuto, por exemplo;
- G – se *ritmo cardíaco* for *baixos*, por exemplo;
- M – associa o valor de G a um conjunto difuso que pode ser por exemplo *braquicardia*.

A Tabela 3 apresenta alguns exemplos de variáveis linguísticas assim como os valores mais comumente usados para estas mesmas variáveis.

**Tabela 3 Exemplos de variáveis linguísticas com valores típicos**

Variáveis linguísticas	Valores típicos
<b>Temperatura</b>	Quente, Confortável, Fria
<b>Peso</b>	Baixo, Médio, Alto
<b>Velocidade</b>	Lenta, Média, Rápida
<b>Distância</b>	Curta, Média, Longa
<b>Idade</b>	Bebé, Criança, Adolescente, Adulto, Idoso
<b>Comida</b>	Intragável, Aceitável, Saborosa
<b>Situação Económica</b>	Pobre, Classe Média, Rico, Milionário

### 3.4. CONJUNTOS DIFUSOS

Ao contrário da teoria dos conjuntos clássicos (lógica booleana) que só permite que uma proposição apenas assuma um de entre dois valores (0 ou 1), a teoria dos conjuntos difusos (lógica difusa) permite que uma proposição assumam um contínuo de valores entre 0 e 1. Desta forma, a teoria dos conjuntos difusos permite que seja possível representar expressões linguísticas qualitativas como negativo, pequeno, grande, positivo, etc [17].

A Figura 20 representa graficamente os conjuntos clássicos e os conjuntos difusos de uma variável linguística denominada de *temperatura*. Como se pode observar, os conjuntos clássicos só assumem dois valores, 0 ou 1. Por sua vez, os conjuntos difusos assumem valores intermédios entre 0 e 1.

Associado aos conjuntos difusos surge o conceito de função de pertença. A função de pertença é um mapeamento matemático de cada valor numérico possível para os conjuntos difusos. Isto é, na representação do conjunto difuso é necessário definir as suas funções de pertença pois são estas que mapeiam os seus elementos no intervalo de  $[0, 1]$ , ficando assim desta forma definido um grau de pertença para cada elemento.

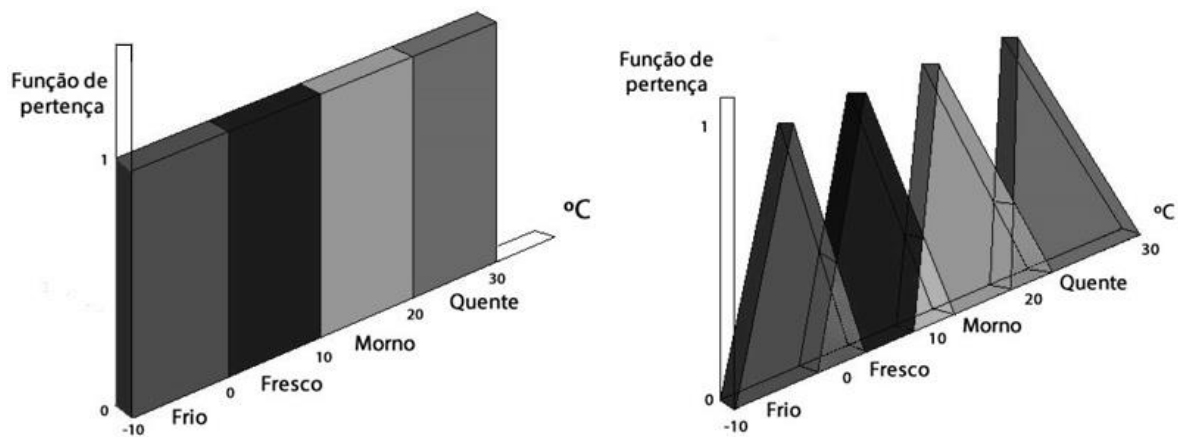


Figura 20 Conjuntos clássicos (esquerda) e conjuntos difusos (direita) que caracterizam por exemplo a temperatura de uma sala [20]

### 3.4.1. MODIFICADORES

O uso de advérbios do género *muito*, *pouco* entre outros, permite modificar matematicamente um conjunto difuso [17]. Por exemplo, considera-se a temperatura numa dada sala. Essa sala encontra-se a 30 °C, o que pode ser considerada uma temperatura quente com um grau de pertença de 0,6. Ou então, a mesma temperatura pode ser considerada muito quente com um grau de pertença de 0,4.

Os modificadores linguísticos apresentados são [17]:

- Concentração – esta operação apresenta uma pequena redução quando o grau de pertença é baixo e uma grande redução quando o grau de pertença é alto. A operação é determinada da forma que se pode ver na equação (4). Exemplo: dado um conjunto difuso de *temperaturas altas*, é possível usar este modificador de forma a criar o conjunto *temperaturas muito altas*.

$$\mu_{Con(A)}(x) = (\mu_A(x))^2 \quad (4)$$

- Expansão – esta operação produz uma expansão grande quando o grau de pertença é baixo e uma baixa expansão quando o grau de pertença é alto. A operação é determinada da forma que se pode ver na equação (5). Exemplo: dado um conjunto difuso de *temperaturas médias*, é possível usar este modificador de forma a criar o conjunto *temperaturas mais ou menos médias*.

$$\mu_{Exp(A)}(x) = (\mu_A(x))^{0,5} \quad (5)$$

- Intensificação – esta operação aumenta o grau de pertença quando o valor é maior que 0,5 e diminui quando é menor que 0,5. A operação é determinada da forma que se pode ver na equação (6). Exemplo: dado um conjunto difuso de *temperaturas médias*, é possível usar este modificador de forma a criar o conjunto *temperaturas realmente médias*.

$$\mu_{Int(A)}(x) = 2(\mu_A(x))^2 \quad \text{para } 0 \leq \mu_A(x) < 0,5 \quad (6)$$

$$\mu_{Int(A)}(x) = 1 - 2(\mu_A(x))^2 \quad \text{para } 0,5 \leq \mu_A(x) \leq 1$$

- Potência – esta operação é uma extensão da operação de concentração. A operação é determinada da seguinte forma como se pode ver na equação (7). Exemplo: dado um conjunto difuso de *temperaturas altas*, é possível usar este modificador (com  $n = 3, 4, 5, \dots$ ) de forma a criar o conjunto *temperaturas muito muito altas*.

$$\mu_{Pot(A)}(x) = (\mu_A(x))^n \quad (7)$$

### 3.4.2. FUNÇÕES DE PERTENÇA

Como já foi referido anteriormente, as funções de pertença permitem representar os conjuntos difusos através de um mapeamento dos seus elementos no intervalo  $[0, 1]$ . As funções de pertença podem ser de dois tipos: contínuas – definidas por funções analíticas – e discretas – valores discretos que correspondem a elementos discretos. Existem diferentes formatos para representar as funções de pertença, no entanto, vão ser apresentados os mais comumente usados e representados nas Figuras 21 e 22 [21]:

- Linear (Figura 21A) – Trata-se do formato de representação mais simples.
- Trapezoidal (Figura 21B) – Onde  $x$  é a variável independente e os parâmetros que definem a função são  $a, b, c$  e  $d$ . A equação que define a função é visível em (8).

$$Trap(x, a, b, c, d) = \begin{cases} 0, & x \leq a \\ 1 - \frac{b-x}{b-a}, & a < x \leq b \\ 1, & b < x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & x > d \end{cases} \quad (8)$$

- Triangular (Figura 21C) – Onde  $x$  é a variável independente e os parâmetros que definem a função são  $e, f$  e  $g$ . A equação que define a função é visível em (9).

$$Tri(x, e, f, g) = \begin{cases} 0, & x \leq e \\ 1 - \frac{f-x}{f-e}, & e < x \leq f \\ \frac{g-x}{g-f}, & f < x \leq g \\ 0, & x > g \end{cases} \quad (9)$$

- Formato em S (Figura 21D) – Onde  $x$  é a variável independente e os parâmetros que definem a função são  $a$ ,  $b$  e  $c$ . A equação que define a função é visível em (10).

$$S(x, a, b, c) = \begin{cases} 0, & x \leq a \\ 2 \left( \frac{x-a}{c-a} \right)^2, & a < x \leq b \\ 1 - 2 \left( \frac{x-c}{c-a} \right)^2, & b < x \leq c \\ 1, & x > c \end{cases} \quad (10)$$

Também é possível representar uma função com formato em S utilizando apenas dois parâmetros como se pode ver na equação (11).

$$S(x, a, b) = \begin{cases} 0, & x \leq a - b \\ \frac{[x - (a - b)]^2}{2b^2}, & a - b < x \leq a \\ 1 - \frac{[(a + b) - x]^2}{2b^2}, & a < x \leq a + b \\ 1, & x > a + b \end{cases} \quad (11)$$

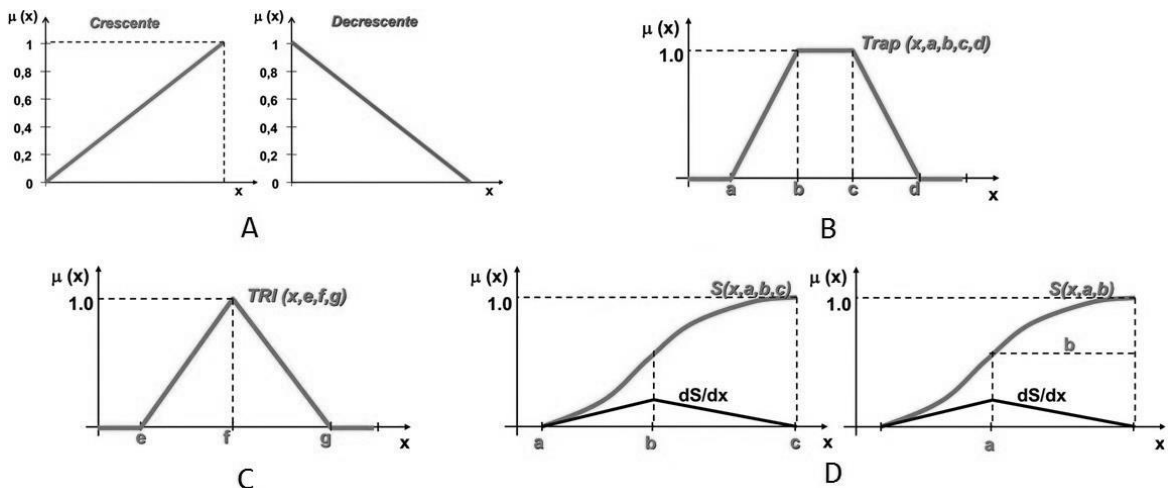


Figura 21 Tipos de funções de pertença [21]

- Formato em  $Z$  (Figura 22A) – Onde  $x$  é a variável independente e os parâmetros que definem a função são  $a$  e  $b$ . A equação que define a função é visível em (12). De realçar que a função de pertença com formato em  $Z$  é o complemento da função de pertença com formato em  $S$  ( $Z(x, a, b) = 1 - S(x, a, b)$ ).

$$Z(x, a, b) = \begin{cases} 1, & x \leq a - b \\ 1 - \frac{[x - (a - b)]^2}{2b^2}, & a - b < x \leq a \\ \frac{[(a + b) - x]^2}{2b^2}, & a < x \leq a + b \\ 0, & x > a + b \end{cases} \quad (12)$$

- Formato em  $\Pi$  (Figura 22B) – Trata-se da junção das curvas  $S$  e  $Z$ . Onde  $x$  é a variável independente e os parâmetros que definem a função são  $a$  e  $b$ . A equação que define a função é visível em (13).

$$\Pi = \begin{cases} S(x, a - \frac{b}{2}, \frac{b}{2}), & x \leq a \\ Z(x, a + \frac{b}{2}, \frac{b}{2}), & x > a \end{cases} \quad (13)$$

- Gaussiana (Figura 22C) – Trata-se da uma distribuição normal. Onde  $x$  é a variável independente,  $\mu$  é a média e  $\sigma$  é o desvio padrão. A equação que define a função é visível em (14).

$$G(x, \mu, \sigma) = e^{-\frac{(x-\mu)^2}{\sigma^2}} \quad (14)$$

- Sigmoidal (Figura 22D) – Onde  $x$  é a variável independente e os parâmetros que definem a função são  $a$  e  $b$ . A equação que define a função é visível em (15).

$$Sig(x, a, b) = \frac{1}{1 + e^{-a(x-b)}} \quad (15)$$

- Singleton (Figura 22E) – Onde  $x$  é a variável independente e o parâmetro que define a função é o  $a$ . Este formato simplifica os cálculos para produzir valores de saída. A equação que define a função é visível em (16).

$$Sgl(x, a) = \begin{cases} 1, & x = a \\ 0, & x \neq a \end{cases} \quad (16)$$

- Irregulares (Figura 22F) – Este formato é usado quando os formatos existentes não conseguem exprimir o significado representado pelo conjunto ao longo do universo.

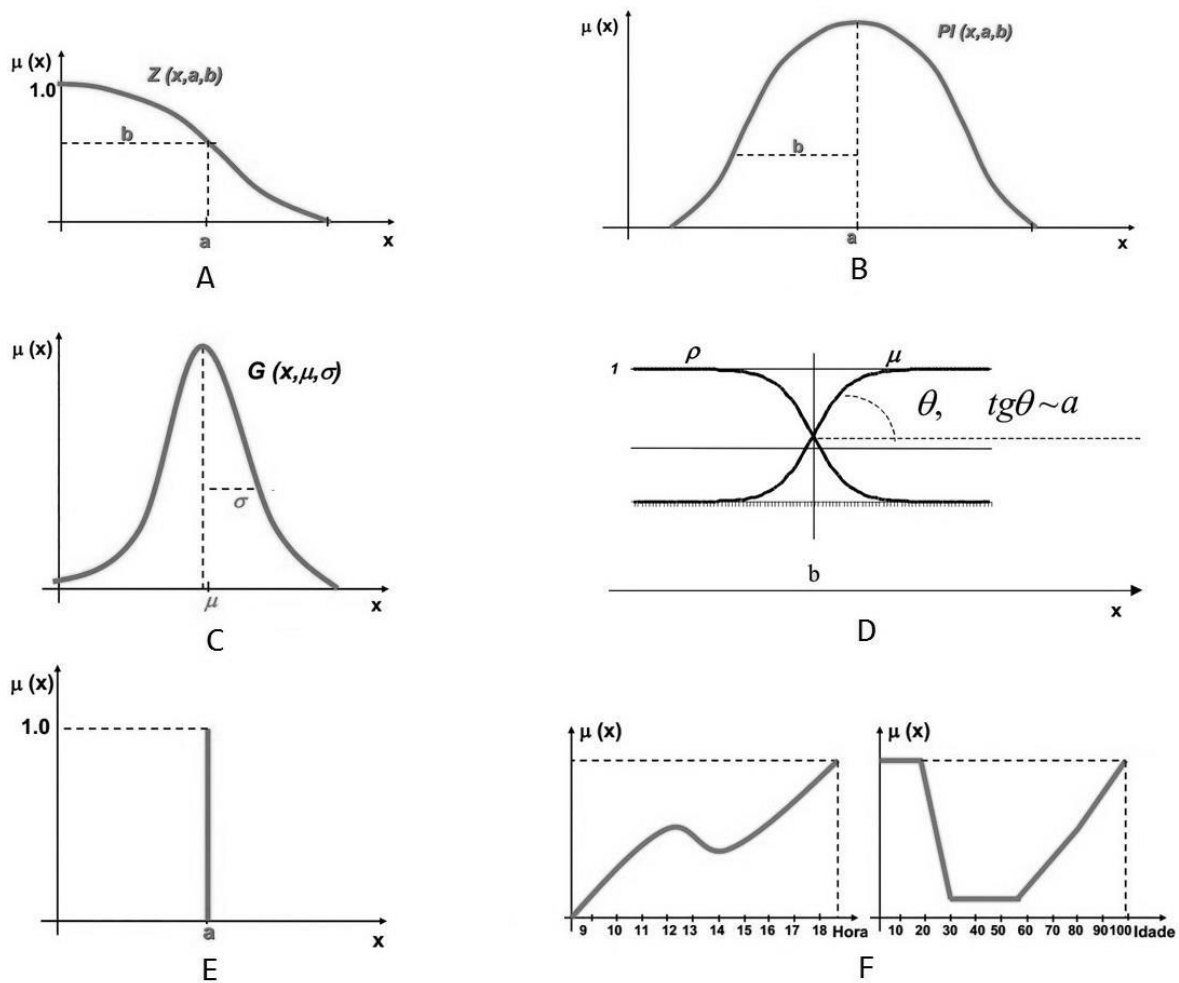


Figura 22 Tipos de funções de pertença [21]

### 3.5. PROPRIEDADES DOS CONJUNTOS DIFUSOS

Apesar da definição de conjunto difuso ser diferente da definição de conjunto clássico, muitas das propriedades dos conjuntos clássicos verificam-se nos conjuntos difusos. Logo verifica-se também que os conjuntos clássicos não são mais do que casos particulares dos conjuntos difusos. A seguir são apresentadas algumas propriedades importantes relativas aos conjuntos difusos [22]:

- Suporte de um conjunto difuso – O suporte de um conjunto difuso  $A$  é um conjunto clássico que contém todos os elementos de  $A$  com grau de pertença positivo (17).

$$S(A) = \{x \in A \mid \mu_A(x) > 0\} \quad (17)$$

- Conjunto difuso convexo – Um conjunto difuso diz-se convexo se for representado por uma função de pertença convexa.

- Largura de um conjunto difuso – A largura de um conjunto difuso  $A$  com suporte  $S(A)$  é dada por (18):

$$Largura(A) = \sup(S(A)) - \inf(S(A)) \quad (18)$$

onde  $\sup$  e  $\inf$  designam o supremo e o ínfimo de  $S(A)$ . No caso particular do conjunto  $S(A)$  ser fechado,  $\sup$  e  $\inf$  podem ser substituídos pelo máximo ( $\max$ ) e pelo mínimo ( $\min$ ).

- Altura de um conjunto difuso – A altura  $h$  de um conjunto  $A$  corresponde ao maior grau de pertença entre todos os elementos do conjunto (19).

$$h(A) = \max\{\mu_A(x), \forall_x\} \quad (19)$$

- Corte de nível  $\alpha$  – Um corte de nível  $\alpha$  é o conjunto  $A_\alpha$  (conjunto clássico) dos elementos de um conjunto difuso  $A$ , num universo de discurso  $U$ , que têm um grau de pertença superior a  $\alpha$  (20).

$$A_\alpha = \{x \in U \mid A(x) > \alpha\} \quad (20)$$

- Núcleo de um conjunto difuso – O núcleo de um conjunto difuso é o conjunto clássico  $Nuc(A)$  definido pela propriedade (21).

$$Nuc(A) = \{x \in U \mid \mu_A(x) = 1\} \quad (21)$$

- Igualdade de dois conjuntos difusos – Dois conjuntos difusos,  $A$  e  $B$ , definidos no mesmo universo de discurso  $U$  são iguais se, e só se, tiverem a mesma função de pertença para todo o  $x$ , conforme mostra a propriedade (22).

$$\forall_{x \in U}: \mu_A(x) = \mu_B(x) \quad (22)$$

- Subconjunto de um conjunto difuso – O conjunto difuso  $A$  é um subconjunto difuso de  $B$  se, e só se, a função de pertença do conjunto  $A$  for menor ou igual à função de pertença do conjunto  $B$  para todo o  $x$ , conforme mostra a propriedade (23).

$$A \subseteq B \Leftrightarrow \forall_{x \in U}: \mu_A(x) \leq \mu_B(x) \quad (23)$$

### 3.6. OPERAÇÕES BÁSICAS SOBRE CONJUNTOS DIFUSOS

As operações dos conjuntos difusos são análogas às operações dos conjuntos clássicos. As operações mais elementares dos conjuntos difusos são a *união*, a *intersecção* e o

*complemento*. Enquanto nos conjuntos clássicos estes operadores são únicos, nos conjuntos difusos para cada uma destas operações existem vários operadores [18]. Suponha-se então dois conjuntos difusos,  $A$  e  $B$ , definidos num universo de discurso  $U$ , como se mostra a seguir nas equações (24) e (25):

$$A = \{(x, \mu_A(x)): x \in U\} \quad (24)$$

$$B = \{(x, \mu_B(x)): x \in U\} \quad (25)$$

- União – A operação difusa de *união* ( $\cup$ ) é representada por uma co-norma triangular, ou S-norma,  $\oplus$ . Formalmente tem-se (26) e (27) [23]:

$$\begin{aligned} \oplus: [0,1] \times [0,1] &\rightarrow [0, 1] \\ (a, b) &\rightarrow a \oplus b \end{aligned} \quad (26)$$

$$\forall_{x \in U}: \mu_{A \cup B}(x) = \mu_A(x) \oplus \mu_B(x) \quad (27)$$

As S-normas são caracterizadas pelas seguintes propriedades (28):

- S-1, comutatividade:  $a \oplus b = b \oplus a$
  - S-2, associatividade:  $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
  - S-3, monotonicidade:  $(a \leq c) \wedge (b \leq d) \Rightarrow a \oplus b \leq c \oplus d$
  - S-4, elemento neutro 0:  $a \oplus 0 = a$
- ,  $\forall_{a,b,c,d \in [0,1]}$  (28)

Os operadores *máximo* (29), *adição algébrica* (30) e *adição limitada* (31) constituem representações habituais da operação de *união* difusa.

$$\forall_{x \in U}: \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (29)$$

$$\forall_{x \in U}: \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \quad (30)$$

$$\forall_{x \in U}: \mu_{A \cup B}(x) = \min(1, \mu_A(x) + \mu_B(x)) \quad (31)$$

- Intersecção – A operação difusa de *intersecção* ( $\cap$ ) é representado por uma norma triangular, ou T-norma,  $\otimes$ . Formalmente tem-se (32) e (33) [23]:

$$\otimes: [0,1] \times [0,1] \rightarrow [0, 1] \quad (32)$$

$$(a, b) \rightarrow a \otimes b$$

$$\forall_{x \in U}: \mu_{A \cap B}(x) = \mu_A(x) \otimes \mu_B(x) \quad (33)$$

As T-normas são caracterizadas pelas seguintes propriedades (34):

$$\begin{aligned} \text{T-1, comutatividade:} & \quad a \otimes b = b \otimes a \\ \text{T-2, associatividade:} & \quad (a \otimes b) \otimes c = a \otimes (b \otimes c) \\ \text{T-3, monotonicidade:} & \quad (a \leq c) \wedge (b \leq d) \Rightarrow a \otimes b \leq c \otimes d \\ \text{T-4, elemento neutro 1:} & \quad a \otimes 1 = a \end{aligned} \quad , \forall_{a,b,c,d \in [0,1]} \quad (34)$$

Nas operações de *intersecção* difusa, os operadores *mínimo* (35) e produto algébrico (36) constituem as operações mais comuns.

$$\forall_{x \in U}: \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (35)$$

$$\forall_{x \in U}: \mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x) \quad (36)$$

- Complemento – A operação difusa de *complemento* ( $c$ ) é representada por uma c-norma, tal como se apresenta a seguir em (37) e (38) [23]:

$$\begin{aligned} c: [0,1] & \rightarrow [0,1] \\ a & \rightarrow c(a) \end{aligned} \quad (37)$$

$$\forall_{x \in U}: \mu_{\bar{A}}(x) = c(\mu_A(x)) \quad (38)$$

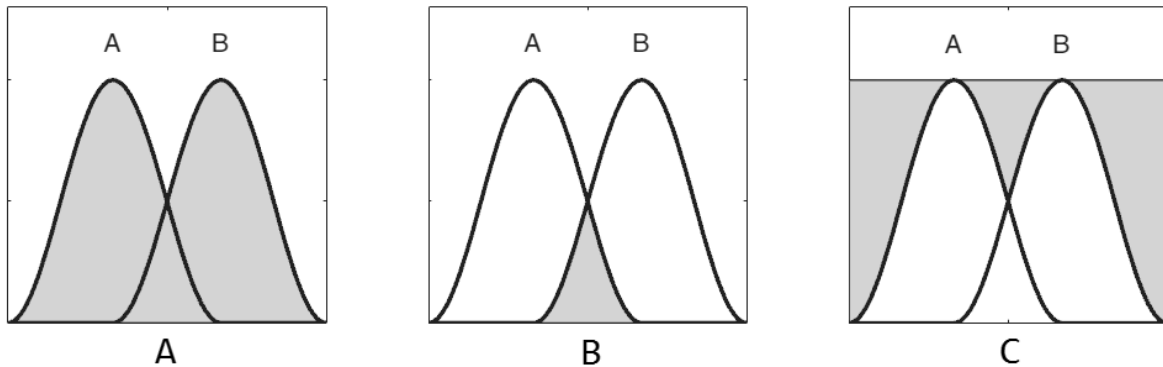
As C-normas são caracterizadas pelas seguintes propriedades (39):

$$\begin{aligned} \text{C-1:} & \quad c(0) = 1 \\ \text{C-2:} & \quad a < b \Rightarrow c(a) > c(b) \\ \text{C-3} & \quad c(c(a)) = a \end{aligned} \quad , \forall_{a,b \in [0,1]} \quad (39)$$

O operador *complemento* é definido conforme a equação (40):

$$\forall_{x \in U}: \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (40)$$

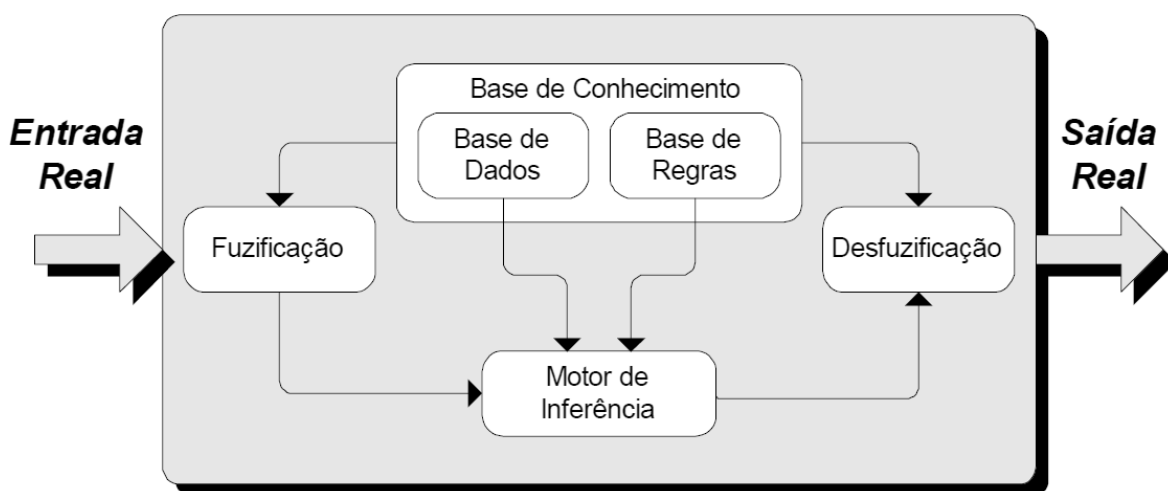
De modo a facilitar a compreensão das operações básicas sobre conjuntos difusos, é apresentada a Figura 23. Nesta figura é possível visualizar as operações atrás enunciadas (*união, intersecção e complemento*).



**Figura 23 Operações básicas sobre conjuntos difusos. A – União, B – Intersecção e C – Complemento [24]**

### 3.7. SISTEMAS DE INFERÊNCIA DIFUSA

Um sistema de inferência difusa (em *inglês*, *Fuzzy Inference System – FIS*) define um mapeamento não linear de um vetor de dados de entrada numa saída escalar utilizando regras difusas. O processo de mapeamento envolve funções de pertinência de entrada/saída, operadores difusos, regras difusas *se-então*, agregação dos conjuntos de saída e a *desfuzificação*. Na Figura 24 é possível visualizar o diagrama do modelo geral de um sistema de inferência difusa.



**Figura 24 Diagrama de um sistema de inferência difusa [23]**

Como se pode verificar na Figura 24, um sistema difuso é composto por quatro elementos fundamentais: um módulo de fuzificação, uma base de conhecimento, um motor de inferência e um módulo de desfuzificação [23][25]. A seguir, vão ser explanados cada um dos elementos apresentando para isso as propriedades e parâmetros do projeto de sistemas difusos.

### 3.7.1. MÓDULO DE FUZIFICAÇÃO

O *módulo de fuzificação* consiste num processo que converte as entradas numéricas em conjuntos difusos, num dado universo de discurso. Trata-se de um módulo fundamental pois a maioria dos sistemas fornecem dados numéricos que necessitam de ser convertidos em conjuntos difusos para serem aplicados nos sistemas difusos. Posto isto, é necessário então *fuzificar* todos os dados numéricos de entrada. Simbolicamente, um dado numérico  $x^*$  é convertido num conjunto difuso  $\tilde{X}^*$  através de um *fuzificador*, como mostra a equação (41) [23].

$$\tilde{X}^* = \text{fuzzificador}(x^*) \quad (41)$$

Consoante o tipo de inferência utilizado é escolhida uma estratégia de *fuzificação*, fomentada em duas regras de inferência [22]:

- *Modus ponens* generalizado – É usado um procedimento *se-então* que representa implicitamente uma relação difusa. Esta regra obedece ao seguinte esquema simbólico:

- $A_1$  é  $Q_1$ ;
- se  $A_1$  é  $P_1$  então  $A_2$  é  $P_2$ ;
- $\therefore A_2$  é  $Q_2$ .

Onde  $A_1$  e  $A_2$  são designações simbólicas de objetos e  $P_1$ ,  $P_2$ ,  $Q_1$  e  $Q_2$  são propriedades. Para melhor compreensão do leitor, é apresentado o seguinte exemplo:

- A corrente elétrica é muito intensa;
  - se a corrente elétrica é intensa então os efeitos eletrodinâmicos são fortes;
  - $\therefore$  os efeitos eletrodinâmicos são muito fortes.
- Regra composicional de inferência – Pode ser considerada um caso especial do *modus ponens* generalizado. É utilizada uma relação difusa que representa explicitamente a

associação entre duas proposições difusas. Simbolicamente, esta regra obedece ao seguinte esquema:

- $A_1$  é  $Q_1$ ;
- $A_1$  R  $A_2$ ;
- $\therefore A_2$  é  $Q_2$ .

Onde  $A_1$  R  $A_2$  significa que  $A_1$  está relacionado com  $A_2$  através de uma relação difusa R com  $\mu_R$ . Segue-se então o seguinte exemplo:

- m é um número pequeno;
- m é um pouco maior que n;
- $\therefore$  n é um número bastante pequeno.

Onde “*um pouco maior que*” é uma relação difusa.

### 3.7.2. BASE DE CONHECIMENTO

A base de conhecimento de um sistema difuso é constituída por dois blocos: o bloco da *base de regras* e o bloco da *base de dados*.

- **Base de regras**

Para modelar difusamente um sistema, é necessário fazer a sua representação tendo por base um conjunto de regras que permitem descrever o sistema. Este conjunto de regras é denominado de *base de regras*. As regras referidas apresentam o formato da equação (42):

$$SE \text{ (estado do sistema)} \text{ ENTÃO (saída do sistema)} \quad (42)$$

onde a primeira parte da regra é a *premissa* e a segunda parte da regra é a *conclusão*. A *premissa* é habitualmente designada por *antecedente* e descreve o estado do sistema. Por sua vez, a *conclusão* é designada de *consequente* e representa as saídas do sistema em resultado das entradas do mesmo.

O desenvolvimento de uma base de regras para modelar difusamente um sistema consiste essencialmente em:

- Seleção das variáveis linguísticas de entrada e de saída do modelo – a seleção das variáveis linguísticas depende essencialmente do sistema. Isto é, com base no

conhecimento disponível sobre o sistema, são escolhidas as variáveis linguísticas que representam as entradas e saídas do mesmo.

- Seleção do formato das regras condicionais – o formato das regras pode ser de dois tipos: *regras de Mamdani* ou *regras de Takagi-Sugeno*. As *regras de Mamdani* caracterizam-se pelo facto de o *consequente*, assim como o *antecedente*, ser um conjunto difuso. O formato deste tipo de regra pode ser visto na equação (43):

$$SE (X \text{ é } A) ENTÃO (Y \text{ é } B) \quad (43)$$

onde os termos linguísticos  $A$  e  $B$  são conjuntos difusos representados pelas funções de pertinência  $\mu_A$  e  $\mu_B$  respetivamente. Por sua vez, nas *regras de Takagi-Sugeno*, apenas os *antecedentes* são conjuntos difusos. Os *consequentes* são definidos como funções dos *antecedentes*, como pode ser visto na equação (44):

$$SE (X_1 \text{ é } A) E (X_2 \text{ é } B) E \dots E (X_m \text{ é } m) ENTÃO y = f(x_1, x_2, \dots, x_m) \quad (44)$$

onde  $x_1, x_2, \dots, x_m$  são valores numéricos associados a cada uma das variáveis linguísticas  $X_1, X_2, \dots, X_m$ . Este tipo de regra é caracterizado pelo valor numérico do *consequente* que é obtido através de uma função  $f$  contendo os valores numéricos dos *antecedentes*.

- Síntese de regras – a síntese de regras é o parâmetro mais importante deste módulo. A seleção das regras a incluir no modelo pode ser feita de duas formas: *manual* e *automática*. A forma *manual* permite que as regras sejam obtidas através do projetista do sistema que utiliza o conhecimento e a experiência que possui do mesmo. Porém, esta forma de obtenção das regras possui a desvantagem de que projetistas diferentes podem apresentar regras diferentes. Isto verifica-se pois os projetistas podem conhecer o sistema de forma diferente. Por sua vez, na forma *automática*, é necessário recolher os dados, analisá-los e conforme a sua análise, verificar quais as relações entre as variáveis do sistema.
- Seleção de termos linguísticos – a seleção dos termos linguísticos é efetuada *a posteriori*, isto é, depois da definição da base de regras e da base de dados a seguir apresentada. Os termos linguísticos a usar podem ser do género *pequeno*, *médio*, *grande*, *zero* entre outros. A estes termos linguísticos podem ser acrescentados os modificadores como por exemplo *muito grande* ou *muito pequeno*. Idealmente, devem ser usados entre 5 a 9 termos linguísticos, sendo o 7 o número ideal. Um número muito pequeno pode resultar numa precisão reduzida do modelo. Por sua vez, um número

muito alto de termos linguísticos pode causar problemas em termos de memorização dos mesmos. Geralmente, o número de termos linguísticos é ímpar para poder utilizar um termo médio entre dois extremos (e.g. *negativo*, *zero* e *positivo*).

