



**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de  
Computadores**



**Sumarização de vídeos de jogos de futebol baseada em  
características audiovisuais e biométricas**

**DAVID ANDRÉ DA SILVA LOURENÇO**

(Licenciado em Engenharia Informática e Multimédia)

Dissertação para obtenção do Grau de Mestre  
em Engenharia de Redes de Comunicação e Multimédia

Orientadores : Prof. Rui Manuel Feliciano de Jesus  
Prof. André Ribeiro Lourenço

Abril, 2019

Júri:

Presidente: Prof. Carlos Jorge de Sousa Gonçalves

Vogal (arguente): Prof. José Manuel Peixoto do Nascimento

Vogal (orientador): Prof. André Ribeiro Lourenço



**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de  
Computadores**



**Sumarização de vídeos de jogos de futebol baseada em  
características audiovisuais e biométricas**

**DAVID ANDRÉ DA SILVA LOURENÇO**

(Licenciado em Engenharia Informática e Multimédia)

Dissertação para obtenção do Grau de Mestre  
em Engenharia de Redes de Comunicação e Multimédia

Orientadores : Prof. Rui Manuel Feliciano de Jesus  
Prof. André Ribeiro Lourenço

Abril, 2019

Júri:

Presidente: Prof. Carlos Jorge de Sousa Gonçalves

Vogal (arguente): Prof. José Manuel Peixoto do Nascimento

Vogal (orientador): Prof. André Ribeiro Lourenço

*Ao meu falecido pai, que tanto me queria ver terminar esta fase, espero que fique orgulhoso onde quer que esteja.*



# Agradecimentos

Agradeço aos meus orientadores Rui Jesus e André Lourenço a paciência e o companheirismo que demonstraram ao longo desta caminhada, que tudo fizeram para levar este trabalho a bom porto. Deixo um agradecimento também ao Tiago Mateus da *CardioID* pelo auxílio que prestou em questões relacionadas com monitorização cardíaca. Agradeço também ao Instituto Superior de Engenharia de Lisboa e a todos os docentes da Licenciatura em Engenharia Informática e Multimédia e do Mestrado em Engenharia de Redes de Comunicação e Multimédia por me terem providenciado as ferramentas necessárias à minha aprendizagem e ao meu enquadramento no mercado de trabalho. Gostaria também de agradecer à Polícia de Segurança Pública por, durante parte do curso, me ter auxiliado atribuindo-me uma bolsa de estudo. Por último mas não menos importante gostaria de agradecer à minha família e os meus amigos pela paciência demonstrada ao longo de todo este tempo e à minha namorada por ser o meu pilar e me fazer acreditar que era possível.



# Resumo

Com o passar dos anos tem-se vindo a tornar um desafio cada vez maior gerir a quantidade de dados gerados pelas várias indústrias tecnológicas, sendo uma das maiores produtoras de conteúdo a área de entretenimento desportivo. Tendo como exemplo o futebol, podemos verificar que é completamente impossível para o consumidor destes conteúdos visualizar todos os jogos, dada a quantidade de equipas existentes e a duração desses jogos.

Existe a necessidade de sumarizar os jogos de futebol num conjunto de segmentos de vídeo que consigam, de forma resumida e estruturada, apresentar ao utilizador um resumo conciso do que de relevante ocorreu no jogo, ao fazer com que este não perca nenhum momento chave mas ao mesmo tempo garantindo que não existe uma sobre-exposição relativa a eventos que não apresentam interesse justificativo.

Parte dos trabalhos existentes sobre a automatização da sumarização dos vídeos assentam apenas sobre o processamento de características audiovisuais que permitam a extração de semânticas relevantes e a sua consequente classificação. É proposto um sistema que adiciona novas características de classificação como a variabilidade do ritmo cardíaco e emoções anotadas manualmente pelo utilizador.

Um dos problemas abordados pelo sistema proposto foi a correta e fiável captação da informação relativa à variabilidade do ritmo cardíaco e das emoções sentidas pelo utilizador a determinada altura do jogo. Para resolver este problema foi implementada como parte do sistema uma aplicação para ser executada exclusivamente em dispositivos móveis *Android* que permite ao utilizador gravar as suas emoções e informação de batimento cardíaco para depois poderem ser processados e utilizados na classificação dos planos.

O sistema proposto inclui o processamento de características visuais para efetuar a segmentação do vídeo em planos, como o cálculo dos histogramas das várias imagens por forma a detetar transições instantâneas e graduais. Estas características são também utilizadas para inferir acontecimentos de relevância como a presença da grande área em determinado plano e a amostragem de cartões amarelos e vermelhos.

Toda esta informação irá ser utilizada em conjunto para melhorar a fiabilidade e robustez dos algoritmos existentes de segmentação e classificação de planos em vídeos relativos a jogos de futebol.

**Palavras-chave:** Variabilidade do ritmo cardíaco, Processamento de vídeo, Características audiovisuais, Segmentação, Classificação

# Abstract

Over the years it has become increasingly challenging to manage the sheer amount of data generated by the many technological industries, one of the largest content producers being the sports entertainment area. Taking football as an example, we can see that it is completely impossible for the average consumer of these contents to visualize all the matches in which it has interest, given the number of existent teams and the long duration of a match.

Given this scenario, there is the need to summarize the football matches in a set of video segments that can, in a short and structured way, tell the consumer of this type of content the "story" of the game to make sure it does not miss any key moments of the match but at the same time ensure that there is no overexposure relating to events that do not present justifying interest.

Much of the existing work on automating video summarization rely only on audiovisual features processing which allow the extraction of relevant semantics and their consequent classification. We propose a system that builds on these themes already studied adding new features such as heart rate variability and emotions manually entered by the user.

One of the problems addressed by the proposed system is the correct and reliable captation of the information on the heart rate variability and the emotions felt by the user at a certain point in the match. To solve this problem a mobile Android application was developed as part of the proposed system that allows the user to record his emotions and heart rate information to be processed and used on the shot classification.

The proposed system includes the processing of visual characteristics to perform segmentation of the video into shots, such as calculating the histograms of the

various images to detect instantaneous and gradual transitions. These characteristics are also used to infer relevant events such as the presence of the penalty box and yellow and red cards on a shot.

All of this information will then be combined and used to improve reliability and robustness of the existing algorithms regarding segmentation and classification of shots on football match videos.

**Keywords:** Heart Rate Variability, Video Processing, Audiovisual Characteristics, Shot Segmentation, Shot Classification

# Índice

<b>Índice</b>	<b>xiii</b>
<b>Lista de Figuras</b>	<b>xvii</b>
<b>Lista de Tabelas</b>	<b>xxi</b>
<b>Listagens</b>	<b>xxv</b>
<b>Lista de Acrónimos</b>	<b>xxvii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Variabilidade do Ritmo Cardíaco . . . . .	2
1.1.1 Enquadramento Teórico . . . . .	2
1.1.2 Ritmo cardíaco e HRV . . . . .	3
1.1.3 Medição do HRV . . . . .	4
1.1.4 HRV em janelas temporais ultra curtas . . . . .	4
1.1.5 Relação de características fisiológicas com as emoções . . . . .	5
1.2 Motivação . . . . .	7
1.3 Objetivos . . . . .	8
1.4 Metodologia . . . . .	8
1.5 Organização do Documento . . . . .	9

<b>2</b>	<b>Estado da Arte</b>	<b>11</b>
2.1	Sistemas baseados em Características Audiovisuais . . . . .	11
2.2	Sistemas baseados em Características Biométricas . . . . .	29
<b>3</b>	<b>Modelo Conceptual do Sistema</b>	<b>35</b>
3.1	Aplicação de captura informação fisiológica e anotação de emoções	36
3.1.1	Especificação de Requisitos . . . . .	36
3.1.2	Casos de utilização . . . . .	38
3.1.3	Histórias de Utilização . . . . .	42
3.1.4	Máquina de Estados da Gravação . . . . .	47
3.1.5	Arquitetura da Solução . . . . .	48
3.2	Módulo de geração de características HRV . . . . .	50
3.2.1	Casos de Utilização . . . . .	50
3.2.2	Histórias de Utilização . . . . .	51
3.3	Módulo de análise de vídeos . . . . .	52
3.3.1	Casos de Utilização . . . . .	52
3.3.2	Histórias de Utilização . . . . .	54
3.3.3	Segmentação . . . . .	55
3.3.4	Classificação de Planos . . . . .	56
3.4	Firestore . . . . .	62
3.4.1	Base de dados em tempo real . . . . .	62
3.4.2	Autenticação . . . . .	63
3.4.3	Armazenamento <i>Firestore</i> . . . . .	64
<b>4</b>	<b>Implementação do Sistema</b>	<b>65</b>
4.1	Aplicação de captura informação fisiológica e anotação de emoções	65
4.1.1	Diagrama de Classes . . . . .	65
4.1.2	Serviço de Gravação de Dados Fisiológicos . . . . .	69
4.1.3	Graphical User Interface . . . . .	70
4.2	Módulo de geração de características HRV . . . . .	73

4.2.1	Diagrama de Classes . . . . .	73
4.3	Módulo de análise de vídeos . . . . .	74
4.3.1	Diagrama de Classes . . . . .	74
4.3.2	Segmentação . . . . .	75
4.3.3	Classificação do Tipo de Plano . . . . .	78
4.3.4	Deteção de Grande Área . . . . .	80
4.3.5	Deteção de Cartões . . . . .	82
4.3.6	Support-Vector Machine . . . . .	84
4.3.7	Graphical User Interface . . . . .	87
<b>5</b>	<b>Recolha de Dados e Avaliação do Sistema</b>	<b>89</b>
5.1	Recolha de Dados . . . . .	89
5.1.1	Participantes e <i>Setup</i> . . . . .	89
5.1.2	Processo de Gravação dos Dados . . . . .	90
5.2	Avaliação do Sistema . . . . .	90
5.2.1	Avaliação de algoritmos que utilizam HRV . . . . .	93
5.2.2	Métricas de avaliação . . . . .	94
5.2.3	Segmentação . . . . .	94
5.2.4	Deteção de Grande Área . . . . .	96
5.2.5	Deteção de Planos Relevantes . . . . .	100
<b>6</b>	<b>Conclusões</b>	<b>121</b>
6.1	Integração da solução proposta em soluções existentes no mercado	122
6.2	Trabalho Futuro . . . . .	123
	<b>Referências</b>	<b>125</b>



# Lista de Figuras

1.1	Intervalo RR. . . . .	2
1.2	Fita de medição Movesense. . . . .	5
1.3	Teoria Humoral. . . . .	6
2.1	Divisão da imagem numa matriz de 3x3 [41]. . . . .	13
2.2	Projeção da máscara binária dos pixels da imagem com árbitro [15].	14
2.3	<i>Frame</i> resultante do algoritmo de deteção de grande área [15]. . . .	14
2.4	<i>Frame</i> resultante do algoritmo de deteção de postes [30]. . . . .	19
2.5	Árvore Hierárquica de atribuição de etiquetas [17]. . . . .	21
2.6	Classificação em plano de campo ou plano fora de campo [17]. . . .	21
2.7	Classificação dos tipos de plano de campo baseado na máscara de movimento [17]. . . . .	22
2.8	Deteção de cartões [17]. . . . .	23
2.9	Rede ISA [38]. . . . .	24
2.10	Diagrama Organizacional do Sistema [37]. . . . .	25
2.11	Exemplo de transição gradual com logótipo. . . . .	28
2.12	Aplicação de EMD [14]. . . . .	31
2.13	Exemplo de PSD [10]. . . . .	33
3.1	Diagrama de Blocos do Sistema. . . . .	36
3.2	Casos de Utilização Geral. . . . .	40

3.3	Casos de Utilização Gravar Jogo. . . . .	41
3.4	Casos de Utilização Anotar Emoção. . . . .	42
3.5	Máquina de Estados da Gravação. . . . .	48
3.6	Arquitetura da Solução. . . . .	49
3.7	Casos de utilização HRV. . . . .	50
3.8	Casos de utilização Análise Vídeos. . . . .	53
3.9	Casos de utilização Reproduzir Vídeo. . . . .	53
3.10	Exemplo de aplicação de máscara a plano longo. . . . .	57
3.11	Exemplo da aplicação da operação de abertura. . . . .	58
3.12	Fluxo da deteção de grande área aplicada a um plano. . . . .	59
3.13	Fluxo da deteção de cartões aplicada a um plano. . . . .	60
3.14	Exemplo Support Vector Machine (SVM). . . . .	61
3.15	Funcionalidades do <i>Firebase</i> . . . . .	62
3.16	Árvore de Dados do <i>Firebase</i> . . . . .	63
4.1	Diagrama de Classes Controladores. . . . .	67
4.2	Diagrama de Classes Acesso a Dados. . . . .	68
4.3	Diagrama de Classes Modelo. . . . .	69
4.4	Interface da aplicação. . . . .	72
4.5	Diagrama de Classes HRV. . . . .	73
4.6	Diagrama de Classes Análise de Vídeos. . . . .	75
4.7	Funcionamento da deteção de grande área. . . . .	81
4.8	Funcionamento da deteção de cartão. . . . .	84
4.9	GUI da interface. . . . .	87
4.10	GUI da interface de Teste. . . . .	88
5.1	Representação gráfica dos algoritmos na Interface de Teste. . . . .	92
5.2	Grande área não totalmente enquadrada na <i>frame</i> . . . . .	97
5.3	Grande área filmada por câmara de repetição. . . . .	97
5.4	Grande área enquadrada corretamente com câmara normal de jogo. . . . .	98

5.5	Exemplo de erro na detecção de cartão amarelo vídeo 1. . . . .	104
5.6	Exemplo de erro na detecção de cartão vermelho vídeo 2. . . . .	104
5.7	Exemplo de erro na detecção de cartão amarelo vídeo 2. . . . .	105
5.8	Exemplo de erro na detecção de cartão vermelho vídeo 4. . . . .	105
5.9	Exemplo de erro na detecção de cartão amarelo vídeo 5. . . . .	106
5.10	Exemplo de erro na detecção de cartão amarelo vídeo 6. . . . .	106



## Lista de Tabelas

3.1	Requisitos Funcionais . . . . .	38
5.1	Resultados da Segmentação. . . . .	95
5.2	Precisão e Cobertura da segmentação. . . . .	95
5.3	Resultados da Detecção de Grande Área. . . . .	98
5.4	Precisão e Cobertura da detecção de grande área. . . . .	99
5.5	Detecção de cartões - Matriz de confusão Vídeo 1. . . . .	101
5.6	Detecção de cartões - Matriz de confusão Vídeo 2. . . . .	101
5.7	Detecção de cartões - Matriz de confusão Vídeo 3. . . . .	102
5.8	Detecção de cartões - Matriz de confusão Vídeo 4. . . . .	102
5.9	Detecção de cartões - Matriz de confusão Vídeo 5. . . . .	102
5.10	Detecção de cartões - Matriz de confusão Vídeo 6. . . . .	102
5.11	Detecção de cartões - Matriz de confusão Vídeo 7. . . . .	102
5.12	Detecção de cartões - Matriz de confusão Vídeo 8. . . . .	103
5.13	Detecção de cartões - Matriz de confusão Vídeo 9. . . . .	103
5.14	Detecção de cartões - Matriz de confusão Total. . . . .	103
5.15	Detecção de cartões com Áudio - Matriz de confusão Vídeo 1. . . .	107
5.16	Detecção de cartões com Áudio - Matriz de confusão Vídeo 2. . . .	107
5.17	Detecção de cartões com Áudio - Matriz de confusão Vídeo 3. . . .	107
5.18	Detecção de cartões com Áudio - Matriz de confusão Vídeo 4. . . .	107

5.19	Deteção de cartões com Áudio - Matriz de confusão Vídeo 5. . . .	108
5.20	Deteção de cartões com Áudio - Matriz de confusão Vídeo 6. . . .	108
5.21	Deteção de cartões com Áudio - Matriz de confusão Vídeo 7. . . .	108
5.22	Deteção de cartões com Áudio - Matriz de confusão Vídeo 8. . . .	108
5.23	Deteção de cartões com Áudio - Matriz de confusão Vídeo 9. . . .	108
5.24	Deteção de cartões com Áudio - Matriz de confusão Total. . . . .	109
5.25	Resultados da Deteção de golos utilizando Áudio. . . . .	110
5.26	Precisão e Cobertura da deteção de golos utilizando áudio. . . . .	110
5.27	Resultados da Deteção de golos utilizando Áudio. . . . .	111
5.28	Precisão e Cobertura da deteção de golos utilizando áudio. . . . .	111
5.29	Resultados da Deteção de golos utilizando Áudio e Grande Área. .	112
5.30	Precisão e Cobertura da deteção de golos utilizando Áudio e Grande Área. . . . .	112
5.31	Resultados da Deteção de golos utilizando Áudio e Heart Rate Va- riability (HRV). . . . .	113
5.32	Precisão e Cobertura da deteção de golos utilizando Áudio e HRV.	113
5.33	Resultados da Deteção de golos utilizando Áudio e HRV com emo- ções. . . . .	114
5.34	Precisão e Cobertura da deteção de golos utilizando Áudio e HRV com emoções. . . . .	114
5.35	Resultados da Deteção de golos baseada em Presença de Grande Área e HRV. . . . .	115
5.36	Precisão e Cobertura da deteção de golos baseada em Presença de Grande Área e HRV. . . . .	115
5.37	Resultados da Deteção de golos baseada em Presença de Grande Área e HRV com emoções. . . . .	116
5.38	Precisão e Cobertura da deteção de golos baseada em Presença de Grande Área e HRV com emoções. . . . .	116
5.39	Resultados da Deteção de golos utilizando Áudio, Grande Área e HRV. . . . .	117

5.40	Precisão e Cobertura da detecção de golos utilizando Áudio, Grande Área e HRV. . . . .	117
5.41	Resultados da Detecção de golos utilizando Áudio, Grande Área e HRV com Emoções. . . . .	118
5.42	Precisão e Cobertura da detecção de golos utilizando Áudio, Grande Área e HRV com Emoções. . . . .	118
5.43	Resumo da Precisão e do Cobertura. . . . .	119



# Listagens

4.1	Início do processo de gravação . . . . .	70
4.2	Segmentação por diferenças de histograma HSV frame a frame . .	76
4.3	Segmentação por diferenças de histograma HSV intervalo frames .	77
4.4	Segmentação Final . . . . .	78
4.5	Classificação do tipo de plano . . . . .	79
4.6	Deteção de grande área . . . . .	80
4.7	Deteção de Cartões . . . . .	82
4.8	Exemplo de dados de treino (valores apenas ilustrativos) . . . . .	84
4.9	Treino SVM . . . . .	84
4.10	Classificação SVM . . . . .	85
4.11	Exemplo de dados de treino com emoções (valores apenas ilustra- tivos) . . . . .	85
4.12	Treino SVM com emoções . . . . .	86
4.13	Classificação SVM com emoções . . . . .	86



# Lista de Acrónimos

**ASM** Active Shape Model.

**AVNN** Average of RR Intervals.

**CART** Classification And Regression Trees.

**CSV** Comma-Separated Values.

**DGPR** Dominant Green Pixel Ratio.

**EEG** Electro-encefalograma.

**EMD** Empirical Mode Decomposition.

**GUI** Graphical User Interface.

**HF** High Frequency.

**HOG** Histogram Of Gradients.

**HRV** Heart Rate Variability.

**HSI** Hue, Saturation, Intensity.

**HSV** Hue, Saturation, Value.

**IMF** Intrinsic Mode Function.

**ISA** Independent Subspace Analysis.

**LF** Low Frequency.

**LF/HF** Low Frequency/High Frequency.

**MVC** Model View Controller.

**OCR** Optical Character Recognition.

**pNN50** Percentage of successive RR intervals that differ by more than 50 ms.

**PSD** Power Spectral Density.

**RGB** Red, Green, Blue.

**RMSSD** Root mean square of successive RR interval differences.

**SDK** Software Development Kit.

**SDNN** Standard deviation of RR Intervals.

**SNA** Sistema Nervoso Autónomo.

**SVM** Support Vector Machine.

**VLF** Very Low Frequency.

**XML** Extensible Markup Language.

# 1

## Introdução

O Futebol é o desporto com maior popularidade a nível mundial [6] o que faz com que os conteúdos relacionados com este desporto sejam cada vez mais requisitados tanto a nível televisivo como na Internet. A natureza estruturada de um jogo de Futebol permite que os seus eventos significativos sejam segmentados e compostos em resumos que permitem ao consumidor ter uma ideia geral do que ocorreu durante o jogo. Esta segmentação é tradicionalmente realizada de forma manual através de processos de pós-produção. A tendência crescente no número de jogos transmitidos faz com que este processo requeira uma enorme quantidade de recursos. A inovação nesta área possui um papel importante na procura de soluções que permitam automatizar todo este processo.