- **Base de dados**

A principal função da *base de dados* é armazenar e fornecer a informação necessária ao correto funcionamento dos diversos blocos de um sistema difuso (*módulo de fuzificação*, *base de regras* e *módulo de desfuzificação*) [23]. Entre essas informações encontram-se as discretizações e normalizações dos universos de discurso, as partições dos espaços de entrada e saída (definição do número de termos que vão caracterizar as variáveis linguísticas) e as definições das funções de pertença [26].

### 3.7.3. MOTOR DE INFERÊNCIA

O motor de inferência tem como função determinar o valor difuso de saída, com base nos parâmetros estabelecidos no *módulo de fuzificação* e na *base de conhecimento*. Para isso, existem duas regras de inferência que são empregues nos mecanismos de inferência: a regra *Modus ponens* generalizado e a regra composicional de inferência. Estas regras foram anteriormente explicadas na Secção 3.7.1.

O projeto de um motor de inferência envolve essencialmente três etapas [23]:

- Seleção de conectivas difusas – Consiste na seleção dos operadores difusos de *união* (definida por uma S-norma), *intersecção* (definida por uma T-norma) e *complemento* (definida por uma C-norma). Estas operações foram anteriormente abordadas na Secção 3.6.
- Representação de um conjunto de regras – existem duas abordagens para representar o conjunto de regras utilizadas num mecanismo de inferência de um sistema difuso: a inferência baseada na combinação de regras e a inferência baseada em regras individuais. Na primeira, as relações difusas são agregadas formando uma única relação que descreve o significado global do conjunto de regras. Por sua vez, na segunda abordagem a inferência ocorre da seguinte forma: em primeiro lugar, determina-se o grau de pertença do valor numérico em causa em cada um dos conjuntos difusos que descrevem o antecedente da regra; o antecedente *fuzificado* é obtido pela aplicação dos operadores lógicos de intersecção, união e negação difusa aos graus de pertença obtidos; em seguida, os conjuntos difusos de saída, relativos a cada

uma das regras, são transformados de acordo com a operação de implicação definida e com o valor de ativação do antecedente correspondente [23].

- Seleção de um método de inferência – O processo de inferência gera um grau de pertinência de ativação para cada regra, onde ocorre a implicação mediante a aplicação do operador escolhido. O significado das implicações presentes nas regras pode ser dado por diferentes métodos entre os quais se encontram o método de implicação de *Lucasiewicz*, o método de implicação de *Zadeh* e o método de implicação de *Mamdani* [22]. O método de implicação de *Mamdani* é o mais popular e baseia-se na operação de intersecção que é definida pelo operador *mínimo*.

### 3.7.4. MÓDULO DE DESFUZIFICAÇÃO

O *módulo de desfuzificação* permite transformar o conjunto difuso obtido através do *motor de inferência* e da *base de conhecimento* num valor real. Isto torna-se útil quando, em problemas de modelação e controlo, é necessário possuir um valor numérico para efetuar a respetiva modelação e controlo. Existem diferentes métodos de *desfuzificação*, como por exemplo [22]:

- Centro de massa / centro de área / centro de gravidade (centróide) – É o método mais popular de todos. Trata-se da média dos pontos de valor de pertinência não nulo pesada pela importância do respetivo valor de pertinência. Logo, se a função de pertinência do conjunto difuso  $U$  for designada como  $\mu(u)$ , então o centro de massa  $u^*$ , correspondente ao valor a tomar como valor colapsado de  $U$ , será dado pela equação (45).

$$u^* = \frac{\int_U u \mu(u) du}{\int_U \mu(u) du} \quad (45)$$

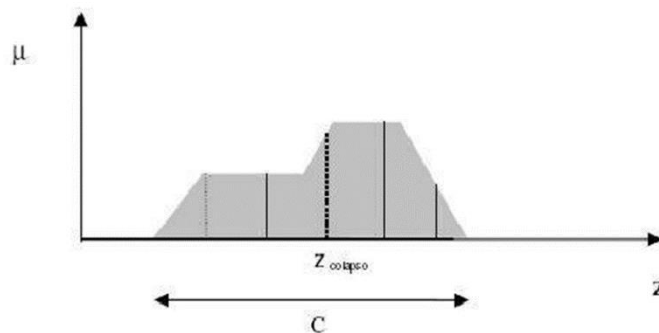


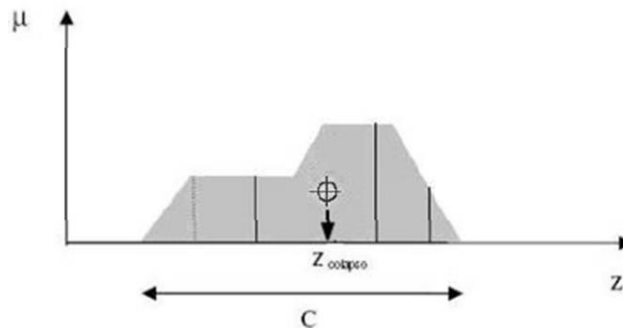
Figura 25 Método do centro de massa

A Figura 25 ilustra graficamente o método do centro de massa. De realçar que a área sobreposta dos dois conjuntos é contada apenas uma única vez.

- Centro das somas das áreas – De forma a abreviar os cálculos, algumas aplicações práticas, em vez de se basearem no conjunto difuso  $U$ , baseiam os cálculos nas áreas dos conjuntos truncados definidas por  $\mu_{TR}^{(k)}(u)$ . A expressão que permite obter o valor *desfuzificado*  $u^*$  é apresentada em (46).

$$u^* = \frac{\sum_k \int_U u \mu_{TR}^{(k)}(u) du}{\sum_k \int_U \mu_{TR}^{(k)}(u) du} \quad (46)$$

Esta expressão corresponde ao cálculo do centro de massa dos centros de massa de cada área individual. Neste caso, as áreas sobrepostas de dois conjuntos truncados contam duas vezes. A Figura 26 ilustra graficamente o método do centro das somas das áreas.

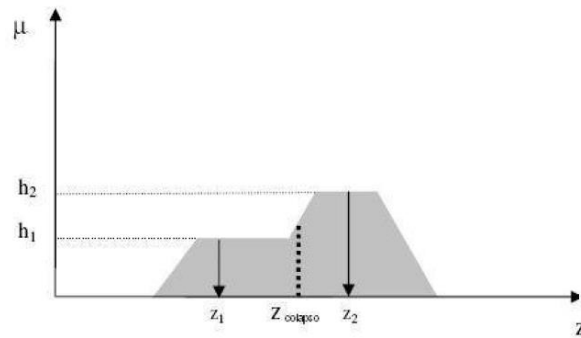


**Figura 26 Método do centro da soma das áreas**

- Centro da média das alturas / média dos máximos – Por fim, a média dos máximos calcula a média pesada dos máximos de cada subconjunto truncado. Como o máximo de cada um dos subconjuntos corresponde a um intervalo, toma-se então o ponto central desse intervalo. Seja então  $c^{(k)}$  o ponto central do máximo do subconjunto difuso  $(k)$ . O seu valor de pertinência é portanto, o máximo atingido nesse subconjunto:  $\mu(c^{(k)}) = \max\{\mu(u)\}$ . O valor *desfuzificado* do conjunto é calculado de acordo com a expressão (47).

$$u^* = \frac{\sum_k c^{(k)} \cdot \mu(c^{(k)})}{\sum_k \mu(c^{(k)})} \quad (47)$$

A Figura 27 ilustra graficamente o método da média dos máximos.



**Figura 27 Método da média dos máximos**

Associadas aos métodos de *desfuzificação* surgem as seguintes propriedades:

- Continuidade – uma pequena alteração na entrada deve ter uma pequena alteração na resposta;
- Inambiguidade – a resposta a uma entrada deve ser única e bem definida;
- Plausibilidade – o valor *desfuzificado* deve corresponder a um elevado valor de pertença no conjunto difuso de resposta;
- Complexidade computacional – o cálculo do valor *desfuzificado* deve ser produzido de forma eficiente, compatível com as aplicações acopladas ao sistema implementado.

### 3.8. APLICAÇÕES

Dadas as suas grandes potencialidades, a lógica difusa é utilizada para controlar máquinas e produtos de consumo. Nas aplicações adequadas, os sistemas difusos são de simples desenvolvimento e podem ser entendidos e implementados por pessoas que não são especialistas na área do controlo. Na maioria dos casos, uma pessoa com conhecimentos técnicos médios é capaz de projetar um sistema difuso. O sistema de controlo não será ótimo mas será aceitável. A lógica difusa também é utilizada em aplicações onde o poder computacional é limitado e apenas é necessário um controlo simples. Este tipo de controlo não é uma resposta para todos os problemas mas apenas para aqueles onde a simplicidade e a rapidez de implementação são importantes. A seguir são apresentadas algumas aplicações que usam a lógica difusa [27]:

- Controlo do ambiente:
  - Equipamentos de ar condicionado;
  - Humidificadores.
- Eletrodomésticos:
  - Máquinas de lavar e secar;

- Aspiradores;
- Torradeiras;
- Micro-ondas;
- Frigoríficos.
- Equipamento eletrónico:
  - Televisões;
  - Fotocopiadoras;
  - Máquinas fotográficas e de vídeo com focagem automática, exposição automática e anti vibração.
- Indústria automóvel:
  - Controlo do clima de um automóvel;
  - Caixas de velocidades automáticas;
  - Direção nas quatro rodas;
  - Sistemas de controlo dos assentos e espelhos.

Para melhor entender como funciona a lógica difusa, é explicado sucintamente o que acontece nas máquinas de lavar, aspiradores, câmaras de vídeo, equipamentos de ar condicionado e nas câmaras fotográficas:

- Numa máquina de lavar, é feita a distinção entre a sujidade da roupa e o tipo de tecido para determinar, automaticamente, as necessidades de água, detergente e potência;
- No caso do aspirador, estes são capazes de variar a pressão da sucção tendo em conta a existência de sujidade no chão e a quantidade de pó no mesmo;
- As câmaras de vídeo permitem gerar imagens “limpas” eliminando as distorções causadas pela não habilidade do utilizador ou pela movimentação dos objetos;
- Os equipamentos de ar condicionado variam a intensidade da operação dependendo do número de pessoas presentes no espaço, mantendo a temperatura homogénea constante;
- Por sua vez, nas câmaras fotográficas a focagem automática, *zoom* e a autoexposição é feita através da lógica difusa.

Como se pode ver, trata-se de um método de controlo usado em diversas aplicações devido à sua simplicidade e ao seu baixo custo.

## 4. ESTUDO E ANÁLISE DOS COMPONENTES DO SDA

Este capítulo engloba a fase inicial do planeamento do sistema. Para além do funcionamento pretendido para o sistema, nele são abordados vários assuntos inerentes ao sistema como a base de dados utilizada e as suas características; é também explicado o *software* usado para manipulação dos dados provenientes da base de dados. Numa fase intermédia é abordada a remoção de ruído presente no sinal eletrocardiográfico e, seguidamente, a identificação do complexo QRS. Por fim, é apresentado o sistema difuso usado assim como o *software* necessário para o seu desenvolvimento.

### 4.1. FUNCIONAMENTO PRETENDIDO

Como já fora referido anteriormente, pretende-se com este projeto desenvolver um sistema que seja capaz de identificar um conjunto de tipo de batimentos em sinais eletrocardiográficos. Para cumprir o funcionamento especificado prevê-se o desenvolvimento de funções e *scripts* em ambiente *MATrix LABORatory* (MATLAB). Numa fase posterior, prevê-se ainda o desenvolvimento de uma aplicação gráfica que torne mais intuitivo e acessível o acesso e manuseamento do sistema.

O ponto de partida para o desenvolvimento do sistema consiste na escolha da base de dados a usar. A plataforma *online PhysioNet* [28] disponibiliza livre acesso a um largo conjunto de sinais fisiológicos (*PhysioBank*) para além de fornecer também *software open-source* (*PhysioToolkit*) que permite o manuseamento dos mesmos. De todas as bases de dados disponíveis no *PhysioBank*, a escolhida foi a *Massachusetts Institute of Technology* (MIT) - *Boston's Beth Israel Hospital* (BIH) *Arrhythmia Database*. A escolha recaiu no facto de ser uma base de dados que contém vários tipos de batimentos, entre os quais os utilizadas neste projeto, e também por ser uma base de dados que se encontra disponível já há alguns anos, o que garante alguma fiabilidade na utilização dos dados. Além disto, esta base de dados disponibiliza anotações que identificam o tipo de batimentos presentes.

Para além da base de dados usada, também vão ser usadas diversas ferramentas que vão permitir o desenvolvimento do sistema. Nesse conjunto de ferramentas estão incluídas o *PhysioBank Automated Teller Machine* (ATM), o *PhysioToolkit* e o MATLAB. No MATLAB irão ser utilizadas várias ferramentas como a *Fuzzy Logic Toolbox* (para o desenvolvimento do sistema difuso) e o *Graphical User Interface Development Environment* (GUIDE) que permite a criação de interfaces gráficas.

#### **4.2. MIT-BIH ARRHYTHMIA DATABASE**

Laboratórios existentes no *Boston's Beth Israel Hospital*, agora denominado de *Beth Israel Deaconess Medical Center*, e no *Massachusetts Institute of Technology* têm suportado pesquisas em análise de arritmias e em outros assuntos relacionados desde 1975 [29]. Um dos primeiros resultados desta pesquisa foi a MIT-BIH *Arrhythmia Database* que foi concluída e começou a ser distribuída em 1980. Esta base de dados foi a primeira do género disponível para a avaliação de detetores de arritmias e tem vindo a ser usada com esse propósito e para investigação básica na dinâmica cardíaca.

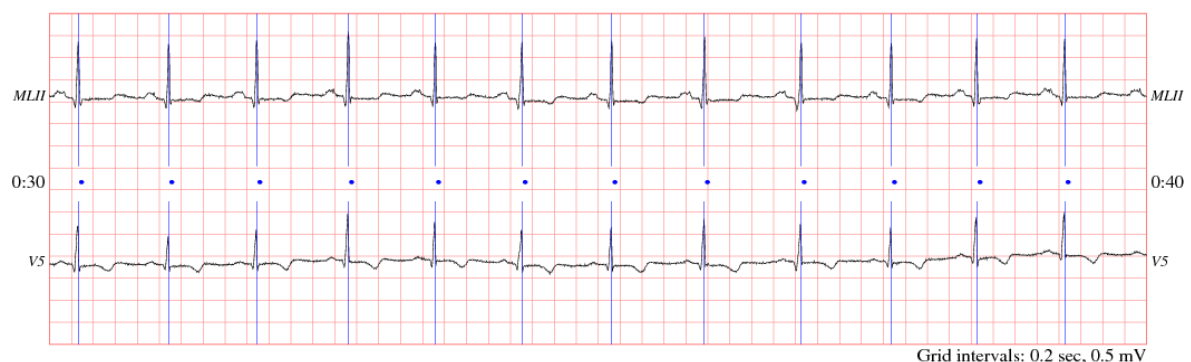
A MIT-BIH *Arrhythmia Database* possui 48 excertos de meia hora (lista completa dos excertos pode ser vista em [29]). Cada excerto possui dois canais correspondentes a duas derivações diferentes. Estes excertos foram obtidos de 47 indivíduos estudados pelo BIH *Arrhythmia Laboratory* entre 1975 e 1979. Dos 47 indivíduos, 25 eram homens com idades compreendidas entre os 32 e 89 anos. Os restantes 22 indivíduos eram mulheres cuja faixa etária ia dos 23 até aos 89 anos. De salientar que existem dois excertos (201 e 202) que são provenientes do mesmo indivíduo masculino.

Dos 48 excertos, 23 deles (numerados de 100 a 124 com alguns números em falta) foram selecionados aleatoriamente de um conjunto de 4000 registos com duração de 24h. Este conjunto de 4000 registos foi obtido a partir de uma população mista de pacientes onde 60% provem de pacientes internos e os restantes 40% de pacientes externos (todos com origem no *Boston's Beth Israel Hospital*). Os restantes 25 excertos (numerados de 200 a 234 com alguns números em falta) foram obtidos a partir do mesmo conjunto mas cuidadosamente escolhidos de forma a incluir as arritmias menos comuns mas clinicamente importantes e que não se encontram bem representadas numa pequena amostra aleatória.

#### 4.2.1. CONFIGURAÇÃO DAS DERIVAÇÕES DOS ECGs

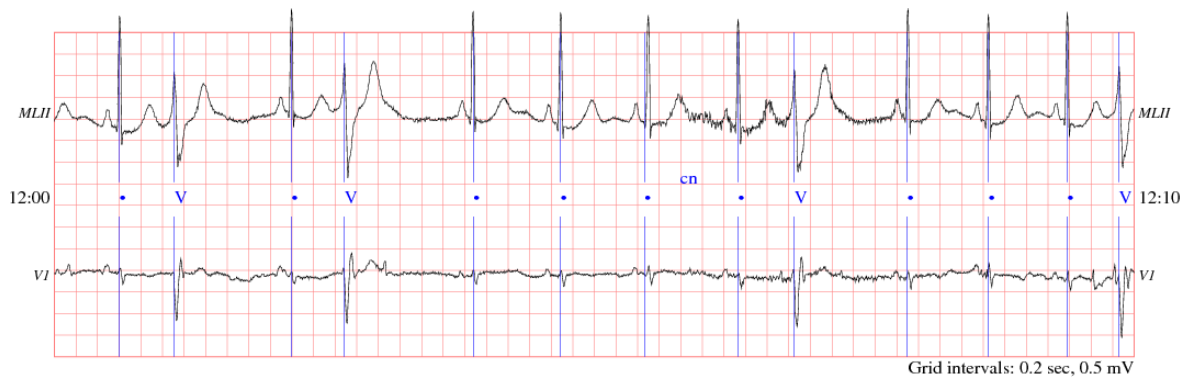
Como já foi referido, cada excerto de meia hora é composto por dois canais (Figura 28). Na maior parte dos excertos, a parte superior do sinal diz respeito à derivação modificada dos membros II (MLII). Esta é obtida colocando os elétrodos sobre o peito. Por sua vez, a parte inferior do sinal é usualmente uma derivação modificada V1 (ocasionalmente V2 ou V5 e em um dos casos V4). Nesta derivação, assim como no caso anterior, os elétrodos também são colocados sobre o peito. Este tipo de configuração é rotineiramente usado pelos laboratórios de arritmias do BIH. Relativamente aos complexos QRS, estes são mais acentuados na parte superior do sinal, isto é, na derivação MLII [30].

Com se pode ver na Figura 28, o excerto do registo 100 possui os tais dois canais anteriormente referidos. Neste caso esses dois canais dizem respeito à derivação MLII e à derivação V5. Ainda nesta figura, verifica-se também que os complexos QRS são mais acentuados na derivação MLII do que na derivação V5.



**Figura 28 Excerto 100**

O eixo da derivação da parte inferior do sinal pode ser quase ortogonal relativamente ao eixo cardíaco elétrico médio. Assim, os batimentos normais são frequentemente mais difíceis de distinguir na parte inferior do sinal. Por sua vez, os batimentos ectópicos (batimentos com distúrbios cardíacos) são mais fáceis de distinguir (o excerto 106 é um exemplo disso, Figura 29).



**Figura 29 Excerto 106**

#### **4.2.2. GRAVAÇÃO E REPRODUÇÃO ANALÓGICA**

As gravações analógicas originais foram feitas usando nove gravadores *Del Mar Avionics*, modelo 445 de dois canais. Durante o processo de digitalização, as gravações analógicas foram reproduzidas num reproduutor *Del Mar Avionics* modelo 660 [30].

Em algumas gravações analógicas foram encontrados desvios fixos (entre a parte alta do sinal e a parte baixa) de até 40 ms provenientes da orientação das cabeças dos gravadores e da unidade de reprodução. Além destes desvios fixos, também foram encontrados desvios variáveis que podem ser comparados em magnitude aos desvios fixos. Estes desvios são um problema que pode causar dificuldades para alguns métodos de análise em tempo real que usem os dois canais dos excertos.

#### **4.2.3. DIGITALIZAÇÃO**

No processo de digitalização dos sinais, as saídas analógicas da unidade de reprodução são filtradas usando um filtro passa-banda (0,1 Hz a 100 Hz) para limitar a saturação e o *anti-aliasing* do conversor analógico/digital (*Analog-to-Digital Converter – ADC*) [30]. O respetivo sinal filtrado foi amostrado a uma frequência de 360 Hz utilizando um *hardware* construído no centro de engenharia biomédica do MIT e no laboratório de engenharia biomédica do BIH. A frequência de amostragem foi escolhida de forma a facilitar a

implementação digital de filtros de *notch* de 60 Hz em detetores de arritmias. Como os gravadores eram alimentados com uma bateria, a maior parte do ruído de 60 Hz presente na base de dados foi introduzida durante a gravação.

As amostras de cada sinal foram obtidas quase simultaneamente devido aos desvios presentes entre a parte inferior do sinal e a parte superior. O ADC utilizado era unipolar com uma resolução de 11 bits numa faixa de 10 mV (de -5 a +5 mV) [30]. A gama de valores vai de 0 a 2047 (inclusive), onde 1024 corresponde a zero volts.

#### 4.2.4. ANOTAÇÕES

Cada registo disponível é devidamente acompanhado das suas anotações [30]. Numa fase inicial, um simples detetor de QRS identificou o primeiro conjunto de anotações. Estas apenas identificavam cada batimento detetado como normal. Numa segunda fase, foram impressos dois registos iguais com o conjunto de anotações anterior e foram entregues a dois cardiologistas que trabalharam neles independentemente. Foram então adicionados os batimentos não detetados pelo detetor de QRS, eliminados os falsos batimentos (se necessário) e mudadas as anotações para cada batimento anormal. Foram adicionadas também anotações sobre o ritmo, qualidade do sinal e comentários.

Estas novas anotações foram adicionadas à base de dados. Após esta adição e a respetiva verificação, um novo registo foi impresso. Nele foram destacadas as divergências encontradas. Cada divergência foi analisada e resolvida por consenso.

Esta base de dados apresenta aproximadamente um total de 110 000 anotações. Estas são geralmente posicionadas no topo da onda R. Na Tabela 4, é possível ver as anotações mais importantes no âmbito deste projeto e o seu respetivo significado. As restantes anotações podem ser vistas em [30].

**Tabela 4 Anotações e respetivos significados**

<b>Símbolo</b>	<b>Significado (Tipo de batimento)</b>
<b>. ou N</b>	Normal
<b>L</b>	Bloqueio do ramo esquerdo
<b>R</b>	Bloqueio do ramo direito
<b>A</b>	Complexos auriculares prematuros
<b>V</b>	Contração prematura ventricular

Também é possível encontrar disponível em [30] tabelas que sumarizam o conteúdo da base de dados. A Tabela 5 é um resumo das tabelas disponíveis em [30] onde apenas são apresentados os tipos de batimentos que se enquadram no âmbito deste documento.

**Tabela 5 Sumário do número de batimentos e o respetivo tipo de batimento**

Registo	Batimento			Tipo de batimento anormal			
	Normal	Anormal	Total	L	R	A	V
100	2239	34	2273	-	-	33	1
101	1860	5	1865	-	-	3	-
102	99	2088	2187	-	-	-	4
103	2082	2	2084	-	-	2	-
104	163	2066	2229	-	-	-	2
105	2526	46	2572	-	-	-	41
106	1507	520	2027	-	-	-	520
107	-	2137	2137	-	-	-	59
108	1739	34	1774	-	-	4	17
109	-	2532	2532	2492	-	-	38
111	-	2124	2124	2123	-	-	1
112	2537	2	2539	-	-	2	-
113	1789	6	1795	-	-	-	-
114	1820	59	1879	-	-	10	43
115	1953	0	1953	-	-	-	-
116	2302	110	2412	-	-	1	109
117	1534	1	1535	-	-	1	-
118	-	2288	2288	-	2166	96	16
119	1543	444	1987	-	-	-	444
121	1861	2	1863	-	-	1	1
122	2476	0	2476	-	-	-	-
123	1515	3	1518	-	-	-	3
124	-	1619	1619	-	1531	2	47
200	1743	858	2601	-	-	30	826
201	1625	375	2000	-	-	30	198
202	2061	75	2136	-	-	36	19
203	2529	451	2980	-	-	-	444
205	2571	85	2656	-	-	3	71
207	-	2332	2332	1457	86	107	105

<b>208</b>	1586	1369	2955	-	-	-	992
<b>209</b>	2621	384	3005	-	-	383	1
<b>210</b>	2423	227	2650	-	-	-	194
<b>212</b>	923	1825	2748	-	1825	-	-
<b>213</b>	2641	610	3251	-	-	25	220
<b>214</b>	-	2262	2262	2003	-	-	256
<b>215</b>	3195	167	3363	-	-	3	164
<b>217</b>	244	1964	2208	-	-	-	162
<b>219</b>	2082	205	2287	-	-	7	64
<b>220</b>	1954	94	2048	-	-	94	-
<b>221</b>	2031	396	2427	-	-	-	396
<b>222</b>	2062	421	2483	-	-	208	-
<b>223</b>	2029	576	2605	-	-	72	473
<b>228</b>	1688	365	2053	-	-	3	362
<b>230</b>	2255	1	2256	-	-	-	1
<b>231</b>	314	1259	1573	-	1254	1	2
<b>232</b>	-	1780	1780	-	397	1382	-
<b>233</b>	2230	849	3079	-	-	7	831
<b>234</b>	2700	53	2753	-	-	-	3

#### 4.2.5. OBTENÇÃO DOS REGISTOS DA MIT-BIH ARRHYTHMIA DATABASE

Os registos desta base de dados podem ser obtidos em [29]. Cada registo é constituído por três ficheiros: ficheiro de anotações “.atr”, ficheiro contendo os dados “.dat” e ficheiro contendo o cabeçalho “.hea”. Estes ficheiros possuem as seguintes características [31]:

- **.atr** – ficheiro do tipo binário que possui todas as anotações que descrevem os sinais de um dado registo.
- **.dat** – ficheiro do tipo binário que contém as amostras digitalizadas dos sinais de um dado registo.
- **.hea** – ficheiro do tipo texto que descreve os sinais, como um cabeçalho. Nele são incluídas algumas informações entre as quais: nome ou URL, tipo de armazenamento, número e tipo de sinais, frequência de amostragem e dados de calibração.

De realçar que o nome de um dado registo é o mesmo para cada um dos três ficheiros anteriormente explicados. A título de exemplo, o registo 100 é composto pelos seguintes ficheiros: 100.atr, 100.dat e 100.hea.

### 4.3. *PHYSIOBANK* ATM

O *PhysioBank Automated Teller Machine* (ATM) é uma ferramenta de *software* que permite visualizar excertos dos sinais disponíveis em todas as bases de dados do *PhysioBank* através de um *browser*. O aspeto desta ferramenta pode ser visto na Figura 30. Como se pode ver, é-nos permitido escolher a base de dados desejada (em “*Database*”), de entre mais de 50 bases de dados disponíveis. Pode escolher-se também o registo desejado (em “*Record*”), o sinal desejado (em “*Signal*”) e as anotações (em “*Annotations*”). No caso da MIT-BIH *Arrhythmia Database* apenas existe uma opção para as anotações. Além disto, também é possível escolher a duração do sinal a apresentar (em “*Length*”), o formato do tempo (em “*Time format*”) e o formato dos dados (em “*Data format*”).

The screenshot shows the PhysioBank ATM interface with the following settings:

- Input:** Database: MIT-BIH Arrhythmia Database (mitdb); Record: 100; Signals: MLII; Annotations: reference beat, rhythm, and signal quality annotations (atr).
- Output:** Length: 10 sec (selected); Time format: time/date; Data format: standard.
- Toolbox:** Plot waveforms (selected).
- Navigation:** Includes buttons for navigation and 'Previous record' / 'Next record'.
- Help:** 'Help' and 'About ATM' buttons.

Figura 30 PhysioBank ATM

É também disponibilizada uma opção (em “*Toolbox*”) que permite escolher o tipo de *output* deste *software*. É possível por exemplo apresentar graficamente o sinal, apresentar um gráfico representativo dos intervalos RR, apresentar os dados em formato texto ou exportar os sinais para o formato “*.mat*” (reconhecido pelo MATLAB). A título de exemplo, o *output* gerado através da Figura 30 pode ser visto na Figura 31. É visível que apenas uma derivação foi apresentada (MLII). As anotações encontram-se localizadas no topo da onda R e estão presentes em cada batimento. É necessário realçar que os comandos utilizados para gerar o *output* também são apresentados.

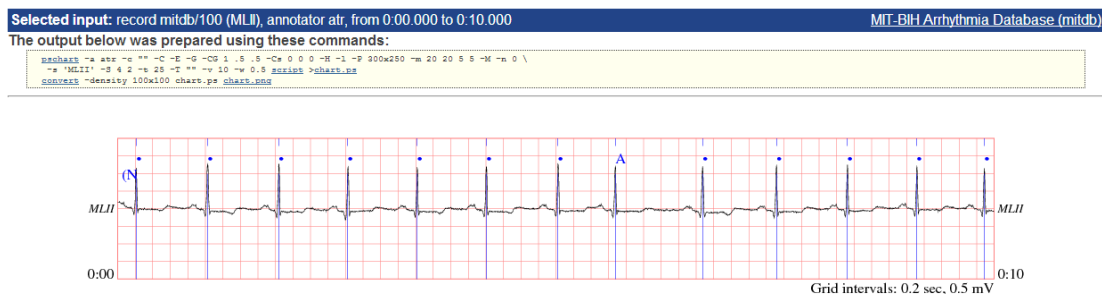


Figura 31 *Output* gerado através dos parâmetros da Figura 30

#### 4.4. *PHYSIO TOOLKIT*

A *PhysioNet* disponibiliza um conjunto de ferramentas para visualização, análise e simulação de sinais fisiológicos [32]. Todo o *software* disponibilizado é livre (*open-source*) ao abrigo da licença GNU *General Public License* (GPL). Além disso, cada componente possui a documentação necessária para a sua perfeita compreensão. Com algumas exceções, o *software* do *PhysioToolkit* é compatível com todas as plataformas populares incluindo GNU/Linux, MAC OS X, MS-Windows e *Unix*.

Neste projeto utilizou-se a ferramenta *WFDB Toolbox* disponível para o MATLAB [33]. A versão utilizada desta ferramenta é a 0.0.3 e pode ser usada em sistemas operativos de 32 bits. É também necessário possuir uma versão do MATLAB igual ou superior à 7 (R14, 2004). Este *software* possui um conjunto de aplicações que permitem leitura, escrita, manipulação e visualização dos dados disponíveis no *PhysioBank*. Estas aplicações encontram-se implementadas como funções no MATLAB e são as seguintes:

- *demo\_wfdb\_tools* – *script* que demonstra algumas das aplicações da *toolbox*;
- *mat2wfdb* – escreve uma variável do MATLAB num ficheiro do tipo *wfdb*;
- *rdann* – lê as anotações de um dado registo da base de dados;
- *rdsamp* – lê um sinal de um dado registo da base de dados;
- *setwfdb* – coloca os diretórios do *wfdb* com os valores por defeito;
- *test\_wfdb* – testa a instalação da *toolbox*;
- *time2sec* – converte o formato dos intervalos de tempo para segundos;
- *wfdb* – imprime menu *help* sobre a *toolbox*;
- *wfdb\_config* – fornece a versão e informação sobre a compilação da *toolbox*;
- *wfdb\_install* – instala ou desinstala a *toolbox*;
- *wfdb\_query* – obtém informação sobre todas as bases de dados e sinais disponíveis em *Physionet*;
- *wfdbdesc* – devolve as especificações dos sinais presentes nos registos;
- *wfdbwhich* – encontra a localização dos ficheiros pertencentes aos registos *wfdb*;
- *wrann* – escreve anotações para os registos *wfdb* no formato tipo;
- *wrsamp* – escreve os dados de um sinal em registos compatíveis com aqueles do *wfdb*.

Atualmente, esta *toolbox* encontra-se na versão 0.9.1 [34]. Ao contrário da versão 0.0.3, esta versão é direcionada para sistemas operativos de 64 bits. Além disso, é necessário possuir uma versão do MATLAB igual ou superior à 7.11 (R2010b). Esta *toolbox* traz

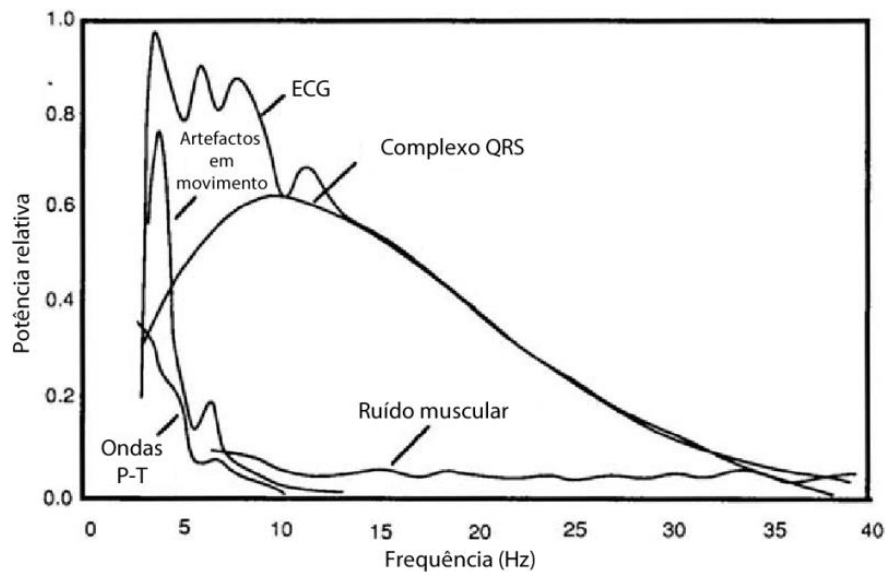
algumas melhorias significativas como a introdução das funções *sqrs* e *wqrs* que permitem a detecção dos complexos QRS num dado sinal.