Existem nesta área duas grandes problemáticas que são abordadas na generalidade dos trabalhos: a segmentação de vídeos em planos (que permitem criar elementos discretos que podem depois ser classificados) e a classificação desses planos [20]. A segmentação do vídeo em planos consiste no agrupamento de *frames* temporalmente adjacentes que possuam características semelhantes, esta tarefa é normalmente realizada utilizando características visuais [16]. A classificação utiliza várias ferramentas, normalmente audiovisuais, para determinar se os planos resultantes da segmentação são relevantes ou não. Esta dissertação constrói sobre ideias de segmentação e classificação utilizando características audiovisuais previamente estudadas e adiciona a variabilidade do ritmo cardíaco e as emoções do consumidor de conteúdos como uma característica de auxílio à

classificação.

A utilização de características biométricas na classificação levanta também outros problemas abordados na dissertação, tais como, a correta captação dos intervalos RR para depois serem utilizados no cálculo da variabilidade cardíaca e também a correta anotação das emoções que o utilizador sente. Para lidar com este problema é criada como parte do sistema uma aplicação *Android* que capta estas informações e as envia para uma base de dados *Firebase* para posterior processamento.

## 1.1 Variabilidade do Ritmo Cardíaco

Não sendo foco principal desta dissertação e sem entrar em grandes detalhes relacionados com questões médicas, é explicada a variabilidade do ritmo cardíaco, a sua relação com o ritmo cardíaco e a relação destes sinais fisiológicos com as emoções.

### 1.1.1 Enquadramento Teórico

A variabilidade do ritmo cardíaco (é comum utilizar-se a expressão e sigla inglesas HRV) consiste na variação do intervalo temporal entre batimentos cardíacos consecutivos, também conhecido como intervalo RR (Figura 1.1). O HRV é uma forma não-invasiva de identificar o estado fisiológico de um indivíduo. Um nível de HRV reduzido pode indicar um estado de *stress* e um nível elevado pode indicar relaxamento [7] [26] [23].

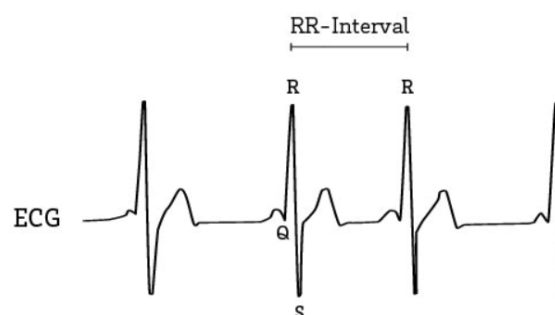


Figura 1.1: Intervalo RR.

HRV é um termo geral constituído por várias características. Estas características são calculadas no domínio do tempo ou no domínio da frequência [34]. As características HRV no domínio do tempo mais comuns são:

- Average of RR Intervals (AVNN) - média dos intervalos RR;
- Standard deviation of RR Intervals (SDNN) - desvio padrão dos intervalos RR (Equação 1.1);

$$SDNN = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (RR_i - AVNN)^2} \quad (1.1)$$

- Root mean square of successive RR interval differences (RMSSD) - raiz quadrada da média do quadrado das diferenças entre intervalos RR consecutivos (Equação 1.2);

$$RMSSD = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (RR_{i+1} - RR_i)^2} \quad (1.2)$$

- Percentage of successive RR intervals that differ by more than 50 ms (pNN50) - percentagem de intervalos RR que diferem em mais de 50ms em relação ao intervalo anterior (Equação 1.3).

$$pNN50 = P(|RR_{i+1} - RR_i| > 50ms) \quad (1.3)$$

As características mais comuns no domínio da frequência são:

- Very Low Frequency (VLF) - potência contida na faixa abaixo de 0,04Hz;
- Low Frequency (LF) - potência contida na faixa de 0,04 a 0,15Hz;
- High Frequency (HF) - potência contida na faixa de 0,15 a 0,4Hz;
- Low Frequency/High Frequency (LF/HF) - Razão entre as potências LF e HF.

### 1.1.2 Ritmo cardíaco e HRV

O batimento cardíaco oferece uma "janela" de visualização para a atividade do Sistema Nervoso Autónomo (SNA) dado que este é controlado pelos sistemas nervosos simpático e parassimpático. O sistema nervoso simpático é responsável por acelerar o batimento cardíaco e pode ser relacionado com o *stress*. O sistema nervoso parassimpático é responsável por desacelerar o batimento cardíaco e pode ser relacionado com o relaxamento [43].

O ritmo cardíaco é medido em batimentos por minuto. Ao contrário do HRV não requer *timestamps* exatos, ou seja, é a média de batimentos cardíacos numa determinada janela temporal. Enquanto o ritmo cardíaco se baseia nos batimentos por minuto, o HRV baseia-se nas variações ao longo do tempo de batimentos cardíacos sucessivos. Como o HRV se baseia nas alterações de tempo entre batimentos cardíacos, é mais complexo de obter e requer mais precisão que o ritmo cardíaco. [46].

O HRV é mais fiável do que o ritmo cardíaco no que diz respeito à inferência de emoções. O HRV normalmente é mais constante em situações em que o indivíduo está mais relaxado e mais variável em situações que envolvem emoções mais intensas [50].

### 1.1.3 Medição do HRV

No passado a medição da variabilidade do ritmo cardíaco era bastante complicada visto que a única forma de efetuar a medição consistia na deslocação do indivíduo a uma clínica e na realização de um eletrocardiograma [7]. Com a evolução dos dispositivos móveis e com o investimento das empresas na produção de aplicações e equipamentos *wearables*, tornou-se bastante mais simples a medição do HRV. Estes novos equipamentos, apesar de não oferecerem uma precisão equiparada aos equipamentos tradicionais, constituem uma alternativa viável [31].

Nesta dissertação, por forma a obter os dados de ritmo cardíaco, é utilizado um monitor cardíaco da *Movesense* (Figura 1.2). Este medidor apenas obtém os intervalos temporais entre batimentos cardíacos para depois poderem ser processados dando origem ao HRV. Mais à frente na dissertação é descrito todo o processo e a metodologia na recolha de dados de HRV.

### 1.1.4 HRV em janelas temporais ultra curtas

Num jogo de futebol a volatilidade no que toca aos acontecimentos é elevada. É necessário efetuar um cálculo de HRV numa janela de poucos segundos, visto que as jogadas num jogo de futebol ocupam janelas temporais muito curtas. Uma janela temporal superior a 24 horas é de longa duração, superior a 5 minutos é de curta duração e inferior a 5 minutos é de ultra-curta duração [34]. Nesta dissertação é utilizada uma janela na ordem dos 15 segundos, o que prevê um enquadramento na categoria da ultra curta duração [40].



Figura 1.2: Fita de medição Movesense.

Salahuddin L. et al. [32] realizou um estudo incidente sobre as várias características que compõem o HRV aplicado em janelas temporais com duração inferior a 5 minutos. O objetivo deste estudo era mostrar que seria possível utilizar o HRV em aplicações móveis aplicado a janelas temporais ultra curtas para monitorizar os níveis de *stress* de um indivíduo. Neste estudo é concluído que efetivamente é possível inferir situações de *stress* sob estas condições e que, das características já discutidas anteriormente, as que oferecem maior precisão em janelas temporais ultra curtas (utilizando sensores móveis, como é o caso do usado nesta dissertação) são: AVNN, RMSSD, pNN50, HF e LF/HF.

### 1.1.5 Relação de características fisiológicas com as emoções

Parte desta dissertação consiste em utilizar as emoções e o HRV num classificador SVM por forma a determinar momentos em que o utilizador sente algum tipo de emoção forte.

Desde a antiguidade que vários estudos tentam relacionar as emoções com aspetos da fisiologia humana. Na China acreditava-se que as emoções residiam no físico e que qualquer excesso emocional poderia resultar em repercussões, por vezes extremas, como o aparecimento de doenças graves e a falência de órgãos vitais. Na antiga Grécia Hipócrates desenvolveu a teoria humoral que assentava na crença de que o corpo humano era constituído por 4 humores: sangue, fleuma, bílis amarela e bílis negra. Qualquer emoção ou comportamento seria causado por desequilíbrios nestes humores. Na figura 1.3 é possível verificar uma representação visual da influência dos vários humores num indivíduo (no sentido horário, a partir da figura superior direita: bílis amarela, bílis negra, sangue, fleuma).

Aristóteles seguia uma teoria semelhante à de Hipócrates em que possuía uma



Figura 1.3: Teoria Humoral.

visão fisiológica das emoções (atribuíam-lhes o nome de "paixões") onde estas podiam ser equiparadas a estados físicos como a fome ou o cansaço [43].

O relacionamento das características fisiológicas com as emoções tem aplicações importantes na medicina, no entretenimento e na implementação de interfaces Homem-máquina. O reconhecimento de padrões fisiológicos pode potencialmente ajudar na quantificação de *stress* e outras emoções. A deteção de emoções através de padrões fisiológicos pode também ser importante na construção de interfaces uma vez que se torna possível inferir emoções de um determinado utilizador utilizando características fisiológicas e ajustar assim os vários componentes por forma a reduzir emoções negativas [36].

Benezeth Y. et al. [39] efetua um estudo da correlação entre estados emocionais e a variabilidade do batimento cardíaco medido de forma remota, através da utilização de uma câmara. Neste artigo são utilizadas as características VLF, LF e HF e dois tipos de estados emocionais: um neutro e um de alta excitação provocada por estímulos audiovisuais. Benezeth Y. et al. conclui que efetivamente é possível estabelecer uma relação entre a variabilidade do ritmo cardíaco e as emoções sentidas, mesmo utilizando formas de medição remotas, que à partida seriam mais ineficazes do que medições que envolvem o contacto físico do utilizador com o sensor. No entanto, os resultados de 4 participantes (num total de 16) neste estudo não são os esperados, podendo dever-se isto ao facto de, na fase de captação de emoções neutras, os utilizadores estarem sob *stress*.

Hristova E. et al. [27] implementou um sistema que também efetua a determinação de emoções através de sinais biológicos, no entanto, ao contrário de Benezeth Y. et al. em [39] aqui não são definidos tipos de emoções, apenas são utilizadas as suas valências na relação com os sinais biológicos. É referido que existem diferenças enormes entre indivíduos, dado que a mesma emoção pode despolar diferentes padrões fisiológicos, tornando-se assim bastante difícil encontrar

padrões que permitam construir um classificador generalizado, ou seja, que funcione com um conjunto de indivíduos sem que os seus parâmetros tenham que ser revistos de indivíduo para indivíduo. Esta conclusão advém do facto de o classificador implementado apenas funcionar para um conjunto muito restrito de utilizadores.

Ragot M. et al. [33] realizou um estudo onde compara sensores de laboratório com sensores do tipo *wearable* na utilização de sinais fisiológicos para reconhecer emoções. Aqui é utilizado um classificador SVM que recebe como *input* as informações fisiológicas e devolve como *output* o estado emocional correspondente. Este trabalho é semelhante ao que é efetuado na presente dissertação, no entanto, nesta dissertação o *input* são as emoções anotadas manualmente e as informações fisiológicas e o *output* é a informação de relevância de determinado segmento.

Brock E. et al. [50] desenvolveu também um sistema que utiliza vários tipos de classificadores e que recebe sinais fisiológicos (neste caso o HRV) e os relaciona com emoções, permitindo assim a construção de interfaces Homem-máquina afetivas. O objetivo final é a construção de sistemas capazes de reconhecer emoções humanas e reagir de forma apropriada, proporcionando assim interações mais naturais.

## 1.2 Motivação

A presente dissertação pretende colmatar uma necessidade crescente de aumentar a fiabilidade dos sistemas responsáveis pela sumarização de vídeos. Pretende também determinar se as emoções anotadas manualmente pelo utilizador permitem melhores resultados na sumarização quando aliadas à informação HRV resultante do processo de gravação.

Parte dos trabalhos na área apenas tentam melhorar a fiabilidade destes sistemas adicionando características audiovisuais ou alterando a forma de implementação das já existentes [9]. Existe uma necessidade de adicionar novos tipos de características que permitam estender a funcionalidade dos sistemas para além daquilo que é possível atingir utilizando características tradicionais.

A motivação desta dissertação tem como base a necessidade de apresentar soluções inovadoras que permitam melhorar os sistemas tradicionais que baseiam o seu funcionamento apenas em características audiovisuais, introduzindo novas características na quantificação da importância dos eventos em vídeos de natureza estruturada, como é o caso dos vídeos desportivos.

## 1.3 Objetivos

Esta dissertação tem como principal objetivo a investigação da viabilidade de utilizar, em conjunto com outras características audiovisuais, a variabilidade do ritmo cardíaco do utilizador na classificação de planos de vídeo. Os planos de um vídeo são obtidos através do processo de segmentação.

Os maiores desafios consistem na implementação um processo uniforme de recolha dos dados de ritmo cardíaco do utilizador, garantindo que não existem interferências que possam levar a conclusões incorretas no que toca à classificação e, dado a natureza subjetiva dos dados e às diferenças que existem entre indivíduos ao nível do ritmo cardíaco, garantir que o sistema funciona de forma generalizada. São utilizados algoritmos de sumarização que utilizam características audiovisuais ao qual é adicionada a característica resultante da variabilidade do ritmo cardíaco do utilizador e as emoções.

Dada esta informação, é também objetivo desta dissertação o desenvolvimento de uma aplicação para o sistema operativo *Android* que permita captar de forma robusta e fidedigna tanto informações de ritmo cardíaco do utilizador como também as emoções anotadas manualmente pelo mesmo.

Um objetivo implícito também desta dissertação é a utilização dos mecanismos propostos anteriormente por forma a minimizar os erros de classificação de planos como relevantes ou não relevantes.

## 1.4 Metodologia

É utilizada no desenvolvimento desta dissertação a tradicional sequência: Análise de Requisitos, Modelação do Sistema, Desenvolvimento do Sistema e Testes ao Sistema.

A análise de requisitos (apresentada no capítulo 3) tem como propósito a produção de um modelo conceptual, completamente abstraído das problemáticas da implementação (tais como a linguagem de programação em que o sistema é implementado ou a sua forma de desenvolvimento). A modelação do sistema tem como objetivo o planeamento da implementação dos requisitos definidos anteriormente. A implementação do sistema consiste na realização do sistema planeado anteriormente ocorrendo em paralelo com a fase de testes, por forma a afinar o funcionamento do sistema.

## 1.5 Organização do Documento

Esta dissertação de Mestrado encontra-se organizada em seis capítulos. Em seguida é apresentado um pequeno resumo dos conteúdos incluídos em cada um:

1. **Introdução** - Introdução ao sistema proposto.
2. **Estado da Arte** - Neste capítulo é feito um estudo das soluções que existem atualmente na área e que contribuíram para o desenvolvimento do sistema proposto nesta dissertação.
3. **Modelo Conceptual do Sistema** - Delineamento da estratégia a abordar para a resolução do problema, definindo os requisitos do sistema.
4. **Implementação do Sistema** - Implementação da abordagem sugerida no capítulo anterior. São apresentados diagramas que permitem ter uma ideia geral do sistema.
5. **Recolha de dados e avaliação do Sistema** - Descrição do processo de recolha de dados e realização de vários testes que permitem inferir a viabilidade da solução.
6. **Conclusões** - Análise enquadrada nos resultados obtidos pelo sistema, sendo propostas possíveis melhorias a realizar futuramente.



# 2

## Estado da Arte

Neste capítulo é feita uma revisão do estado da arte atual, dos algoritmos que existem neste momento e que são estudados de forma extensiva. Os trabalhos relacionados revistos neste capítulo utilizam características audiovisuais, biométricas ou até a combinação destas.

Samuel F. de Sousa Júnior et al. [9], Yasmin S. Khan e Soudamini Pawar [18], Amjad Rehman e Tanzila Saba [21], Huang-Chia Shih [48], Dharmesh Tank [49] e Sachan P. Rajendra [12] realizaram uma recolha e estudo aprofundados de alguns dos vários sistemas implementados até hoje para a sumarização automática de vídeos o que constitui um ponto de partida para o estudo mais aprofundado e individual das conquistas alcançadas até hoje e dos problemas que ainda são necessários enfrentar.

### 2.1 Sistemas baseados em Características Audiovisuais

Ekin, A. e Tekalp. M. [15] implementaram um sistema para a análise e sumarização de vídeos utilizando algoritmos de processamento de vídeo como a deteção de cor dominante, segmentação e classificação de planos. Este sistema implementa também algoritmos de deteção de eventos com relevância semântica como a deteção de golos, do árbitro e da grande-área.

O algoritmo de detecção de cor dominante tem o objetivo de detetar quais os pixels pertencentes ao relvado. O algoritmo não assume a cor dominante do relvado, esta é determinada no arranque e adapta-se com a mudança das condições de luminosidade do estádio. A cor dominante do relvado é descrita como a média de um intervalo de valores (definido empiricamente) vizinhos do pico absoluto do histograma de cada componente de cor no espaço de cor Hue, Saturation, Intensity (HSI). Os pixels de campo em cada *frame* são detetados calculando a distância da cor de cada pixel à cor média.

No algoritmo de segmentação são utilizadas duas características:

- A diferença do rácio de pixels de cor dominante entre *frames* adjacentes (os pixels de cor dominante calculados anteriormente são utilizados para calcular esta característica);
- A diferença da interceção de histogramas entre *frames* adjacentes (a interceção de histogramas é calculada recorrendo à equação 2.1 onde  $N$  representa o número de componentes de cor,  $Bn$  o número de intervalos no histograma,  $H_i^m$  representa o histograma da componente de cor  $m$  da imagem  $i$  e  $H_{i-k}^m$  representa o histograma da componente de cor  $m$  da imagem  $i-k$ ).

$$H(i, k) = \frac{1}{N} \sum_{m=1}^N \sum_{j=0}^{Bn-1} \min(H_i^m[j], H_{i-k}^m[j]) \quad (2.1)$$

Estas características são calculadas para todas as *frames* de um vídeo e depois comparadas com dois limites definidos empiricamente. Caso as duas características de determinada *frame* sejam maiores que os limites, é declarada uma transição de plano nessa *frame*.

Após a segmentação dos planos é executado um algoritmo que insere o plano numa de três categorias: plano longo, plano médio ou plano próximo/fora de campo. O rácio de pixels de cor dominante é utilizado para inferir o tipo do plano. Um rácio de pixels de cor dominante elevado pode indicar plano longo e um rácio de pixels de cor dominante reduzido pode indicar um plano próximo/fora de campo. Este algoritmo parte do princípio básico que quanto mais a cor verde está presente num plano, maior é a quantidade de relva visível, o que permite inferir o seu tipo visto que uma elevada quantidade de relva visível normalmente indica um plano longo, uma quantidade média indica um plano médio e uma quantidade reduzida indica um plano próximo/fora de campo. Os planos do

tipo médio possuem um elevado rácio de pixels da cor dominante o que muitas vezes pode levar a que sejam classificados de forma errada como planos do tipo longo. Como forma de contornar este problema é implementada a técnica de *Golden Region* que divide a imagem numa matriz de 3x3 (Figura 2.1) e é efetuado o mesmo procedimento de avaliação da cor dominante no elemento central da matriz. A aplicação desta técnica tem a vantagem de permitir distinguir com maior facilidade entre planos do tipo médio e longo uma vez que num plano médio habitualmente existe um jogador no centro da *frame* e num plano longo isto não acontece.

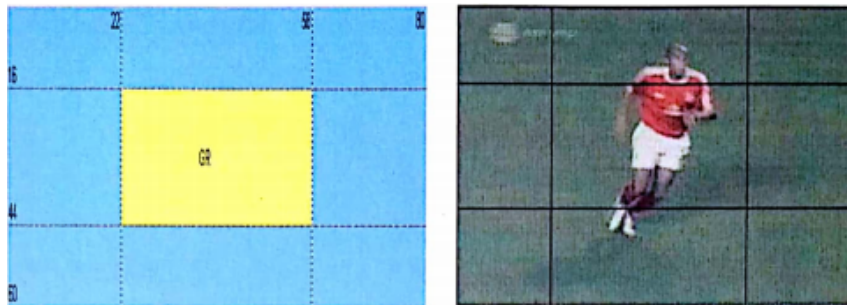


Figura 2.1: Divisão da imagem numa matriz de 3x3 [41].

Para a deteção do golo é definido o padrão de características cinematográficas que normalmente quando ocorrem indicam a existência de um golo (de acordo com as técnicas de realização normalmente empregues nos jogos de futebol). Por exemplo, quando é detetada uma repetição são encontrados os planos longos que antecedem e que precedem esta repetição. Se o período de tempo entre estes dois planos tiver entre 30 e 120 segundos (intervalo definido experimentalmente) e existir pelo menos um plano próximo/fora de campo o segmento é declarado como golo.

Habitualmente quando o árbitro está em destaque num plano é indicativo de um momento interessante como a amostragem de um cartão. A técnica *Golden Region* é utilizada também aqui para melhorar a deteção do árbitro. O árbitro apenas é procurado em planos do tipo médio ou próximo/fora de campo. Quando se verifica um plano deste tipo, a máscara binária dos seus pixels é projetada horizontalmente e verticalmente (Figura 2.2) e depois, dependendo dos valores de pico horizontais e verticais e do seu espalhamento, é definido se existe árbitro na *frame* ou não. A decisão da existência de árbitro baseia-se totalmente na área de projeção dos dados, ou seja, se a área de projeção for uma área reduzida e existir pouco espalhamento dos dados, é declarada a existência do árbitro na *frame*, caso seja uma área dispersa, a *frame* é descartada.

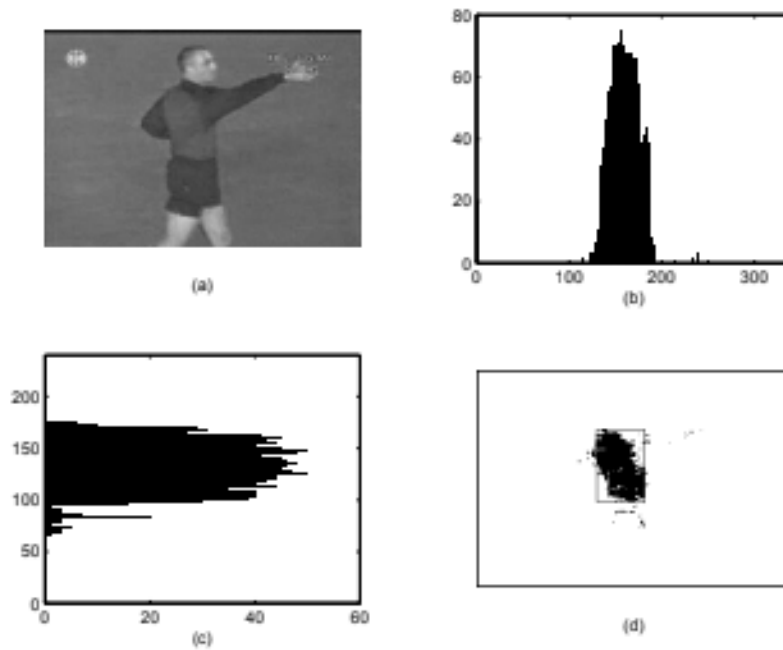


Figura 2.2: Projeção da máscara binária dos pixels da imagem com árbitro [15].

Nesta dissertação é também realizada a detecção da grande área partindo do pressuposto que quando está visível na sua totalidade é indicativo de um momento de interesse. Esta procura é apenas efetuada em planos longos e consiste na detecção de 3 linhas paralelas: a linha de final de campo, uma linha da pequena área e uma linha da grande área. É calculada uma máscara dos pixels da cor do campo (cor dominante) para depois efetuar a binarização da imagem, ficando os pixels do campo a branco e os jogadores e as linhas a negro (Figura 2.3). Para detetar apenas as linhas é aplicada uma máscara de *Laplace* de detecção de contornos [5]. Em último lugar é utilizada uma transformação de *Hough* [28] com restrições de paralelismo, permitindo assim a extração das três linhas paralelas e a conseqüente detecção da grande área.

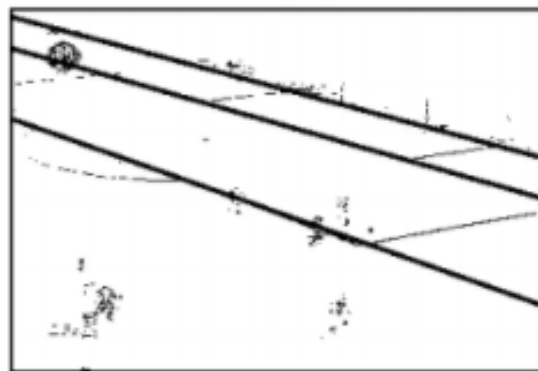


Figura 2.3: *Frame* resultante do algoritmo de detecção de grande área [15].

Ekin A. [22] implementa uma norma descritiva do conteúdo de um vídeo, o MPEG-7. Neste trabalho propôs um modelo que permite a descrição de características de alto e baixo nível de um vídeo. Esta descrição é efetuada com recurso à norma MPEG-7 que providencia um conjunto de esquemas de descrição e descritores que permitem a interoperabilidade entre diversas bases de dados e a extensão da norma.

Para definir os descritores MPEG-7 é necessário utilizar características multimodais de baixo nível como características baseadas em imagem, áudio ou texto. A deteção dos pixels que pertencem ao relvado, a segmentação de planos e a deteção do tipo de plano são realizadas utilizando algoritmos semelhantes aos implementados por Ekin A. e Tekalp M. em [15]. No entanto os histogramas utilizados são do espaço de cor Hue, Saturation, Value (HSV).

A deteção de repetições em câmara lenta é conseguida partindo do princípio que é utilizada a técnica de inserção de imagens repetidas na criação da mesma, se for utilizada uma câmara de alta velocidade para o efeito esta técnica não produz resultados satisfatórios. Em primeiro lugar são calculadas as intensidades dos pixels de duas *frames* consecutivas e em seguida é aplicado o método *Zero Crossing* [1]. Se o número de transições por zero for maior que um limite é definida uma repetição.

As características visuais são utilizadas para detetar paragens no jogo enquanto que as características auditivas e as textuais são utilizadas para detetar eventos como golos.

O processamento do áudio parte do princípio base de que quando existe algum lance de interesse, existe um pico no som produzido. Para medir a intensidade do som é calculada a energia do áudio (Equação 2.2 onde E representa a energia do sinal áudio  $x[n]$ , T é a duração do sinal e N a janela temporal. É utilizada uma janela temporal de 5 segundos).

$$E = T \sum_{n=0}^{N-1} x^2[n] \quad (2.2)$$

Como é necessário limitar a energia a determinada janela temporal é calculada a potência do sinal (Equação 2.3 onde E é a energia do sinal áudio  $x[n]$  e T é a duração do sinal).

$$P = \frac{1}{N} \sum_{n=0}^{N-1} x^2[n] \quad (2.3)$$

Após o cálculo são determinados os máximos locais e avaliados se os valores destes são maiores que um limite calculado dinamicamente com base na média e no desvio padrão. Em caso afirmativo é definido que naquele ponto temporal existe um momento de interesse.

Este sistema introduz a detecção de texto presente num plano como uma novidade em relação aos sistemas revistos anteriormente. O texto é utilizado para detetar momentos relevantes num jogo dado que quanto estes ocorrem normalmente é exibido texto no ecrã, como num golo por exemplo. O processamento de texto envolve a detecção de regiões de texto, que são regiões normalmente caracterizadas por um alto contraste. A estas regiões de texto é aplicada uma binarização e filtragem por forma a realçar os caracteres textuais para serem extraídos utilizando uma técnica chamada Optical Character Recognition (OCR).

Hanjalic A. [42] propõe um método de extração de momentos relevantes num jogo de futebol que não utiliza características semânticas de alto nível, ou seja, pode ser utilizado em vídeos de qualquer tipo de desporto. Para isso é implementado um sistema que efetua a extração de *highlights* completamente baseada na excitação do público.

Para detetar a excitação do público são utilizadas três características de baixo nível: o movimento medido nas transições de *frame*, a densidade de transições abruptas (Equação 2.5 onde  $p(k)$  e  $n(k)$  são as posições em índices de *frames* das transições abruptas mais próximas à direita e à esquerda da *frame* analisada respetivamente. O parâmetro  $\delta$  representa a forma de distribuição dos valores de  $c(k)$ ) e a energia do áudio. A atividade de movimento numa determinada *frame* é calculado utilizando uma estimacão de movimento baseada em blocos entre a *frame* atual e a *frame* seguinte e calculando a magnitude média normalizada de todos os vetores de movimento (Equação 2.4 onde  $B$  é o número de blocos utilizados na estimacão do movimento numa *frame*,  $|v_{max}|$  é o comprimento máximo de um vetor de movimento e  $v_i(k)$  é o valor médio dos vetores de movimento).

$$m(k) = \frac{100}{B \times |v_{max}|} \times \left( \sum_{i=1}^B |v_i(k)| \right) \% \quad (2.4)$$

$$c(k) = 100e^{\frac{1-(n(k)-p(k))}{\delta}} \% \quad (2.5)$$

Às duas características anteriores é aplicada uma janela de *Kaiser* (Equações 2.6 para a estimacão de movimento e 2.7 para a densidade de transições bruscas) para diminuir o ruído que possa eventualmente interferir na classificação.

$$G_1(k) = \frac{\max(m(k))}{\max(\tilde{m}(k))} \tilde{m}(k) \% \quad (2.6)$$

$$G_2(k) = \frac{\max(c(k))}{\max(\tilde{c}(k))} \tilde{c}(k) \% \quad (2.7)$$

Por forma a detetar momentos relevantes utilizando o áudio são calculados os máximos absolutos da curva de energia (Equação 2.8 onde  $e_n(k)$  representa a normalização da curva de energia do áudio e  $\bar{e}$  representa a média da curva da energia do áudio).

$$G_3(k) = 100e_n(k)(1 - \bar{e}_n) \% \quad (2.8)$$

As três curvas obtidas dos processamentos anteriores são correlacionadas e é obtida uma curva no domínio do tempo que permite definir se existem momentos relevantes. Para garantir que a seleção dos momentos não-relevantes não é totalmente rígida (o que levaria a selecionar momentos que não são relevantes ou deixar de fora momentos relevantes) é implementada uma função de ponderação ao invés de ser definido um limite estático que é comparado com a curva final.

Em [41] Fortio R. utiliza técnicas semelhantes de deteção de cor dominante, de deteção do tipo de plano e de segmentação às utilizadas por Ekin, A. e Tekalp, A. em [15] e Ekin A. em [22]. Este projeto não realiza deteção de características semânticas de alto nível como golos ou cartões, apenas efetua sumarização baseada no tipo de planos inferindo situações de interesse aquando da ocorrência de planos próximos ou repetições. Os segmentos de câmara lenta são detetados utilizando um algoritmo semelhante ao implementado por Ekin A. em [22].

Zawbaa H. et al. [30] propõem um algoritmo de sumarização de jogos de futebol utilizando uma SVM. O sistema é dividido em 6 fases:

- Pré-processamento onde o vídeo é segmentado em planos;
- Processamento de planos onde são aplicados dois tipos de classificação aos planos resultantes da fase anterior;
- Deteção de repetições que utiliza o algoritmo SVM;
- Deteção de *score board* que também utiliza o algoritmo SVM;

- Detecção de eventos de excitação e a fase de deteção de eventos baseada em logótipo (um logótipo é, por exemplo, o símbolo da liga que por vezes é mostrado nas transições);
- Sumarização

Na fase de pré-processamento são utilizadas as características de baixo nível para segmentar o vídeo. O processamento desta fase é apenas aplicado a cada 10 *frames* para garantir que transições graduais também são detetadas, processo semelhante ao utilizado por Ekin, A. e Tekalp. A. em [15]. Todas as *frames* são divididas em blocos de 32 por 32 pixels e é calculada a diferença média de todos os blocos pertencentes a uma determinada *frame* e a *frame* 10 posições adiante, calculando em seguida a percentagem de blocos que mudaram entre as *frames*. De seguida é calculada a diferença média do rácio de cor dominante entre uma determinada *frame* e a *frame* 10 posições adiante. Em último lugar é verificado se estes valores são maiores que determinados limites definidos de forma arbitrária e em caso afirmativo é definida uma transição de plano.

Na fase de processamento de planos são classificados os planos resultantes da fase anterior. É efetuada a classificação do tipo de plano (longo, médio, próximo ou fora de campo) utilizando o rácio de pixels de cor dominante e 4 limites pré-definidos. É também efetuada a classificação relativa aos momentos de paragem do jogo, utilizando a localização das *frames* de início e fim dos planos.

Na fase de deteção de repetições são utilizados os logótipos por forma a marcar o acontecimento de uma repetição, pois normalmente esta ocorre entre duas amostragens de logótipos. Para detetar as repetições, em primeiro lugar o classificador SVM é treinado com imagens que possuam logótipos visíveis e imagens onde tal não ocorre. Esta classificação é conseguida aumentando o contraste das *frames* de entrada e calculando a percentagem de pixels brancos nas *frames*. Se alguma possuir uma percentagem acima dos 50%, a *frame* original (sem aumento de contraste) é passada ao classificador SVM que a classifica como repetição ou não. Na fase de deteção de *score board* é utilizado um método semelhante. Esta fase complementa a anterior para confirmar com mais segurança a existência de um golo, visto que quando ocorre um golo o *score board* é normalmente exibido no plano seguinte.

Na fase de deteção de eventos parte-se do princípio que configuram momentos de interesse aqueles em que a grande área e a baliza são visíveis à semelhança do que acontece com Ekin, A. e Tekalp. A. em [15]. Mas ao contrário da abordagem

de Ekin, A. e Tekalp. A. que apenas deteta as três linhas paralelas que compõem os limites longitudinais da área, aqui é efetuada a detecção dos postes (Figura 2.4) e da rede da baliza que é aliada à variação sonora para inferir momentos de interesse. Para efetuar a detecção dos postes é utilizado o algoritmo k-médias [3] para transformar cada *frame* numa imagem binária e em seguida é utilizada a transformação de *Hough*. Para efetuar a detecção da rede da baliza é aplicado um filtro de *Gabor* [] que permite detetar padrões repetidos, como é o caso. O volume de áudio de cada *frame* é calculado de forma semelhante ao que acontece com Ekin A. em [22].

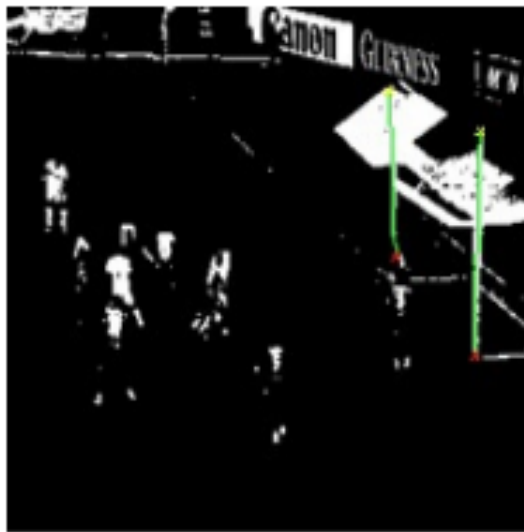


Figura 2.4: *Frame* resultante do algoritmo de detecção de postes [30].

Na última fase são utilizados os dados extraídos das fases anteriores e é aplicado conhecimento de alto nível acerca do jogo para inferir a ocorrência de eventos conceptuais como é o caso de um golo, um ataque ou uma falta.