#### 4.5. REMOÇÃO DO RUÍDO DO SINAL ELETROCARDIOGRÁFICO

Tendo em conta que todos os sinais provenientes da MIT-BIH *Arrhythmia Database* são caracterizados pela presença de ruído, torna-se então importante a remoção do mesmo. Para proceder à sua remoção, é necessário distinguir as principais fontes causadoras da distorção do sinal [35], que são:

- Rede elétrica (60 Hz) – Ruído inserido através dos sistemas de medição apesar da correta ligação à terra. Um filtro de *Notch* pode ser usado para rejeitar este ruído.
- Desvio da linha de base – Esta distorção é essencialmente causada pela respiração do paciente, elétrodos impróprios e má colocação dos elétrodos, entre outros. A gama de frequências é usualmente menor que 0,5 Hz. Um filtro passa-alto com frequência de corte de 0,5 Hz pode ser usado para remover a interferência inserida devido ao desvio da linha de base.
- Eletromiograma (EMG) – É produzido pela atividade elétrica muscular. Num sinal eletrocardiográfico a interferência vinda do EMG aparece com rápidas flutuações que variam mais rápido que as ondas típicas do ECG. As frequências presentes situam-se numa gama entre 0 Hz e 10 kHz. Um filtro morfológico pode ser usado de modo a remover a interferência do EMG.
- Artefactos em movimento – Resulta do movimento do eléctrodo em relação à pele do paciente. É capaz de produzir sinais de maior amplitude no eletrocardiograma com durações entre 100 e 500 ms. Um filtro adaptativo pode ser usado para remover a interferência dos artefactos em movimento.

A Figura 32 representa o espectro de potência relativa do ECG, do complexo QRS, das ondas P e T, do ruído muscular e dos artefactos em movimento. É possível visualizar que o complexo localiza-se principalmente na faixa [5, 15] Hz. A figura mostra que o sinal ECG possui maior concentração de potência abaixo dos 25 Hz. Também é possível visualizar que os artefactos em movimento encontram-se sobrepostos numa pequena parte do ECG enquanto o ruído muscular se sobrepõe na totalidade do sinal eletrocardiográfico. A remoção de ruído presente no sinal eletrocardiográfico será então um dos passos a executar de modo a obter um sinal eletrocardiográfico mais limpo e desta forma ser mais fácil identificar os parâmetros que irão ser utilizados pelo sistema difuso desenvolvido.



**Figura 32** Espectro de potência relativa do ECG, do complexo QRS, das ondas P e T, do ruído muscular e dos artefactos em movimento [36]

#### 4.6. ALGORITMO DE IDENTIFICAÇÃO DO COMPLEXO QRS

Neste projeto será utilizado um algoritmo de detecção do complexo QRS. Este algoritmo é disponibilizado por Aman Chadha [37] e foi desenvolvido com base no algoritmo de Pan-Tompkins [38] para detecção do complexo QRS. Depois de efetuados alguns testes com sinais da base de dados usada, verificou-se que o algoritmo detetava corretamente a onda R, no entanto, em alguns batimentos as ondas Q e S não eram corretamente identificadas. Com base nisto, modificou-se o algoritmo removendo a capacidade de identificação das ondas Q e S. Desta forma, o algoritmo permite identificar e guardar a localização de todas as ondas R presentes num sinal eletrocardiográfico. Esta característica irá permitir posteriormente o cálculo da taxa de batimentos por minuto que será utilizada para dizer se o sinal analisado possui um ritmo normal ou anormal (arritmia).

Este algoritmo é constituído por seis etapas: Filtro passa-baixo, Filtro passa-alto, Diferenciação, Integração, Quadrar e Integração de janela móvel [38]. As duas primeiras etapas permitem atenuar o ruído do sinal. Este irá passar pelos dois filtros que, em cascata, formam um filtro passa-banda. Depois de filtrar, o processo seguinte consiste na diferenciação do sinal que será seguidamente elevado ao quadrado para que seja finalmente aplicada uma integração de janela móvel. O processo de elevar ao quadrado aumenta o declive da curva de resposta em frequência do sinal depois de ter sido afetado de uma diferenciação. Isto permite atenuar o aparecimento de falsos positivos causados pelas

ondas T com energias espectrais maiores que o normal. A integração de janela móvel produz um sinal que contém informação tanto do declive como da largura do complexo QRS. Depois de implementados estes procedimentos são então definidos *thresholds* para identificação do complexo QRS (onde apenas a identificação da onda R será usada). A Figura 33 ilustra um diagrama representativo do algoritmo de deteção do complexo QRS.



Figura 33 Diagrama representativo do algoritmo de deteção do complexo QRS

#### 4.7. SISTEMA DIFUSO

O sistema difuso utilizado neste projeto pode ser consultado em [39] e irá ser sucintamente explicado a seguir. Com uma precisão de 93,78%, este sistema difuso permite classificar batimentos cardíacos normais (NORM) e anormais. A precisão de deteção dos batimentos cardíacos normais é de 95,06%. Os batimentos anormais passíveis de serem identificados por este sistema difuso e a sua precisão são:

- Bloqueio do ramo esquerdo (*Left bundle branch block – LBBB*), 91,03%;
- Bloqueio do ramo direito (*Right bundle branch block – RBBB*), 90,50%;
- Contração prematura ventricular (*Ventricular premature contractions – VPC*), 92,63%;
- Contração prematura auricular (*Atrial premature contractions – APC*), 93,77%.

Antes de mais, convém lembrar os principais parâmetros do ECG e introduzir algumas características do mesmo cuja compreensão é fulcral para o entendimento do sistema difuso enunciado. Na Figura 34 podem ser vistos dois batimentos cujas características identificadas podem ser consultadas na Tabela 6.

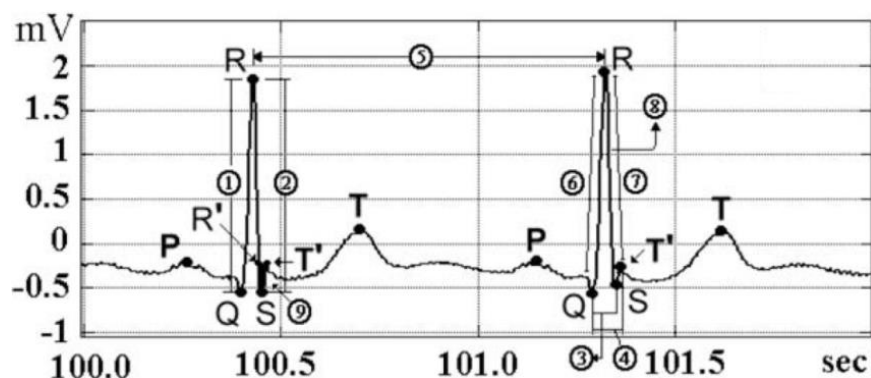


Figura 34 Parâmetros do eletrocardiograma [39]

**Tabela 6 Descrição das características do eletrocardiograma**

<b>Caract.</b>	<b>Símbolo</b>	<b>Descrição</b>	<b>Unidade</b>
<b>1</b>	H-QR	Amplitude entre a onda Q e a onda R	mV
<b>2</b>	H-RS	Amplitude entre a onda R e a onda S	mV
<b>3</b>	QRS-dur	Duração da despolarização dos ventrículos (desde a onda Q até à onda S)	ms
<b>4</b>	QTP-int	Intervalo de tempo desde a onda Q até à onda T'	ms
<b>5</b>	Ratio-RR	Razão entre RRs e RRa, onde RRs é a duração de um único intervalo RR e RRa é a duração média de todos os intervalos RR	-
<b>6</b>	Slope-QR	Declive entre a onda Q e a onda R	mV/ms
<b>7</b>	Slope-RS	Declive entre a onda R e a onda S	mV/ms
<b>8</b>	Area-QRS	Área formada pelo complexo QRS	mV×ms
<b>9</b>	Area-R'ST'	Área formada pelos parâmetros R', S e T'. O ponto R' é o próximo ponto anterior que possui a mesma amplitude que T'	mV×ms

A primeira fase do desenvolvimento deste sistema difuso consistiu na extração das características enunciadas na Tabela 6. Para esta recolha das características foi usada a seguinte amostra de registos:

- Registos 103, 113, 123 e 234 para caracterizar os batimentos do tipo NORM;
- Registos 111 e 214 para caracterizar os batimentos do tipo LBBB;
- Registos 118, 212 e 231 para caracterizar os batimentos do tipo RBBB;
- Registos 200, 221 e 233 para caracterizar os batimentos do tipo VPC;
- Registos 222 e 232 para caracterizar os batimentos do tipo APC.

Na caracterização do tipo de batimento NORM, foram recolhidas as características do registo 103 para cada batimento de modo a obter um valor mínimo, valor máximo e um valor médio. Na Tabela 7 é possível visualizar os valores dos intervalos de cada característica para o registo 103. Este procedimento foi efetuado para os restantes registos do batimento do tipo NORM (113, 123 e 234). A Tabela 8 apresenta os resultados obtidos para o caso NORM. Este processo foi efetuado para os restantes casos (LBBB, RBBB, VPC e APC). Por fim, na Tabela 9 é possível visualizar os valores de todos os intervalos para cada tipo de batimento e o respetivo valor médio ( $V_M$ ). Estes valores irão ser futuramente necessários para a definição das funções de pertinência dos conjuntos difusos do sistema. A Figura 35 representa graficamente todos os intervalos de valores e o respetivo

valor médio para cada característica de cada tipo de batimento. Cada gráfico da figura representa uma característica onde: A é H-QR, B é H-RS, C é QRS-dur, D é QTP-int, E é Ratio-RR, F é Slope-QR, G é Slope-RS, H é Area-QRS e I é Area-R'ST. Na Figura 35A são apresentados os eixos (mínimo, média e máximo) que são iguais para os restantes casos (Figuras 35B, 35C, 35D, 35E, 35F, 35G, 35H e 35I).

**Tabela 7 Intervalo de valores das características para o registo 103**

<b>Característica:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>Mínimo</b>	2,125	2,050	44,0	58,0	0,828	0,061	0,079	51,81	2,70
<b>Máximo</b>	2,665	2,625	64,0	75,0	1,145	0,095	0,156	79,20	6,01
<b>Unidade</b>	mV	mV	ms	ms	–	mV/ms	mV/ms	mV×ms	mV×ms

**Tabela 8 Intervalo de valores das características para o caso NORM**

<b>Característica:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>Mínimo</b>	0,695	0,800	33,0	43,0	0,800	0,019	0,017	20,0	0,00
<b>Máximo</b>	2,690	3,645	79,0	90,0	1,200	0,134	0,214	82,0	24,50
<b>Unidade</b>	mV	mV	ms	ms	–	mV/ms	mV/ms	mV×ms	mV×ms

**Tabela 9 Intervalo de valores das características para cada tipo de batimento**

<b>Carct.</b>	<b>Tipo de Batimento</b>									
	<b>NORM</b>		<b>LBBB</b>		<b>RBBB</b>		<b>VPC</b>		<b>APC</b>	
	<b>Intervalo</b>	<b>V<sub>M</sub></b>	<b>Intervalo</b>	<b>V<sub>M</sub></b>	<b>Intervalo</b>	<b>V<sub>M</sub></b>	<b>Intervalo</b>	<b>V<sub>M</sub></b>	<b>Intervalo</b>	<b>V<sub>M</sub></b>
<b>1</b>	[0,695; 2,690]	2,178	[0,205; 2,060]	1,313	[0,695; 2,240]	1,629	[0,105; 3,095]	1,303	[0,275; 1,870]	0,882
<b>2</b>	[0,800; 3,645]	2,630	[0,705; 2,815]	1,750	[0,955; 3,300]	2,309	[0,870; 3,575]	2,473	[0,490; 2,345]	1,263
<b>3</b>	[33,0; 79,0]	55,0	[86,0; 153,0]	109,2	[46,0; 130,0]	74,6	[52,0; 210,0]	120,4	[34,0; 61,0]	54,3
<b>4</b>	[43,0; 90,0]	70,0	[135,0; 230,0]	172,8	[110,0; 210,0]	144,3	[120,0; 485,0]	291,8	[50,0; 112,0]	91,3
<b>5</b>	[0,80; 1,20]	0,995	[0,85; 1,30]	1,002	[0,775; 1,50]	1,028	[0,45; 0,76]	0,640	[0,41; 0,76]	0,706
<b>6</b>	[0,019; 0,134]	0,067	[0,004; 0,050]	0,023	[0,012; 0,111]	0,049	[0,002; 0,061]	0,021	[0,006; 0,079]	0,022
<b>7</b>	[0,017; 0,214]	0,138	[0,043; 0,055]	0,048	[0,023; 0,136]	0,063	[0,011; 0,108]	0,053	[0,013; 0,162]	0,064
<b>8</b>	[20,0; 82,0]	24,9	[0,0; 146,15]	73,88	[25,01; 120,96]	60,17	[4,16; 289,92]	85,0	[10,59; 69,35]	27,43
<b>9</b>	[0,0; 24,5]	9,3	[20,0; 32,0]	25,0	[34,0; 88,0]	55,8	[0,0; 265,0]	149,9	[0,0; 155,0]	8,00

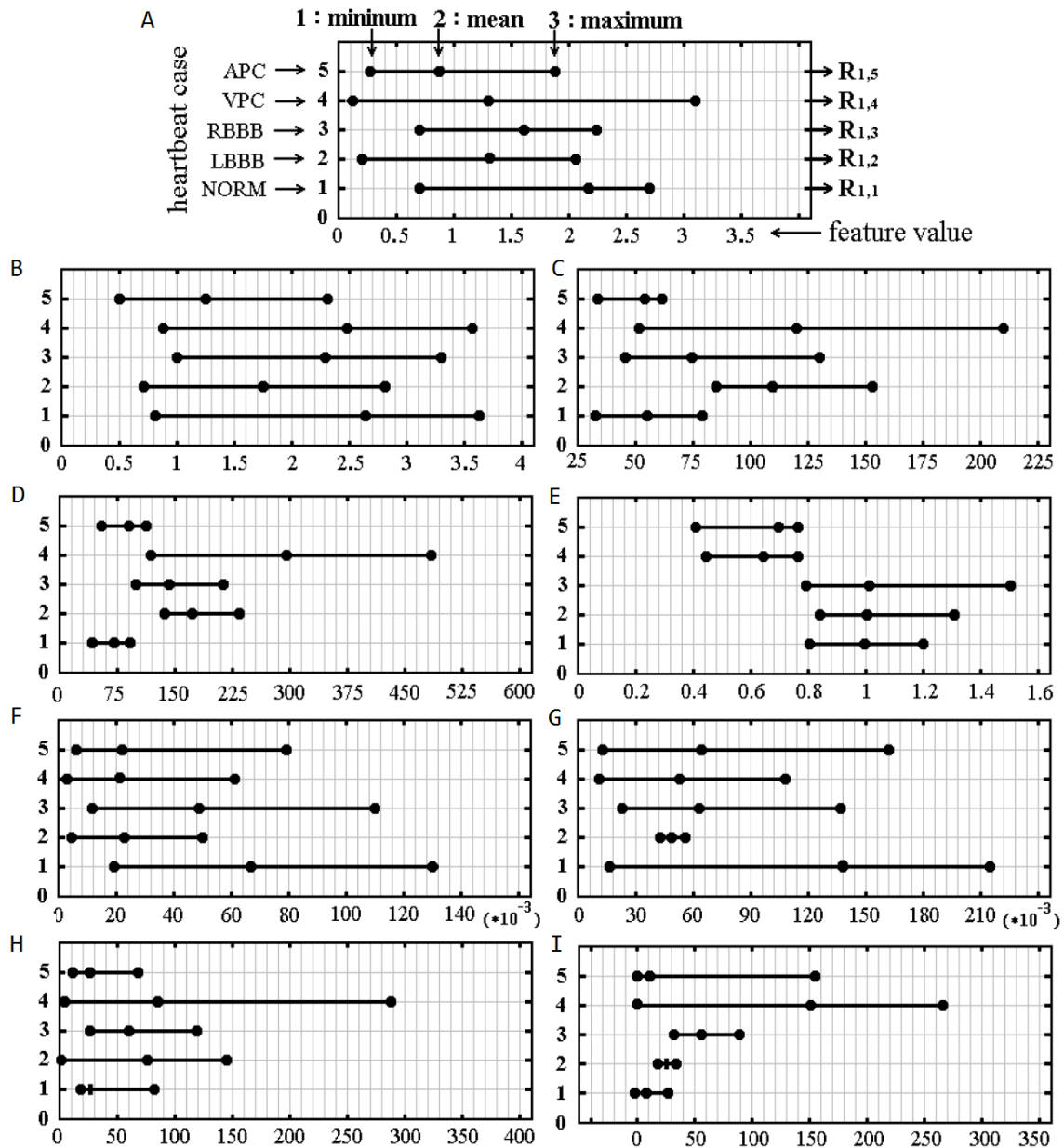


Figura 35 Gráficos dos intervalos de valores das características para cada tipo de batimento [39]

A seguinte fase do desenvolvimento do sistema difuso consistiu na seleção das características qualitativas a usar para a classificação do tipo de batimentos. O objetivo desta seleção reside na procura de um subconjunto de características  $m$  dentro do conjunto total de características  $n$  satisfazendo a condição  $m \leq n$ . Este processo de seleção é dividido em 6 etapas:

- Etapa 0 – Definir  $R_{i,j}$ . Esta variável define o valor do intervalo da característica  $i$  ( $i = 1, 2, 3, \dots, 9$ ) para o tipo de batimento  $j$ . O índice  $i$  identifica as nove características anteriormente faladas. Por sua vez, o índice  $j$  pode tomar os valores 1, 2, 3, 4 e 5 onde 1 é o tipo de batimento NORM, o 2 é LBBB, o 3 é RBBB, o 4 é VPC e o 5 é APC. A

título de exemplo,  $\mathbf{R}_{1,3} = [0,695; 2,24]$  mV diz respeito à característica 1 (H-QR) do tipo de batimento 3 (RBBB).

- Etapa 1 – Obter a característica  $i$  usando o seguinte algoritmo: Se  $\tilde{\mathbf{R}} = 1$ , então a característica é obtida quando  $1 \leq i \leq 9$ ,  $1 \leq k \leq 5$ ,  $1 \leq j \leq 5$ ,  $k \neq j$ , onde  $\tilde{\mathbf{R}} = \mathbf{R}_{i,k} \cap \mathbf{R}_{i,j}$  e o índice  $i,j$  (ou  $i,k$ ) é o mesmo que o de  $\mathbf{R}_{i,j}$ . Observação:  $\tilde{\mathbf{R}} = \mathbf{R}_{i,k} \cap \mathbf{R}_{i,j} = 1$  se os dois intervalos de valores  $\mathbf{R}_{i,k}$  e  $\mathbf{R}_{i,j}$  não se sobrepõem. Desta forma, a característica  $i$  é obtida para discriminar batimentos do tipo  $k$  e  $j$ , e  $N_i$  é incrementado de uma unidade.  $N_i$  indica o número total de casos que a característica  $i$  pode discriminar dois tipos de batimentos distintos ( $k \neq j$ ). De modo a compreender o que foi dito, é apresentado o seguinte exemplo: Se  $\mathbf{R}_{3,1} = [33, 79]$  ms e  $\mathbf{R}_{3,2} = [86, 153]$  ms, logo  $\tilde{\mathbf{R}} = \mathbf{R}_{3,1} \cap \mathbf{R}_{3,2} = 1$ ; isto significa que a característica 3 (QRS-dur) pode discriminar batimentos do tipo 1 (NORM) e 3 (RBBB). Por sua vez, se  $\tilde{\mathbf{R}} = \mathbf{R}_{i,k} \cap \mathbf{R}_{i,j} = 0$ , isto é, se  $\mathbf{R}_{i,k}$  e  $\mathbf{R}_{i,j}$  se sobrepõem, significa que a característica  $i$  não pode ser obtida e não pode discriminar batimentos do tipo  $k$  e  $j$ . Neste caso o valor de  $N_i$  permanece inalterado. Um exemplo disso é: Se  $\mathbf{R}_{1,1} = [0,695; 2,69]$  mV e  $\mathbf{R}_{1,2} = [0,205; 2,06]$  mV, logo  $\tilde{\mathbf{R}} = \mathbf{R}_{1,1} \cap \mathbf{R}_{1,2} = 0$ ; isto significa que a característica 1 (H-QR) não pode discriminar batimentos do tipo 1 (NORM) e 3 (RBBB).
- Etapa 2 – Ordenar  $N_i$  (com  $i = 1, 2, \dots, 9$ ) de forma decrescente de valores e depois selecionar o índice do valor mais alto que será  $i = \arg\{Max(N_i), i = 1, 2, \dots, 9\}$ . Por exemplo, supondo que  $N_1 = 6$ ,  $N_2 = 1$  e  $N_3 = 3$ , a ordenação de acordo com os valores seria  $N_1$ ,  $N_3$  e  $N_2$ . Assim, o índice com o maior valor é  $N_1$  e a sequência de sub-índices  $N_i$  com ordem decrescente de valores é 1, 3 e 2.
- Etapa 3 – Obtenção da característica qualitativa  $i$ . A característica  $i$  é selecionada como característica qualitativa caso satisfaça as duas seguintes condições:
  - Condição 1 – Característica  $i$  pode discriminar batimentos do tipo  $j$  e  $k$  onde os sub-índices são obtidos na etapa 2.
  - Condição 2 – A característica qualitativa que discrimina batimentos do tipo  $j$  e  $k$  ainda não foi encontrada ( $k,j = 1, 2, 3, 4, 5$  e  $k \neq j$ ).

Caso a característica  $i$  não cumpra as condições e não possa ser selecionada como característica qualitativa, então é necessário ir para a Etapa 4.

- Etapa 4 – Obter a próxima característica qualitativa. Se as características qualitativas obtidas a partir da etapa 3 são suficientes para discriminar todos os tipos de batimentos então passar para a etapa 5. Caso contrário, voltar para a etapa 3.

- Etapa 5 – Fim do processo de seleção das características qualitativas.

As características qualitativas resultantes deste processo são então: QRS-dur (caract. 3), QTP-int (caract. 4), Ratio-RR (caract. 5) e area-R'ST' (caract. 9). Por fim, a última fase do desenvolvimento do sistema difuso consistiu em definir: os conjuntos difusos, a base de regras, o motor de inferência difusa e a desfuzificação.

As variáveis de entrada do sistema difuso são as já identificadas características qualitativas (características 3, 4, 5 e 9) enquanto a variável de saída do sistema é o tipo de batimento. Os conjuntos difusos definidos para as variáveis de entrada são apresentados a seguir por ordem crescente de acordo com os valores médios da Tabela 9:

- PS – positivo pequeno;
- PM – positivo médio;
- PB – positivo grande;
- PE – positivo enorme.
- PV – positivo muito grande;

As funções de pertença de cada conjunto difuso das variáveis de entrada são definidas através de formas triangulares cujos parâmetros podem ser visualizados na Tabela 9. Relativamente à variável de saída, esta contém os seguintes conjuntos difusos:

- NORM;
- LBBB;
- RBBB;
- VPC;
- APC.

As funções de pertença dos conjuntos difusos da variável de saída são representadas por formas triangulares ao contrário do que acontece em [39] onde são representadas por funções do tipo *Singleton*. Esta modificação no sistema difuso deve-se ao facto de que, quando usadas funções de pertença do tipo *Singleton* no método de *Mamdani*, o sistema não respondeu corretamente aos exemplos disponibilizados em [39]. O mesmo não aconteceu quando usadas funções de pertença triangulares onde se verificou que os resultados obtidos eram os desejados. Relativamente à base de regras, estas são elaboradas com base na Tabela 10. A Tabela 11 apresenta a sintaxe das regras para este sistema difuso.

Tabela 10 Tabela de regras

Características qualitativas				
Tipo de Batimento	QRS-dur (3)	QTP-int (4)	Ratio-RR (5)	Area-R'ST' (9)
NORM	PM	PS	PB	PM
LBBB	PE	PE	PE	PB
RBBB	PB	PB	PV	PE
VPC	PV	PV	PS	PV
APC	PS	PM	PM	PS

Tabela 11 Regras do sistema difuso

Regra N°	Regra
1	<i>Se QRS-dur é PM, e QTP-int é PS, e Ratio-RR é PB, e Area-R'ST' é PM, então tipo de batimento é NORM</i>
2	<i>Se QRS-dur é PE, e QTP-int é PE, e Ratio-RR é PE, e Area-R'ST' é PB, então tipo de batimento é LBBB</i>
3	<i>Se QRS-dur é PB, e QTP-int é PB, e Ratio-RR é PV, e Area-R'ST' é PE, então tipo de batimento é RBBB</i>
4	<i>Se QRS-dur é PV, e QTP-int is PV, e Ratio-RR é PS, é Area-R'ST' é PV, então tipo de batimento é VPC</i>
5	<i>Se QRS-dur é PS, e QTP-int is PM, e Ratio-RR é PM, e Area-R'ST' é PS, então tipo de batimento é APC</i>

Por fim, falta referir os métodos usados para o motor de inferência difusa e para a desfuzificação. Apenas os métodos de implicação, agregação e desfuzificação são disponibilizados em [39]. Os restantes métodos foram escolhidos depois de verificar o correto funcionamento do sistema. Posto isto, os métodos usados são:

- Método operador lógico *and* – *min*;
- Método operador lógico *or* – *max*;
- Método de implicação – *min*;
- Método de agregação – *max*;
- Método de desfuzificação – *centroid* (centro de massa).

De modo a poder implementar o sistema difuso até agora descrito, é necessário uma ferramenta com potencialidades para isso. A ferramenta utilizada é a *Fuzzy Logic Toolbox* disponível no MATLAB. A secção seguinte apresenta as características principais desta ferramenta.

#### **4.8. TOOLBOX “FUZZY LOGIC” DO MATLAB**

A *toolbox* utilizada neste projeto para a criação de sistemas difusos encontra-se disponível no MATLAB. Além desta ferramenta, existem várias, tanto comerciais como *open-source*, para implementar este tipo de sistemas, entre as quais se encontram:

- *jFuzzyLogic*;
- *fuzzyTECH*;
- *mbFuzzIT*;
- *Xfuzzy*;
- *Fuzzy Logic Tools* (FLT);
- *Fuzzy Logic* para a plataforma *Mathematica*.

##### **4.8.1. DESCRIÇÃO DA TOOLBOX “FUZZY LOGIC”**

O *software Fuzzy Logic Toolbox* [40][41] é um conjunto de funções construídas no ambiente técnico de computação MATLAB (Figura 36). Este *software* fornece ferramentas para criar e editar sistemas de inferência difusa no MATLAB. Outro aspeto interessante é a capacidade de integrar os sistemas difusos desenvolvidos em simulações no *software Simulink*. Podem ainda ser criados programas em linguagem C que invoquem os sistemas difusos desenvolvidos no MATLAB. Isto é possível graças a uma máquina de inferência difusa que faz a leitura dos sistemas difusos guardados numa sessão do MATLAB (Figura 36). Esta *toolbox* depende muito do seu ambiente gráfico que permite desenvolver os sistemas difusos de uma forma muito intuitiva, no entanto, é possível trabalhar inteiramente a partir da linha de comandos. Fica à escolha do projetista do sistema a forma com a qual prefere trabalhar.

Posto isto, pode afirmar-se que esta *toolbox* fornece três tipos de categorias de ferramentas:

- Funções para a linha de comandos;
- Ferramentas gráficas;
- Blocos para o *Simulink*.

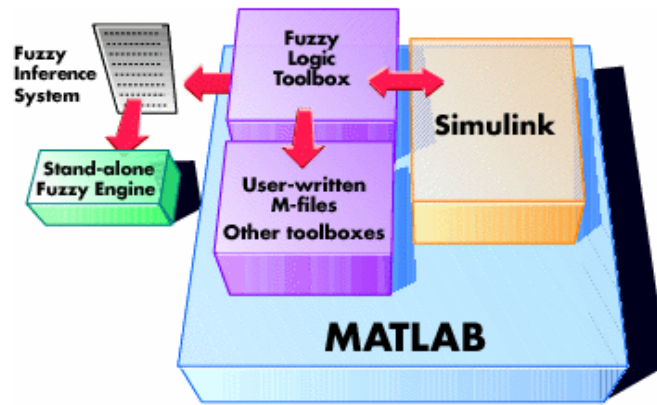


Figura 36 Integração da *toolbox fuzzy logic* com outras ferramentas [41]

No Anexo A é feita uma apresentação detalhada destas ferramentas onde o principal alvo de escrutínio é o sistema de inferência difusa. Este sistema permite um fácil e simples desenvolvimento de sistemas difusos através de uma interface gráfica denominada de *Fuzzy Inference System (FIS) Editor*. A Figura 37 mostra o aspeto principal do *FIS Editor*. Neste editor é possível escolher que métodos de fuzificação e desfuzificação devem ser usados. É também possível alterar o nome das variáveis de entrada e de saída. O controlador de *Mamdani* é escolhido por defeito, no entanto, é possível escolher o controlador de *Sugeno* através do menu de ferramentas criando um novo FIS. É também possível exportar e importar sistemas de inferência de e para ficheiros ou do e para o *workspace*. Além disto, também é possível adicionar ou remover variáveis de entrada e saída.

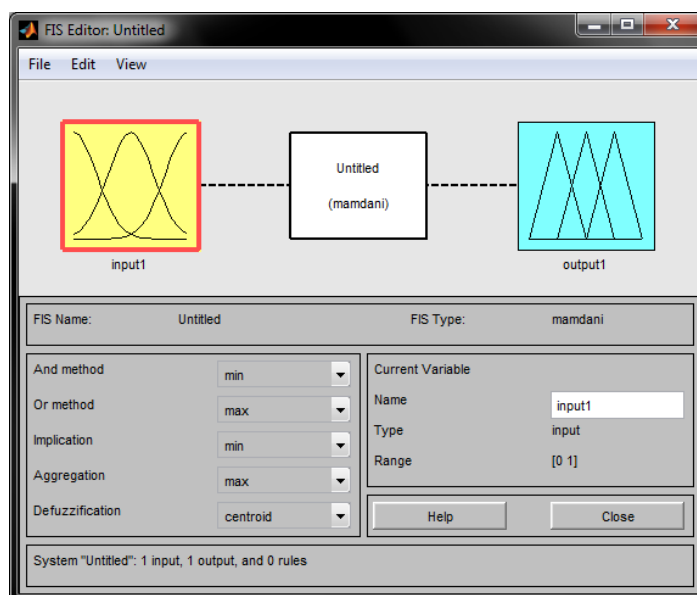
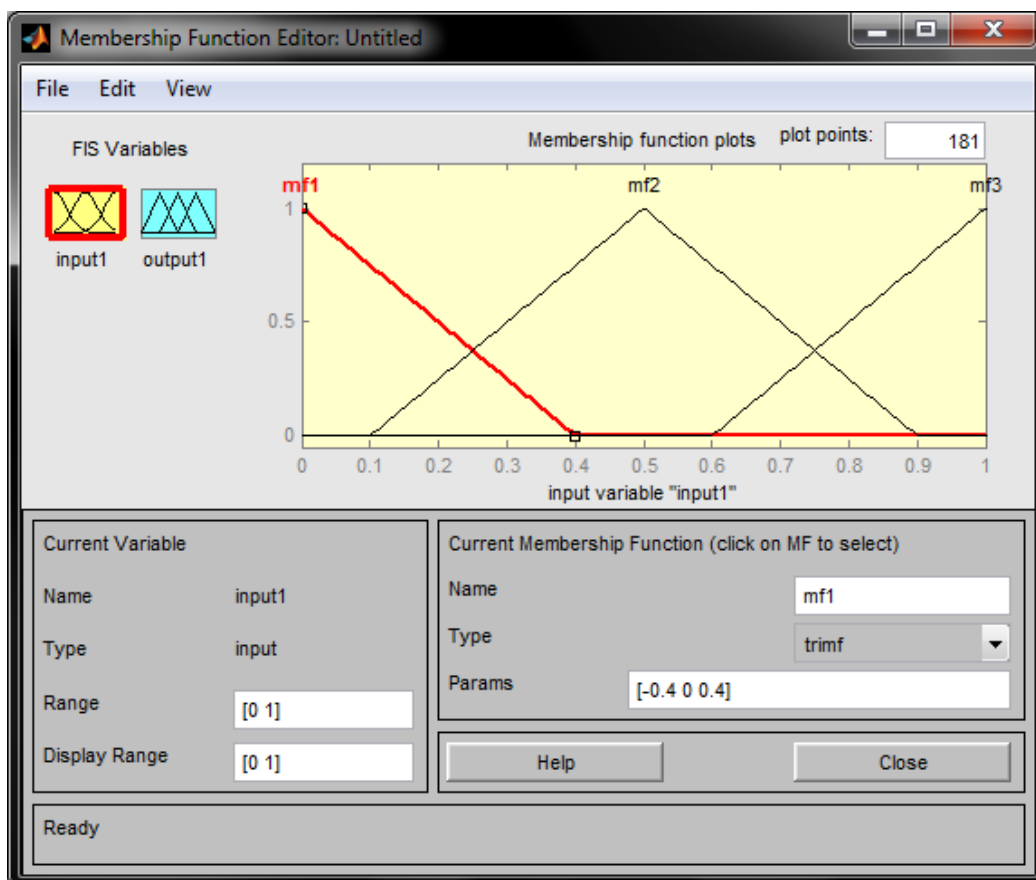


Figura 37 *FIS Editor*

Esta interface gráfica também possui mais funcionalidades como por exemplo a manipulação das funções de pertença dos conjuntos difusos do sistema difuso. A manipulação das funções de pertença é feita através de um editor gráfico criado para o efeito (*Membership Function Editor*). Este editor pode ser visto na Figura 38. Neste editor é possível alterar o número de funções de pertença, escolher o formato das mesmas e definir o seu intervalo. Por último, e não menos importante, é também possível editar as regras através de um editor disponível para o efeito. Este editor é denominado de *Rule Editor* e pode ser visualizado na Figura 39. O *Rule Editor* permite definir as regras do sistema difuso de uma forma bastante intuitiva onde também é possível eliminá-las ou modificá-las consoante as decisões do utilizador.