Kolekar M. e Segupta S. [17] propõem um sistema que utiliza uma rede Bayesiana para detetar eventos relevantes num jogo de futebol utilizando características audiovisuais. Neste sistema são definidas três camadas de abstração no que toca à atribuição de etiquetas:

- Uma camada de características de baixo nível que neste caso é a cor;
- Uma camada onde são definidos eventos. Esta camada utiliza a primeira camada para definir eventos, como por exemplo o agrupamento de jogadores de determinada equipa;

- Uma camada de conceitos de alto nível que utiliza os eventos definidos pela camada anterior para inferir conceitos semânticos como golos de uma determinada equipa ou a amostragem de cartões.

Ao contrário da maioria dos sistemas revistos anteriormente, este não utiliza procedimentos de deteção de limites de plano ou algoritmos de agrupamento como k-médias, a extração dos momentos relevantes numa primeira instância é baseada apenas em áudio.

De referir ainda que dos sistemas estudados, este é o único em que as características semânticas possuem uma granularidade ao nível da equipa, ou seja, permite identificar se um momento relevante se refere à equipa A ou à equipa B, permitindo assim que a classificação não seja só efetuada em relação ao tipo de evento, mas também em relação à equipa responsável por esse evento. Por exemplo, ao invés de um momento relevante ser classificado apenas como "golo", é classificado como "golo da equipa X".

Para atribuir as etiquetas de evento referidas na segunda camada aos segmentos de vídeos utilizando as características de baixo nível gerados pela primeira camada é definido um sistema composto por 5 níveis (Figura 2.5). Antes do primeiro nível do sistema é necessário efetuar pré-processamento dos dados onde são extraídos segmentos de vídeo relevantes utilizando a energia do áudio numa janela temporal semelhante ao que acontece com Ekin A. em [22] e Zawbaa H. et al. em [30]. Os segmentos de vídeo relevantes são extraídos como um conjunto de *frames* consecutivas onde a energia de áudio média normalizada é maior que um determinado limite pré-definido. Após a extração, os segmentos de vídeo onde é detetada excitação de público são passados ao primeiro nível onde é calculada a diferença de histogramas no espaço de cor HSV entre as *frames* do segmento de vídeo e uma *frame template* de um logótipo, sendo depois detetados os *frames* de transição de logótipo comparando os histogramas com limite pré-definido. Em seguida são selecionados os segmentos de vídeo entre duas transições de logótipo e é calculada a frequência de *frames* para definir se o segmento de vídeo se trata de uma repetição ou não. Em caso afirmativo é atribuída a etiqueta de repetição ao mesmo.

Caso não seja uma repetição, o segmento de vídeo é passado ao segundo nível do sistema de atribuição de etiquetas da camada de eventos. No nível 2 é utilizado o rácio de cor dominante para classificar os segmentos de vídeo em plano de campo ou plano exterior ao campo. Caso este valor seja elevado este é classificado

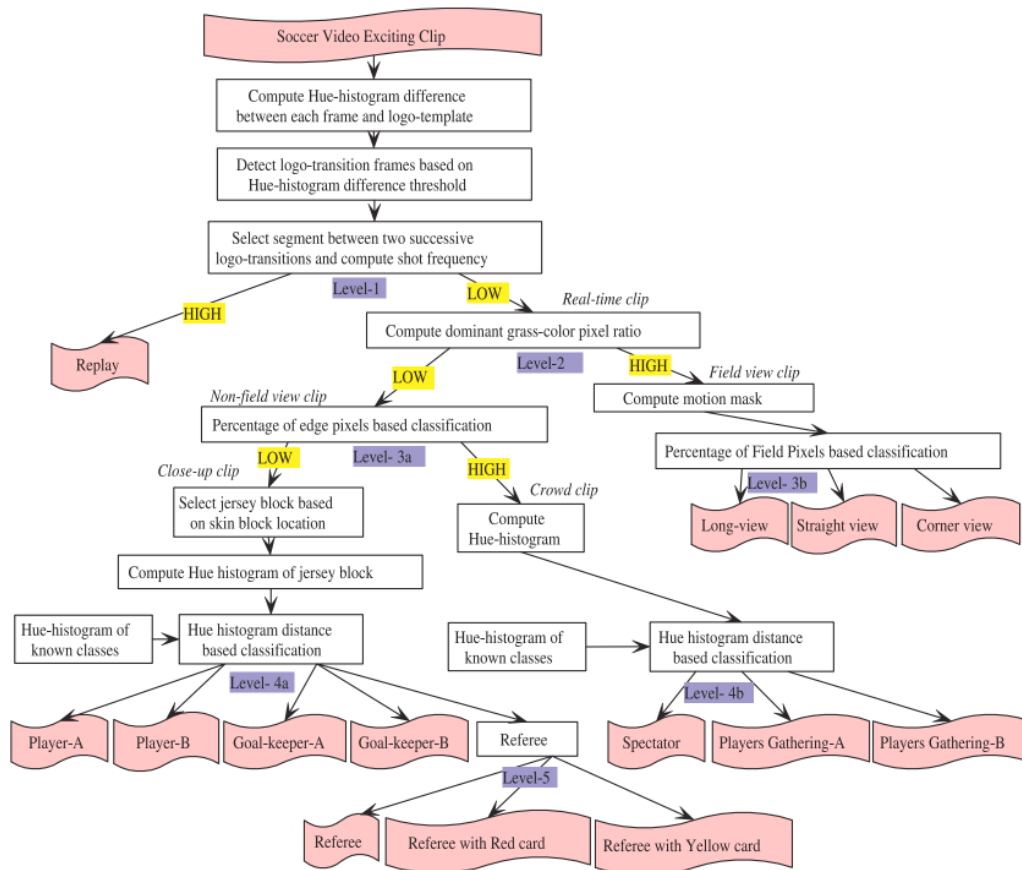


Figura 2.5: Árvore Hierárquica de atribuição de etiquetas [17].

como um plano de campo, caso contrário é classificado como um plano exterior ao campo (Figura 2.6).

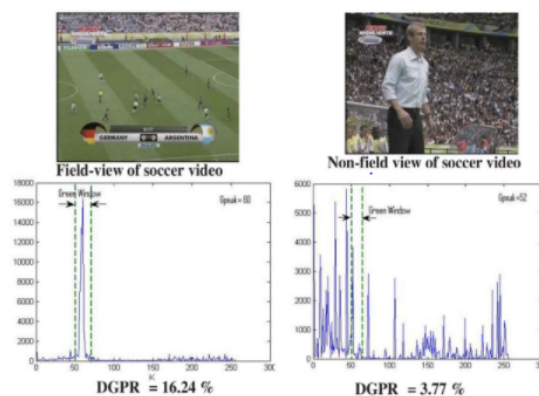


Figura 2.6: Classificação em plano de campo ou plano fora de campo [17].

No terceiro nível, caso o resultado do nível anterior seja classificado como um plano de campo, é aplicada uma máscara de movimento para calcular a percentagem de pixels de campo permitindo assim determinar o tipo de plano longo e

atribuir uma etiqueta da camada de eventos. Neste caso são utilizados três tipos de planos de campo: se apenas é visível o campo e não é visível o público é atribuída uma etiqueta ao plano denominada "plano longo", se o campo ocupa cerca de 40 por cento da imagem e a linha lateral se encontra nivelada é atribuída uma etiqueta denominada "plano a direito" e se for visível o canto do campo e algum público é atribuída uma etiqueta denominada "plano de canto" (Figura 2.7).

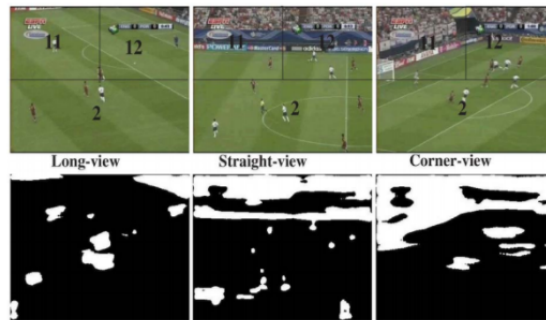


Figura 2.7: Classificação dos tipos de plano de campo baseado na máscara de movimento [17].

Ainda no terceiro nível, caso o rácio de pixels de cor dominante seja reduzido, é calculada a percentagem de pixels que pertencem a arestas das *frames* do segmento de vídeo recebido do nível anterior. Partindo do princípio que uma imagem do público possui mais densidade de arestas, caso exista uma elevada percentagem este segmento é definido como contendo imagens do público, caso contrário é definido como sendo composto por planos próximos dos jogadores ou do árbitro.

No quarto nível são atribuídas etiquetas de acordo com a classificação resultante do nível anterior. Se o segmento de vídeo é classificado como tendo o público visível é marcado com uma das seguintes etiquetas: "público", "agrupamento de jogadores da equipa A" ou "agrupamento de jogadores da equipa B". Esta atribuição baseia-se na distância de histogramas da componente *Hue* no espaço de cor HSV. No caso de o segmento gerado pelo nível anterior ser classificado como sendo plano próximo, as *frames* pertencentes a esse plano são divididas em 16 blocos e é utilizada a informação de cor da pele do jogador para detetar em que bloco se encontra a face do mesmo. Com esta informação é possível inferir em que bloco se encontra a camisola do jogador e determinar a cor da mesma, sendo esta comparada com as cores pré-definidas de forma arbitrária relativas às duas equipas, aos guarda-redes e ao árbitro. De acordo com o resultado da comparação é atribuída uma das seguintes etiquetas: "jogador equipa A", "jogador equipa B", "guarda-redes equipa A" ou "guarda-redes equipa B". No caso do plano ser

classificado como sendo plano com árbitro visível este é passado a um quinto e último nível.

No quinto nível da árvore hierárquica são recebidos os segmentos de vídeo onde é detetado um árbitro e também aqui as *frames* são divididas em 16 blocos e são aplicados filtros vermelhos e amarelos (Equações 2.9 e 2.10, respetivamente) aos 4 blocos superiores para detetar cartões amarelos ou vermelhos (Figura 2.8). Dependendo do resultado da técnica anterior é atribuída ao segmento uma das seguintes etiquetas: "árbitro" no caso de não ser detetado nenhum cartão, "árbitro com cartão amarelo" caso seja detetado um cartão amarelo e "árbitro com cartão vermelho" caso seja detetado um cartão vermelho.

$$R_c = \frac{r}{\sqrt{r^2 + g^2 + b^2}} \quad (2.9)$$

$$Y_c = \frac{1 - b}{\sqrt{(1 - r)^2 + (1 - g)^2 + (1 - b)^2}} \quad (2.10)$$

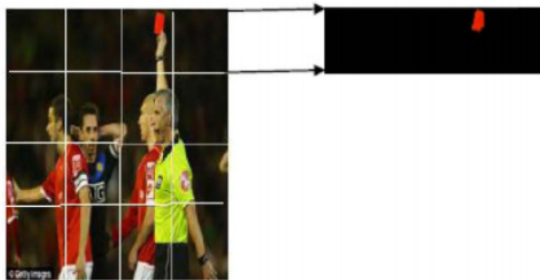


Figura 2.8: Detecção de cartões [17].

Após da atribuição de etiquetas aos vários segmentos de vídeo é necessária uma forma de associar os eventos a conceitos semânticos de alto nível. Para isso é utilizada uma rede Bayesiana por forma a calcular a probabilidade *à posteriori* dos conceitos de alto nível tendo em conta os eventos resultantes do processo identificado anteriormente. O conceito de alto nível com a probabilidade *à posteriori* mais elevada é atribuído ao segmento de vídeo. Neste processo é utilizado conhecimento do domínio como, por exemplo, o facto de que quando existe um agrupamento de jogadores pertencentes a determinada equipa, normalmente ocorre um golo dessa mesma equipa.

Cunha J. e Graça H. [13] desenvolveram um sistema que tem como objetivo a deteção de jogadas num jogo de futebol recorrendo ao processamento de imagem

e de áudio. O sistema é dividido em dois módulos: um baseado em imagem com o propósito de efetuar a segmentação do vídeo em planos e um baseado em áudio para detetar quais os planos relevantes do jogo.

A segmentação das *frames* em planos é semelhante à utilizada por Ekin, A. e Tekalp, A. em [15], ou seja, utiliza a diferença de percentagem de relva utilizando o espaço de cor HSV e nos histogramas utilizando o espaço de cor Red, Green, Blue (RGB) para definir quais as *frames* que delimitam um plano. É também utilizado o método *Golden Region* para determinar com maior precisão os planos médios. O método da deteção de câmara lenta é idêntico ao implementado por Ekin A. [22] e Fortio R. em [41].

O áudio é processado de forma semelhante ao que acontece com Ekin A. em [22], Kolekar M. e Segupta S. em [17] e Zawbaa H. et al. em [30] onde é calculada a potência do áudio numa janela temporal e posteriormente são encontrados os máximos locais.

Sripada S. et al. [38] utilizam um método de aprendizagem não-supervisionada baseado na técnica Independent Subspace Analysis (ISA) e que incorpora ideias relacionadas com redes neuronais. É também implementado um sistema que utiliza Histogram Of Gradients (HOG) 3D e amostragem densa cujos resultados são depois comparados com os resultados do algoritmo ISA. O algoritmo ISA é uma rede composta por duas camadas: a primeira com não-linearidade quadrada e a segunda com não-linearidade de raiz quadrada (Figura 2.9).

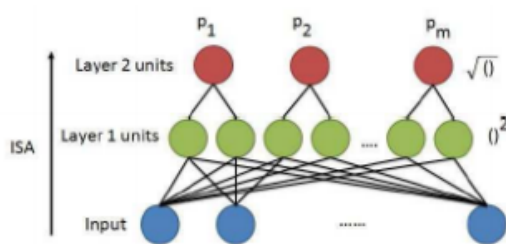


Figura 2.9: Rede ISA [38].

Neste artigo ocorre a anotação manual dos vídeos no que toca à sua classe (golo, falta, pontapé de livre, penalty e canto). As anotações são depois utilizadas como *Ground Truth* para comparação com os resultados dos algoritmos. São realizados vários testes para aferir o funcionamento dos dois algoritmos, tanto para várias classes como apenas para duas classes (relevante ou não-relevante). No entanto os resultados obtidos não são os desejados, o que segundo o autor, estará possivelmente relacionado com os parâmetros utilizados.

Rathod G. e Nikam D. [11] propõem um modelo que utiliza conhecimento humano para implementar um sistema de inferência difusa para extrair eventos relevantes de um jogo de futebol. A lógica difusa permite a especificação de um problema em termos linguísticos ao invés de implementar relações numéricas. A primeira fase deste modelo consiste na leitura do vídeo e na extração de *key frames*. Em seguida é utilizado um algoritmo de detecção de relva para classificar as *frames* em plano longo, médio ou próximo como visto em projetos revistos anteriormente. Às *frames* do tipo médio é aplicado um algoritmo de extração de linhas (semelhante ao efetuado por Ekin, A. e Tekalp. M. em [15]). A percentagem de relva e as linhas são depois passadas ao sistema de inferência difusa que aplica um conjunto de regras por forma a inferir os eventos importantes. Este modelo parte do princípio que só são importantes os eventos que possuem nas suas *frames* as linhas da área.

Jai-Andaloussi S. et al. [37] propõem um sistema inovador nesta área que ao invés de utilizar informação contida no jogo, utiliza a informação textual presente na rede social *Twitter* para a detecção de eventos relevantes. Este sistema analisa os sentimentos expressos na rede social *Twitter* para detetar a equipa de cada utilizador bem como as equipas e os jogadores que estão associados a cada evento. Esta informação é utilizada para produzir um resumo textual do jogo, ao invés de um resumo audiovisual como acontece com os restantes sistemas implementados nesta área. Este resumo é obtido utilizando métodos de aprendizagem automática que permitem a extração de conhecimento proveniente de bases de dados de grandes dimensões. Na figura 2.10 é possível visualizar o diagrama representativo do funcionamento do sistema.

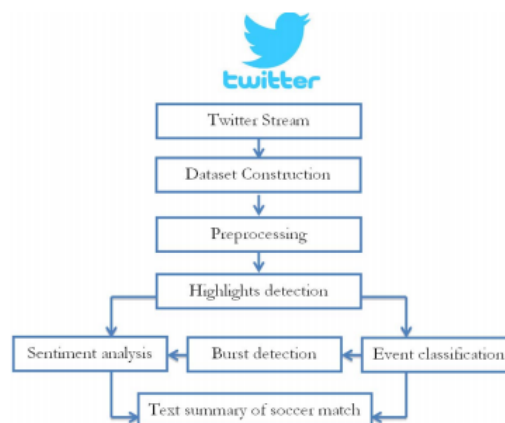


Figura 2.10: Diagrama Organizacional do Sistema [37].

Neste sistema ocorre a classificação de *tweets* em cinco classes (golo, cartão vermelho, cartão amarelo, penalty e falta). Como é utilizado um algoritmo de aprendizagem supervisionado, é necessário anotar manualmente os *tweets*, atribuindo uma das classes definidas a cada um. Foram testados três algoritmos para a classificação de *tweets*: SVM, *Naive Bayes* e redes neuronais (os *tweets* foram obtidos diretamente do *Twitter* em tempo real). No entanto, após a realização de alguns testes pelo autor do artigo, este chegou à conclusão que o algoritmo com melhor performance foi o SVM, o que levou a que fosse o utilizado no sistema final.

O sistema produziu resultados muito satisfatórios na deteção de golos, cartões e penaltys. No entanto os resultados não foram os melhores na deteção de faltas.

Bhavsar B. et al. [25] apresenta um projeto que utiliza técnicas semelhantes às revistas anteriormente para inferir características semânticas de alto nível que permitem a deteção automática de um tipo específico de evento num jogo de futebol, os pontapés de canto. O método de deteção de pixeis pertencentes ao campo aqui utiliza o espaço de cores HSV (é utilizado este espaço de cor em detrimento do RGB porque este modelo se enquadra melhor à perceção humana) e é obtido ao calcular a média de cada componente de cor à volta dos respetivos picos de histograma.

O algoritmo de agrupamento K-médias é aplicado aos valores de rácio de pixeis de cor dominante de uma *frame* por forma a classificá-la como plano longo ou não-longo. Este rácio refere-se ao número de pixeis de cor dominante numa *frame* quando em comparação com a totalidade dos pixeis nessa mesma *frame*. O algoritmo recebe as *frames* e dois valores iniciais arbitrários de rácio de pixeis de cor dominante. Seguidamente atribui as *frames* ao *cluster* com o valor de rácio mais próximo e recalcula os centros baseando-se no valor de média dos valores de rácio das *frames*. Este processo é repetido até não existirem alterações no valor dos centros de cada *clusters*. O resultado deste algoritmo é composto por dois *clusters* onde estão agrupadas as *frames* de plano longo e noutra as *frames* de plano não-longo.

Após efetuada esta classificação o algoritmo k-médias é aplicado novamente às *frames* definidas como planos não-longos para os classificar como planos médios ou fora de campo.

Depois de definido o tipo de cada *frame* são determinadas as *frames* onde ocorrem transições de planos, o que permite a segmentação do vídeo. Estas transições são determinadas utilizando processos idênticos aos utilizados em projetos revistos anteriormente, baseando-se na diferença de rácios de pixeis de cor dominante

entre duas *frames* consecutivas e na diferença de histogramas de cor no espaço de cor HSV. Se estes valores forem superiores a um limite empiricamente definido, conclui-se que existe uma transição de plano nessa *frame*. Depois de definidas as transições de planos é possível segmentar o vídeo, sendo o resultado um conjunto de planos que contêm *frames* com características idênticas. O tipo de um plano é obtido determinando qual o tipo de *frames* predominante nesse mesmo plano, ou seja, se existirem *frames* do tipo longo em maior número, o plano é classificado como um plano do tipo longo.

Para efetuar a detecção de pontapés de canto, é aplicada a todos os planos identificados como planos não-longos um filtro de *Sobel*, uma operação de erosão e uma transformação de *Hough* para detetar linha verticais (neste caso o objetivo é detetar o poste da bandeirola de canto, à semelhança do que acontece com Zawbaa H. et al. em [30]). Se for encontrada uma linha vertical e essa linha tiver um determinado comprimento pré-definido é detetado o evento de pontapé de canto.

Hong Shao et al. [29] propõe um método de segmentação de um vídeo em planos que combina a segmentação baseada na diferença de histogramas no espaço de cor HSV com a segmentação baseada numa característica denominada HOG (semelhante ao que acontece em [38]). No sistema proposto é efetuada uma primeira segmentação utilizando os histogramas no espaço de cor HSV. Em seguida é utilizada a característica HOG com o objetivo de eliminar possíveis segmentações erradas provenientes da fase anterior e eventualmente adicionar novas. O HOG é mais robusto no que toca a alterações geométricas e de luminosidade, o que resulta numa segmentação com maior grau de acerto.

A característica HOG é obtida seguindo os seguintes passos:

1. Dividir a imagem em pequenas regiões conectadas chamadas células (constituídas por 16x16 pixels) e calcular para cada uma o histograma de orientações de bordas;
2. Para cada célula quantificar num conjunto de 9 ângulos pré-definidos o valor de orientação de borda de cada pixel, atribuindo em seguida a cada célula o valor do ângulo que estiver em maioria nos seus pixels;
3. Conjuntos de 4 células adjacentes (2x2) são agrupadas em blocos que constituem a característica HOG final.

Na segunda fase de segmentação é obtida esta característica para todas as *frames* do vídeo e calculada a diferença entre *frames* consecutivas, e onde esta diferença

for maior que um determinado limite adaptativo é definida uma transição de plano. É também efetuada uma comparação dos resultados da segunda fase de segmentação com os resultados da primeira, ou seja, se numa determinada *frame* a primeira fase de segmentação determinar que existe uma transição de plano e nessa mesma *frame* o resultado da segunda fase de segmentação seja muito díspar ao da primeira, a transição de plano proveniente da primeira fase de segmentação é removida.

Sigari M. et al. [35] propõem um algoritmo que utiliza um sistema de inferência difusa semelhante ao proposto em [11] por Rathod G. e Nikam D. Neste projeto são definidos três níveis de processamento dos dados extraídos do vídeo. No primeiro nível são extraídos os histogramas das componentes de cor R, G e B de cada *frame* e é utilizado o método de diferença de histogramas para detecção de transições de plano, semelhante ao utilizado em alguns projetos revistos anteriormente. As transições graduais são detetadas recorrendo a histogramas 3D aplicados às *frames* de transição. Os histogramas são utilizados num classificador do tipo Classification And Regression Trees (CART) para detetar a presença de logótipos. O método parte do princípio que as transições graduais são acompanhadas da amostragem de animações de logótipos (Figura 2.11).



Figura 2.11: Exemplo de transição gradual com logótipo.

Em caso de deteção de um logótipo é declarada a existência de uma transição gradual. As repetições são detetadas como um conjunto de *frames* entre duas amostragens de logótipos (partindo do mesmo princípio que Zawbaa H. et al. em [30]).

Em último lugar é calculada a energia do áudio no plano imediatamente antes das repetições detetadas. Esta característica é utilizada para determinar a excitação do público.

Janelas temporais onde a energia do áudio é maior que 80% da energia máxima

nesse intervalo são indicativas da existência de excitação do público e potencialmente representam um evento relevante no jogo.

O sistema de inferência difusa processa as características produzidas pelos métodos anteriores: a duração da repetição, o rácio de planos próximos e fora de campo e a excitação do público imediatamente antes da repetição e calcula uma probabilidade de um acontecimento importante utilizando heurísticas definidas anteriormente. O sistema implementa uma máquina de estados hierárquica constituída por 3 macro-estados onde o primeiro estado analisa todas as *frames* do vídeo e apenas invoca o segundo estado quando deteta uma transição de plano. O terceiro estado é invocado quando é detetado um logótipo no segundo estado. É no terceiro estado que são passadas as informações ao sistema de inferência difusa para que este determine se o segmento é relevante ou não.

O sistema de inferência difusa utiliza funções de associação que permitem associar os *inputs* recebidos a valores definidos previamente. Após este processo o sistema aplica regras aos valores resultantes das funções de associação para inferir a relevância. Esta classificação é utilizada depois para inferir quais os planos antecedentes a repetições e as respetivas repetições que são selecionadas para o vídeo de resultado final.

Esta estrutura permite que os estados que requerem mais poder computacional apenas sejam executados quando é necessário, o que faz com que este sistema seja adequado a aplicações de tempo real visto que o único processamento transversal a todas as *frames* é o referente ao primeiro estado, ou seja, apenas é calculado o histograma no espaço de cores RGB e a consequente deteção de transições de plano.

## 2.2 Sistemas baseados em Características Biométricas

Nos últimos tempos tem sido estudada a hipótese de, para além das características utilizadas tradicionalmente neste tipo de sistemas como a imagem e o áudio, serem utilizadas também características biométricas. Exemplos de características biométricas são as ondas produzidas pelo cérebro e o batimento cardíaco de um sujeito quando este é exposto a determinado estímulo audiovisual.

Salehin M. e Paul M. [14] propõem um sistema capaz de detetar o interesse de um espetador num vídeo de qualquer tipo. Este sistema utiliza a Electro-encefalograma (EEG) para obter a atividade neuronal do indivíduo enquanto este assiste a um vídeo. O sistema é composto por 4 fases:

- Recolha de dados através da realização da encefalografia durante a visualização de um vídeo;
- Extração de características relevantes da informação recolhida;
- Geração da curva de atenção neuronal;
- Sumarização do vídeo utilizando a informação das etapas anteriores.

Na primeira fase é realizada a EEG. Este processo consiste na colocação de vários elétrodos no escalpe do indivíduo para capturar a eletricidade libertada pelos neurónios no cérebro humano quando existe algum tipo de estímulo exterior.

Na fase de extração de características é utilizada a informação recolhida na fase anterior para obter características que possam ajudar a determinar se o utilizador está interessado em determinado segmento de vídeo ou não. Partindo do princípio que o sinal produzido pela encefalografia contém informação mais relevante nas frequências elevadas, nesta fase é utilizada a técnica Empirical Mode Decomposition (EMD) [24] que decompõe o sinal num finito número de componentes ordenados da frequência mais elevada para a mais reduzida (Figura 2.12 ). Cada componente tem o nome de Intrinsic Mode Function (IMF). O comprimento destes componentes é igual ao do sinal original e o seu domínio é também temporal. Para este sistema é utilizada uma combinação do primeiro e segundo IMFs na geração da curva de atenção neuronal.

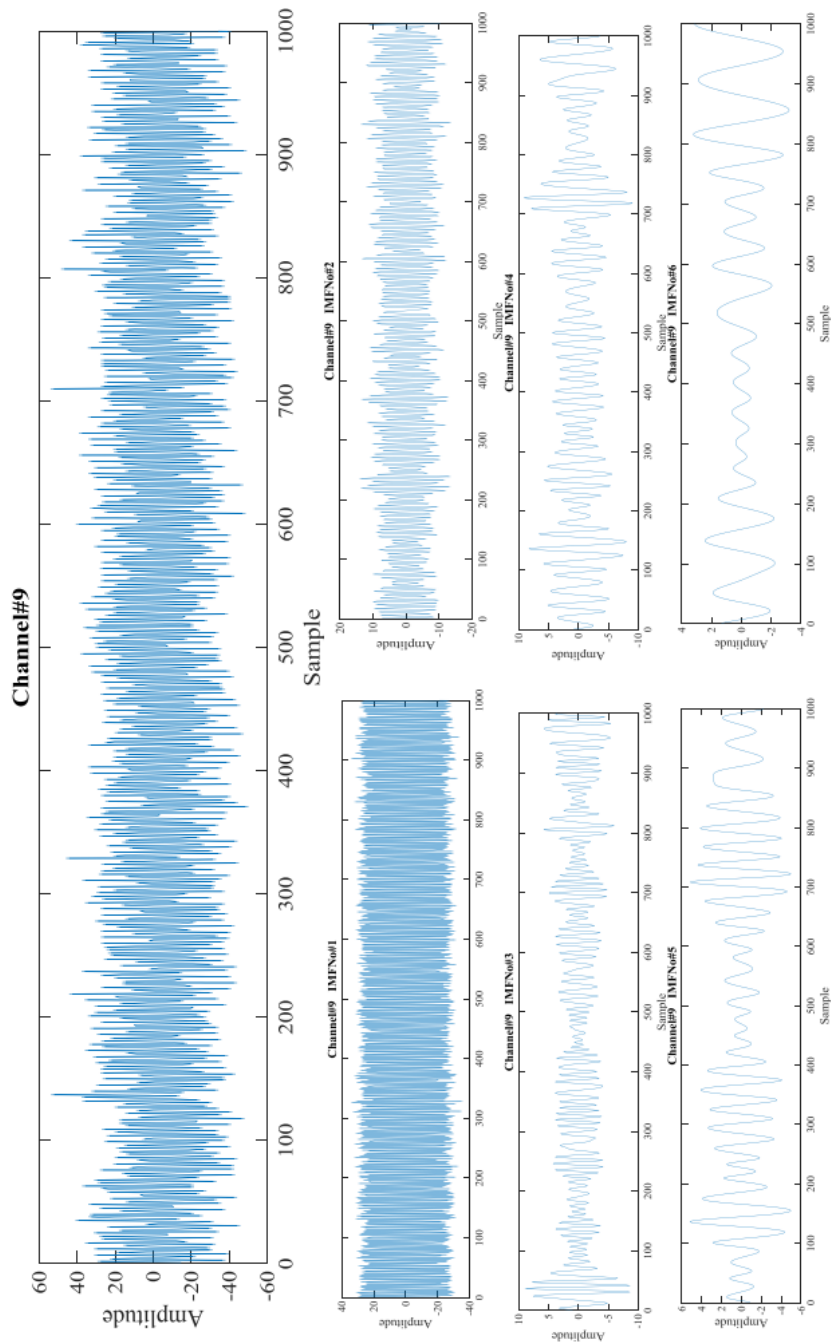


Figura 2.12: Aplicação de EMD [14].

Na fase de geração da curva de atenção neuronal é calculada a Power Spectral Density (PSD) normalizada dos dois IMFs recebidos da fase anterior resultantes do sinal combinado da encefalografia aplicada ao lóbulo frontal e occipital (Figura 2.13). O valor de atenção neuronal é calculado e os seus valores são ordenados de forma decrescente para que as *frames* com os valores mais elevados de atenção neuronal apareçam primeiro na lista de valores. Em último lugar são seleccionadas as *frames* correspondentes aos  $n$  maiores valores dessa lista, sendo

$n$  um valor definido pelo utilizador do sistema. Por último é produzida uma sumarização do vídeo utilizando as *frames* resultantes na sua ordem cronológica original.

Chakraborty P. [8, 10] e Tjondronegoro D. [10] propuseram sistemas capazes de detetar o interesse de um espetador em vídeos utilizando características biométricas como o ritmo cardíaco e as expressões faciais.

Nestes sistemas são atribuídas pontuações a cada *frame* relativas a três categorias de emoção: positiva, negativa ou neutra. São extraídas também características temporais dos dados de batimento cardíaco. Os dados são depois fundidos com os dados relativos às expressões faciais obtendo um conjunto de características como funções temporais. Instâncias consecutivas de *frames* que possuam valores desta função elevados serão identificados como um segmento de vídeo.

O algoritmo de deteção facial deteta a zona da face utilizando o detetor de face *Viola-Jones* e extrai também os movimentos dos pontos faciais utilizando um *tracker* Active Shape Model (ASM). A distância entre os pontos faciais internos é utilizada como característica de textura e geometria num classificador SVM para classificar nas três classes de emoções definidas anteriormente (positiva, negativa ou neutra). O *output* da classificação é uma probabilidade *frame a frame* de pontuações para cada classe de emoção que varia entre 0 e 1, sendo a soma destas três pontuações igual a 1.

Em relação às características relacionadas com o ritmo cardíaco é utilizado um filtro passa-baixo para alisar o sinal e remover flutuações rápidas. Através deste sinal são obtidas as características que serão utilizadas na classificação: frequência de variação, máximos locais e a energia da banda de baixa frequência. As amostras do ritmo cardíaco são comparadas com um limite pré-definido e caso apresentem um maior valor são selecionadas. As amostras selecionadas consecutivas são convertidas em segmentos de vídeo.

Em último lugar é efetuada uma operação de união *frame a frame* entre os segmentos resultantes do módulo relativo à expressão facial e do módulo relativo ao ritmo cardíaco para obter o conjunto final de segmentos que representam a sumarização final do vídeo.

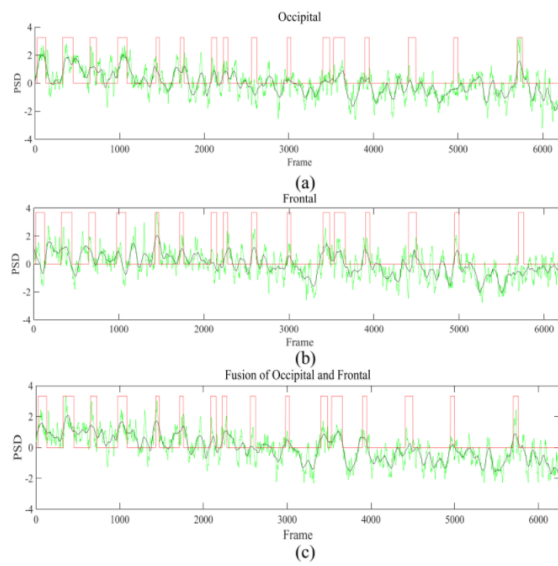


Figura 2.13: Exemplo de PSD [10].



# 3

## Modelo Conceptual do Sistema

Neste capítulo é apresentado de forma detalhada o sistema que permite implementar um fluxo sistemático que começa com a captura dos dados fisiológicos e a anotação de emoções e termina com a geração de um conjunto de segmentos de um vídeo que representem o seu conteúdo relevante, tentando assim solucionar todas as questões mencionadas no capítulo 1.

O sistema proposto divide-se em 3 módulos principais que são independentes entre si mas que em conjunto contribuem para o objetivo final. Estes módulos são:

- Aplicação de captura de informação fisiológica e anotação de emoções;
- Módulo de geração de características HRV;
- Módulo de análise de vídeos.

De um modo geral o seu funcionamento é o seguinte:

1. A aplicação capta as informações fisiológicas e as anotações de emoções e efetua o *upload* para uma base de dados;
2. O módulo de geração de características HRV obtém as informações fisiológicas da base de dados e processa-as gerando assim as características HRV que são depois passadas ao módulo de análise de vídeo;

3. O módulo de análise de vídeo utiliza-as em conjunto com as emoções e as características audiovisuais para gerar os segmentos de vídeo relevantes (Figura 3.1).

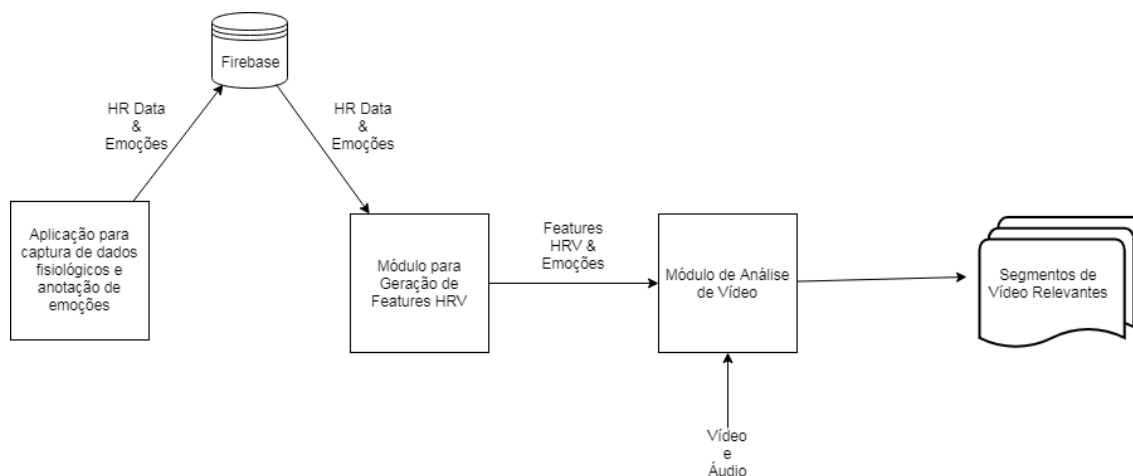


Figura 3.1: Diagrama de Blocos do Sistema.

## 3.1 Aplicação de captura informação fisiológica e anotação de emoções

### 3.1.1 Especificação de Requisitos

Um bom ponto de partida para a modelação de um sistema é a definição dos seus requisitos funcionais (Tabela 3.1). Os requisitos funcionais permitem ter uma ideia geral das funcionalidades específicas que definem o sistema e que idealmente este é capaz de realizar finda a sua implementação.