A interface gráfica do sistema de inferência difusa possui ainda visualizadores (*Rule Viewer* e *Surface Viewer*) que permitem, como o próprio nome indica, visualizar o comportamento das regras e visualizar um gráfico tridimensional onde os eixos X e Y são variáveis de entrada do sistema e o eixo Z uma variável de saída. Estes dois visualizadores são apresentados no Anexo A.



**Figura 38** Editor de funções de pertença

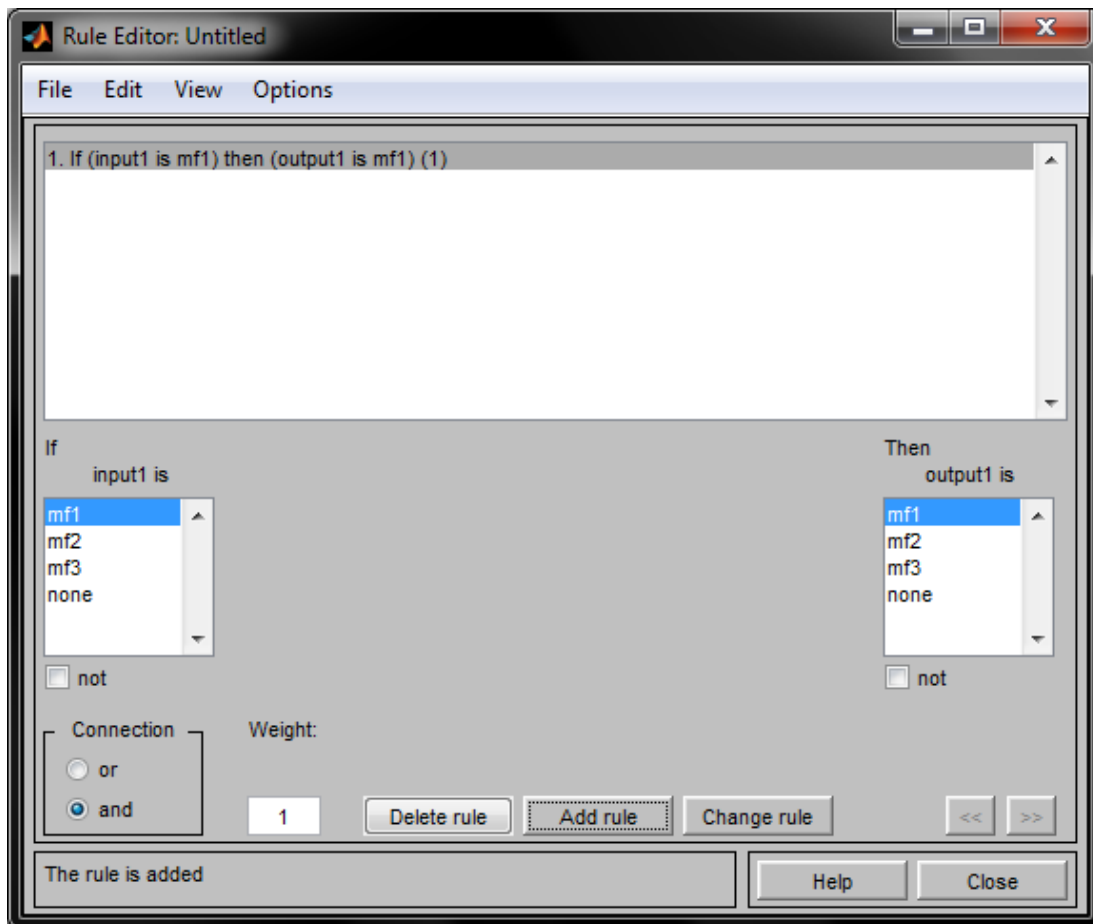


Figura 39 Editor de regras

## 5. DESENVOLVIMENTO DOS COMPONENTES DO SDA

O desenvolvimento do Sistema de Detecção de Arritmias pode ser distinguido em várias partes. A primeira parte, essencial ao sistema, consiste na obtenção dos sinais disponíveis na MIT-BIH *Arrhythmia Database*. Segue-se a segunda parte que abrange a representação gráfica do sinal sem e com anotações, a remoção do ruído dos sinais e além disso, também é introduzida a possibilidade de guardar os dados para um posterior tratamento dos mesmos. Por sua vez, a terceira parte do desenvolvimento do sistema engloba a identificação do complexo QRS do batimento que se pretende analisar e a obtenção dos parâmetros de entrada do sistema difuso. Por fim, não menos importante que as anteriores, surge a quarta parte. Esta parte consiste num sistema difuso que deteta cinco tipos de batimentos diferentes. A Figura 40 apresenta um diagrama representativo onde pode ser visto o atrás enunciado. Estas quatro partes serão expostas no decorrer de todo este capítulo. Neste capítulo é ainda apresentada a interface gráfica desenvolvida e que, de uma forma bastante intuitiva, permite utilizar todas as potencialidades desenvolvidas. Por fim, este capítulo é finalizado com a exibição de alguns resultados práticos obtidos em situações de teste e que permitem atestar a funcionalidade do sistema.

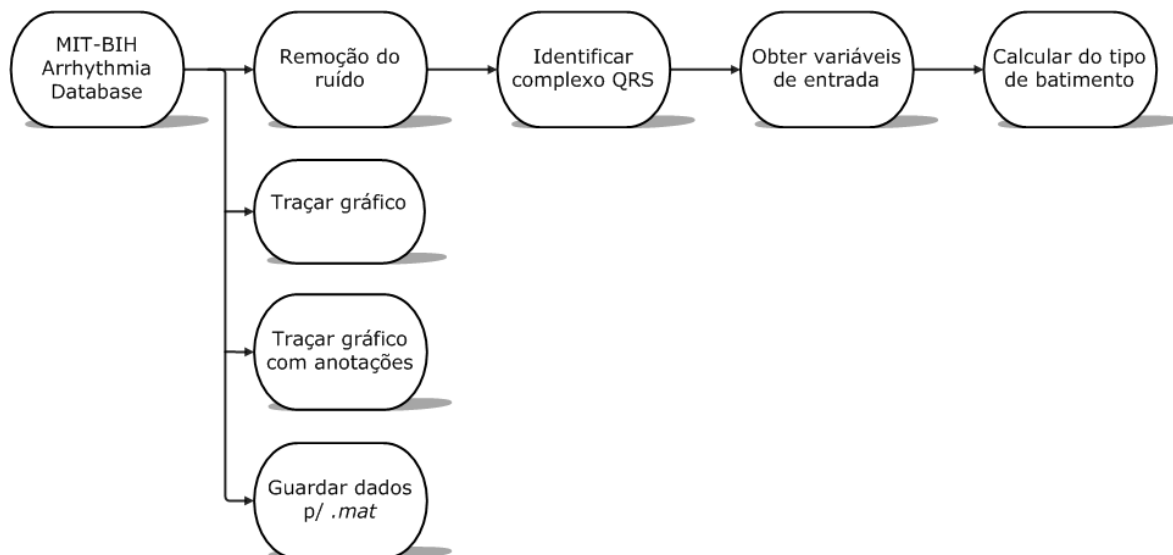


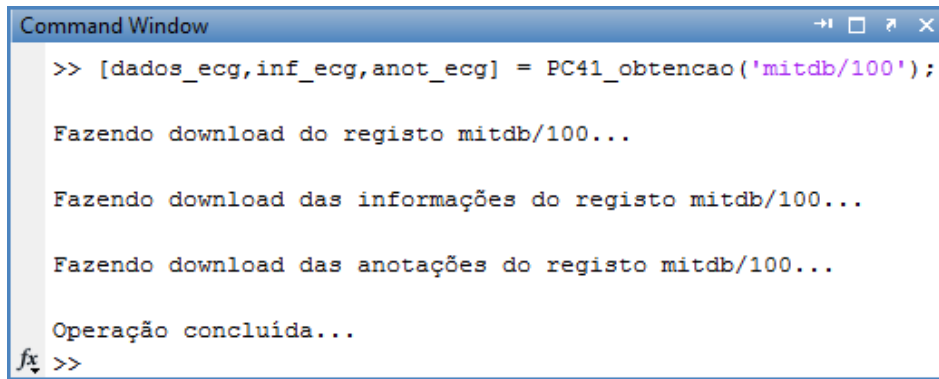
Figura 40 Diagrama representativo do sistema

## 5.1. OBTENÇÃO DOS REGISTOS

A obtenção dos registos é uma parte fulcral do sistema. Estes são obtidos graças à *WFDB Toolbox*. A função que permite obter o sinal diretamente através da Internet é a função *rdsamp*. Com base nesta função foi então desenvolvida outra função que obtém os registos desejados. Além dos registos, também são obtidas as informações (através da função *wfdbdesc*) e as anotações (através da função *rdann*) do sinal caso sejam necessárias posteriormente. A seguinte linha de código mostra como deve ser chamada a função no MATLAB:

```
[dados_ecg, inf_ecg, anot_ecg] =
PC41_obtencao(sinal_mitbih);
```

Esta função necessita apenas de um parâmetro de entrada denominado de *sinal\_mitbih*. Trata-se de uma *string* que contém o nome do sinal que se deseja obter. A sintaxe a utilizar deverá ser a seguinte: *'mitdb/100'*. Onde *mitdb* indica a base de dados a utilizar e *100* o registo que se pretende obter. O resultado desta função é a obtenção das três variáveis *dados\_ecg*, *inf\_ecg* e *anot\_ecg* que representam os dados, as informações e as anotações respetivamente do sinal desejado. No Anexo B é possível visualizar as informações essenciais sobre esta função. De modo a exemplificar o que foi anteriormente dito, é apresentada a Figura 41. Esta figura apresenta o resultado da obtenção do registo 100. Como se pode verificar, à medida que vão sendo obtidos os dados, são impressas informações no ecrã que indicam o estado da obtenção dos dados. As três variáveis obtidas são disponibilizadas no *workspace* (Figura 42) aquando do fim da função.



```
Command Window
>> [dados_ecg,inf_ecg,anot_ecg] = PC41_obtencao('mitdb/100');

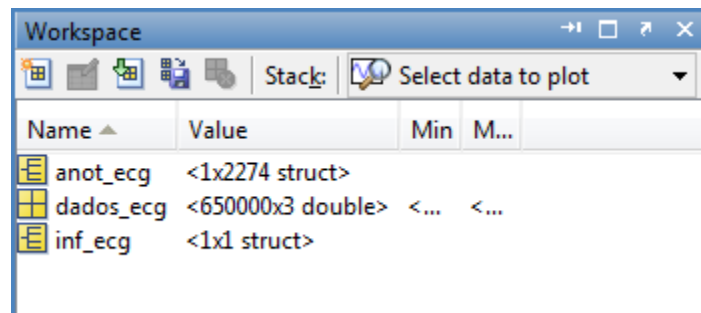
Fazendo download do registo mitdb/100...

Fazendo download das informações do registo mitdb/100...

Fazendo download das anotações do registo mitdb/100...

Operação concluída...
fx >>
```

Figura 41 Exemplo da obtenção do registo 100



Name	Value	Min	M...
anot_ecg	<1x2274 struct>		
dados_ecg	<650000x3 double>	<...>	<...>
inf_ecg	<1x1 struct>		

Figura 42 Output da execução da função *PC41\_obtencao*

## 5.2. REPRESENTAÇÃO GRÁFICA E GRAVAÇÃO DOS DADOS

De modo a poder visualizar os registos obtidos através da função *PC41\_obtencao*, foram desenvolvidas duas funções independentes. A primeira função permite representar uma derivação (sem anotações) do registo obtido em dois gráficos: um deles com uma janela limitada com valores fornecidos pelo utilizador e o outro com a representação completa do sinal. Esta função é denominada de *PC42\_representacao*. Esta função é invocada utilizando a seguinte sintaxe:

```
PC42_representacao(dados_ecg,inf_ecg,derivacao_de  
sejada,seg_min,seg_max);
```

Como se pode ver através da análise da sintaxe da função *PC42\_representacao*, esta recebe os parâmetros *dados\_ecg* e *inf\_ecg* provenientes da função *PC41\_obtencao*. Além destes, também recebe um parâmetro denominado *derivacao\_desejada* que só admite dois valores (1 ou 2). Se for passado o valor 1, esta função irá representar a parte superior do registo, caso contrário, se receber o valor 2, irá ser representada a parte inferior do registo. Por fim, recebe mais dois parâmetros (*seg\_min* e *seg\_max*) que indicam o tamanho da janela para representar o sinal escolhido. Estes dois últimos parâmetros devem ser

fornecidos em segundos e cujo tamanho máximo é 1805,6 segundos o que equivale sensivelmente a 30 minutos. A título de exemplo, apresenta-se a Figura 43 que mostra a execução do comando *PC42\_representacao*. O *output* desta execução pode ser visto na Figura 44. Esta figura apresenta os dois gráficos anteriormente falados onde um deles tem uma janela que mostra o sinal dos 6 aos 10 segundos. O Anexo B permite observar informações essenciais sobre esta função.

```

Command Window
>> PC42_representacao(dados_ecg,inf_ecg,1,6,10);

Fazendo o plot da derivação MLII do registo 100.dat...

Operação concluída...
fx >>

```

Figura 43 Exemplo da representação do sinal 100

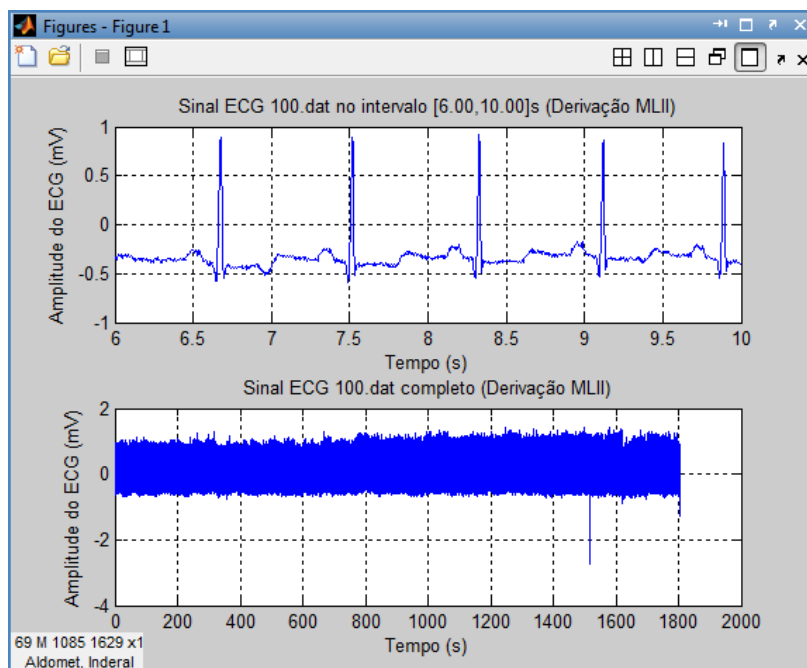


Figura 44 Output da execução da função *PC42\_representacao*

A segunda função desenvolvida tem como objetivo apresentar uma derivação, com anotações, do registo obtido através da função *PC41\_obtencao*. Esta função possui a seguinte sintaxe:

```

PC43_rep_annot(dados_ecg,inf_ecg,anot_ecg,derivacao_desejada,seg_min,seg_max);

```

Analisando a sintaxe desta função, facilmente se visualiza que esta recebe como parâmetros de entrada os valores obtidos da função *PC41\_obtencao*. Além destes parâmetros, também é necessário indicar a *derivacao\_desejada* (1 - parte alta do sinal ou 2 - parte baixa do sinal) e o intervalo da janela (*seg\_min* e *seg\_max*). O valor máximo da janela são os já referidos 1805,6 segundos. A Figura 45 representa a execução da função *PC43\_rep\_annot*. O *output* obtido pode ser visto na Figura 46 onde é possível visualizar cinco batimentos e as respetivas anotações para cada batimento no topo da onda R. O Anexo B contém informações úteis sobre esta função.

```

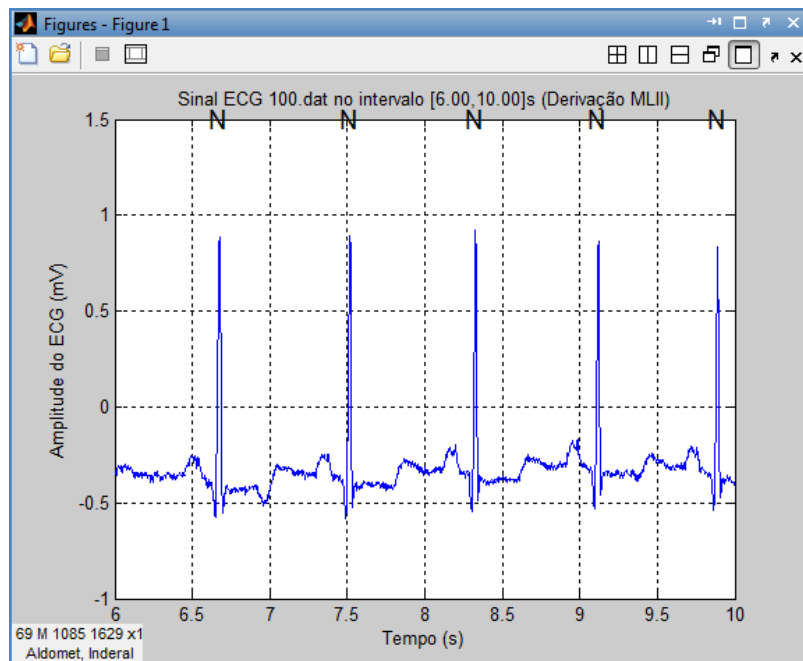
Command Window
>> PC43_rep_annot(dados_ecg,inf_ecg,anot_ecg,1,6,10);

Fazendo o plot das anotações do sinal da derivação MLII no intervalo [6.00,10.00]s...

Operação concluída...
fx >>

```

**Figura 45** Exemplo da representação do sinal 100 com anotações



**Figura 46** Output da execução da função *PC43\_rep\_annot*

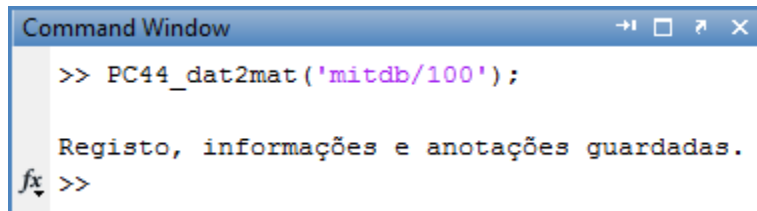
Por fim, foi ainda desenvolvida uma função que permite obter o registo e efetuar a sua gravação em formato “.mat”. A função desenvolvida para o efeito é denominada de *PC44\_dat2mat* e a sua sintaxe é a seguinte:

```

PC44_dat2mat(sinal_mitbih);

```

O parâmetro de entrada recebido pela função é o registo desejado. Este parâmetro tem a seguinte sintaxe: *'mitdb/100'*. Onde *mitdb* indica a base de dados a utilizar e *100* o registo que se pretende obter. A Figura 47 representa a execução da função *PC44\_dat2mat* e a Figura 48 o seu *output*. Como se pode ver na Figura 48, são guardados três ficheiros “.mat” cada um com a respetiva identificação do registo e do tipo de ficheiro (dados, informações e anotações). No Anexo B é possível visualizar informações úteis sobre a função *PC44\_dat2mat*.



```
Command Window
>> PC44_dat2mat('mitdb/100');

Registo, informações e anotações guardadas.
fx >>
```

Figura 47 Exemplo da gravação do registo 123

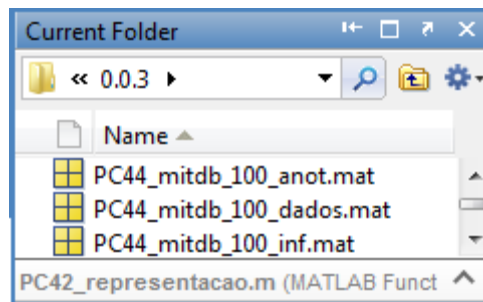


Figura 48 Output da execução da função *PC44\_dat2mat*

### 5.3. REMOÇÃO DO RUÍDO DO SINAL

De forma a remover grande parte do ruído do sinal eletrocardiográfico foi desenvolvida a função *PC45\_rem\_ruido* que aplica três filtros ao sinal obtido através da função *PC41\_obtencao*. Depois de várias tentativas, chegou-se à conclusão que os filtros aplicados ao sinal e que apresentariam resultados satisfatórios são: um filtro de *Notch*, um filtro passa-alto e um filtro passa-baixo. Os parâmetros usados para o filtro de *Notch* são fixos: um fator de qualidade igual a 2 e a componente de frequência que se deseja remover (60 Hz). Por sua vez, tanto o filtro passa-alto como o filtro passa-baixo possuem parâmetros ajustáveis que são inseridos no cabeçalho da função. Esses parâmetros ajustáveis são a ordem e a frequência de corte a usar em cada filtro. A sintaxe da função *PC45\_rem\_ruido* é apresentada a seguir.

```
[sinal_notch_baixo_alto] = PC45_rem_ruido(
dados_ecg, inf_ecg, derivacao_desejada, seg_min, seg_
max, ordema, ordemb, wca, wcb)
```

Como se pode ver através das linhas de código anteriores, esta função recebe os seguintes parâmetros de entrada: *dados\_ecg* e *inf\_ecg* provenientes da função *PC41\_obtencao*, o tamanho da janela que se deseja apresentar e os parâmetros dos filtros passa-alto e passa-baixo. Os parâmetros *ordema*, *ordemb*, *wca* e *wcb* são as ordens e as frequências de corte dos filtros passa-alto e passa-baixo respetivamente. A função devolve um vetor com o sinal filtrado. A Figura 49 mostra a execução da função *PC45\_rem\_ruido* cujos parâmetros são os dados e as informações do sinal *100.dat*, uma janela com intervalo [53,5; 55,2] s, a ordem dos filtros é de 2º grau e as frequências de corte são 0,05 Hz para o filtro passa-alto e 35 Hz para o filtro passa-baixo.

```
Command Window
>> [sinal_notch_baixo_alto] = PC45_rem_ruido(dados_ecg, inf_ecg, 1, 53.5, 55.2, 2, 2, 0.05, 35);

Aplicando um filtro de Notch para remover a interferência da rede elétrica

Aplicando um filtro passa-alto para remover o desvio da linha de base

Aplicando um filtro de passa-baixo para remover a interferência da atividade elétrica muscular

Fazendo o plot do sinal 100.dat filtrado...
fx >>
```

Figura 49 Exemplo da remoção de ruído de um sinal

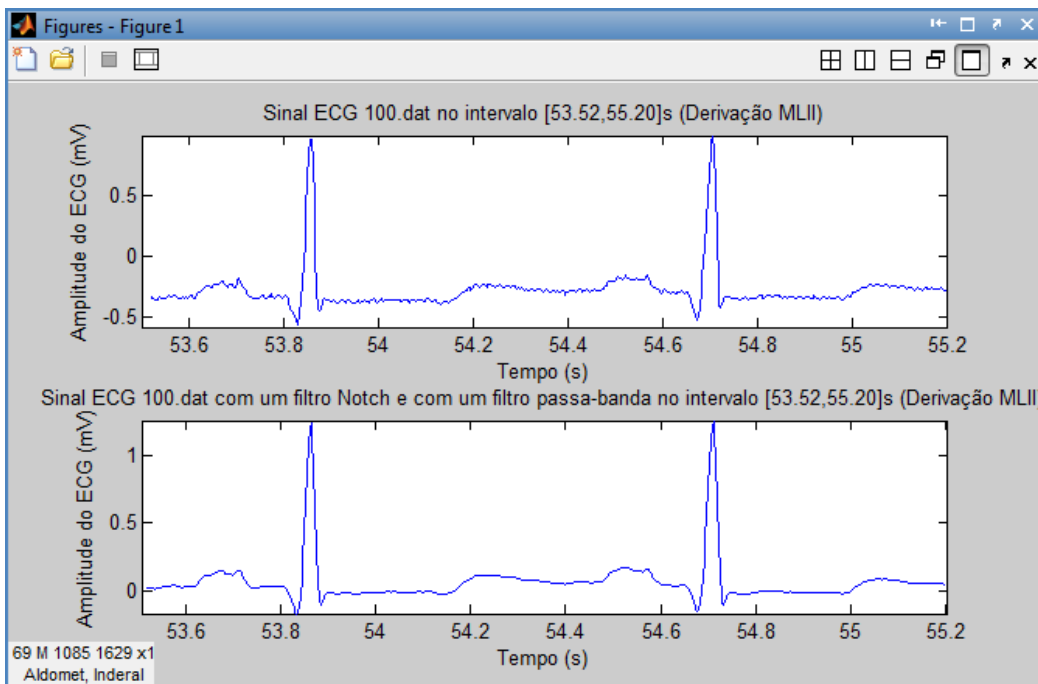


Figura 50 Output da execução da função *PC45\_rem\_ruido*

A Figura 50 representa o *output* da execução da função *PC45\_rem\_ruido*. Como se pode analisar pela figura, o sinal superior é o sinal antes de ser submetido aos filtros e o sinal inferior é o sinal depois de ser submetido aos filtros. Verifica-se que o sinal filtrado encontra-se mais limpo e centrado em torno do zero o que não acontecia no sinal por filtrar. O Anexo B contém informações úteis sobre a função *PC45\_rem\_ruido*.

#### 5.4. IDENTIFICAÇÃO DO COMPLEXO QRS E OBTENÇÃO DAS VARIÁVEIS PARA O SISTEMA DIFUSO

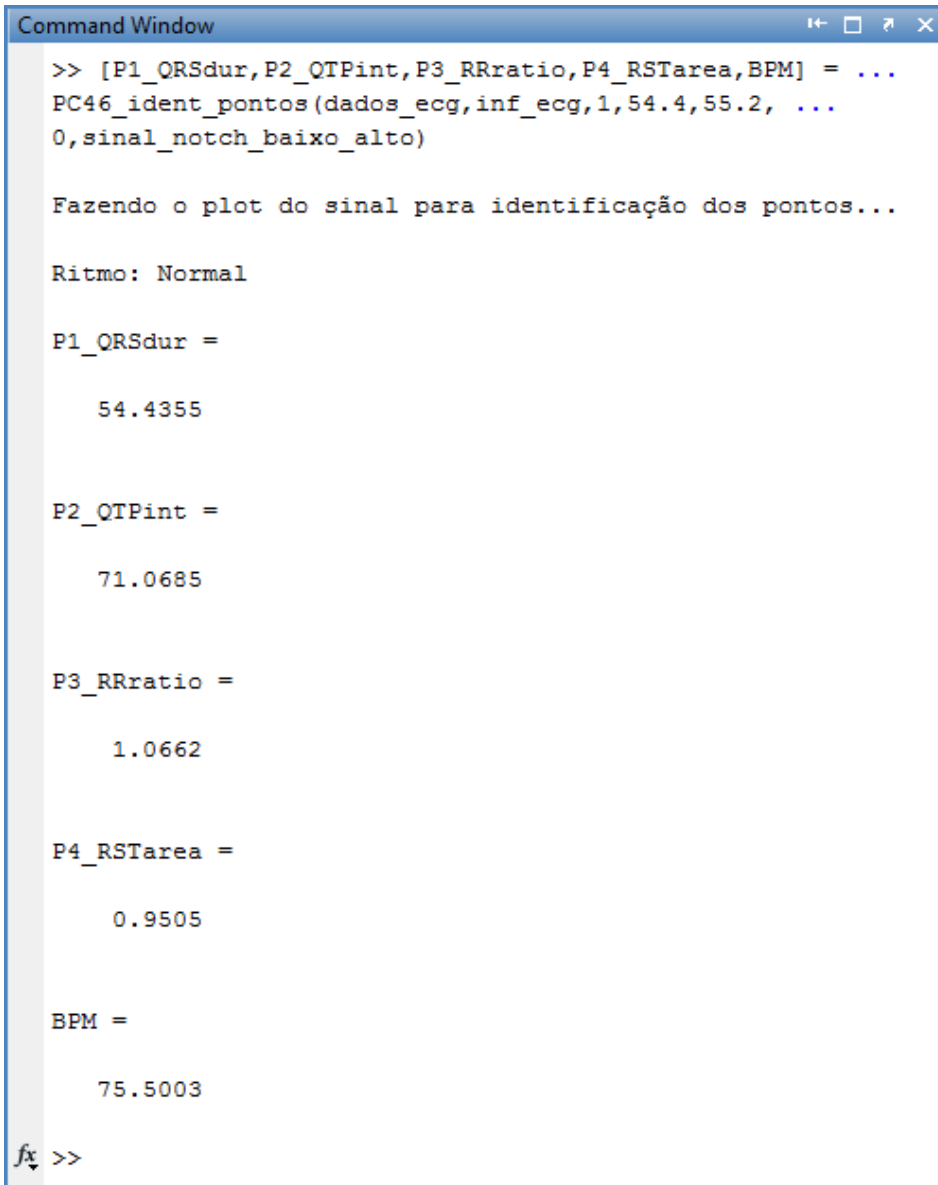
De modo a fazer a ligação entre os registos obtidos até à entrada do sistema difuso, foi necessário desenvolver uma função que permita obter as variáveis de entrada do sistema difuso. Essas variáveis de entrada tanto podem ser obtidas a partir de um sinal onde não foram aplicados filtros como de um sinal onde foram efetivamente aplicados filtros. A função desenvolvida tem o nome de *PC46\_ident\_pontos* e deve ser chamada de acordo com a seguinte instrução:

```
[P1_QRSdur, P2_QTPint, P3_RRratio, P4_RSTarea, BPM] =  
PC46_ident_pontos (dados_ecg, inf_ecg,  
derivacao_desejada, seg_min, seg_max, filtra,  
sinal_notch_baixo_alto);
```

Analisando a sintaxe da função, vê-se que lhe são passados dois dos dados obtidos através da função *PC41\_obtencao* (*dados\_ecg* e *inf\_ecg*). Esta função também recebe as seguintes variáveis: *derivacao\_desejada*, *seg\_min*, *seg\_max*, *filtra*, *sinal\_notch\_baixo\_alto*. A variável *derivacao\_desejada* recebe o tipo de derivação desejada (1 ou 2). O tamanho da janela que se pretende escolher de modo a visualizar o batimento desejado é dado pelas variáveis *seg\_min* e *seg\_max*. A variável *filtra* toma dois valores (0 ou 1) e indica o uso (0) ou não (1) do sinal filtrado para efetuar a identificação do complexo QRS. Por fim, a variável *sinal\_notch\_baixo\_alto* contém o sinal filtrado proveniente da função *PC45\_rem\_ruido*. De salientar que as duas últimas variáveis, *filtra* e *sinal\_notch\_baixo\_alto* não são obrigatórias, isto é, não é necessária a sua inclusão no cabeçalho. Desta forma, caso não sejam incluídas no cabeçalho, será utilizado o sinal por filtrar que se encontra na variável *dados\_ecg*.