Requisito	Descrição	Obrigatório
R1	Os dados de <i>login</i> são constituídos por email e <i>password</i> .	Sim
R2	Existem dois tipos de utilizadores: Utilizador normal e Administrador.	Sim
R3	Para efetuar qualquer ação na aplicação o utilizador tem de ter <i>login</i> efetuado.	Sim
R4	Se o utilizador tiver <i>login</i> efetuado a primeira página mostrada é a do menu principal, em caso contrário é a de <i>login</i> .	Sim

R5	O menu principal permite as seguintes ações: gravar o jogo, verificar gravações realizadas e fazer <i>logout</i> .	Sim
R6	A página de <i>login</i> permite efetuar <i>login</i> na aplicação e realizar o registo de um novo utilizador.	Sim
R7	Os dados necessários para o registo do utilizador são os seguintes: nome, data de aniversário, email, password, género e clube.	Sim
R8	Só é possível passar da página de <i>scan</i> de dispositivos para a página de seleção dos clubes se a aplicação conseguir conectar-se com sucesso a um dispositivo <i>Bluetooth</i> .	Sim
R9	O utilizador normal apenas consegue consultar as suas próprias gravações.	Sim
R10	O administrador consegue consultar as gravações de todos os utilizadores.	Sim
R11	Durante o jogo o utilizador pode terminar a gravação a qualquer altura.	Sim
R12	No fluxo normal de gravação, durante o tempo regulamentar o utilizador apenas pode iniciar o intervalo uma única vez.	Sim
R13	Sempre que o utilizador para a gravação (quer pare definitivamente ou inicie um intervalo) é efetuado o <i>upload</i> de 2 ficheiros: o relativo aos dados fisiológicos e o relativo às emoções.	Sim
R14	Se o utilizador iniciar intervalo durante o tempo regulamentar a gravação pausa e é efetuado o <i>upload</i> dos ficheiros relativos à primeira parte.	Sim
R15	Se o utilizador terminar gravação durante o tempo regulamentar sem que tenha ocorrido o intervalo, é feito <i>upload</i> de ficheiros relativos à primeira parte e o processo de gravação termina.	Sim
R16	Se o utilizador terminar gravação e tenha ocorrido o intervalo, é feito <i>upload</i> de ficheiros relativos à segunda parte e é perguntado ao utilizador se o jogo possui tempo extra.	Sim

R17	O processo de gravação no tempo extra funciona de forma semelhante ao do tempo regulamentar (o tempo extra também possui um intervalo).	Sim
R18	Quando o utilizador termina a gravação no tempo extra é perguntado se o jogo vai a pontapés de grande penalidade.	Sim
R19	A gravação do jogo durante os pontapés de grande penalidade é um pouco diferente da gravação durante o tempo extra e o tempo regulamentar dado não existir intervalo, neste caso o utilizador apenas pode iniciar e terminar a gravação.	Sim
R20	Caso ocorra uma desconexão da aplicação com o sensor durante a gravação, esta mostra uma mensagem de desconexão e fica em espera até que o sensor seja conectado novamente.	Sim
R21	Os ficheiros resultantes são gravados no sistema em formato Comma-Separated Values (CSV) antes de serem carregados no <i>Firebase</i> com o mesmo formato.	Sim
R22	A aplicação necessita de estar ligada à Internet.	Sim
R23	O utilizador pode definir 5 níveis de emoção negativa e positiva.	Sim
R24	O utilizador pode definir 1 nível de emoção neutra.	Sim
R25	Caso o utilizador num espaço de x segundos (configurável) não definir uma emoção, o sistema pergunta ao utilizador o que está a sentir ao mostrar um <i>popup</i> .	Sim

Tabela 3.1: Requisitos Funcionais

### 3.1.2 Casos de utilização

Após a recolha dos requisitos funcionais do sistema, é possível identificar os diagramas de casos de utilização, onde são descritas as funcionalidades do sistema

proposto, Dado que não há diferenciação no acesso às várias funcionalidades entre o utilizador normal e o utilizador administrador, é apenas definido um ator: o utilizador genérico. Em primeiro lugar é apresentado um diagrama de Casos de Utilização geral onde são apresentadas as funcionalidades gerais do sistema (Figura 3.2):

- **Registar Utilizador** - Criação de um novo *login*, sendo os dados inseridos carregados posteriormente na base de dados *Firebase*;
- **Autenticação** - Capacidade do sistema de lidar com funcionalidades relacionadas com a autenticação, implementando um sistema de *login*, em que o utilizador previamente registado insere o seu email e a sua *password* para aceder às restantes funcionalidades da aplicação;
- **Gravar Jogo** - Consiste na funcionalidade principal do sistema. Permite a conexão a um dispositivo *Bluetooth* de medição cardíaca e a consequente captação da informação fisiológica bem como a anotação de emoções por parte do utilizador. Esta funcionalidade pode ser decomposta num conjunto de funcionalidades;
- **Visualizar Gravações** - Consiste na amostragem das várias gravações efetuadas pelo utilizador que tem *login* efetuado. Se o utilizador for administrador o sistema mostra todas as gravações.

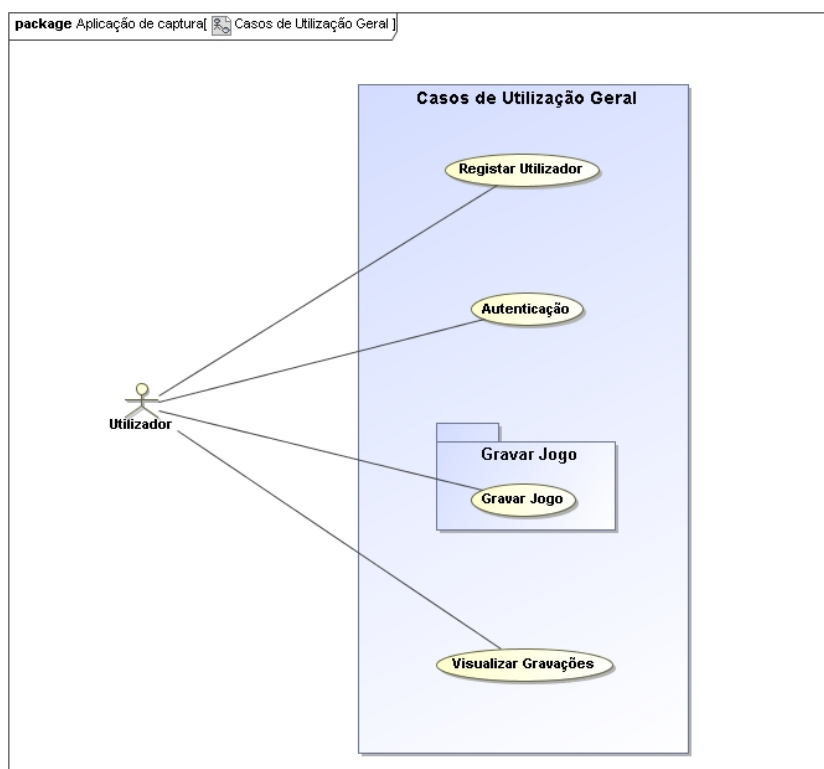


Figura 3.2: Casos de Utilização Geral.

A funcionalidade "Gravar Jogo" pode ser decomposta em 5 outras funcionalidades que em conjunto têm o objetivo de providenciar um sistema capaz de implementar todo o fluxo de gravação de um jogo (Figura 3.3).

- **Iniciar Gravação** - Começa o processo de gravação, dando início ao cronómetro, iniciando a captura de dados do sensor e gerando os ficheiros de emoções e dados fisiológicos em disco;
- **Parar Gravação** - Termina o processo de gravação, efetuando *upload* do ficheiro para o sistema de armazenamento *Firebase*;
- **Iniciar Intervalo** - Pausa a gravação, efetuando *upload* do ficheiro relativo à primeira parte e criando um novo relativo à segunda parte;
- **Terminar Gravação** - Inicia a gravação, começando o cronómetro nos 45 minutos e escrevendo desta vez no ficheiro de segunda parte;
- **Anotar Emoção** - Permite a escrita no ficheiro do tipo de emoção e intensidade.

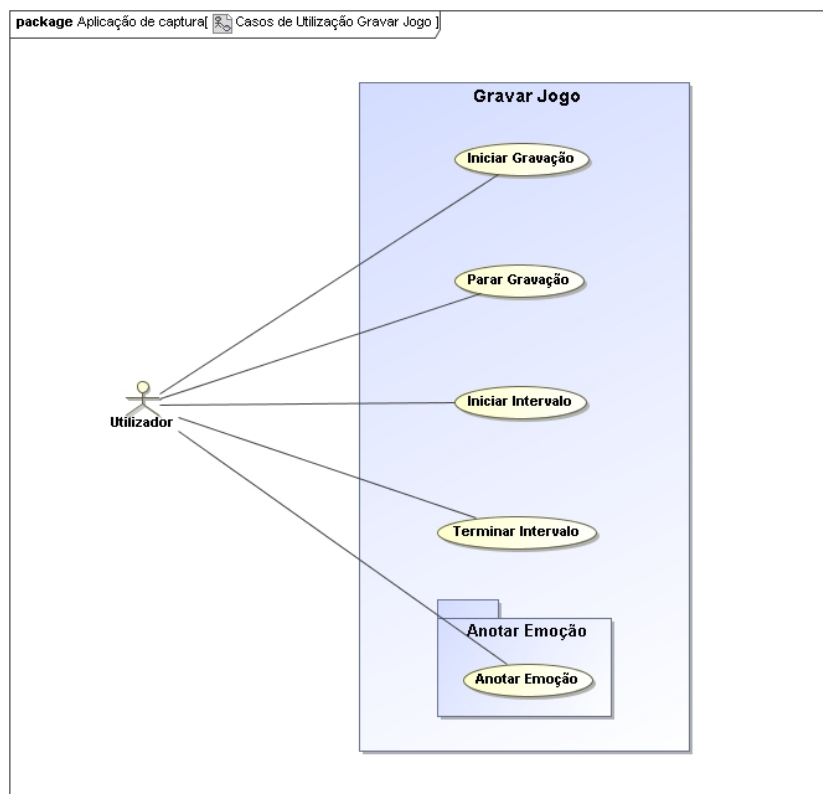


Figura 3.3: Casos de Utilização Gravar Jogo.

A funcionalidade "Anotar Emoção" pode ser decomposta em 3 funcionalidades, uma para cada tipo de emoção que se pode definir (Figura 3.4).

- **Anotar Emoção Positiva** - Anotação de emoção positiva e da sua intensidade que são gravadas no ficheiro CSV;
- **Anotar Emoção Neutra** - Anotação de emoção neutra e da sua intensidade que são gravadas no ficheiro CSV;
- **Anotar Emoção Negativa** - Anotação de emoção negativa e da sua intensidade que são gravadas no ficheiro CSV.

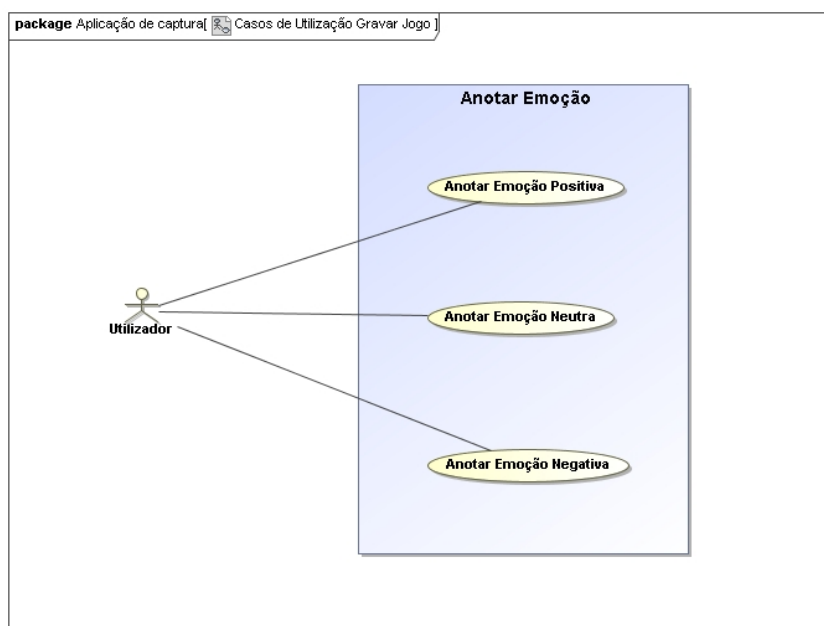


Figura 3.4: Casos de Utilização Anotar Emoção.

### 3.1.3 Histórias de Utilização

Para definir com maior detalhe os casos de utilização torna-se necessário identificar histórias de utilização que permitem uma visão sequencial do processo e também a definição de qualquer pré-condição que seja necessária à sua execução.

#### 3.1.3.1 Registrar Utilizador

Pré Condições:

- Não existem.

Processo:

1. O utilizador seleciona a opção "No account? Register";
2. O sistema solicita um conjunto de dados (ver requisito funcional R7, tabela 3.1);
3. O utilizador insere os dados solicitados e posteriormente carrega no botão "Register";
4. O sistema valida os dados inseridos;

5. O sistema cria um novo utilizador na base de dados *Firebase* utilizando os dados inseridos;
6. O sistema mostra uma mensagem de confirmação da criação do utilizador;
7. O caso de utilização termina.

### 3.1.3.2 Login

Pré Condições:

- Não existem.

Processo:

1. O sistema solicita um conjunto de dados (ver requisito funcional R1, tabela 3.1);
2. O utilizador insere os dados solicitados;
3. O sistema mostra o menu principal;
4. O caso de utilização termina.

### 3.1.3.3 Iniciar Gravação

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Idle".

Processo:

1. O utilizador pressiona o botão "START GAME";
2. O caso de utilização termina.

### 3.1.3.4 Parar Gravação

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Jogo", "Tempo Extra" ou "Penalties".

Processo:

1. O utilizador pressiona o botão "STOP GAME";
2. O caso de utilização termina.

### 3.1.3.5 Iniciar Intervalo

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Jogo" ou "Tempo Extra".

Processo:

1. O utilizador pressiona o botão "START BREAK";
2. O caso de utilização termina.

### 3.1.3.6 Terminar Intervalo

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Intervalo".

Processo:

1. O utilizador pressiona o botão "START BREAK";
2. O caso de utilização termina.

### 3.1.3.7 Anotar Emoção Positiva

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Jogo", "Tempo Extra" ou "Penalties".

Processo:

1. O utilizador pressiona sobre o botão de emoção positiva e arrasta até aparecer a intensidade pretendida (ver requisito funcional R23, tabela 3.1), largando para anotar a emoção;
2. O caso de utilização termina.

### 3.1.3.8 Anotar Emoção Neutra

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Jogo", "Tempo Extra" ou "Penalties".

Processo:

1. O utilizador pressiona sobre o botão de emoção neutra e arrasta, largando para anotar a emoção (ver requisito funcional R24, tabela 3.1);
2. O caso de utilização termina.

### 3.1.3.9 Anotar Emoção Negativa

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;
- A aplicação tem que estar conectada a um dispositivo *Bluetooth* de medição cardíaca;
- O estado de gravação da aplicação tem que ser "Jogo", "Tempo Extra" ou "Penalties".

Processo:

1. O utilizador pressiona sobre o botão de emoção positiva e arrasta até aparecer a intensidade pretendida (ver requisito funcional R23, tabela 3.1), largando para anotar a emoção;
2. O caso de utilização termina.

### 3.1.3.10 Visualizar Gravações

Pré Condições:

- O utilizador tem que ter *login* válido efetuado;

Processo:

1. O utilizador seleciona no menu principal a opção "My Recordings";
2. O utilizador pode visualizar todas as suas gravações fazendo *scroll* na janela;
3. O caso de utilização termina.

### 3.1.4 Máquina de Estados da Gravação

Para exemplificar todo o fluxo de gravação durante um jogo é apresentada uma máquina de estados onde são indicados todos os estados e transições que ocorrem na execução normal da aplicação. Quando a atividade de gravação inicia esta encontra-se inicialmente num estado "Idle" em que não ocorre nada enquanto não existirem estímulos ao sistema.

Quando ocorre um estímulo, que neste caso consiste no utilizador premir o botão "START GAME" o sistema passa para o estado "Jogo" em que se inicia a gravação e os ficheiros necessários são criados em sistema. Do estado "Jogo" o sistema pode passar para o estado "Intervalo" se o utilizador premir o botão "START BREAK" ou então, caso o utilizador prima o botão "STOP GAME", pode ir para o estado "Tempo Extra" ou terminar a execução, dependendo da escolha do utilizador.

No estado "Tempo Extra" o jogador pode também passar ao estado "Intervalo" caso o utilizador prima o botão "START BREAK" ou caso o utilizador prima o botão "STOP GAME" o sistema pode passar ao estado "Penalties" ou terminar a execução, dependendo da escolha do utilizador (ver requisito R18 na tabela 3.1).

Do estado "Intervalo" o sistema pode passar ao estado "Tempo Extra" ou "Jogo", dependendo do estado onde se encontrava previamente (por exemplo, se o sistema se encontrava no estado "Jogo" antes de ir para "Intervalo", este volta para o estado "Jogo" quando o utilizador prime o botão "STOP BREAK" e o mesmo acontece com o estado "Tempo Extra").

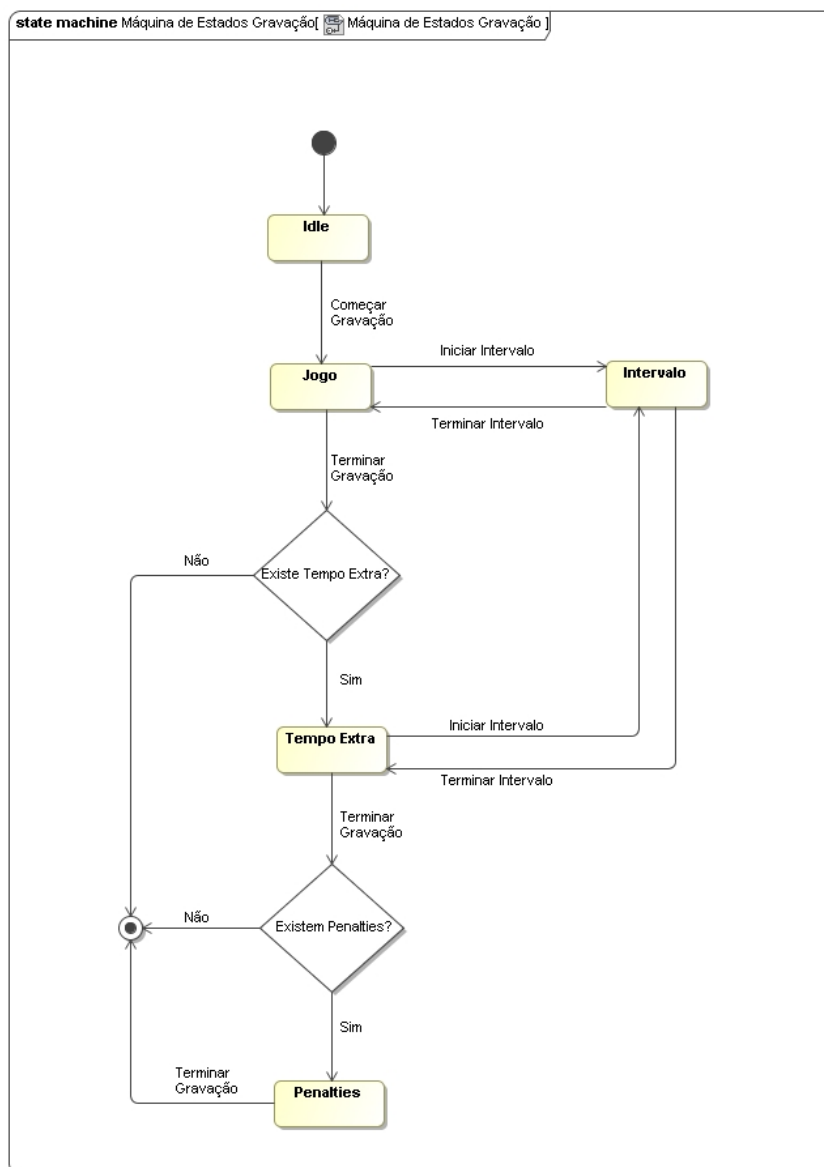


Figura 3.5: Máquina de Estados da Gravação.