Relativamente à funcionalidade desta função, será apresentada uma janela com a derivação escolhida do registo obtido; o sinal obtido poderá encontrar-se ou não filtrado. A duração do sinal a apresentar será definida pelos parâmetros *seg\_min* e *seg\_max*. Nessa janela apresentada, será pedido ao utilizador para identificar os parâmetros P, Q, R, R', S, T' e T

através da função *ginput*. Depois de obtidos os parâmetros, serão efetuados alguns cálculos de modo a obter as variáveis de entrada do sistema difuso: *QRS-dur*, *QTP-int*, *ratio-RR* e *area-R'ST*'. Além de devolver as variáveis de entrada para o sistema difuso, esta função também calcula a taxa de batimentos por minutos (BPM) do eletrocardiograma e indica qual o tipo de ritmo cardíaco do sinal. A Figura 51 mostra a execução da função desenvolvida e a obtenção dos parâmetros e da taxa de batimentos por minutos. De realçar que o sinal utilizado para identificação dos pontos é o sinal filtrado proveniente da função *PC45\_rem\_ruido*. O Anexo B contém informações úteis acerca da função *PC46\_ident\_pontos*.



```
Command Window
>> [P1_QRSdur,P2_QTPint,P3_RRratio,P4_RSTarea,BPM] = ...
PC46_ident_pontos(dados_ecg,inf_ecg,1,54.4,55.2, ...
0,sinal_notch_baixo_alto)

Fazendo o plot do sinal para identificação dos pontos...

Ritmo: Normal

P1_QRSdur =

    54.4355

P2_QTPint =

    71.0685

P3_RRratio =

    1.0662

P4_RSTarea =

    0.9505

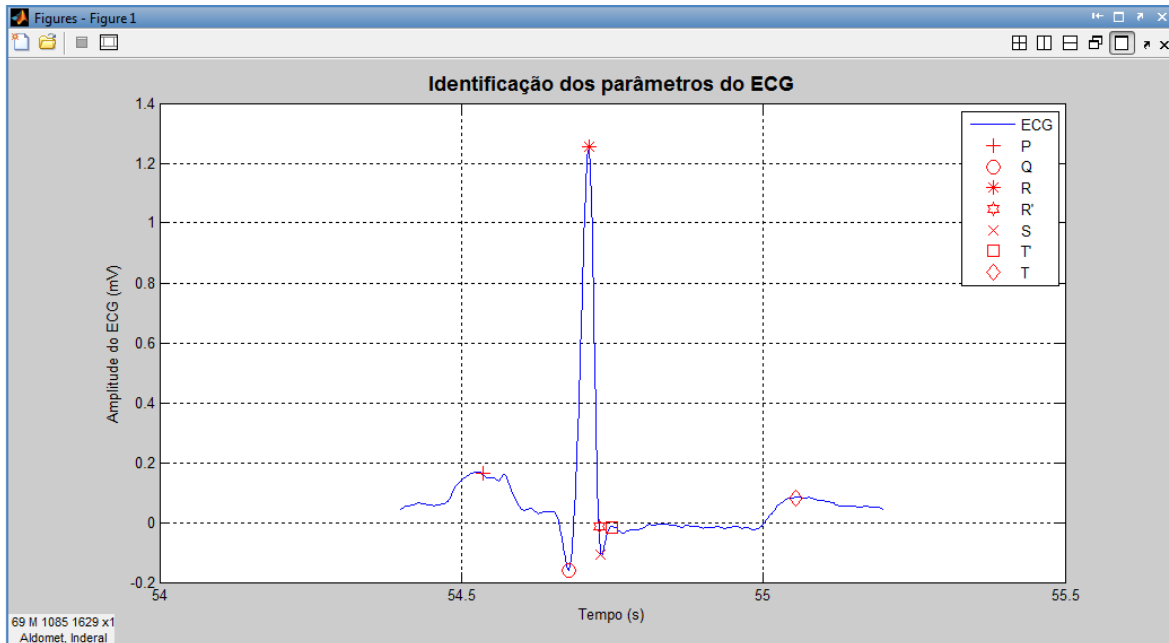
BPM =

    75.5003

fx >>
```

**Figura 51** Exemplo da identificação do complexo QRS e obtenção das variáveis de entrada para o sistema difuso

A janela que surge para que o utilizador identifique os parâmetros do sinal tem o aspeto da Figura 52. Neste caso, os parâmetros já foram identificados pelo utilizador como se pode verificar. De realçar que, o tamanho do sinal usado para apresentar graficamente o sinal (Figura 52) é o definido nos parâmetros *seg\_min* e *seg\_max* (visíveis na Figura 51).



**Figura 52 Identificação do complexo QRS**

Os cálculos dos parâmetros são obtidos depois de introduzidos os pontos pedidos ao utilizador. As linhas de código seguintes traduzem os cálculos necessários para obter os parâmetros *QRS-dur*, *QTP-int* e *area-R'ST'*. Para o parâmetro *QRS-dur* é subtraído o valor temporal do parâmetro Q ao valor temporal do parâmetro S. O cálculo do parâmetro *QTP-int* é semelhante; é subtraído o ponto T' ao ponto Q. Por sua vez, o cálculo da área R'ST' é feito em três etapas: cálculo da base, cálculo da altura e aplicação da fórmula.

```
% Calcular parâmetros de entrada no sistema
difuso (QRSdur, QTPint e RRSTTarea)
P1_QRSdur = (pS(1) - pQ(1))*10^3;
P2_QTPint = (pTT(1) - pQ(1))*10^3;
base = (pTT(1) - pRR(1))*10^3;
altura = (pTT(2) - pS(2));
P4_RSTarea = (base*altura)/2;
```

Para o cálculo do parâmetro *ratio-RR* é necessário determinar todas as ondas R do sinal. Para esse efeito foi utilizado o algoritmo explicado no capítulo anterior que permite a deteção de todas as ondas R e a sua respetiva localização. As linhas de código seguintes

calculam a duração de um único intervalo RR (*RRs*) e a média das durações de todos os intervalos RR (*RRa*). O parâmetro *ratio-RR* é dado pela razão entre *RRs* e *RRa*.

```
% Calcular parâmetros de entrada no sistema
difuso (RRratio)
for i=1:length(R_loc)-1
    if R_loc(i) > x1a && R_loc(i) < x2a
        R_locs = [R_loc(i-1) R_loc(i)];
    end
end
RRs = t(R_locs(2))-t(R_locs(1));
RRa = [];
for i=1:length(R_loc)-1
    RRa(i) = t(R_loc(i+1)) - t(R_loc(i));
end
RRa = sum(RRa) / length(RRa);
P3_RRratio = RRs / RRa;
```

## 5.5. DETERMINAÇÃO DO TIPO DE BATIMENTO

Para efetuar a determinação do tipo de batimento é necessário, numa primeira fase, implementar o sistema difuso que foi apresentado no Capítulo 4 para que, numa segunda fase, se desenvolva o código necessário para complementar o mesmo. De modo a facilitar a implementação do sistema, foi utilizada a ferramenta gráfica FIS. Primeiramente foi escolhido um sistema difuso do tipo *Mamdani* para que as variáveis de saída possam ser definidas por conjuntos difusos. Seguidamente foram definidas tanto as variáveis de entrada como as variáveis de saída do sistema. Através da Figura 53 é possível visualizar a existência das quatro variáveis de entrada (*QRSdur*, *QTPint*, *ratioRR* e *areaRST*) e da variável de saída (*TipoBatimento*). De realçar que os métodos dos operadores lógicos, o método de implicação, de agregação e de desfuzificação são os apresentados na Figura 53 e introduzidos no Capítulo 4. O passo seguinte contempla a definição dos conjuntos difusos das variáveis de entrada e de saída. No caso das variáveis de entrada, são definidos de acordo com a Tabela 9 e a Tabela 10 do capítulo 4. As funções de pertença dos conjuntos difusos das variáveis de entrada são do tipo triangular. A Figura 54 apresenta os conjuntos difusos da variável *QRS-dur*. As três figuras seguintes (Figura 55, 56 e 57) representam, respetivamente, a implementação dos conjuntos difusos para as variáveis *QTP-int*, *ratio-RR* e *area-R'ST'*. Relativamente à variável de saída, *TipoBatimento*, foram também implementados cinco conjuntos difusos relativos ao tipo de batimento. O tipo de função de pertença usado para os conjuntos difusos da variável de saída é do tipo triangular. Na Figura 58 é possível visualizar os conjuntos difusos da variável de saída. De

realçar que estes encontram-se centrados em 2, 4, 6, 8 e 10 para os respetivos conjuntos NORM, LBBB, RBBB, VPC e APC.

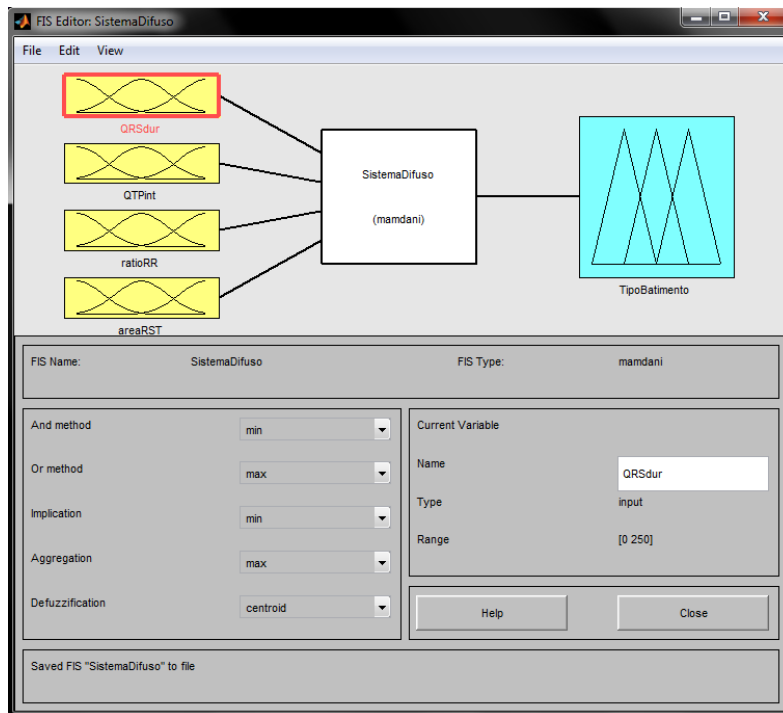


Figura 53 Implementação do sistema difuso

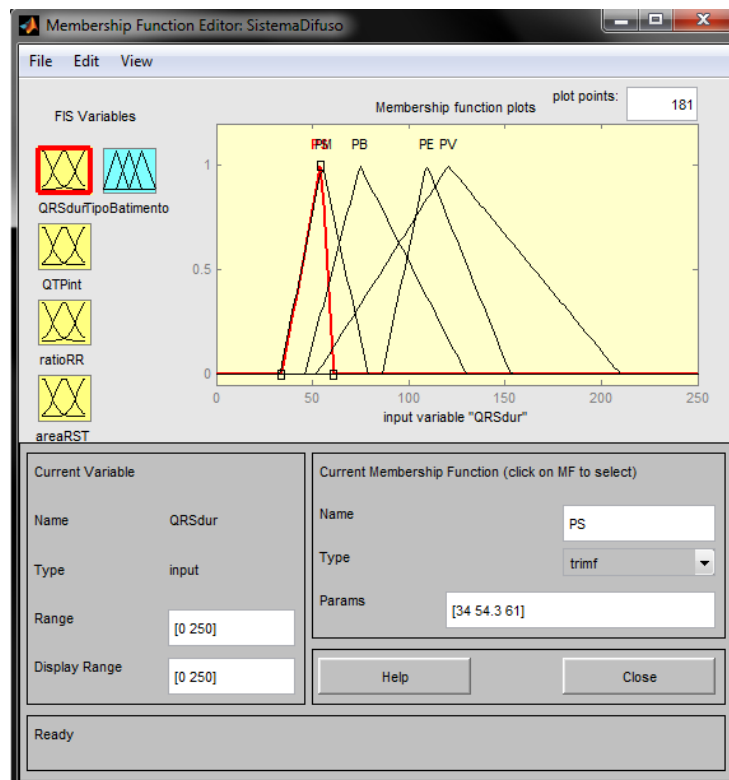


Figura 54 Conjuntos difusos da variável de entrada *QRSdur*

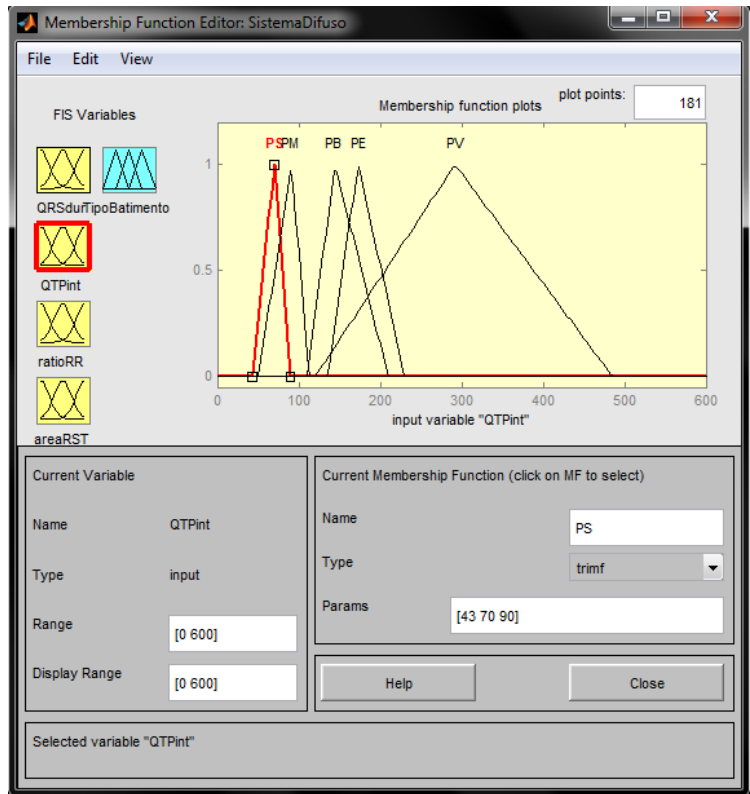


Figura 55 Conjuntos difusos da variável de entrada *QTPint*

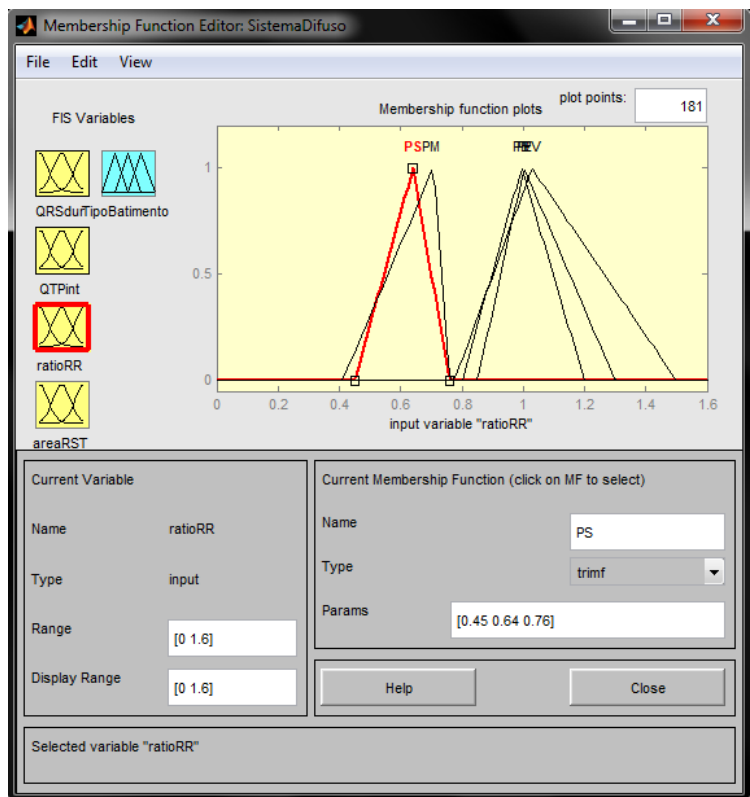


Figura 56 Conjuntos difusos da variável de entrada *ratioRR*

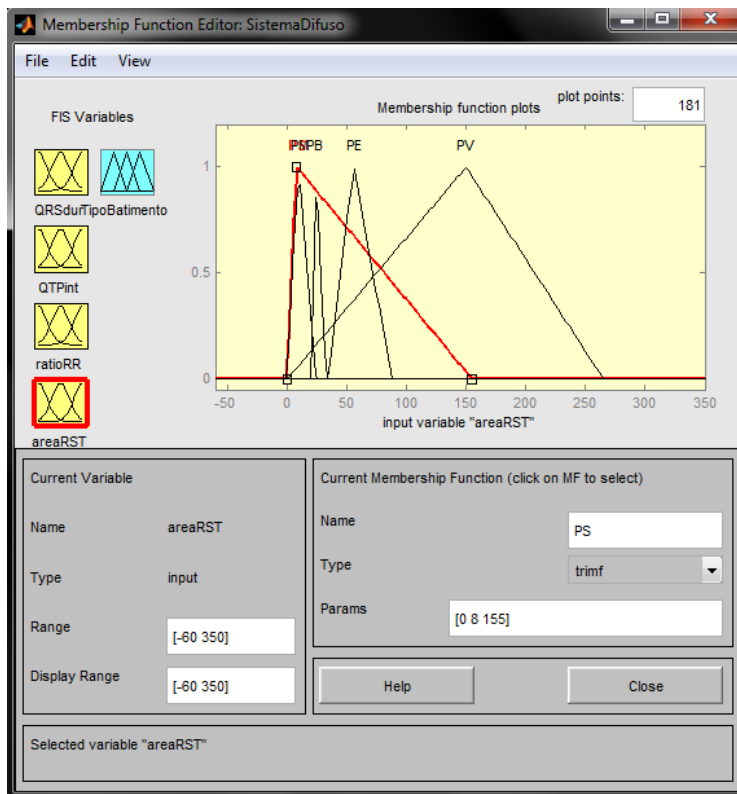


Figura 57 Conjuntos difusos da variável de entrada *areaRST*

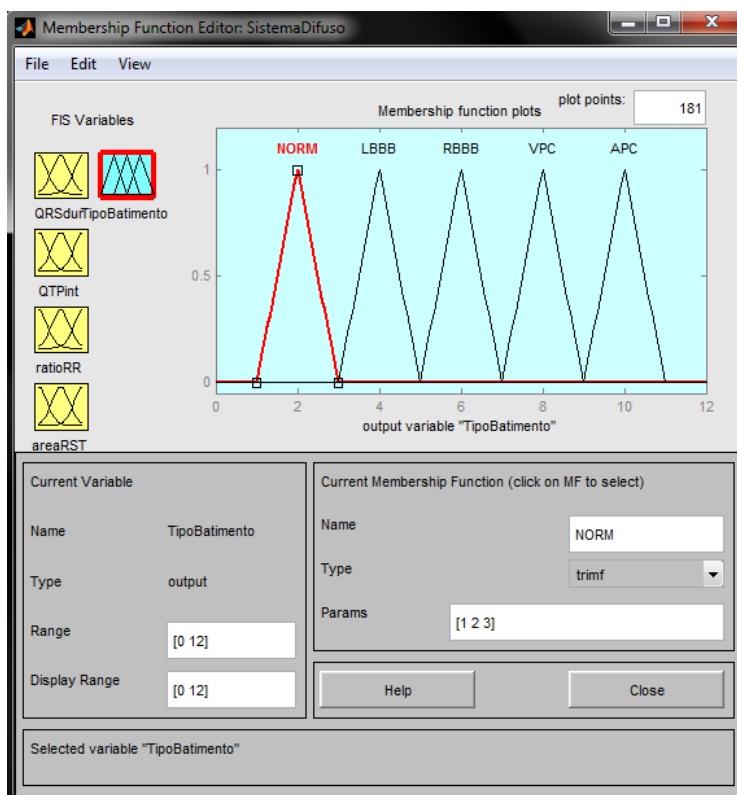
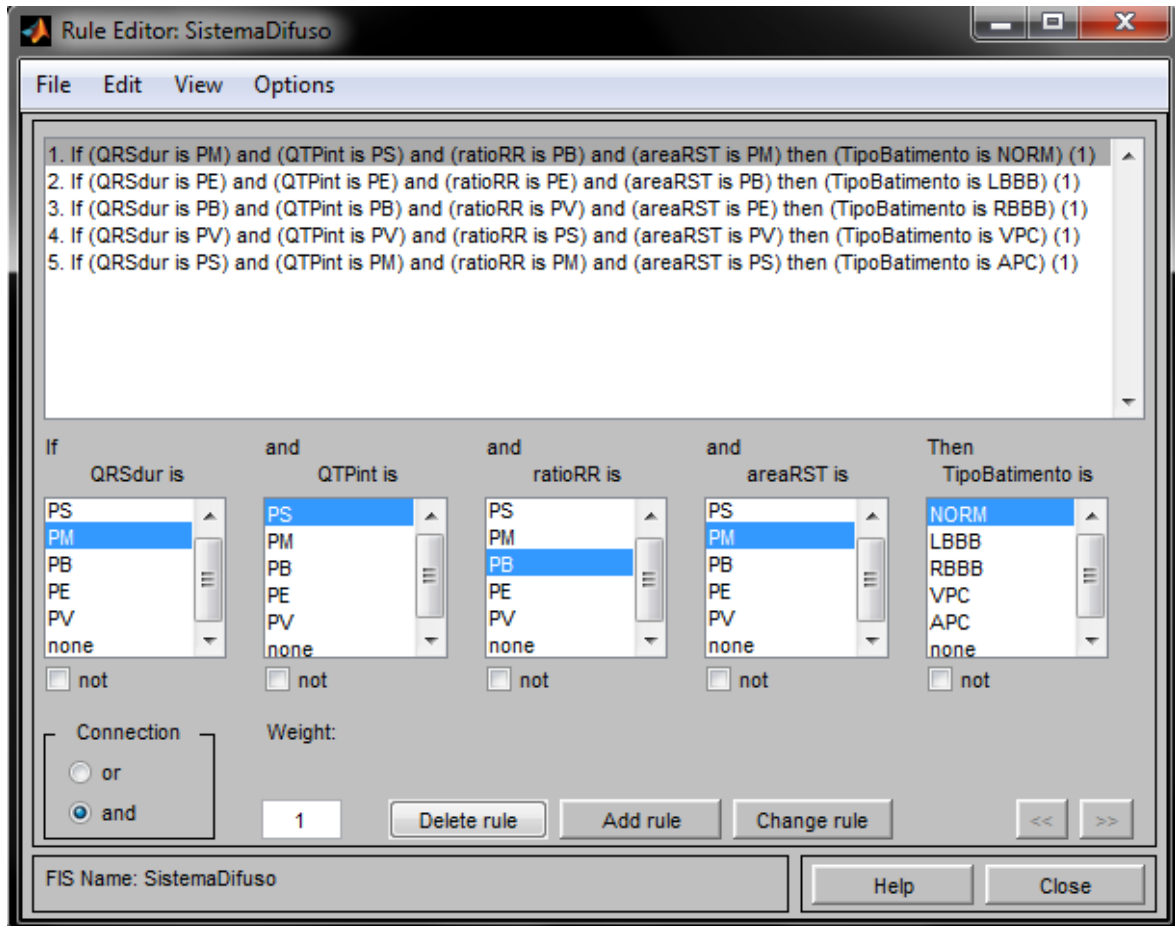


Figura 58 Conjuntos difusos da variável de saída *TipoBatimento*

Por fim, é necessário definir as regras do sistema difuso. As regras são as que se encontram presentes na Tabela 11 do Capítulo 4 e a sua implementação na *Fuzzy Logic Toolbox* pode ser vista na Figura 59. Desta forma é criado o sistema difuso que irá ser utilizado na deteção do tipo de batimento.



**Figura 59** Regras do sistema difuso

O código necessário para dar seguimento à seção anterior (secção 5.4), vai ser apresentado a seguir. A função criada para esse efeito é denominada de *PC47\_tipo\_batimento*. A sintaxe para invocar esta função é a seguinte:

```
PC47_tipo_batimento(P1_QRSdur,P2_QTPint,P3_RRratio, P4_RSTarea)
```

Esta função recebe os quatro parâmetros provenientes da função *PC46\_ident\_pontos* que serão os parâmetros de entrada do sistema difuso. O Anexo B contém informações úteis sobre esta função. Posto isto, as linhas de código seguintes traduzem o processo que é necessário de modo a obter o resultado do tipo de batimento.

```

% Carregar o sistema difuso
sistema_difuso = readfis('SistemaDifuso');

% Fazer os cálculos para obter o tipo de
batimento
tipo_batimento = evalfis([P1_QRSdur P2_QTPint
P3_RRratio P4_RSTarea],sistema_difuso);

% Arredondar o resultado do sistema difuso às
unidades
tipo_batimento = round(tipo_batimento);

```

Através da análise das linhas de código anteriores, verifica-se que, a primeira instrução (função *readfis*) carrega o sistema difuso para a variável *sistema\_difuso*. A instrução seguinte permite obter o tipo de batimento (faz os cálculos) de acordo com o valor das variáveis de entrada. O resultado da função *evalfis* é arredondado às unidades para que, usando as seguintes instruções, seja impresso no ecrã o tipo de batimento obtido:

```

% Apresentar resultado
if tipo_batimento == 2
    fprintf('\n Batimento normal.\n\n');
else
    fprintf('\nTipo de batimento: ');
    switch tipo_batimento
        case 4
            fprintf('LBBB.\n\n');
        case 6
            fprintf('RBBB.\n\n');
        case 8
            fprintf('VPC.\n\n');
        case 10
            fprintf('APC.\n\n');
        otherwise
            fprintf('\n\n Erro na deteção do tipo
de batimento.\n\n');
    end
end

```

De modo a exemplificar o uso do sistema difuso, foram utilizados os dados provenientes do exemplo da secção anterior, secção 5.4 (dados visíveis na Figura 51). A Figura 60 mostra a execução da função *PC47\_tipo\_batimento* cujos parâmetros de entrada são:

- P1\_QRSdur = 54,4355;
- P2\_QTPint = 71,0685;
- P3\_RRratio = 1,0662;
- P4\_RSTarea = 0,9505.

```
Command Window
>> PC47_tipo_batimento(P1_ORSdur,P2_QTPint,P3_RRratio, P4_RSTarea);

Batimento normal.

fx >>
```

Figura 60 Exemplo da classificação do tipo de batimento

## 5.6. INTERFACE GRÁFICA

De modo a poder controlar todas as funções implementadas de uma forma mais intuitiva, desenvolveu-se uma interface gráfica com ajuda da ferramenta GUIDE. Esta ferramenta encontra-se disponível no MATLAB logo, a programação da interface, foi efetuada em linguagem MATLAB. O GUIDE permite implementar menus, barra de ferramentas, botões e gráficos entre outros.

A interface gráfica desenvolvida pode ser visualizada através da Figura 61. Através da sua análise, rapidamente se observa que esta interface se encontra dividida em seis grupos: obter ou abrir ficheiros, mostrar sinal, identificar parâmetros e calcular tipo de batimento, tipo de batimento e batimentos/minuto, representação gráfica e a barra de estado. O Anexo B contém informações úteis acerca da função que implementa a interface gráfica. Convém realçar que esta interface encontra-se otimizada apenas para sinais provenientes da base de dados utilizada.

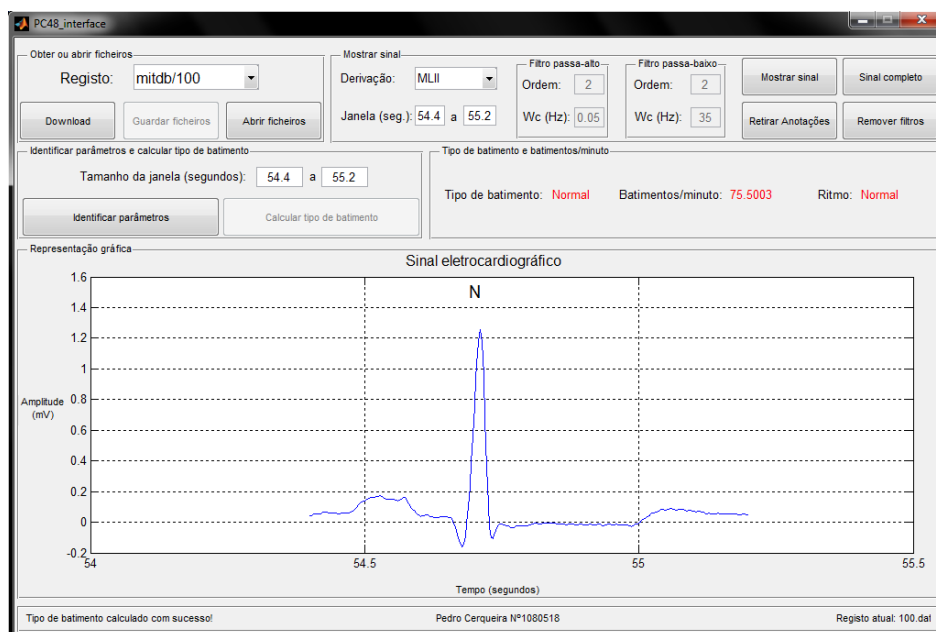


Figura 61 Interface gráfica desenvolvida

### 5.6.1. OBTER OU ABRIR FICHEIROS

O primeiro grupo, que se encontra no topo superior esquerdo da interface, é o grupo para a obtenção ou abertura dos ficheiros (Figura 62). Neste grupo, e como o próprio nome indica, é possível efetuar a obtenção (*download* diretamente da base de dados) dos registos ou abrir registos previamente guardados. Além disto, existe também a possibilidade de, depois de efetuado o *download* de um registo, guardar os respetivos ficheiros em formato “.mat”. Uma vez guardado o registo descarregado da base de dados, o botão *Guardar ficheiros* é desativado e apenas é ativado novamente aquando da obtenção de um novo ficheiro da base de dados. O botão *Abrir ficheiros*, quando utilizado, solicita o carregamento de ficheiros previamente guardados. Estes devem ser inseridos com a seguinte ordem: primeiro abrir os dados, depois as informações e finalmente as anotações do registo que se pretende analisar. Na primeira utilização da interface, apenas os botões de *Download* e de *Abrir ficheiros* se encontram ativos. Uma vez utilizado algum deles, os restantes botões da interface são ativados automaticamente.



Figura 62 Obter ou abrir ficheiros

### 5.6.2. MOSTRAR SINAL

O grupo seguinte, *Mostrar sinal* (Figura 63), permite mostrar o sinal aberto assim como apresentar anotações e aplicar os filtros implementados. O sinal é posteriormente mostrado no grupo *Representação gráfica*. Inicialmente e depois de abrir ou descarregar um determinado registo, o grupo *Mostrar sinal* apenas possui os botões *Mostrar sinal* e *Sinal completo* ativos. Uma vez que se executa o botão *Mostrar sinal*, tanto as caixas de texto dos filtros passa-alto e passa-baixo como os botões *Mostrar anotações* e *Filtrar sinal* são ativados.

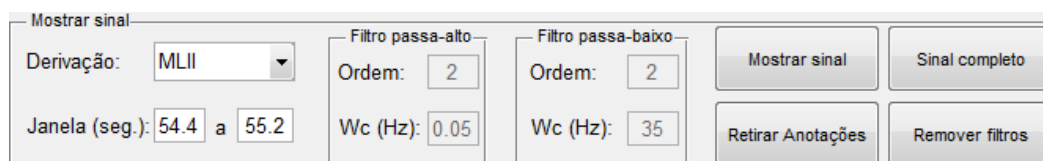


Figura 63 Mostrar sinal

### 5.6.3. IDENTIFICAR PARÂMETROS E CALCULAR TIPO DE BATIMENTO

O terceiro grupo, *Identificar parâmetros e calcular tipo de batimento* (Figura 64), permite escolher um batimento do sinal atual e identificar nele os parâmetros que permitirão, posteriormente, calcular o tipo de batimento e a taxa de batimentos por minuto do sinal. Posto isto, neste grupo apenas é possível especificar a janela desejada para a identificação dos parâmetros. Uma vez executado o botão *Identificar parâmetros*, será apresentada uma janela que pedirá para identificar os parâmetros do sinal. Depois da inserção correta de todos os parâmetros, a janela encerra automaticamente e este botão é desativado. O botão *Calcular tipo de batimento* é então ativado de modo a poder efetuar o cálculo do tipo de batimento. Caso os parâmetros tenham sido identificados incorretamente, uma janela de aviso será mostrada e o botão *Identificar parâmetros* será mantido ativado assim como o botão *Calcular tipo de batimento* será mantido desativado.

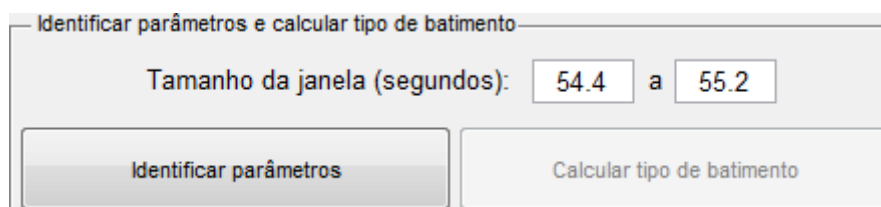


Figura 64 Identificar parâmetros e calcular tipo de batimento

### 5.6.4. TIPO DE BATIMENTO E BATIMENTOS/MINUTO

O grupo *Tipo de batimento e batimentos/minuto* mostra o resultado da execução do botão *Calcular tipo de batimento* que implementa o cálculo do tipo de batimento e a taxa de batimentos/minuto. A Figura 65 mostra o resultado do mesmo exemplo prático apresentado durante a explicação das funções desenvolvidas. O batimento analisado provém da derivação MLII situado no intervalo  $[54,4; 55,2]$  s como se pode ver na Figura 64.

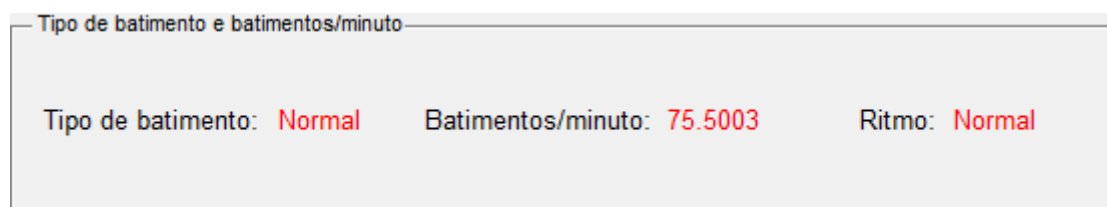
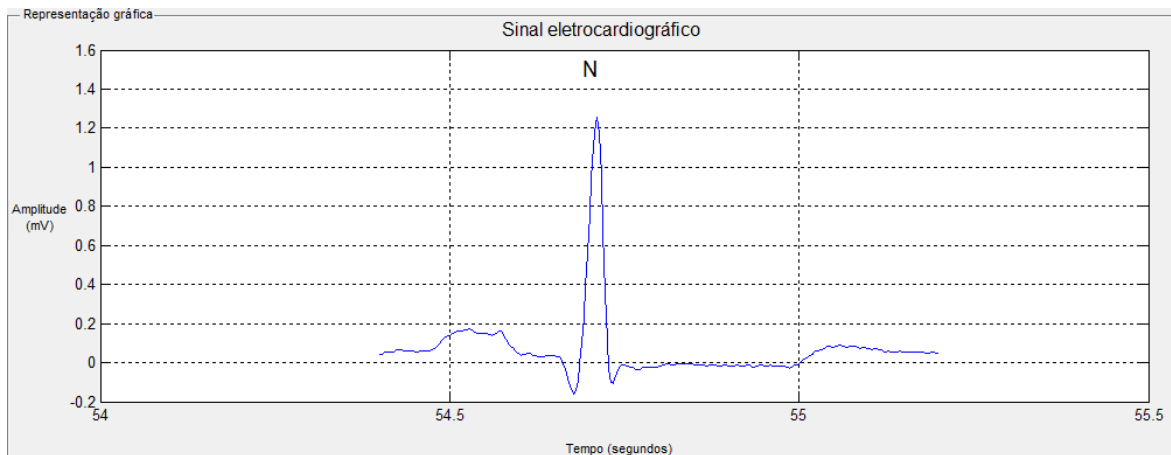


Figura 65 Tipo de batimento e batimentos/minuto

### 5.6.5. REPRESENTAÇÃO GRÁFICA

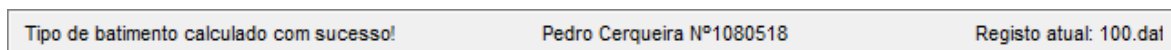
O grupo *Representação gráfica* é onde, como se encontra referenciado no nome, se efetua a representação gráfica do sinal com os parâmetros definidos no grupo *Mostrar sinal*. Na Figura 66 é possível ver a representação da derivação MLII no intervalo [54,4; 55,2] s onde foram aplicados filtros com os parâmetros definidos conforme a Figura 63.