### 3.1.5 Arquitetura da Solução

O sistema segue o padrão de arquitetura de *software* Model View Controller (MVC). Este padrão divide uma aplicação em 3 partes interconectadas que permitem separar as representações de informação internas dos modos como estas são apresentadas ao utilizador (Figura 3.6). Isto permite uma maior reutilização de código e uma maior modularidade dos vários módulos que compõem o sistema. O modelo consiste nos dados e lógica da aplicação. Uma visão é a informação apresentada ao utilizador (que no caso de uma aplicação *Android* são os ficheiros Extensible Markup Language (XML) responsáveis por implementar o *layout*

da mesma). Os controladores (no caso de uma aplicação *Android* as atividades possuem funções semelhantes aos controladores) basicamente são responsáveis por receber *inputs* do utilizador e convertê-los em comandos que são aplicados ao modelo ou a uma visão [19].

Nesta arquitetura o modelo interage com o módulo de acesso a dados que possui repositórios que utilizam o sistema *Firebase* para adicionar, ler, atualizar e apagar dados.

Tal como indicado anteriormente, o sistema também possui um módulo de conexão ao sensor de medição cardíaca *Movesense* que funciona como uma "caixa negra". Apenas é utilizada para efetuar o *scan* dos dispositivos, a conexão a um dispositivo e para realizar a captação dos dados fisiológicos.

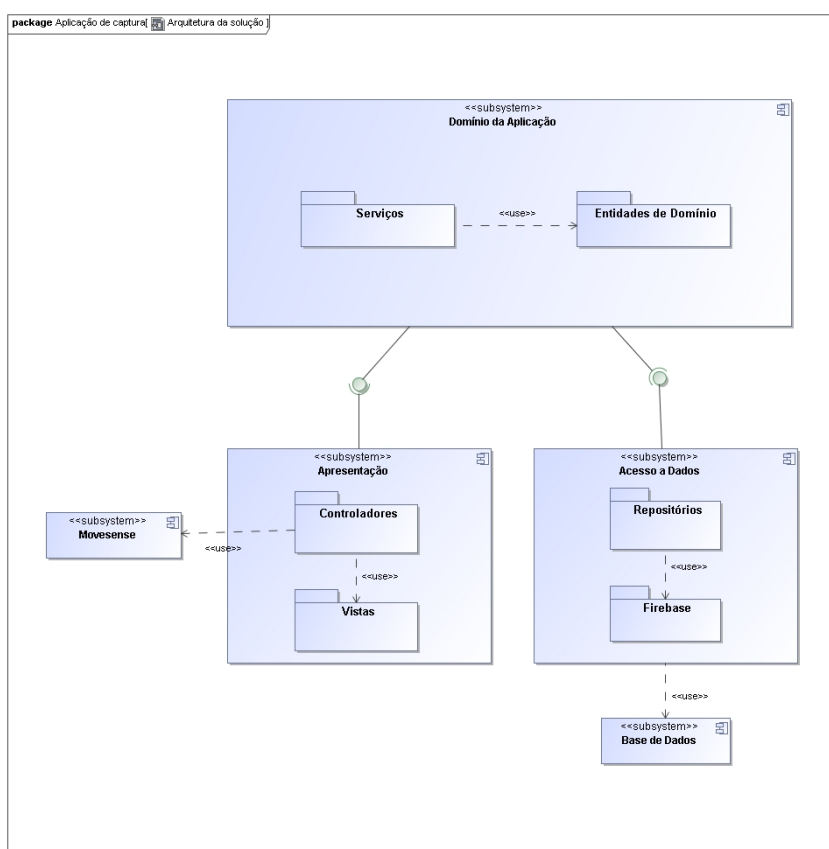


Figura 3.6: Arquitetura da Solução.

## 3.2 Módulo de geração de características HRV

### 3.2.1 Casos de Utilização

O módulo de geração de características HRV trata-se apenas de uma simples aplicação de consola implementada na linguagem de programação *Python*. Esta aplicação tem apenas duas funcionalidades base: a de efetuar *download* dos ficheiros do sistema *Firebase* e processá-los para gerar os ficheiros que contêm as características HRV (Figura 3.7). Neste caso não há distinção entre utilizadores, logo apenas existe um ator nos casos de utilização.

- **Efetuar Download *Firebase*** - Permite ao utilizador obter uma listagem dos seus ficheiros que estão presentes no sistema *Firebase* e selecionar um por forma a fazer o *download*;
- **Gerar Ficheiros HRV** - Permite ao utilizador escolher um ficheiro descarregado anteriormente do sistema *Firebase* e processá-lo para gerar um ficheiro que contém as características HRV.

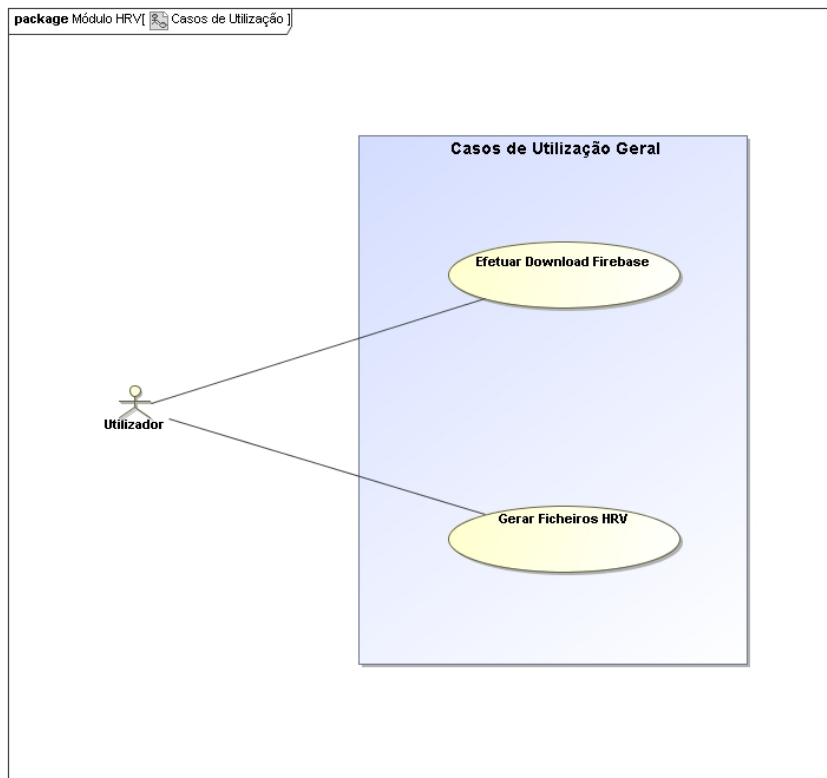


Figura 3.7: Casos de utilização HRV.

## 3.2.2 Histórias de Utilização

### 3.2.2.1 Efetuar Download Firebase

Pré Condições:

- Não existem.

Processo:

1. O utilizador seleciona a opção "Download folder from Firebase";
2. O sistema solicita um conjunto de dados (ver requisito funcional R1, tabela 3.1);
3. O utilizador insere os dados solicitados;
4. O sistema mostra uma lista de ficheiros;
5. O utilizador seleciona um ficheiro;
6. O sistema descarrega o ficheiro para uma pasta local;
7. O caso de utilização termina.

### 3.2.2.2 Gerar Ficheiros HRV

Pré Condições:

- Não existem.

Processo:

1. O utilizador seleciona a opção "Generate HRV Files";
2. O sistema mostra uma lista de ficheiros presentes numa pasta local;
3. O utilizador seleciona um dos ficheiros;
4. O sistema gera o ficheiro com as características HRV e guarda-o numa pasta local.
5. O caso de utilização termina.

### 3.3 Módulo de análise de vídeos

Como já referido anteriormente neste capítulo, todo o trabalho relativo a este módulo é realizado sobre o código desenvolvido por Cunha J. e Graça H. em [13]. Devido a este facto não são aprofundadas nesta secção questões relacionadas com a arquitetura do sistema, o foco é nas alterações realizadas ao sistema original que são ao nível da Graphical User Interface (GUI) e do algoritmo de segmentação e de classificação de planos relevantes (o modo de segmentação é alterado e são adicionadas à classificação novas características visuais e de HRV como é possível verificar mais à frente).

Este módulo é responsável por implementar uma interface que permite o carregamento de um vídeo, a sua segmentação e a classificação dos planos resultantes. De uma forma geral o fluxo de processamento deste módulo é o seguinte:

1. É carregado um vídeo no sistema;
2. É efetuada a segmentação do vídeo em planos;
3. O algoritmo de deteção de grande área é aplicado a todos os planos;
4. O algoritmo de deteção de cartões é aplicado a todos os planos;
5. As emoções e a informação de HRV relativas ao intervalo temporal do vídeo são utilizadas no algoritmo SVM;
6. As informações geradas pelos algoritmos anteriores são utilizadas para definir os planos relevantes.

#### 3.3.1 Casos de Utilização

- **Carregar Ficheiro** - Permite seleccionar um vídeo e carregá-lo no sistema para depois ser processado;
- **Reproduzir Vídeo** - Funcionalidade que permite reproduzir o vídeo carregado. Pode ser decomposta num conjunto de funcionalidades;
- **Gerar Segmentos** - Funcionalidade principal do sistema. Permite utilizar um algoritmo de segmentação e classificação por forma a gerar todos os planos de um jogo e indicar quais destes são relevantes.

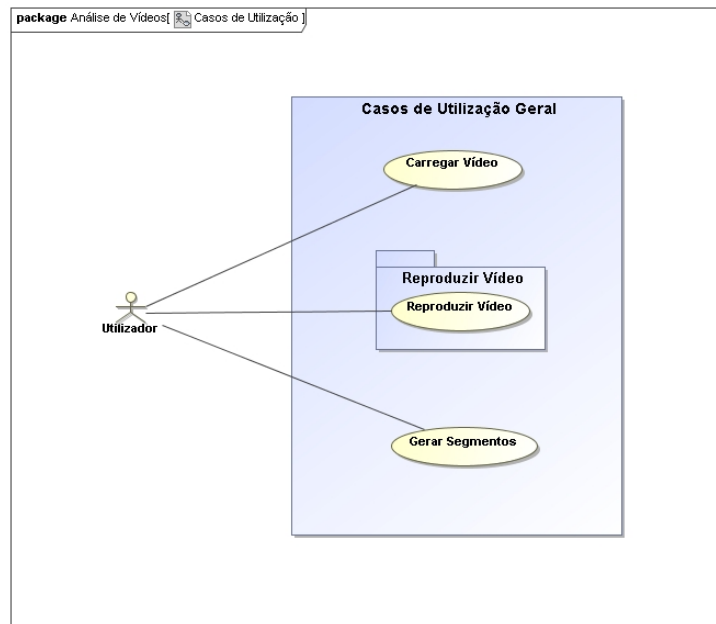


Figura 3.8: Casos de utilização Análise Vídeos.

- **Reproduzir Vídeo Completo** - Permite reproduzir o vídeo carregado na sua totalidade. O vídeo é reproduzido numa janela à parte;
- **Reproduzir Segmento** - Permite reproduzir os segmentos gerados pelo algoritmo de segmentação. O vídeo é reproduzido na mesma janela.

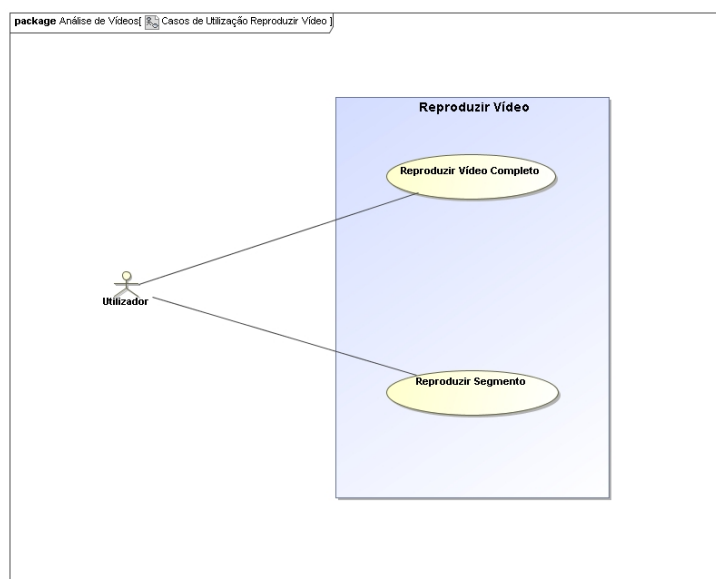


Figura 3.9: Casos de utilização Reproduzir Vídeo.

## 3.3.2 Histórias de Utilização

### 3.3.2.1 Carregar Vídeo

Pré Condições:

- Não existem.

Processo:

1. O utilizador seleciona a opção "OPEN";
2. O sistema solicita a escolha do ficheiro de vídeo;
3. O utilizador seleciona o ficheiro de vídeo pretendido;
4. O caso de utilização termina.

### 3.3.2.2 Reproduzir Vídeo Completo

Pré Condições:

- Existir um vídeo carregado.

Processo:

1. O utilizador seleciona a opção "Full Movie";
2. O vídeo é exibido no painel superior na sua totalidade;
3. O caso de utilização termina.

### 3.3.2.3 Reproduzir Segmento

Pré Condições:

- Existir um vídeo carregado;
- Existir segmentos resultantes do algoritmo de segmentação e classificação.

Processo:

1. O utilizador clica num dos segmentos no painel inferior;
2. O vídeo é exibido no painel superior;
3. O caso de utilização termina.

### 3.3.2.4 Gerar Segmentos

Pré Condições:

- Existir um vídeo carregado.

Processo:

1. O utilizador seleciona a opção "RUN";
2. O sistema executa o algoritmo de segmentação e classificação de planos;
3. O sistema exhibe os segmentos gerados no painel inferior;
4. O caso de utilização termina.

### 3.3.3 Segmentação

Como já foi referido parte deste módulo consiste na segmentação do vídeo carregado no sistema. De uma forma geral, a segmentação consiste no agrupamento de *frames* de um vídeo tendo em conta a sua semelhança ao nível de características visuais. Estes agrupamentos de *frames* irão gerar os planos que são classificados nas fases seguintes deste módulo. O algoritmo de segmentação deteta as *frames* de transição, ou seja, as *frames* onde existe transição de plano.

As transições de plano normalmente são de dois tipos: transições abruptas ou graduais. As transições abruptas são mais simples de detetar pois normalmente quando ocorrem existe uma grande diferença nas características de duas *frames* consecutivas ao contrário das transições graduais que devido à sua natureza implicam a ausência de *frames* consecutivas com diferenças acentuadas.

O método de deteção de transições utilizado é semelhante ao implementado por Hong Shao et al. em [29]. Neste método em primeiro lugar todas as *frames* são convertidas no espaço de cor HSV. Em seguida os valores de HSV são quantificados, sendo a matiz (*Hue*) quantificada em 16 níveis e a saturação (*Saturation*) e o valor (*Value*) quantificados em 8 níveis. Esta diferença nos níveis de quantificação deve-se ao facto de o olho humano ser mais sensível à matiz do que aos restantes valores.

Esta quantificação vai resultar num *array* de dimensões 16x8x8. Depois da quantificação o *array* resultante é convertido num *array* de de 1024 elementos. Este

processo é efetuado para todos os *frames* e em seguida é calculada a distância *Chi-square* (utilizando a equação 3.1 onde H representa o histograma gerado anteriormente) entre *frames* adjacentes.

$$d_j(H_j, H_{j+1}) = \sum_{i=1}^{1024} \frac{(H_j(i) - H_{j+1}(i))^2}{H_j(i) + H_{j+1}(i)} \quad (3.1)$$

Onde esta distância for maior que um limite calculado dinamicamente utilizando a média e o desvio padrão de todas as distâncias entre *frames*, é definida uma transição abrupta.

A detecção de transições graduais ocorre utilizando exatamente o mesmo método da detecção de transições abruptas, mas neste caso este é apenas aplicado de 15 em 15 *frames*. Isto faz com que eventuais transições, onde as diferenças entre *frames* consecutivas não sejam acentuadas, sejam também consideradas.

Por fim é efetuada uma união das transições abruptas e graduais sendo assim obtidas todas as *frames* de transição do vídeo, o que o permite segmentá-lo em planos.

### 3.3.4 Classificação de Planos

#### 3.3.4.1 Classificação do Tipo de Plano

Depois da obtenção dos planos, é necessário determinar qual o tipo de cada um. Este processo é efetuado utilizando em cada *frame* de um plano a técnica *Golden Region* para obter apenas a zona central dessa *frame* e depois converter para o espaço de cor HSV, aplicar uma máscara da cor verde (os limites da cor verde no espaço de cor HSV são obtidos pelo sistema ao mostrar ao utilizador uma *frame* para que este selecione uma área referente ao relvado), aplicar uma operação morfológica de abertura (para remoção do ruído) e contar a percentagem de pixels brancos presentes (o resultado da aplicação da máscara faz com que os pixels da cor verde apareçam a branco, sendo assim possível obter os pixels pertencentes ao relvado, como se pode ver na figura 3.10).

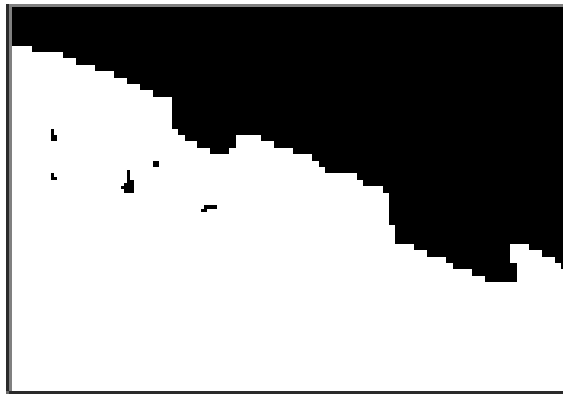


Figura 3.10: Exemplo de aplicação de máscara a plano longo.

Depois desta contagem é calculada a percentagem destes pixels quando em comparação com todos os pixels da zona central e é assim obtida a percentagem de relvado presente numa *frame*. Por fim o valor de percentagem é comparado com limites definidos experimentalmente e de acordo com o resultado a *frame* é declarada do tipo longo, médio ou próximo/fora de campo. O plano é do tipo predominante das *frames* que o compõem (se a maioria das *frames* pertencentes ao plano são do tipo longo, o plano é declarado como sendo do tipo longo).