**Figura 66 Representação gráfica**

### 5.6.6. BARRA DE ESTADO

Por fim, o último grupo é uma simples barra de estado onde no extremo esquerdo da Figura 67 vão sendo apresentadas mensagens à medida que se vai clicando nos diversos botões; por sua vez, no extremo direito da mesma figura é apresentado qual o registo atual disponível para identificação do tipo de batimento.



**Figura 67 Barra de estado**

## 5.7. TESTES EFETUADOS AO SISTEMA

De modo a verificar a funcionalidade do Sistema de Detecção de Arritmias, foram efetuados alguns testes que consistem em selecionar de forma aleatória batimentos de sinais distintos para detetar o tipo de batimento. Logicamente serão apenas testados os batimentos que são passíveis de serem detetados pelo sistema difuso (NORM, LBBB, RBBB, VPC e APC). Não será efetuada nenhuma análise estatística ao sistema uma vez que, como é o utilizador que identifica os parâmetros de entrada do sistema difuso, estes podem ter sido

incorretamente identificados o que levaria a uma tomada de decisão errada. Seguidamente é apresentado para cada caso, a identificação do respetivo tipo de batimento.

### 5.7.1. BATIMENTO NORM

Para testar a resposta do sistema ao tipo de batimento NORM, foi selecionado o registo 100. O intervalo aleatório selecionado para identificar o batimento foi [205; 205,8] s. A identificação dos parâmetros do sinal foi efetuada conforme a Figura 68. Na identificação destes pontos usou-se a derivação V5 de um sinal filtrado com os seguintes parâmetros para cada filtro:

- Filtro passa-alto – Ordem = 2 e frequência de corte = 0,05 Hz;
- Filtro passa-baixo – Ordem = 2 e frequência de corte = 35 Hz.

Através da Figura 69, é possível confirmar o tipo de batimento (NORM) através do grupo criado para o efeito. A taxa de batimentos/minuto é de aproximadamente 75 BPM o que indica um ritmo normal para o paciente em causa. De realçar também que no grupo *Representação gráfica* é apresentada a derivação V5 filtrada no intervalo [200, 210] s com as respetivas anotações.

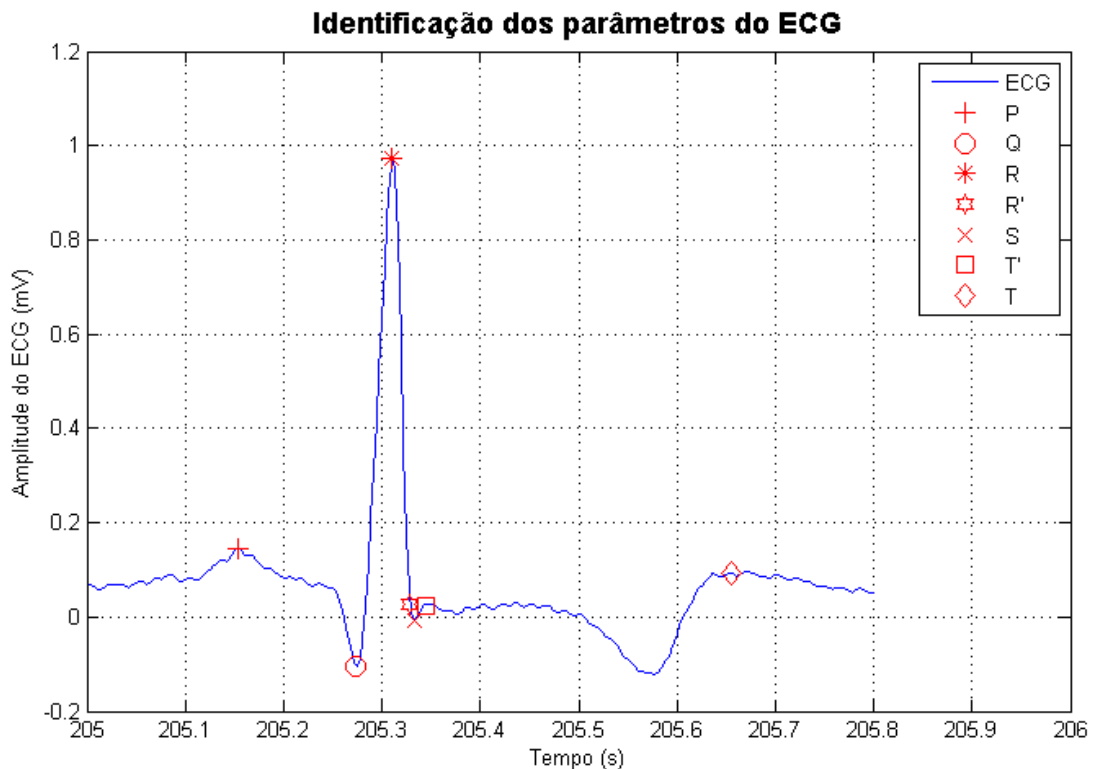
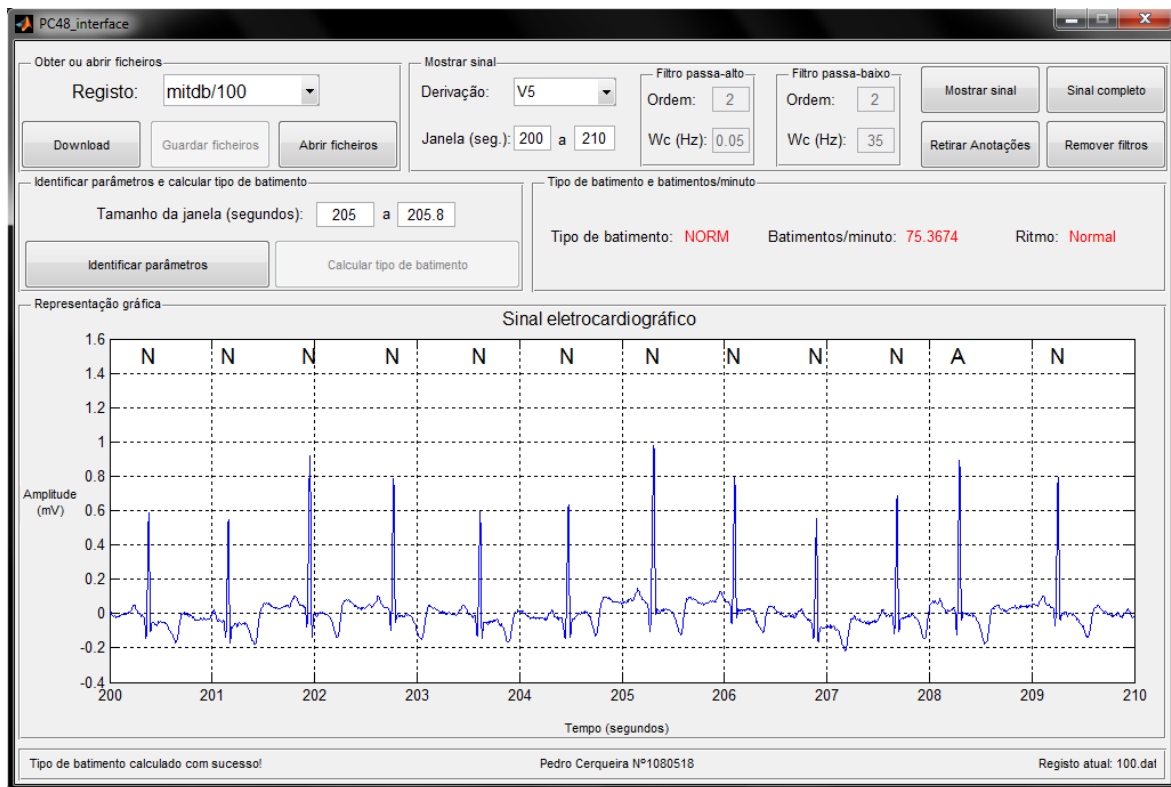


Figura 68 Identificação dos parâmetros para o caso NORM



**Figura 69 Interface gráfica no caso NORM**

### 5.7.2. BATIMENTO LBBB

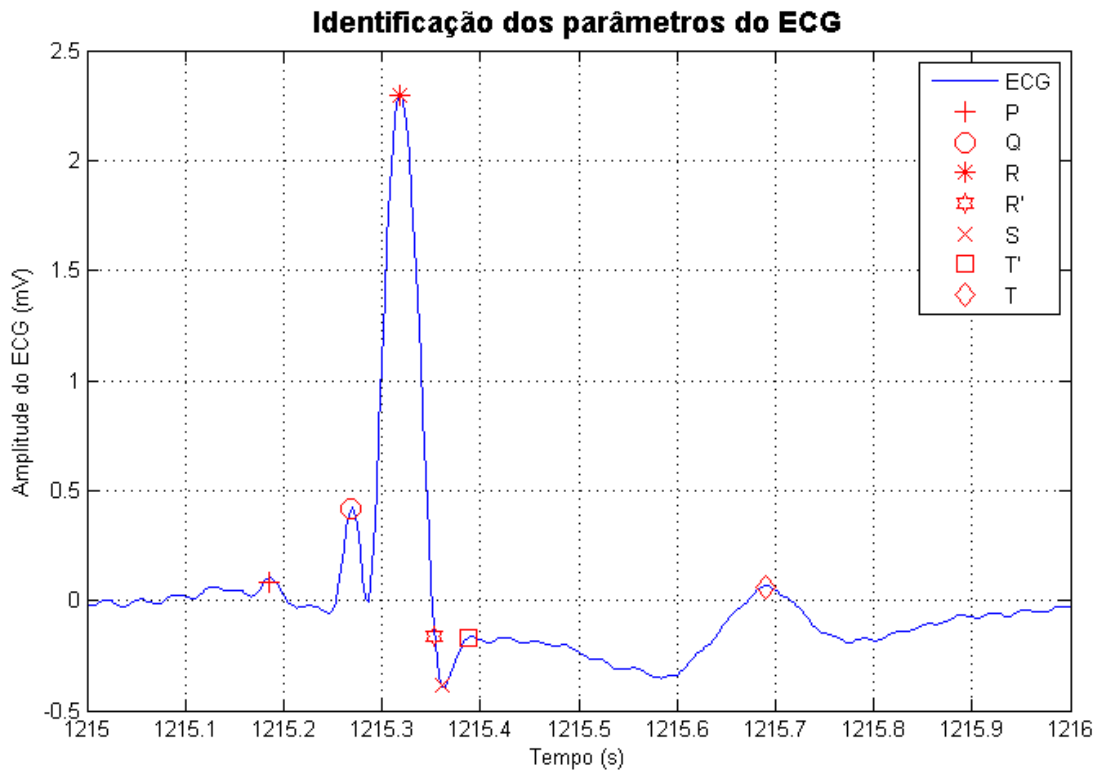
Para este tipo de batimento, dada a dificuldade em identificar os parâmetros do ECG corretamente, não é possível apresentar nenhum exemplo da deteção de LBBB.

### 5.7.3. BATIMENTO RBBB

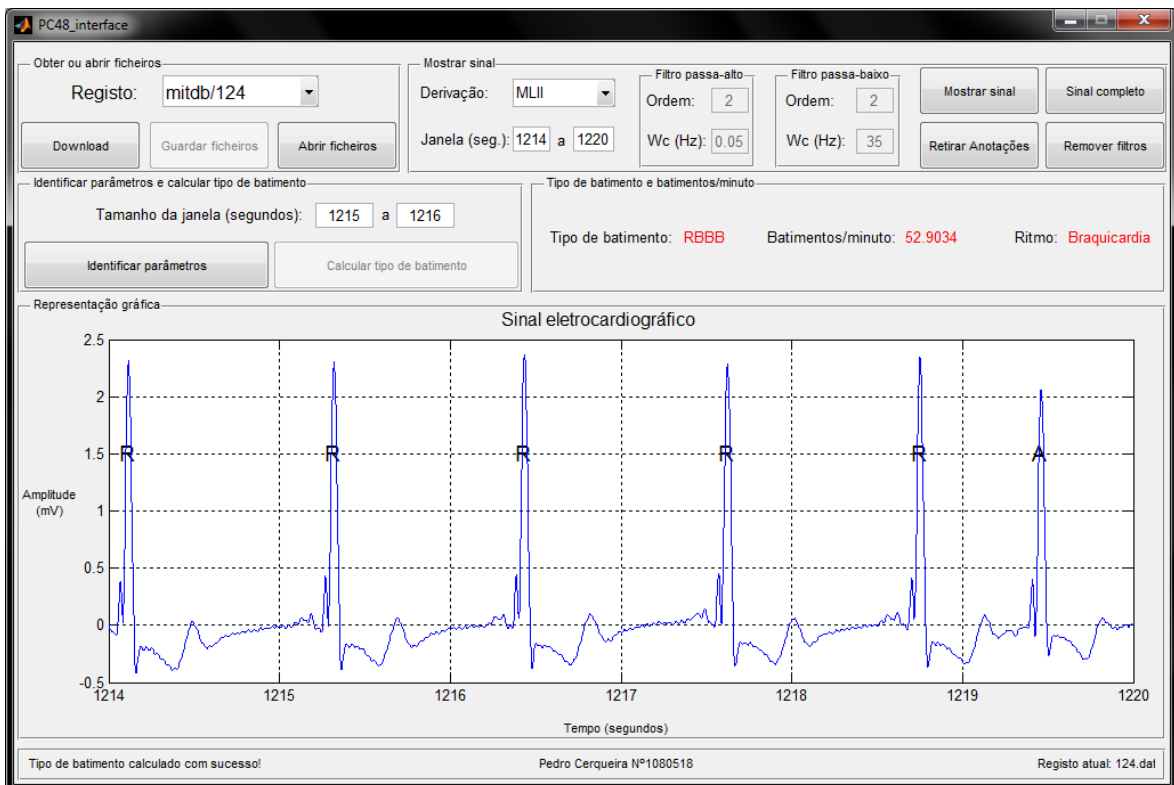
Para o tipo de batimento RBBB, foi seleccionada a derivação MLII do registo 124. Seleccionou-se o batimento presente no intervalo [1215, 1216] s para a respetiva análise. A identificação dos parâmetros pode ser vista na Figura 70. Utilizou-se o sinal filtrado com as seguintes características relativas aos filtros:

- Filtro passa-alto – Ordem = 2 e frequência de corte = 0,05 Hz;
- Filtro passa-baixo – Ordem = 2 e frequência de corte = 35 Hz.

Como se pode ver através da Figura 71, o resultado da identificação do tipo de batimento foi o desejado, RBBB. De realçar que este paciente possui uma taxa de batimentos/minuto aproximadamente igual a 53 BPM logo encontramos na presença de uma braquicardia. Por fim, verifica-se ainda que a anotação do batimento analisado corresponde com o resultado obtido.



**Figura 70** Identificação dos parâmetros para o caso RBBB



**Figura 71** Interface gráfica no caso RBBB

#### 5.7.4. BATIMENTO VPC

O registo 100 possui um batimento do tipo VPC. Com ajuda da interface, analisou-se esse batimento para ver se o sistema difuso respondia corretamente. O batimento encontra-se presente no intervalo [1518,5; 1519,5] s. O batimento assim como a respetiva identificação dos parâmetros pode ser visto na Figura 72. Para a identificação dos parâmetros usou-se os valores dos filtros que têm vindo a ser habituais, isto é:

- Filtro passa-alto – Ordem = 2 e frequência de corte = 0,05 Hz;
- Filtro passa-baixo – Ordem = 2 e frequência de corte = 35 Hz.

O resultado do cálculo do tipo de batimento foi o esperado, VPC, como se pode ver na Figura 73. Como se utilizou o registo 100, a taxa de batimentos/minutos é a mesma que no exemplo do batimento NORM.

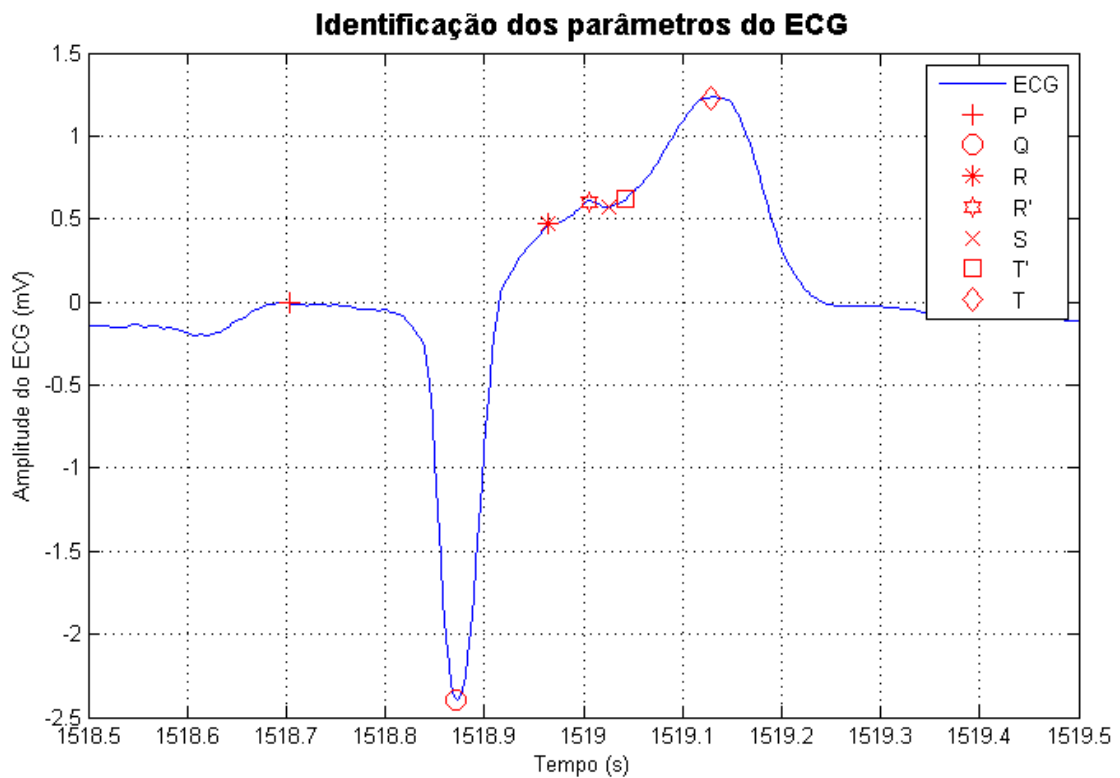


Figura 72 Identificação dos parâmetros para o caso VPC



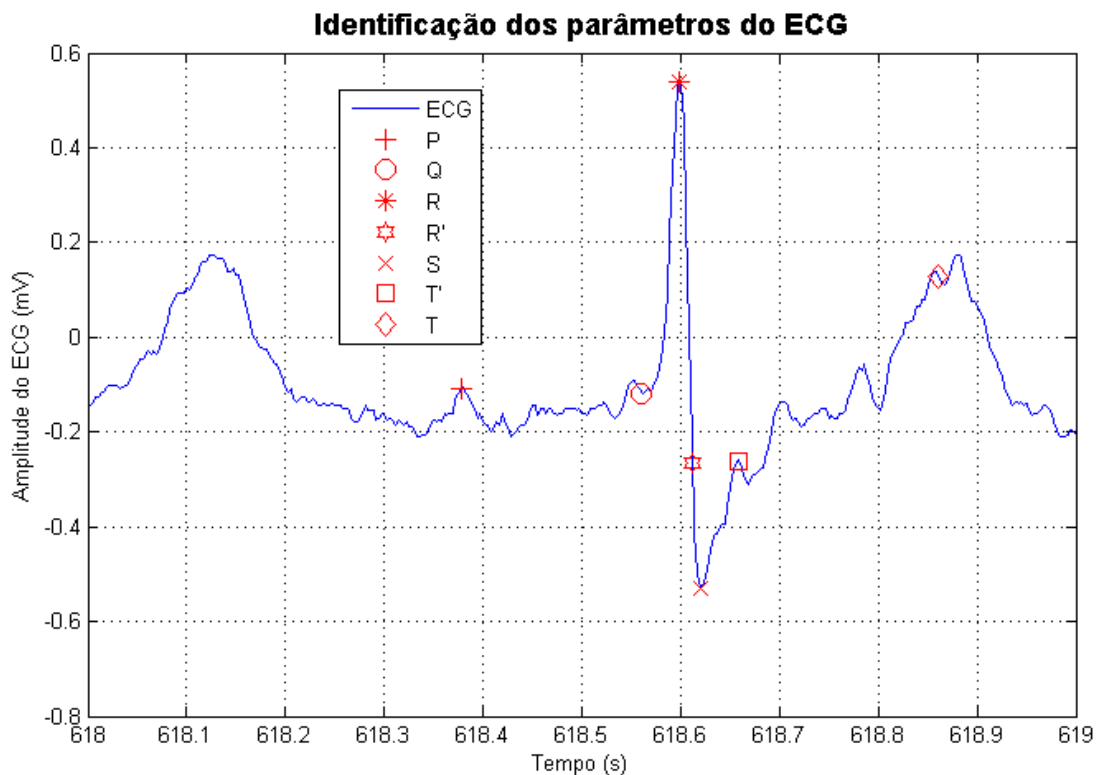
**Figura 73 Interface gráfica no caso VPC**

### 5.7.5. BATIMENTO APC

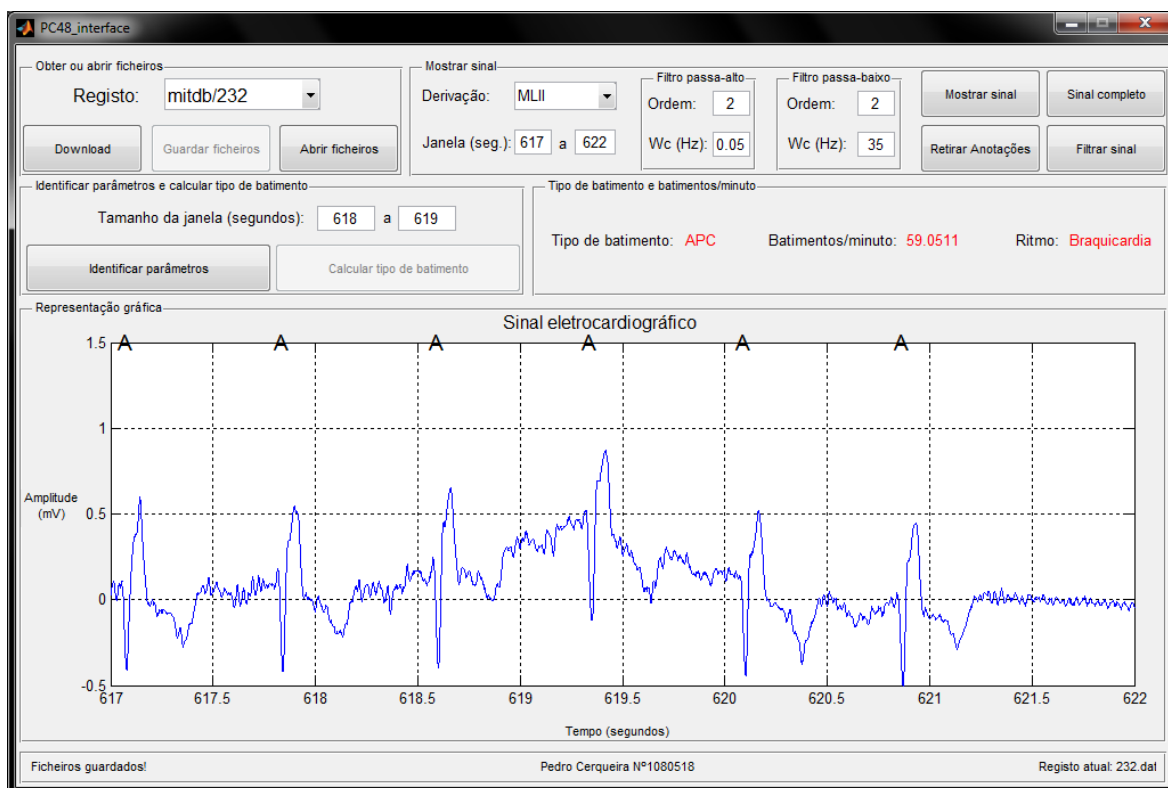
Por fim, para último batimento analisado, APC, utilizou-se o registo 232. Ao contrário dos outros exemplos, neste batimento não se removeu o ruído. A não remoção do ruído teve como base mostrar que, desde que se consiga identificar os pontos corretamente, o sistema é capaz de determinar com sucesso o tipo de batimento.

O batimento analisado foi escolhido aleatoriamente e encontra-se presente no intervalo [618, 619] s. A identificação das ondas no sinal eletrocardiográfico foi efetuada de acordo com a Figura 74. Por sua vez, a Figura 75 mostra o resultado do cálculo do tipo de batimento. Verifica-se que o resultado calculado foi APC e encontra-se de acordo com as anotações presentes no grupo *Representação gráfica*.

Neste sinal, a taxa de batimentos por minuto calculada foi aproximadamente de 59 BPM. Também neste caso, o paciente analisado possuía um ritmo cardíaco anormal, neste caso, braquicardia.



**Figura 74** Identificação dos parâmetros para o caso APC



**Figura 75** Interface gráfica no caso APC

## 6. CONCLUSÕES

Sendo o coração um dos órgãos mais importantes, o sinal eletrocardiográfico torna-se uma ferramenta essencial no que toca à análise e prevenção de eventuais anomalias que possam surgir. Para pessoas não entendidas na área, o ECG pode ser dificilmente compreendido sendo então necessário a ajuda de outro meio para identificar possíveis patologias. Ao longo deste documento foi sendo apresentado um sistema de deteção de arritmias.

### 6.1. OBJETIVOS REALIZADOS

Tendo em conta os objetivos inicialmente propostos, considera-se que estes foram satisfatoriamente cumpridos.

Numa fase inicial desenvolveram-se funções que permitiram obter (*PC41\_obtencao*) e visualizar os sinais provenientes da base de dados utilizada (MIT-BIH *Arrhythmia Database*). Uma das funções desenvolvidas permite mostrar o sinal sem anotações (*PC42\_representacao*) e a outra função permite mostrar o sinal com anotações (*PC43\_rep\_annot*). Foi também desenvolvida uma função que permite guardar os dados obtidos através da base de dados (sinal, informações e anotações) em formato “.mat” (*PC44\_dat2mat*). Além destas funções que permitem apenas a visualização do sinal, foi desenvolvida uma função que implementa a remoção de ruído no ECG (*PC45\_rem\_ruido*) e onde é aplicado um filtro de *Notch*, um filtro passa-alto e um filtro passa-baixo. Estes

filtros removem o ruído da rede elétrica (60 Hz), eliminam o desvio da linha de base e eliminam a interferência vindo do EMG respetivamente.

Posteriormente foi desenvolvida uma função para identificar os parâmetros do batimento que se deseja analisar e calcular os variáveis de entrada que serão usadas pelo sistema difuso (*PC46\_ident\_pontos*). A função desenvolvida solícita ao utilizador que identifique os parâmetros (P, Q, R, R', S, T' e T) do batimento que se pretende analisar. Esses parâmetros são identificados graficamente através do rato. Além disto, também é implementado um algoritmo que deteta todas as ondas R do sinal eletrocardiográfico e armazena a sua localização. A deteção e armazenamento da localização das ondas R, para além da média dos intervalos RR, também permitirá o cálculo da taxa de batimentos por minuto para determinar o ritmo do paciente, normal ou anormal (braquicardia ou taquicardia). Esta função calcula ainda as variáveis de entrada que serão introduzidas no sistema difuso.

Seguidamente foi implementada uma função que toma a decisão de que tipo é o batimento analisado (*PC47\_tipo\_batimento*). Utilizando as variáveis de entrada calculadas com a função anterior, *PC46\_ident\_pontos*, a função *PC47\_tipo\_batimento* invoca o sistema difuso implementado e efetua os cálculos necessários de modo a obter o tipo do batimento analisado. Relativamente ao sistema difuso, modificaram-se os conjuntos difusos da variável de saída uma vez que, depois de efetuados testes, verificou-se que o sistema não obtinha os resultados desejados. A variável de saída inicial possuía conjuntos difusos com formas *Singleton* tendo sido alterados para formas triangulares.

Por fim, e de forma a tornar estas funções mais acessíveis para o utilizador, desenvolveu-se uma interface gráfica que implementa todas as potencialidades das funções enunciadas anteriormente.

## **6.2. LIMITAÇÕES DO SISTEMA E PERSPETIVAS DE DESENVOLVIMENTO**

A principal limitação do sistema consiste na possibilidade de erro por parte do utilizador aquando a identificação dos parâmetros do batimento a analisar. Esta má identificação pode levar a uma tomada de decisão errada por parte do sistema difuso. Outra limitação é o fato da interface estar apenas otimizada para análise de sinais provenientes da MIT-BIH *Arrhythmia Database*.

Relativamente às perspectivas de desenvolvimento, será uma mais-valia melhorar o sistema eliminando as suas limitações. Uma das possíveis melhorias é a adição de um algoritmo que detete todos os parâmetros necessários ao cálculo das variáveis de entrada do sistema difuso. A inclusão de um algoritmo que obtivesse todos os parâmetros automaticamente, seria benéfico ao ponto de permitir analisar todos os batimentos do sinal e determinar a patologia do doente em causa. Outra sugestão de melhoramento seria a possibilidade de análise de sinais provenientes de outras fontes ou até mesmo sinais vindos dos sistemas de aquisição de sinais eletrocardiográficos. Por fim, e não menos importante que as anteriores, a adição de mais anomalias à base de conhecimento do sistema difuso de modo a efetuar a deteção de um número maior de tipos de batimentos.

### **6.3. APRECIÇÃO FINAL**

Tendo em consideração o descrito anteriormente, verificou-se satisfatoriamente o cumprimento dos objetivos propostos. Espera-se que este projeto, acima de tudo, sirva de base de conhecimento para aqueles cujo interesse reside na manipulação de sinais eletrocardiográficos e na análise de batimentos. Para finalizar, é com apreço que se obtém este documento depois de todo o trabalho desenvolvido e que aporta um elevado grau de satisfação e de autorrealização.



## Referências Documentais

- [1] Manual Merck, Biblioteca Médica Online. — *Diagnóstico das doenças cardíacas, Exames de diagnóstico*. [Consult. 27 Ago. 2013]. Disponível em:  
<URL:<http://www.manualmerck.net/?id=41&cn=610&ss=>>.
- [2] World Health Organization — *Cardiovascular diseases (CVDs)*. [Consult. 11 Fev. 2013]. Disponível em:  
<URL:<http://www.who.int/mediacentre/factsheets/fs317/en/index.html>>.
- [3] World Health Organization — *Types of cardiovascular disease*. [Consult. 11 Fev. 2013]. Disponível em:  
<URL:[http://www.who.int/cardiovascular\\_diseases/en/cvd\\_atlas\\_01\\_types.pdf](http://www.who.int/cardiovascular_diseases/en/cvd_atlas_01_types.pdf)>.
- [4] GUYTON, Arthur C. HALL, John E. — *Textbook of Medical Physiology*. 11<sup>a</sup> ed. Philadelphia, Pennsylvania: Elsevier Saunders, 2006. ISBN 0-7216-0240-1.
- [5] KLABUNDE, Richard E. — *Cardiovascular Physiology Concepts*. 2<sup>a</sup> ed. United States, Philadelphia: Wolters Kluwer - Lippincott Williams & Wilkins, 2007. ISBN 978-1-4511-1384-6.
- [6] Wikipedia. — *Membrane potential*. [Consult. 18 Abr. 2013]. Disponível em:  
<URL:[http://en.wikipedia.org/wiki/Membrane\\_potential](http://en.wikipedia.org/wiki/Membrane_potential) >.
- [7] ORACLE ThinkQuest. — *Introduction: Understanding Action Potential*. [Consult. 18 Abr. 2013]. Disponível em:  
<URL:<http://library.thinkquest.org/C003758/Function/UnderstandingActionPotential.htm> >.
- [8] PINNELL, Jeremy. TURNER, Simon. HOWELL, Simon. — *Cardiac muscle physiology*. [Consult 18 Abr. 2013]. Disponível em:  
<URL :<http://ceaccp.oxfordjournals.org/content/7/3/85.full.pdf+html>>.
- [9] MALMIVUO, Jaakko. PLONSEY, Robert. — *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields*. New York: Oxford University Press, 1995. [Consult 18 Abr. 2013]. Disponível em:  
<URL:<http://www.bem.fi/book/>>.
- [10] ZYWIETZ, Chr. — *A Brief History of Electrocardiography: Progress through Technology*. [Consult. 23 Fev. 2013]. Disponível em:  
<URL:[http://www.openecg.net/Tutorials/A\\_Brief\\_History\\_of\\_Electrocardiography.pdf](http://www.openecg.net/Tutorials/A_Brief_History_of_Electrocardiography.pdf)>.
- [11] CHAVES, Paulo Castro. MOREIRA, Adelino Leite. — *Electrocardiografia*. Faculdade de Medicina da Universidade do Porto, Serviço de Fisiologia. Porto, 2001/2002. [Consult. 25 Fev. 2013]. Disponível em:  
<URL:[http://fisiologia.med.up.pt/Textos\\_Apoio/cardiaco/ECG.pdf](http://fisiologia.med.up.pt/Textos_Apoio/cardiaco/ECG.pdf) >.

- [12] POCOCK, Gillian. RICHARDS, Christopher D. — *Fisiologia Humana: A Base da Medicina*. 2ª ed. Brasil, Rio de Janeiro: Guanabara Koogan, 2006. ISBN 85-277-1138-9.
- [13] RIBEIRO, Ricardo Luiz. REIS, Pablo Ferreira. Bomfim, Alfredo de Souza. BARBOSA, Eduardo Corrêa. BOGHOSSIAN, Silvia Helena Cardoso. VELOSO, Henrique H. GINEFRA, Paulo. — *Antigos e Novos Conceitos sobre a Onda U do Eletrocardiograma*. [Consult. 28 Fev. 2013]. Disponível em: <URL:[http://www.rbconline.org.br/wp-content/uploads/a2004\\_v17\\_n03\\_art05.pdf](http://www.rbconline.org.br/wp-content/uploads/a2004_v17_n03_art05.pdf)>.
- [14] Dicionário da Língua Portuguesa. — *lógica In Infopédia*. Porto: Porto Editora, 2003-2013. [Consult. 1 Abr. 2013]. Disponível em: <URL:<http://www.infopedia.pt/lingua-portuguesa/l%C3%B3gica>>.
- [15] KOHAGURA, Tiago. — *Lógica Fuzzy e Suas Aplicações*. Universidade Estadual de Londrina. Londrina, 2007. [Consult. 1 Abr. 2013]. Disponível em: <URL:<http://www2.dc.uel.br/nourau/document/?view=601>>.
- [16] GARRIDO, Angel. — *A Brief History of Fuzzy Logic*. Faculty of Sciences. Madrid. [Consult. 1 Abr. 2013]. Disponível em: <URL:<http://brain.edusoft.ro/index.php/brain/article/viewFile/308/390>>.
- [17] JESUS, Isabel S. — *Controlo Lógico Difuso*. Instituto Superior de Engenharia do Porto, 2012.
- [18] GIL, Paulo. — *Introdução ao Controlo Difuso*. Universidade Nova de Lisboa, Departamento de Engenharia Eletrotécnica da Faculdade de Ciências e Tecnologia. [Consult. 3 Abr. 2013]. Disponível em: <URL:[http://www-scd.dee.fct.unl.pt/leec/cintel/Arquivo/slides\\_difuso.pdf](http://www-scd.dee.fct.unl.pt/leec/cintel/Arquivo/slides_difuso.pdf)>.
- [19] TANSCHKEIT, Ricardo. — *Sistemas Fuzzy*. Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica. [Consult. 3 Abr. 2013]. Disponível em: <URL:<http://www2.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas%20Fuzzy.pdf>>.
- [20] AZIZ, Shahariz Abdul. PARTHIBAN, Jeyakody. — *Fuzzy Sets & Fuzzy Operations*. [Consult. 3 Abr. 2013]. Disponível em: <URL:[http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/sbaa/report.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.html)>.
- [21] Inteligência Computacional Aplicada. — *Conjuntos Fuzzy*. Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica. [Consult. 3 Abr. 2013]. Disponível em: <URL:<http://www2.ica.ele.puc-rio.br/Downloads/40/LN-Cursop3-Formatos&Opera%C3%A7%C3%B5es2.pdf>>.
- [22] MIRANDA, Vladimiro. — *Fundamentos de Controlo Difuso*. Faculdade de Engenharia da Universidade do Porto, Departamento de Engenharia Eletrotécnica e de Computadores. Porto, 1998.
- [23] PAIVA, Rui. — *Identificação Neuro-Difusa: Aspectos de Interpretabilidade*. Faculdade de Ciências e Tecnologias da Universidade de Coimbra, Departamento de Engenharia Informática. Coimbra, 1999. [Consult. 5 Abr. 2013]. Disponível em:

- <URL:<https://estudogeral.sib.uc.pt/bitstream/10316/15368/1/Identifica%C3%A7%C3%A3o%20Neuro-Difusa.pdf>>.
- [24] JANTZEN, Jan. — *Foundations of Fuzzy Control*. 1ª ed. England, West Sussex: John Wiley & Sons Ltd, 2007. ISBN: 978-0-470-02963-3.
- [25] AGUIAR, Manuel. — *Modelos para produção distribuída*. Faculdade de Engenharia da Universidade do Porto, Departamento de Engenharia Eletrotécnica e de Computadores. [Consult. 15 Abr. 2013]. Disponível em: <URL:<http://repositorio-aberto.up.pt/bitstream/10216/12481/2/Texto%20integral.pdf>>.
- [26] GOMIDE, Fernando A. GUDWIN, Ricardo R. TANSCHHEIT, Ricardo. — *Conceitos fundamentais da teoria dos conjuntos fuzzy, lógica fuzzy e aplicações*. Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, Departamento de Engenharia de Computação e Automação Industrial. [Consult. 7 Abr. 2013]. Disponível em: <URL:<ftp://ftp.dca.fee.unicamp.br/pub/docs/gudwin/publications/ifsa95.pdf>>.
- [27] VERNON, James — *Fuzzy Logic Systems*. [Consult. 20 Nov. 2012]. Disponível em: <URL:<http://www.control-systems-principles.co.uk/whitepapers/fuzzy-logic-systems.pdf>>.
- [28] Physionet. — *the research resource for complex physiologic signals*. [Consult. 29 Abr. 2013]. Disponível em: <URL:<http://www.physionet.org/>>.
- [29] Physionet. — *MIT-BIH Arrhythmia Database*. [Consult. 29 Abr. 2013]. Disponível em: <URL:<http://www.physionet.org/physiobank/database/mitdb/>>.
- [30] Physionet. — *MIT-BIH Arrhythmia Database Directory*. [Consult. 15 Mai. 2013]. Disponível em: <URL:<http://www.physionet.org/physiobank/database/html/mitdbdir/intro.htm>>.
- [31] FERREIRA, Kathy. — *MANIPULAÇÃO DE ARQUIVOS PhysioBank*. Universidade Federal de Minas Gerais, Departamento de Engenharia Eletrônica. [Consult. 20 Mai. 2013]. Disponível em: <URL:<http://www.cpdee.ufmg.br/~MACSIN/services/reports/mac304.pdf>>.
- [32] Physionet. — *An Introduction to PhysioToolkit*. [Consult. 1 Jun. 2013]. Disponível em: <URL: <http://www.physionet.org/physiotools/getting-started.shtml>>.
- [33] Physionet. — *The WFDB Toolbox for MATLAB*. [Consult. 1 Jun. 2013]. Disponível em: <URL: <http://www.physionet.org/physiotools/matlab/wfdb-swig-matlab/>>.
- [34] Physionet. — *The WFDB Toolbox for MATLAB*. [Consult. 15 Jul. 2013]. Disponível em: <URL: <http://www.physionet.org/physiotools/matlab/wfdb-app-matlab/>>.
- [35] YEH, Yun-Chi. WANG, Wen-June. — *QRS complexes detection for ECG signal: The Difference Operation Method*. Computer Methods and Programs in Biomedicine, April 2008. [Consult. 17 Jun. 2013]. Disponível em: <URL:<http://www.sciencedirect.com/science/article/pii/S0169260708001004>>.
- [36] HENRION, Stéphane. MAILHES, Corinne. — *Biomedical signal processing and analysis*. Information Society Technologies, 2001. [Consult. 17 Jun. 2013]. Disponível em:

- <URL:[http://ursafe.tesa.prd.fr/ursafe/new/PublicDeliverable/Ursafe\\_Signal\\_Processing.pdf](http://ursafe.tesa.prd.fr/ursafe/new/PublicDeliverable/Ursafe_Signal_Processing.pdf)>.
- [37] CHADHA, Aman. — *engineer, web developer & computer enthusiast*. [Consult. 1 Jun. 2013]. Disponível em: <URL: <http://www.amanchadha.com/>>.
- [38] PAN, Jiapu. TOMPKINS, Willis J. — *A Real-Time QRS Detection Algorithm*. IEEE Transactions on Biomedical Engineering, vol. BME-32, NO. 3, March 1985. [Consult. 2 Jun. 2013]. Disponível em: <URL:<http://www.projectcuris.com/files/Pan%20and%20Tompkins%201985.pdf>>.
- [39] YEH, Yun-Chi. WANG, Wen-June. CHIOU, Che Wun. — *Heartbeat Case Determination Using Fuzzy Logic Method on ECG Signals*. International Journal of Fuzzy Systems, December 2009. [Consult. 12 Jun. 2013]. Disponível em: <URL:[http://www.ijfs.org.tw/ePublication/2009\\_paper\\_4/ijfs09-4-r-4\\_IJFS\\_template-0907-proof%20ok-.pdf](http://www.ijfs.org.tw/ePublication/2009_paper_4/ijfs09-4-r-4_IJFS_template-0907-proof%20ok-.pdf)>.
- [40] MATLAB, The MathWorks. — *Fuzzy Logic Toolbox™ 2, User's Guide*. Setembro 2009.
- [41] MATLAB, The MathWorks. — *Fuzzy Logic Toolbox*. [Consult. 30 Nov. 2012]. Disponível em: <URL:[http://www.mathworks.com/help/fuzzy/index.html?s\\_cid=BB](http://www.mathworks.com/help/fuzzy/index.html?s_cid=BB)>.

# Anexo A. *Toolbox* “Fuzzy Logic” do MATLAB

Neste anexo são identificados os principais aspetos da *toolbox* que permitem o desenvolvimento de sistemas difusos.

## A.1. FERRAMENTAS GRÁFICAS

A *toolbox fuzzy logic* fornece algumas ferramentas gráficas que permitem a utilização de algumas funções através de uma interface gráfica. Nesta secção são apresentadas as seguintes ferramentas gráficas disponíveis nesta *toolbox*:

- Sistema de Inferência Difusa;
- Sistema Adaptativo de Inferência Neuro-Difusa;
- *Clustering*.

Será dada uma maior importância ao sistema de inferência difusa pois é esta a interface gráfica que é usada no âmbito deste projeto.

### A.1.1. SISTEMA DE INFERÊNCIA DIFUSA

Nesta secção vão ser identificadas as várias particularidades do sistema de inferência difusa para desenvolvimento de sistemas difusos.

Como já anteriormente referido, a interface gráfica desta *toolbox* é uma das mais-valias evidenciadas. Através do editor *Fuzzy Inference System* (FIS) o desenvolvimento de sistemas difusos torna-se bastante intuitivo e fácil de compreender. Para poder iniciar esta interface gráfica, basta digitar na linha de comandos (*Command Window*) do MATLAB a seguinte instrução:

```
>> fuzzy
```

Caso esta *toolbox* se encontre presente no MATLAB, a janela seguinte (Figura 76) é apresentada no ecrã.

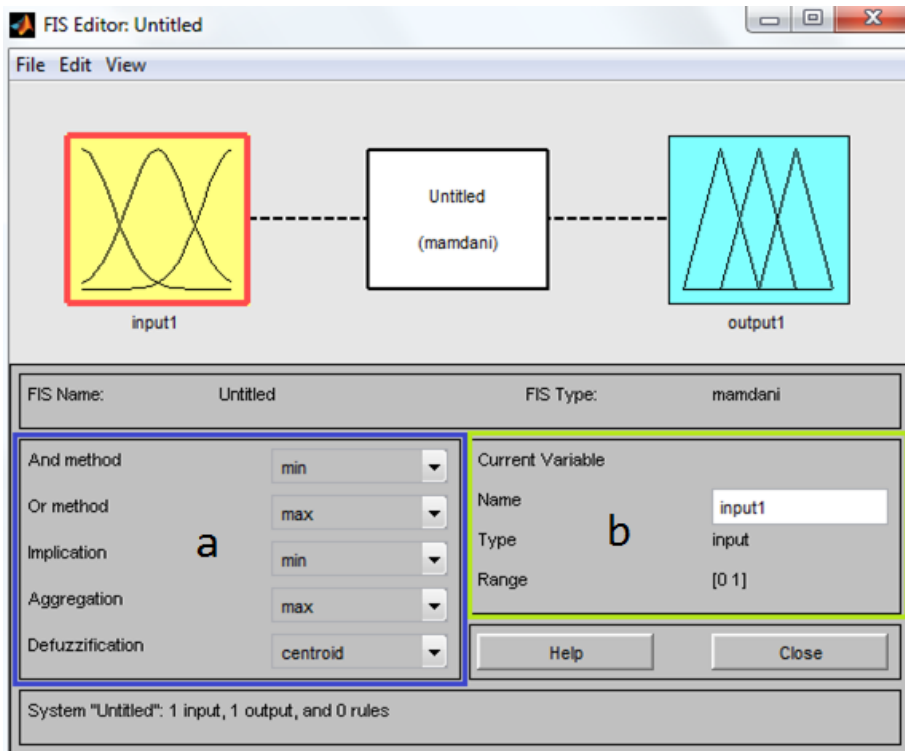


Figura 76 Janela principal da *toolbox fuzzy logic*

Como se pode comprovar através da análise da Figura 76, trata-se de uma interface bastante amigável. Dentro da caixa **a**, encontram-se os métodos dos operadores lógicos *and* e *or*, os métodos de implicação, agregação e desfuzificação. Por sua vez, dentro da caixa **b** é onde se pode proceder à alteração do nome das variáveis linguísticas. Atualmente, na Figura 76, encontra-se selecionada uma variável linguística de entrada com o nome *input1*. Este nome pode ser alterado para outro da escolha do projetista (e.g. tipo de batimento).

A escolha do tipo de controlador a usar, *Sugeno* ou *Mamdani*, é feita através do menu *File* e seguidamente em *New FIS*, como se pode ver na Figura 77.

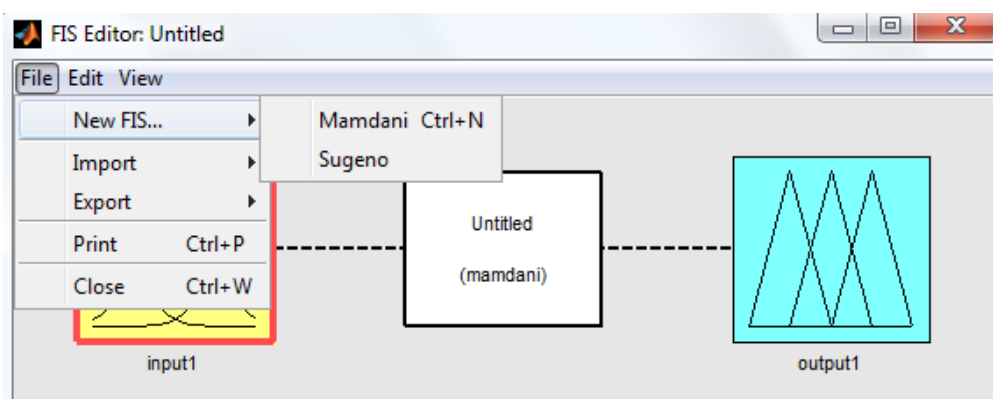
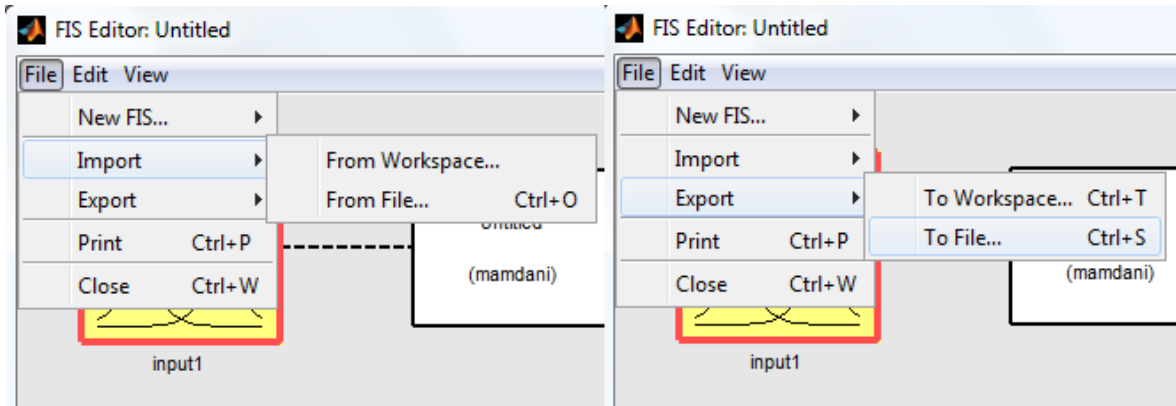


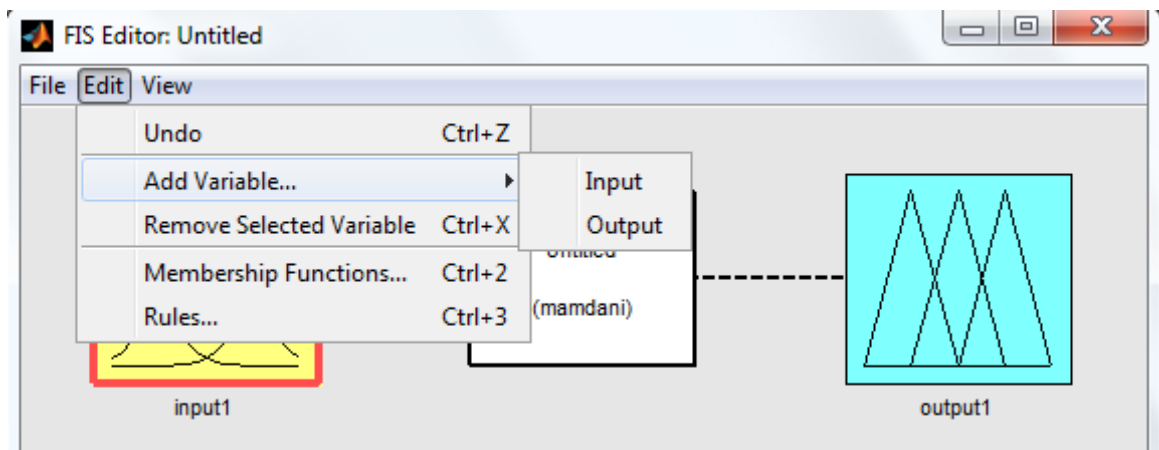
Figura 77 Escolha do tipo de controlador

Ainda no que diz respeito à Figura 77, podem ver-se os menus *Import* e *Export*. Estes dois menus permitem como o próprio nome indica, importar o controlador do *workspace* ou de um ficheiro; e exportar o controlador para *workspace* ou para um ficheiro. Na Figura 78 pode ver-se o atrás enunciado.



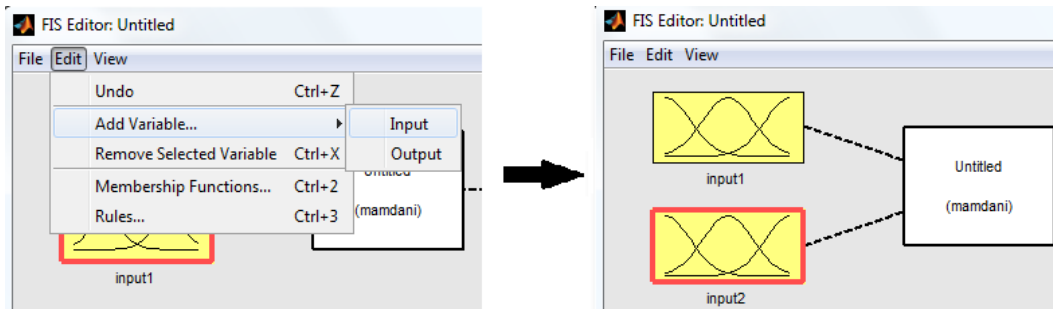
**Figura 78 Importar e exportar controladores difusos**

Por sua vez, no menu *Edit* é possível adicionar variáveis linguísticas tanto de entrada como de saída (Figura 79). O processo contrário, remoção da variável, pode ser feito clicando na opção *Remove Selected Variable*. Nesta opção, a variável atualmente selecionada será aquela que irá ser removida. Ainda no menu *Edit*, existe a possibilidade de aceder ao editor das funções de pertinência e ao editor das regras (Figura 79).



**Figura 79 Menu *Edit* do editor FIS**

Ao adicionar uma variável de entrada, a interface gráfica apresentará o resultado através do diagrama de blocos presente no editor FIS, como se pode ver na Figura 80.

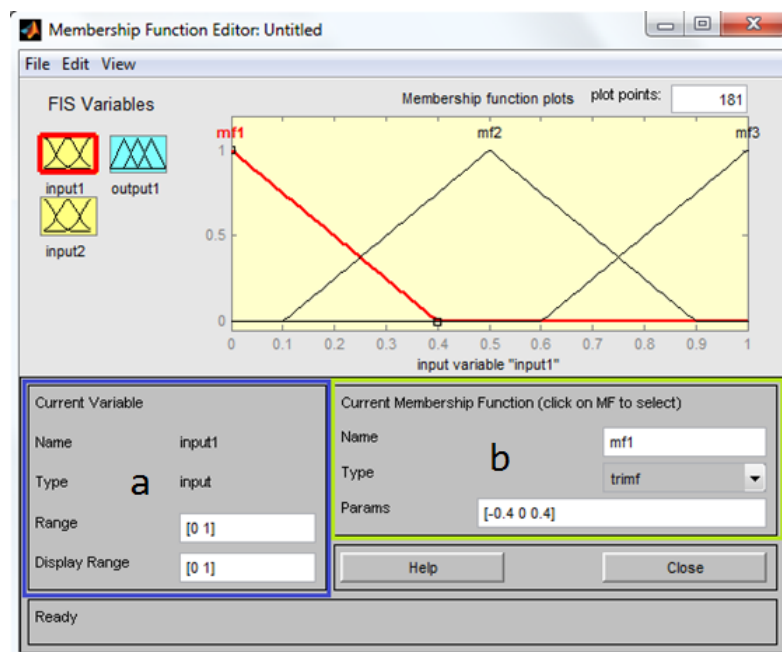


**Figura 80 Adição de uma variável linguística de entrada**

O acesso ao editor das funções de pertença é feito através do menu *Edit* clicando em *Membership Functions*, através do atalho *CTRL+2* ou ainda clicando duas vezes numa das variáveis existentes. Depois de efetuadas uma destas três ações, o editor de funções de pertença irá aparecer no ecrã (Figura 81).

Neste editor pode:

- Ver-se as variáveis existentes tanto de entrada como de saída;
- Ver-se um gráfico com as funções de pertença existentes para uma dada variável selecionada;
- Alterar-se o universo de discurso de uma dada variável selecionada;
- Alterar-se as propriedades (nome, tipo e parâmetros) de uma função de pertença selecionada.

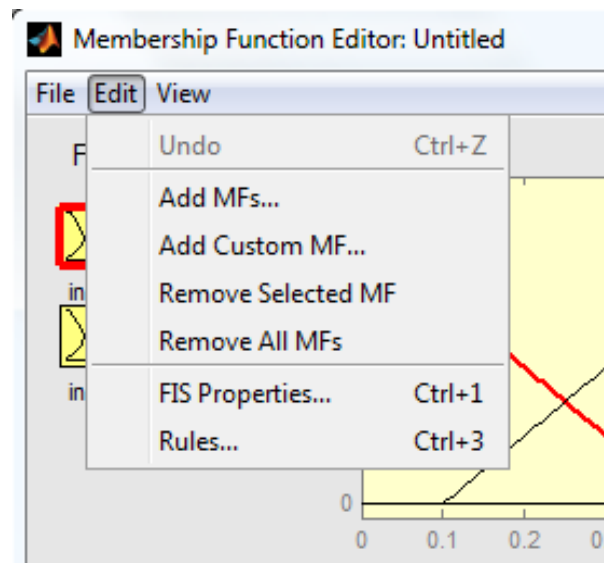


**Figura 81 Editor das funções de pertença**

Relativamente à Figura 81, é dentro da caixa **a** que se pode alterar o universo de discurso de uma dada variável. A seleção da variável é feita através de um clique na variável desejada, na parte superior esquerda (*FIS Variables*). Para alterar as funções de pertença, é necessário escolher a variável que contém a função de pertença a alterar e, dentro da caixa **b**, pode proceder-se à alteração do nome, do tipo (triangular, trapezoidal, etc.) e dos parâmetros da função de pertença. De notar que na Figura 81 a variável selecionada é a *input1* e a função de pertença selecionada é a *mfl*.

No menu *Edit* do editor das funções de pertença (Figura 82) existe a possibilidade de:

- Adicionar funções de pertença na variável selecionada;
- Adicionar uma função de pertença personalizada na variável selecionada;
- Remover a função de pertença selecionada na variável selecionada;
- Remover todas as funções de pertença existentes na variável selecionada;
- Abrir o editor FIS;
- Abrir o editor de regras.



**Figura 82 Menu *Edit* do editor de funções de pertença**

Esta *toolbox* disponibiliza um editor de regras que vai ser analisado a seguir. Para aceder a este editor, é necessário clicar em *Rules* no menu *Edit*, através do atalho *CTRL+3* ou ainda clicando duas vezes no controlador presente no *FIS Editor*.

O editor de regras (Figura 83) permite adicionar e modificar as regras de um controlador difuso. As regras criadas podem ser vistas dentro da caixa **a** na Figura 83. O número de

regras que é possível criar é igual ao número de combinações possíveis entre as variáveis de entrada e saída e as funções de pertinência existentes. Na caixa **b**, é escolhido o valor (função de pertinência) que toma a primeira variável de entrada. Por sua vez, na caixa **c**, é escolhido o valor que toma a segunda variável de entrada. Caso seja necessário estes valores podem ser negados através da *check box* NOT. Dentro da caixa **d**, é escolhido o tipo de operação lógica entre ambas as variáveis selecionadas. Por último, a caixa **e** indica qual a saída que o controlador deverá entregar para os valores indicados das variáveis de entrada. Através do botão *Add rule*, a regra é adicionada ao controlador. Os botões *Delete rule* e *Change rule* permitem eliminar e modificar uma regra já existente. É possível ainda alterar o peso da regra (entre 0 a 1) através da caixa de texto que se encontra dentro da caixa **f**.

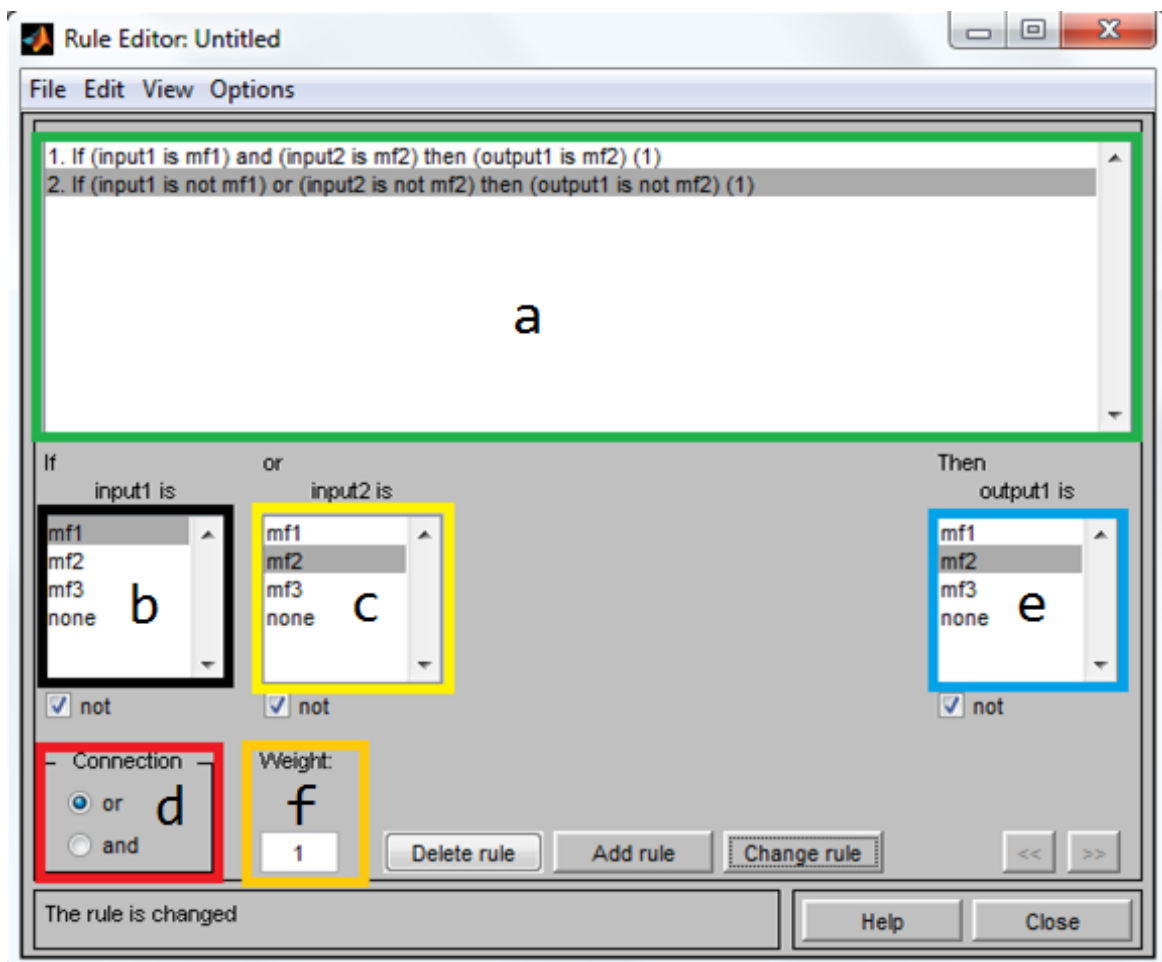
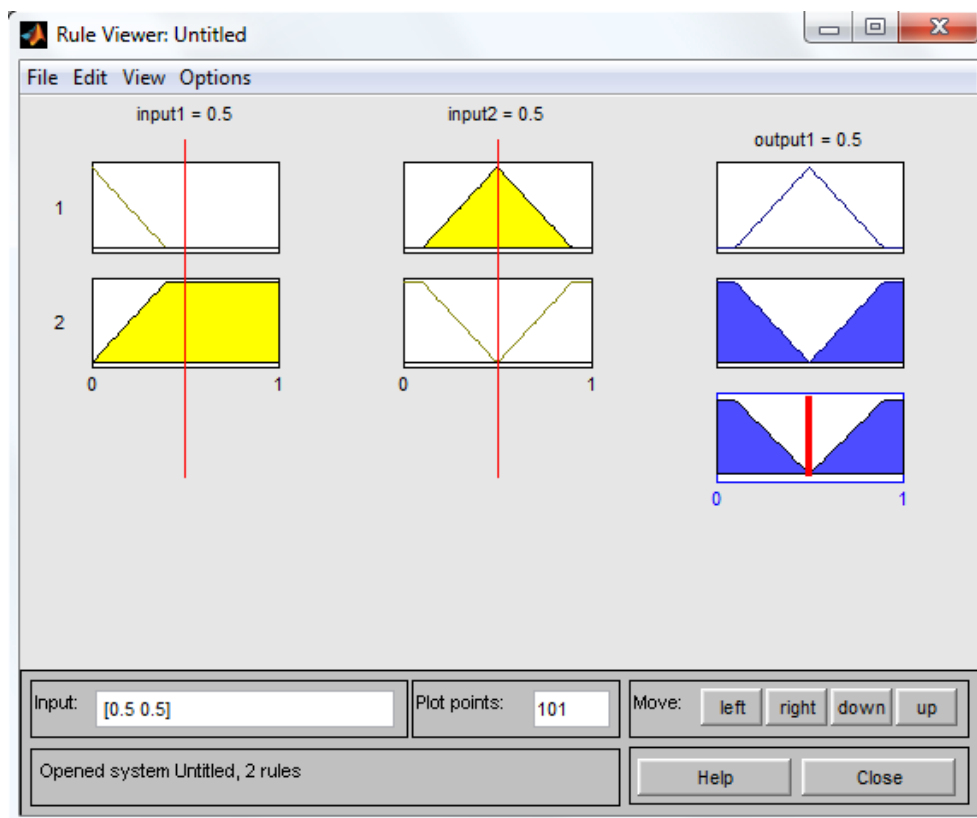


Figura 83 Editor de regras

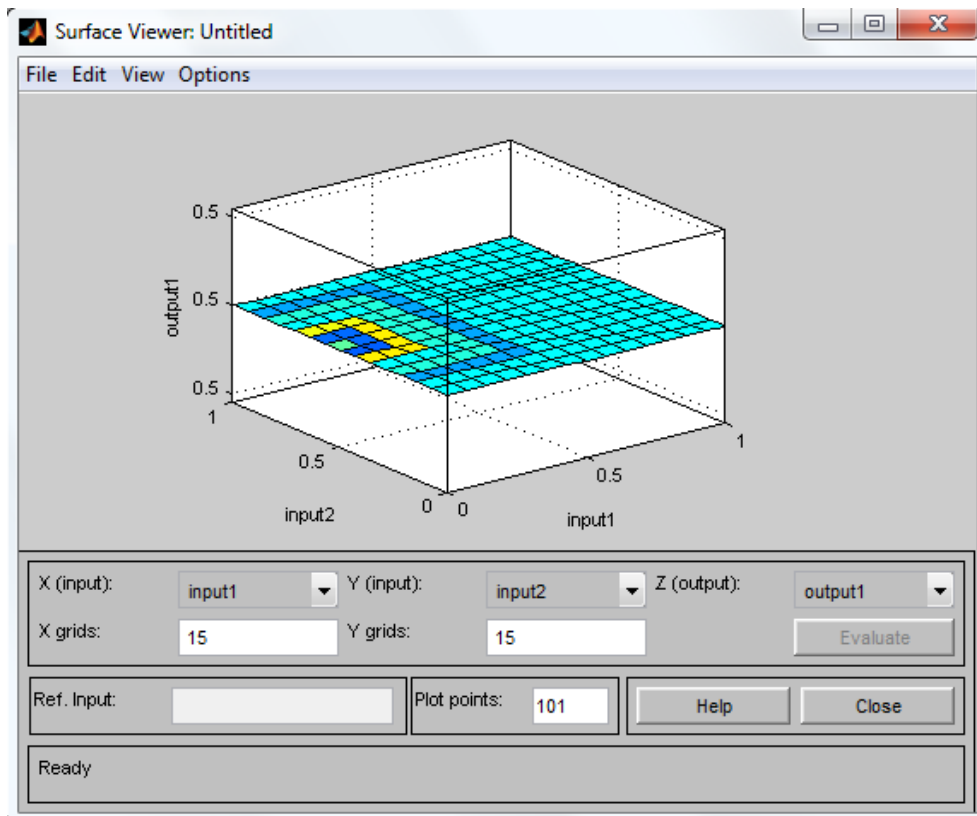
De modo a poder ver o resultado do controlador difuso criado, esta *toolbox* tem duas ferramentas que permitem visualizar o comportamento do controlador consoante as variáveis de entrada tomam diferentes valores. Essas duas ferramentas são o *Rule Viewer* e o *Surface Viewer*.

O *Rule Viewer* permite alterar as variáveis de entrada de modo a simular todos os casos possíveis e ver o resultado da ou das variáveis de saída. A variação dos valores que as variáveis podem tomar é feita através da alteração da posição da reta vertical vermelha. O *Rule Viewer* pode ser visto na Figura 84.

Por sua vez, o *Surface Viewer*, permite ver um gráfico tridimensional cujos eixos X e Y são variáveis de entrada e o eixo Z é uma variável de saída. Na eventualidade do sistema possuir mais do que duas variáveis de entrada e uma de saída, pode escolher-se que variáveis podem ser apresentadas no gráfico. O *Surface Viewer* pode ser visto na Figura 85.



**Figura 84** *Rule Viewer*



**Figura 85** *Surface Viewer*

Todos os editores e visualizadores anteriormente apresentados podem ser chamados através da linha de comandos. Além do comando já apresentado para abrir o editor FIS (*fuzzy*) existem os seguintes:

- *mfedit* – abre o editor das funções de pertinência (Figura 81);
- *ruleedit* – abre o editor de regras (Figura 83);
- *ruleview* – abre o *Rule Viewer* (Figura 84);
- *surfview* – abre o *Surface Viewer* (Figura 85).

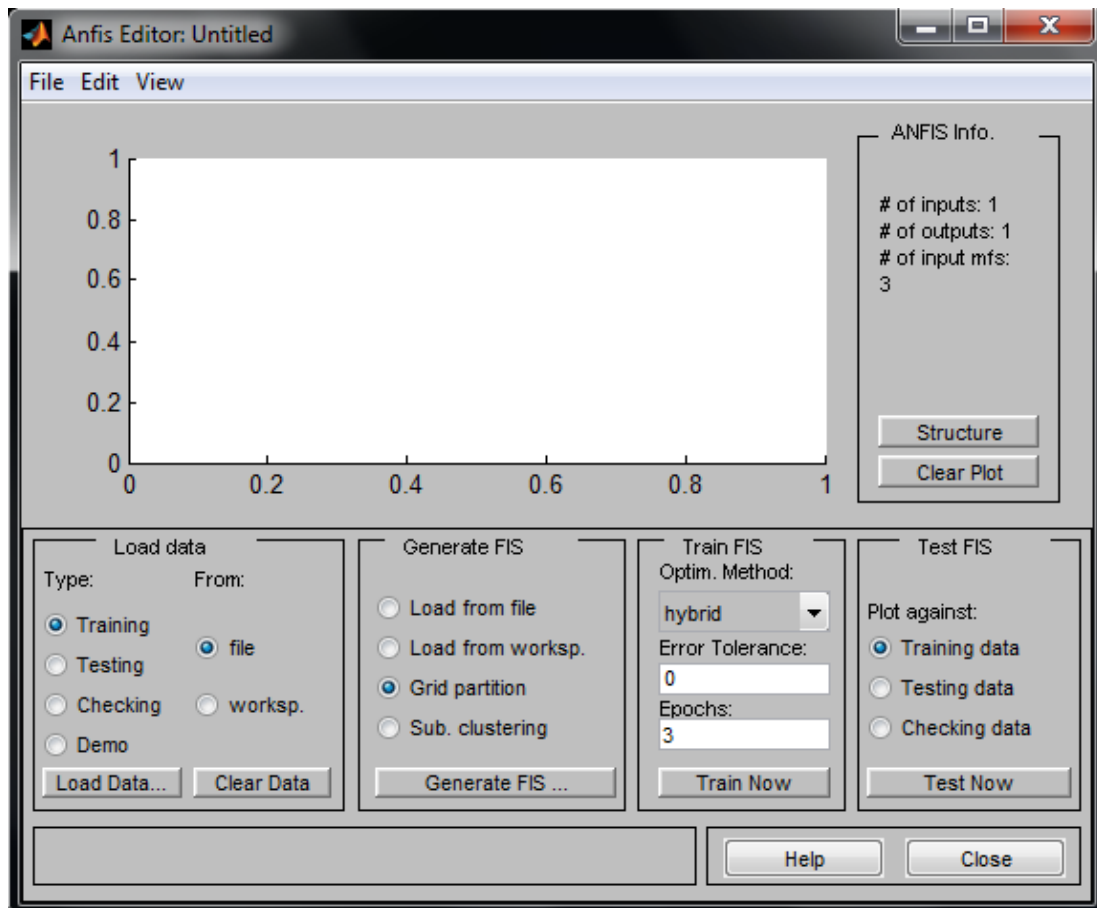
### **A.1.2. SISTEMA ADAPTATIVO DE INFERÊNCIA NEURO-DIFUSA**

A presente *toolbox* também disponibiliza uma interface gráfica para os sistemas adaptativos de inferência neuro-difusas (*Adaptive Neuro-Fuzzy Inference System – ANFIS*) denominada de editor ANFIS. Este editor (Figura 86) permite implementar sistemas adaptativos de inferência neuro-difusa. São usados modelos adaptativos onde um sistema de inferência difusa é estruturado através de um modelo neuronal e cujos parâmetros (do FIS) podem ser ajustados com base em diversas técnicas de aprendizagem.

Para abrir esta interface gráfica, pode recorrer-se ao comando seguinte:

```
>> anfisedit
```

De uma forma resumida, dado um conjunto de dados entrada/saída, esta interface gráfica permite construir um FIS cujas funções de pertinência são ajustadas usando um algoritmo de retropropagação sozinho ou combinado com um método dos mínimos quadrados (método híbrido). Isto permite que o sistema difuso desenvolvido aprenda com os dados que são modelados.



**Figura 86** Editor ANFIS

Através do editor ANFIS presente na Figura 86 é possível:

- Carregar dados (treino, teste e verificação) selecionando os botões desejados na caixa *Load data* da interface gráfica e clicar em *Load data...* para efetuar esse carregamento. Os dados carregados são apresentadas no gráfico presente no editor.
- Gerar um modelo FIS ou carregar um modelo FIS usando as opções da caixa *Generate FIS* da interface gráfica.
- Visualizar a estrutura do modelo FIS gerado ou carregado clicando no botão *Structure* da caixa *ANFIS info*.

- Escolher qual o método de otimização para o modelo FIS: retropropagação ou método híbrido. Isto é feito na caixa *Train FIS*.
- Escolher o número de fases de treino e a tolerância do erro de treino ainda na caixa *Train FIS*.
- Treinar o modelo FIS clicando no botão *Train Now*. Durante o treino do modelo, os parâmetros das funções de pertinência são ajustados e os resultados são apresentados no gráfico disponível para o efeito.
- Ver a saída do modelo FIS em comparação com o treino, ou o teste, ou a verificação, clicando no botão *Test Now*. Os resultados são apresentados no gráfico disponível para o efeito.

Também pode ser usada a barra de ferramentas da interface gráfica para carregar a inicialização do treino de um modelo FIS, guardar o treino do modelo FIS, abrir um novo sistema *Sugeno* ou abrir qualquer outra interface gráfica de modo a poder interpretar o treino do modelo FIS atual.

### A.1.3. CLUSTERING

A *toolbox fuzzy logic* também fornece uma interface gráfica para o efetuar o *clustering* (Figura 87). O *clustering* de dados numéricos é a base de várias classificações e de algoritmos para modelação de sistemas. Tem por objetivo identificar grupos naturais de dados em grandes conjuntos de dados para reproduzir uma representação concisa do comportamento do sistema.

Esta *toolbox* permite encontrar *clusters* no treino de dados de entrada/saída. As informações provenientes do *cluster* podem ser usadas para gerar sistemas de inferência difusa do tipo *Sugeno* que melhor representam o comportamento dos dados usando um menor número de regras.

A interface gráfica disponibilizada para efetuar o *clustering* é apresentada na Figura 87 e pode ser chamada através da seguinte instrução a ser executada na linha de comandos do MATLAB:

```
>> findcluster
```

Esta interface gráfica implementa as funções *fcm* e *subclust*. São funções implementadas nesta *toolbox* que permitem fazer o *clustering* dos dados. Através da interface, estas duas

funções são executadas de forma transparente ao utilizador. As tarefas que podem ser realizadas na interface gráfica são:

- Carregar e apresentar os dados no gráfico;
- Iniciar o *clustering*;
- Guardar o centro do *cluster*.

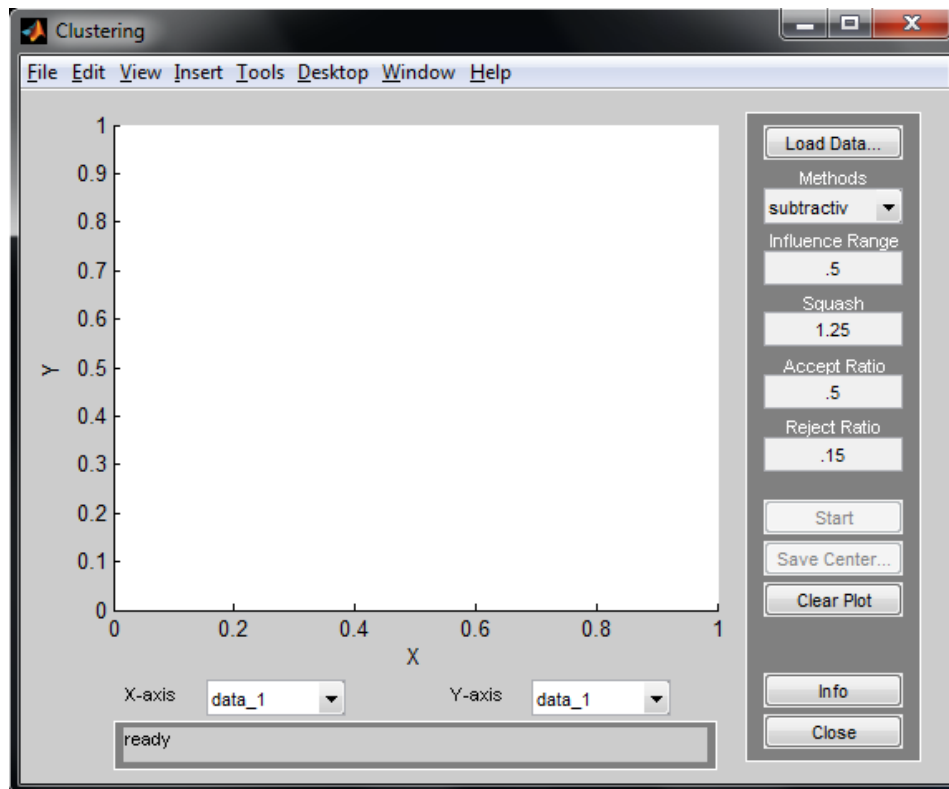


Figura 87 Interface gráfica para efectuar o *Clustering*

## A.2. FUNÇÕES PARA A LINHA DE COMANDOS

Esta categoria é composta por funções que podem ser chamadas através da linha de comandos do MATLAB ou através de aplicações desenvolvidas pelo utilizador. Algumas dessas funções são *m-files* que implementam algoritmos especializados na lógica difusa. O código de cada função pode ser visto usando a seguinte instrução:

```
>> type function_name
```

Onde *function\_name* é o nome da função da qual se quer visualizar o código. O funcionamento de qualquer função da *toolbox* pode ser alterado copiando o código para um novo *m-file*, renomeando o ficheiro e modificando a cópia criada. A *toolbox* também pode ser expandida adicionando mais *m-files*.

Para uma maior compreensão destes comandos, adotou-se a divisão apresentada no manual da *toolbox*:

- Visualização de gráficos;
- Funções de pertinência;
- Estrutura de dados FIS;
- Técnicas avançadas de inferência difusa.

A seguir são descritas sucintamente as funções de cada um dos tópicos apresentados.

### **A.2.1. VISUALIZAÇÃO DE GRÁFICOS**

São disponibilizados dois comandos para traçar o sistema de inferência difusa e para traçar todas as funções de pertinência para uma dada variável. Esses dois comandos são:

- *plotfis* – esta função apresenta um diagrama de alto nível para um FIS. As entradas e as suas funções de pertinência são apresentadas à esquerda do diagrama. Por sua vez, as saídas e as suas respectivas funções de pertinência são apresentadas à direita do diagrama.
- *plotmf* – esta função permite traçar todas as funções de pertinência de uma dada variável de entrada ou saída.

### **A.2.2. FUNÇÕES DE PERTENÇA**

Existem também funções que permitem a criação das funções de pertinência. Ao total são 11 funções que são apresentadas a seguir:

- *dsigmf* – diferença entre duas funções de pertinência sigmoidais;
- *gauss2mf* – função de pertinência de uma combinação gaussiana;
- *gaussmf* – função de pertinência de uma curva gaussiana;
- *gbellmf* – função de pertinência generalizada em forma de sino;
- *pimf* – função de pertinência em forma de  $\Pi$ ;
- *psigmf* – produto de duas funções de pertinência sigmoidais;
- *sigmf* – função de pertinência sigmoidal;
- *smf* – função de pertinência em forma de S;
- *trapmf* – função de pertinência trapezoidal;
- *trimf* – função de pertinência triangular;
- *zmf* – função de pertinência em forma de Z.

### A.2.3. ESTRUTURA DE DADOS FIS

Esta *toolbox* disponibiliza um conjunto de comandos essenciais para o manuseamento da estrutura de dados FIS. A seguir é apresentada a lista de comandos disponibilizada com uma explicação sucinta do seu funcionamento:

- *addmf* – permite adicionar uma função de pertinência ao FIS;
- *addrule* – permite adicionar uma regra ao FIS;
- *addvar* – permite adicionar uma variável linguística ao FIS;
- *defuzz* – permite efetuar a desfuzificação de uma função de pertinência;
- *evalfis* – realiza os cálculos para obter a saída do sistema;
- *evalmf* – avaliação das funções de pertinência do FIS;
- *gensurf* – permite gerar o gráfico tridimensional de saída do FIS;
- *getfis* – permite obter as propriedades do FIS;
- *mf2mf* – transporta os parâmetros de uma função de pertinência para outra;
- *newfis* – permite criar um novo FIS;
- *parsrule* – analisa as regras do FIS;
- *readfis* – carrega o FIS a partir de um ficheiro;
- *rmmf* – remove uma função de pertinência do FIS;
- *rmvar* – remove uma variável linguística do FIS;
- *setfis* – permite alterar as propriedades do FIS;
- *showfis* – apresenta uma listagem dos dados do FIS;
- *showrule* – apresenta as regras do FIS;
- *writefis* – permite guardar o FIS criado.

É útil referir que a interface gráfica utiliza cada uma destas funções. Isto torna a tarefa de criação de um sistema difuso mais fácil e intuitiva para o utilizador visto que este não necessita de conhecimentos ao nível da linha de comandos para criar um sistema. Pode unicamente utilizar as interfaces gráficas disponíveis para o efeito.

### A.2.4. TÉCNICAS AVANÇADAS PARA A INFERÊNCIA DIFUSA

Por fim, esta *toolbox* disponibiliza ainda 6 comandos que representam técnicas avançadas para a inferência difusa. Estes comandos realizam funções como algoritmos de treino e *clustering* e são apresentados a seguir:

- *anfis* – algoritmo de treino para sistemas FIS do tipo *Sugeno*;

- *fcm* – algoritmo que permite a realização do *clustering*;
- *genfis1* – gera a estrutura de um FIS a partir dos dados usando o método *grid partition*;
- *genfis2* – gera a estrutura de um FIS a partir dos dados usando o método *subtractive clustering*;
- *genfis3* – gera a estrutura de um FIS a partir dos dados usando o método *FCM clustering*;
- *subclust* – encontra o centro dos *clusters* usando o método *subtractive clustering*.

### A.3. BLOCOS PARA O SIMULINK

Além de todas as funções e potencialidades anteriormente atrás abordadas, a *toolbox fuzzy logic* do MATLAB permite a utilização de sistemas difusos no *Simulink*. Isto é possível graças a biblioteca *fuzzy logic* no *Simulink*. Para poder usar os blocos disponibilizados por esta biblioteca deve abrir-se o *Simulink* e arrastar os blocos desejados para o modelo ou então pode digitar-se o comando seguinte:

```
>> fuzblock
```

O resultado deste comando é uma janela (Figura 88) que contém dois blocos que podem ser usados para implementar um controlador difuso e uma sub-biblioteca que contém blocos para as funções de pertença.

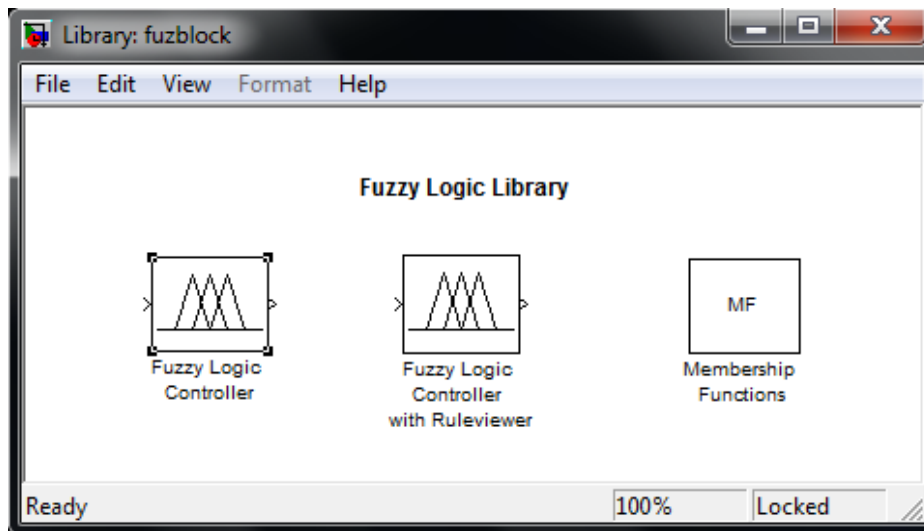


Figura 88 *Fuzzy Logic Library*

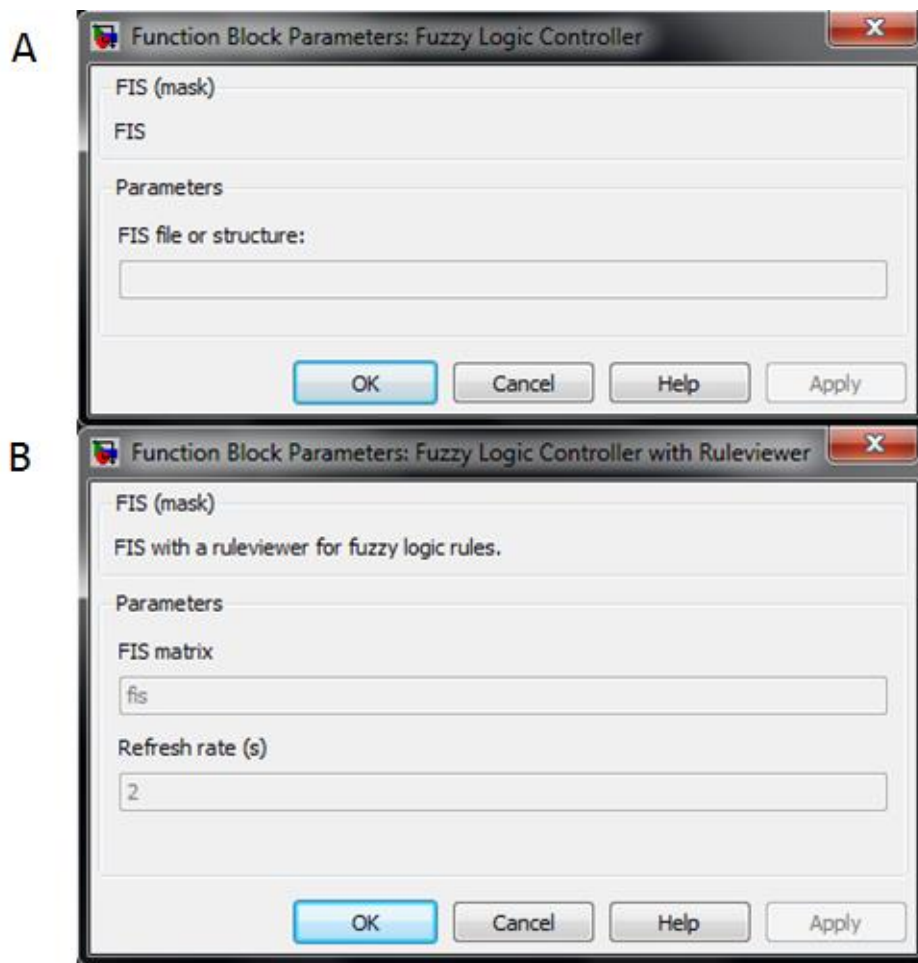
Os blocos que implementam um controlador difuso são:

- *Fuzzy Logic Controller*;

- *Fuzzy Logic Controller with Ruleviewer.*

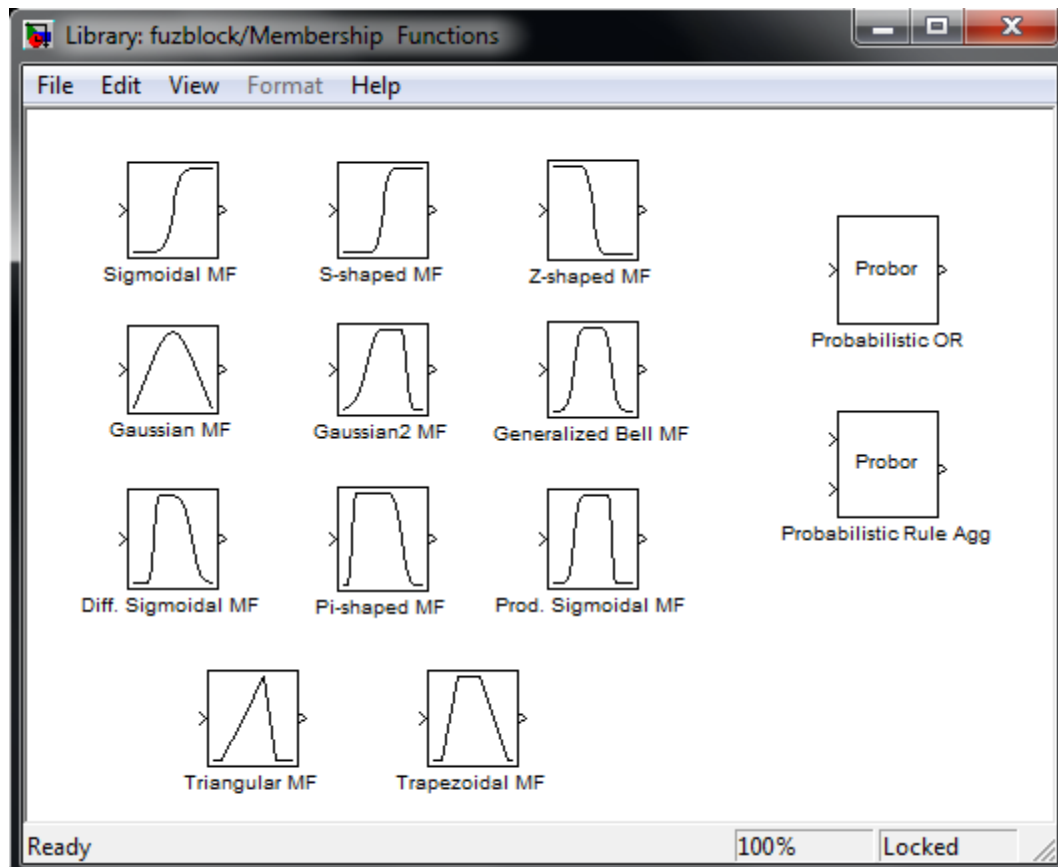
A diferença entre eles reside na capacidade do segundo (*with Ruleviewer*) apresentar automaticamente uma janela com o *Rule Viewer* (Figura 84) aquando da simulação do modelo no *Simulink*.

Através da Figura 89 é possível visualizar os parâmetros que são precisos alterar. Na Figura 89A é possível ver a caixa de diálogo para o *Fuzzy Logic Controller* e onde apenas é necessário especificar o sistema difuso desenvolvido como ficheiro ou estrutura do *workspace*. Por sua vez, o *Fuzzy Logic Controller with Ruleviewer* (Figura 89B), também apresenta uma caixa de texto onde é necessário especificar qual o sistema difuso a usar (ficheiro ou estrutura do *workspace*). Além desta caixa de texto, apresenta outra onde é necessário colocar a taxa de atualização para que, durante a simulação, o *Rule Viewer* apresente os valores tomados pelas variáveis de acordo com a taxa definida.



**Figura 89** Parâmetros do *Fuzzy Logic Controller* (A) e do *Fuzzy Logic Controller with Ruleviewer* (B)

Por sua vez, a sub-biblioteca *Membership Functions* (Figura 90), inclui as funções de pertinência usadas para caracterização das variáveis linguísticas e dois blocos de operadores lógicos. Através da Figura 90 podem ver-se várias funções de pertinência como a sigmoideal, a triangular ou a trapezoidal e os dois blocos de operadores lógicos (*Probabilistic OR* e *Probabilistic Rule Agg*).



**Figura 90** Sub-biblioteca para as funções de pertinência

Torna-se útil realçar que, quando se usa um *Fuzzy Logic Controller*, não é necessário saber utilizar os blocos presentes nesta sub-biblioteca. Estes são implementados automaticamente aquando do início da simulação. Pode, eventualmente, verificar-se como se encontram estruturados os diagramas onde são utilizados estes blocos através da ligação *Look Under Mask*. Na Figura 91 pode ver-se um diagrama de blocos gerado automaticamente pelo *Simulink*. Este diagrama é relativo a um sistema que possui uma variável de entrada denominada de *erro*. Esta variável tem três funções de pertinência (duas trapezoidais e uma triangular) definidas através do editor FIS. Como se pode ver, este processo é transparente ao utilizador, não sendo necessário possuir conhecimentos aprofundados sobre a sub-biblioteca *Membership Functions*.

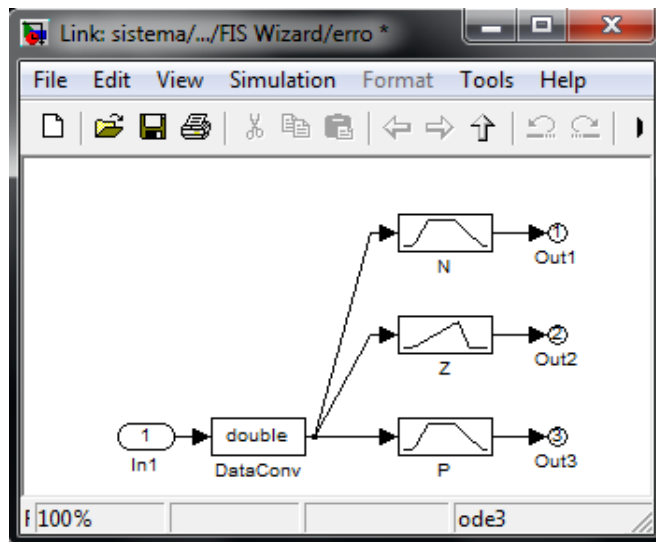


Figura 91 Exemplo prático da implementação de blocos de funções de pertinência



## Anexo B. Informações úteis sobre as funções e a interface gráfica desenvolvida

Este anexo pretende fornecer algumas informações úteis sobre as funções e a interface gráfica desenvolvidas. Dada a significativa extensão do código desenvolvido para o sistema, as funções não serão então incluídas em anexo sendo disponibilizadas em formato digital e armazenados num *Compact Disc - Recordable* (CD-R). Na Tabela 12 é possível visualizar um quadro resumo com todas as funções desenvolvidas e presentes no CD-R. De salientar que todas as funções possuem um *help* em caso de dúvida na sua utilização.

**Tabela 12** Quadro resumo de todas as funções desenvolvidas

Função	Característica	Descrição
PC41_obtencao.m	Funcionalidade	Permite obter em formato “.mat” os dados, as informações e anotações de um determinado sinal .
	Sintaxe	[dados_ecg,inf_ecg,anot_ecg] = PC41_obtencao(sinal_mitbih)
	Exemplo	>>[dados_ecg,inf_ecg,anot_ecg] = PC41_obtencao('mitdb/100')
PC42_representacao.m	Funcionalidade	Permite representar o sinal obtido com <i>PC41_obtencao</i> . Apresenta dois gráficos: um deles com o sinal todo e o outro com um excerto do sinal com uma janela definida pelo utilizador.
	Sintaxe	PC42_representacao(dados_ecg,inf_ecg,derivacao_desejada,seg_min,seg_max)
	Exemplo	>> PC42_representacao(dados_ecg,inf_ecg,1,0,2)
PC43_rep_anot.m	Funcionalidade	Permite representar um sinal obtido através de <i>PC41_obtencao</i> e as suas respetivas anotações num intervalo

		definido pelo utilizador.
	Sintaxe	<code>PC43_rep_annot(dados_ecg, inf_ecg, anot_ecg, derivacao_desejada, seg_min, seg_max)</code>
	Exemplo	<code>&gt;&gt; PC43_rep_annot(dados_ecg, inf_ecg, anot_ecg, 1, 0, 2)</code>
	Funcionalidade	Função que permite fazer o <i>download</i> de um dado ECG e guardar no diretório atual os vários ficheiros correspondentes em formato “.mat”.
<b>PC44_dat2mat.m</b>	Sintaxe	<code>PC44_dat2mat(sinal_mitbih)</code>
	Exemplo	<code>&gt;&gt; PC44_dat2mat('mitdb/100')</code>
	Funcionalidade	Função que permite remover o ruído de um sinal obtido a partir da função <i>PC41_obtencao</i> . Esta função implementa três filtros: filtro de <i>Notch</i> , filtro passa-alto e filtro passa-baixo.
<b>PC45_rem_ruido.m</b>	Sintaxe	<code>[sinal_notch_baixo_alto]=PC45_rem_ruido(dados_ecg, inf_ecg, derivacao_desejada, seg_min, seg_max, ordema, ordemb, wca, wcb)</code>
	Exemplo	<code>&gt;&gt;[sinal_notch_baixo_alto] = PC45_rem_ruido(dados_ecg, inf_ecg, 1, 3, 4, 2, 2, 35, 0.05)</code>
	Funcionalidade	Função que permite obter as variáveis de entrada do sistema difuso. Nesta função tanto pode ser usado o sinal original proveniente da base de dados como o sinal filtrado através da função <i>PC45_rem_ruido</i> . Para usar o sinal original basta apenas não colocar os dois últimos parâmetros da função ( <i>filtra</i> e <i>sinal_notch_baixo_alto</i> ).
<b>PC46_ident_pontos.m</b>	Sintaxe	<code>[P1_QRSdur, P2_QTPint, P3_RRratio, P4_RSTarea, BPM]=PC46_ident_pontos(dados_ecg, inf_ecg, derivacao_desejada, seg_min, seg_max, filtra, sinal_notch_baixo_alto)</code>
	Exemplo	<code>&gt;&gt;[P1_QRSdur, P2_QTPint, P3_RRratio, P4_RSTarea, BPM]=PC46_ident_pontos(dados_ecg, inf_ecg, 1, 3, 4, 0, sinal_notch_baixo_alto)</code>  <code>&gt;&gt;[P1_QRSdur, P2_QTPint, P3_RRratio, P4_RSTarea, BPM]=PC46_ident_pontos(dados_ecg, inf_ecg, 1, 3, 4)</code>
<b>PC47_tipo_batimento.m</b>	Funcionalidade	Esta função permite calcular o tipo de batimento que se está a analisar. A função recebe os parâmetros

		resultantes da execução da função <i>PC46_ident_pontos</i> e o seu <i>output</i> é o tipo de batimento.
	Sintaxe	<code>PC47_tipo_batimento(P1_QRSdur,P2_QTPint,P3_RRratio, P4_RSTarea)</code>
	Exemplo	<code>&gt;&gt; PC47_tipo_batimento(P1_QRSdur,P2_QTPint,P3_RRratio, P4_RSTarea)</code>

**PC48\_interface.fig**      Funcionalidade      Figura que contém o *layout* da interface gráfica desenvolvida.

	Funcionalidade	Função desenvolvida que contém todo o código necessário para o correto funcionamento da interface gráfica.
<b>PC48_interface.m</b>	Sintaxe	<code>&gt;&gt; PC48_interface</code>