#### 3.3.4.2 Detecção de Grande Área

A deteção de grande área é implementada de forma semelhante ao que acontece em [15] com Ekin, A. e Tekalp. M. . Esta deteção parte do princípio de que quando a grande área se encontra visível num plano, existem nesse plano 3 linhas de campo paralelas: a linha de final de campo, uma linha da pequena área e uma linha da grande área. Para isso é necessário extrair as linhas de uma *frame* e verificar se existem no mínimo 3 linhas paralelas entre si com uma distância pré-definida. Esta verificação é apenas aplicada a todas *frames* de planos identificados como planos do tipo longo visto que é sempre neste tipo de planos que as linhas estão completamente visíveis.

Em primeiro lugar é necessário utilizar uma máscara da cor verde (os limites da cor verde são definidos da mesma forma que na classificação de planos) para eliminar todos os pixels desta cor restando assim apenas os pixels relativos às linhas de campo. Uma vez que esta operação raramente produz resultados satisfatórios é necessário proceder à realização de operações morfológicas para a remoção do ruído como a operação de abertura (esta operação pode ser definida matematicamente utilizando a equação 3.2 onde  $\ominus$  representa a operação de erosão e  $\oplus$  representa a operação de dilatação).

$$A \circ B = (A \ominus B) \oplus B \quad (3.2)$$

A operação de abertura pode ser definida como a aplicação da operação de erosão seguida da aplicação da operação de dilatação aos pixels de uma imagem (exemplo de resultado pode ser visualizado na Figura 3.11).



Figura 3.11: Exemplo da aplicação da operação de abertura.

Depois de efetuada a operação de abertura, a imagem resultante é convertida numa imagem em níveis de cinza. Após esta conversão é-lhe aplicado um limite adaptativo com o objetivo de gerar uma imagem binária à qual é depois aplicada uma transformação de *Hough*. A transformação de *Hough* é um método de extração de características utilizada no processamento de imagem. Esta técnica permite encontrar objetos pertencentes a certas classes de formas geométricas (a versão inicial desta transformação apenas permitia detetar linhas mas entretanto foi estendida para poder detetar outras formas geométricas) através de um procedimento de votação. O procedimento de votação é executado numa representação paramétrica de onde os objetos candidatos são obtidos como máximos locais de um acumulador [28].

Depois de aplicada a cada *frame* do plano a transformação de *Hough* para a extração das linhas é verificado também se existem pelo menos três linhas paralelas entre si e em caso afirmativo é declarada a existência de grande área nessa *frame*. Por fim, se num plano existir um número considerável de *frames* em que seja detetada a grande área, é declarada a presença de grande área nesse plano. O fluxo normal de processamento de um plano pode ser visualizado na Figura 3.12.

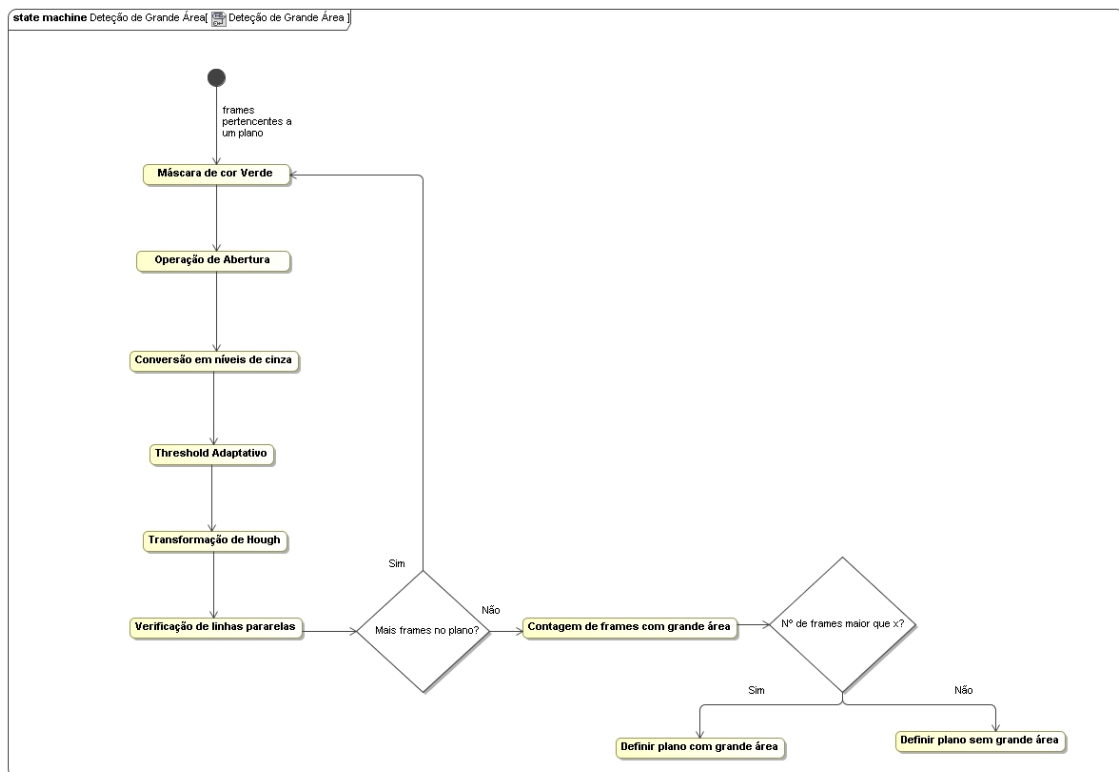


Figura 3.12: Fluxo da detecção de grande área aplicada a um plano.

### 3.3.4.3 Detecção de Cartões

O processo de detecção de cartões é implementado de forma semelhante ao implementado por Kolekar M. e Segupta S. em [17]. O processo de detecção de cartões parte do princípio que quando é amostrado por parte do árbitro um cartão, este aparece sempre presente na parte superior do plano ao centro. Posto isto, é necessário dividir uma *frame* por forma a obter apenas a parte superior da mesma e aplicar filtros da cor amarela e vermelha que permitam determinar se existem áreas consideráveis de pixels desta cor presentes.

De uma forma geral, o processo tem o seguinte fluxo:

1. As *frames* pertencentes a planos identificados como planos do tipo próximo são divididas em 16 blocos (4 colunas e 4 linhas);
2. Desses 16 blocos apenas são considerados os 2 blocos centrais do topo da imagem;
3. A estes dois blocos são aplicadas máscaras de cor vermelha e amarela, o que vai resultar em imagens onde a única cor existente são estas duas cores;

- Por último é verificado se existe alguma concentração considerável de pixels destas cores. Se existir alguma destas concentrações e esta constituir um polígono com 4 vértices, é calculada a área deste polígono e comparada com um limite pré-definido (determinado experimentalmente). Caso se enquadre é declarada a existência de cartão nessa *frame* (amarelo ou vermelho, dependente do resultado);

Depois de processadas todas as *frames* de um plano, se nele existirem um número de *frames* considerável em que tenha sido declarada a existência de um cartão, é definida a existência de um cartão nesse plano. Este processamento é aplicado depois a todos os planos do tipo próximo/fora de campo de um vídeo. O fluxo normal de processamento de um plano pode ser visualizado na Figura 3.13.

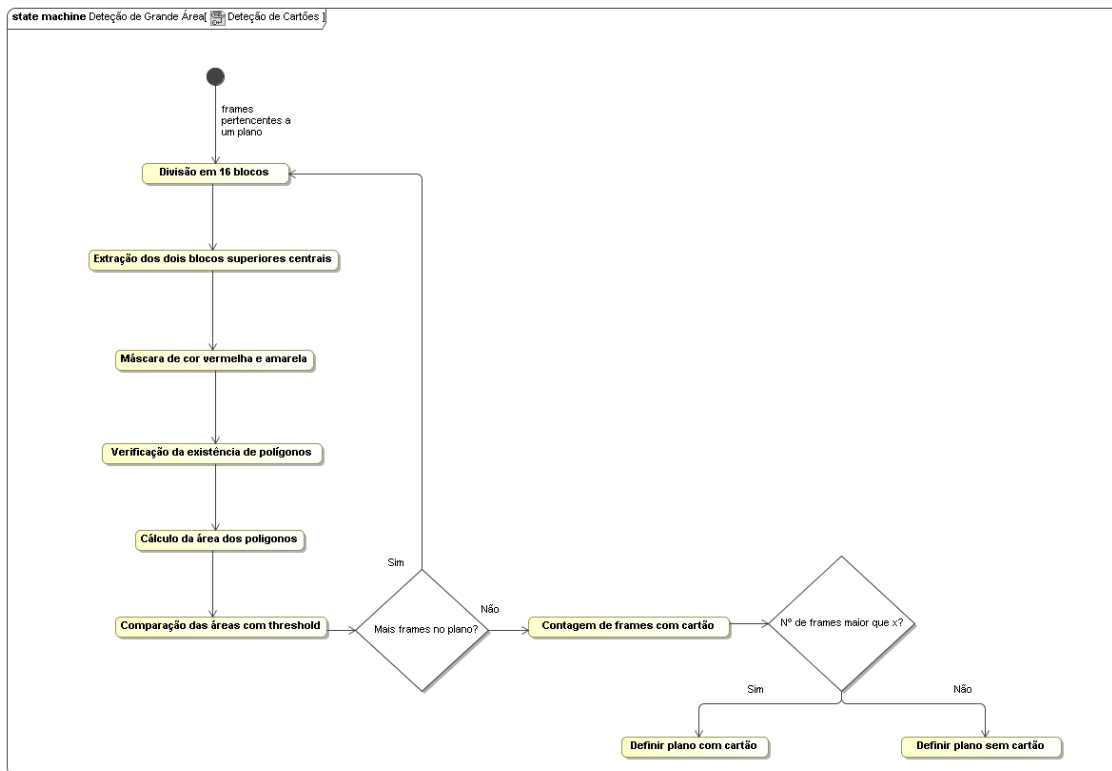


Figura 3.13: Fluxo da detecção de cartões aplicada a um plano.

#### 3.3.4.4 Support-Vector Machine

Visto que este sistema utiliza um algoritmo SVM para classificar os dados gerados pela aplicação, torna-se necessário oferecer uma pequena contextualização no que toca ao funcionamento deste algoritmo.

O processo de classificação utilizando as características HRV e as emoções anotadas é implementado utilizando um SVM. Um SVM é um algoritmo de aprendizagem automática supervisionado, ou seja, é um tipo de aprendizagem em que são analisados dados de treino (que consistem nos dados de *input* e nas classes desejadas relativos a esses *inputs*) e produz a função de associação que pode ser utilizada para associar dados novos nunca recebidos pelo algoritmo. Num cenário ótimo o classificador depois de treinado é sempre capaz de associar novos dados à classe correta [44].

Num SVM cada *input* passado ao algoritmo é representado por um ponto num espaço n-dimensional (onde  $n$  representa o número de características) com o valor de cada característica a representar as coordenadas nesse espaço. Os vetores de suporte neste algoritmo são os pontos de cada classe neste espaço que servirão como pontos de "fronteira" com outras classes (Figura 3.14). A classificação é efetuada ao encontrar o hiperplano que melhor segrega as classes (o hiperplano que melhor segrega as classes é aquele que maximiza a distância possível entre si próprio e todos os vetores de suporte como é possível verificar na Figura ??). Se existirem duas classes é necessário encontrar a linha que melhor separa estas classes mas no caso de existirem múltiplas classes, é necessário encontrar um hiperplano n-dimensional [47].

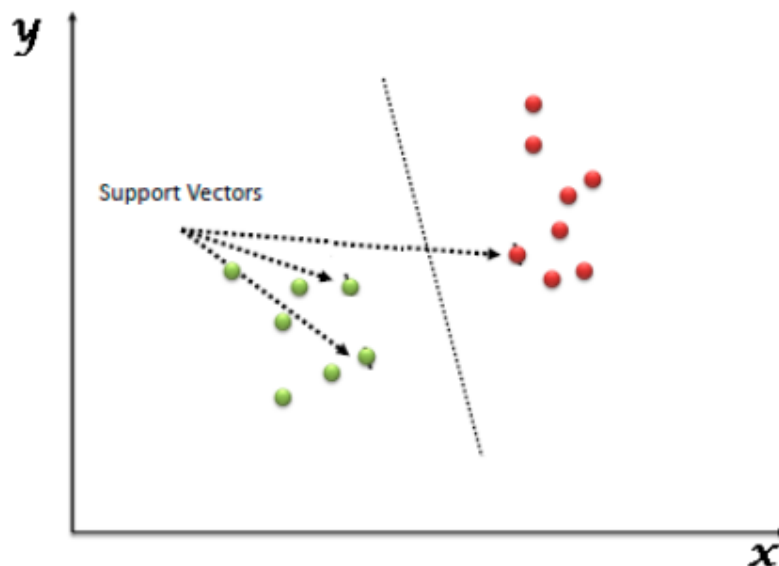


Figura 3.14: Exemplo SVM.

No caso desta dissertação, é implementado um algoritmo SVM cujo treino consiste na receção um ficheiro com dados de treino constituído por várias linhas

sendo que cada linha contem a classe (relevante ou não relevante) e os *inputs* que desejamos associar a essa classe (os *inputs* são constituídos pelas características HRV e pelas emoções anotadas). Depois de treinado o algoritmo é possível passar ao mesmo combinações de valores de características HRV e anotações de emoções nunca vistos anteriormente e obter a classe ao qual pertencem estes valores (num caso ótimo devolve sempre a classe correta).

## 3.4 Firebase

O *Firebase* é uma plataforma de desenvolvimento que providencia aos *developers* várias ferramentas e serviços para a implementação de aplicações *web* e aplicações móveis.

Todas as funcionalidades do *Firebase* podem ser visualizadas na figura 3.15.



Figura 3.15: Funcionalidades do *Firebase*.

Destas funcionalidades as que são utilizadas nesta dissertação são: a base de dados em tempo real (sincroniza em tempo real a todos os utilizadores ligados), a autenticação e o armazenamento multimédia.

### 3.4.1 Base de dados em tempo real

A base de dados em tempo real é uma base de dados NoSQL (não-relacional, ou seja, é modelada de forma diferente das relações tabulares utilizadas nas bases de dados relacionais) alojada na *cloud* que permite o armazenamento e sincronização de dados entre vários utilizadores em tempo real (ou seja, uma alteração à base de

dados por determinado utilizador atualiza os dados imediatamente para todos os utilizadores). A informação é guardada numa árvore de valores como pode ser evidenciado na figura 3.16.



Figura 3.16: Árvore de Dados do Firebase.

Uma das maiores vantagens desta base de dados é a integração de Software Development Kit (SDK) para aplicações móveis e de *web* que permitem a construção de aplicações sem a necessidade de *hardware* dedicado. Quando os utilizadores ficam sem ligação à Internet, estes SDKs permitem que a informação seja guardada em *cache* e depois carregada na base de dados quando a ligação à Internet é reestabelecida.

### 3.4.2 Autenticação

A autenticação do *Firebase* providencia um conjunto de ferramentas e serviços que permitem a abstração de todo o processo de implementação e manutenção de um sistema de autenticação seguro. Isto faz com que todo o processo seja implementado utilizando poucas linhas de código e também com que operações complexas como a união de contas sejam executadas de forma simples.

A autenticação *Firebase* permite a utilização de várias formas de autenticação, entre as quais a autenticação tradicional em que se utiliza um email e uma password (é a utilizada nesta dissertação), através do número de telefone ou através de redes sociais (*Google*, *Facebook* ou *Twitter*).

Uma vantagem da utilização da autenticação *Firebase* é o facto desta ter sido desenvolvida por uma empresa com vasta experiência nesta área, como é o caso da *Google*.

### 3.4.3 Armazenamento *Firestore*

O *Firestore* providencia também uma solução para o armazenamento de conteúdo multimédia. Esta é uma solução *standalone* que permite o carregamento de conteúdos gerados pelo utilizador a partir de dispositivos móveis e de *web browsers*.

O armazenamento do *Firestore* utiliza um simples sistema de pastas para estruturar os ficheiros carregados.

# 4

## Implementação do Sistema

Neste capítulo é aprofundada a solução para cada uma das propostas apresentadas nos capítulos anteriores. São apresentados com especificidade os vários diagramas de classes implementados para cada um dos módulos do sistema bem como apresentados e explicados alguns segmentos de código considerados relevantes na construção dos mesmos.

Aqui ao contrário do que ocorre no capítulo anterior as questões relacionadas com a implementação já não são abordadas de uma forma abstrata mas sim de uma forma mais técnica e dependente das ferramentas utilizadas. São também abordadas questões mais técnicas relativas ao sistema utilizado para base de dados e armazenamento de conteúdos multimédia, o *Firebase*.

### 4.1 Aplicação de captura informação fisiológica e anotação de emoções

#### 4.1.1 Diagrama de Classes

Para providenciar uma ideia geral do código pertencente aos vários módulos da aplicação (modelo, controladores e acesso a dados) são apresentados separadamente por módulo os vários diagramas de classes bem como dada uma explicação do propósito de cada classe utilizada.

As classes implementadas no contexto dos controladores são (Figura 4.1):

- **SplashscreenActivity** - Responsável por carregar em memória os dados presentes na base de dados *Firebase* no arranque da aplicação. É também responsável por reproduzir uma animação cada vez que a aplicação é aberta, efetuando depois a transição para o ecrã de *login*;
- **LoginActivity** - Responsável por receber *inputs* do utilizador relativos ao ecrã de *login* (neste caso são a introdução do texto relativo ao email e password e o pressionar do botão de *login* ou de transição para o ecrã de registo) e produzir as ações correspondentes. É também responsável por realizar o processo de *login* através da funcionalidade de autenticação do *Firebase*;
- **RegisterActivity** - Responsável por receber *inputs* do utilizador relativos ao ecrã de registo (neste caso são a introdução do nome, do aniversário, do email, da *password*, do género, do clube e da seleção nacional bem como o pressionar do botão de registo) e produzir as ações correspondentes. É também responsável por realizar o processo de validação das informações introduzidas e realizar o processo de registo através da funcionalidade de autenticação do *Firebase*;
- **MainMenuActivity** - Responsável por receber *inputs* do utilizador relativos ao ecrã de menu principal (neste caso é um dos botões pressionados pelo utilizador) e efetuar a transição para o ecrã correspondente ao botão pressionado;
- **MyRecordingsActivity** - Responsável por mostrar todas as gravações efetuadas pelo utilizador com *login* efetuado;
- **ScanActivity** - Responsável por receber *inputs* do utilizador (neste caso é o pressionar do botão de *scan*) e mostrar todos os dispositivos detetados através do *Bluetooth*. É também responsável por utilizar o módulo do *Movesense* para efetuar o *scan*;
- **SelectClubsActivity** - Responsável por receber *inputs* do utilizador (neste caso são a seleção do clube da casa e do clube visitante e o pressionar do botão de confirmação) e transitar para o ecrã de gravação;
- **RecordActivity** - Responsável por receber *inputs* do utilizador relativos ao ecrã de gravação (neste caso são o pressionar dos vários botões relativos ao processo de gravação e dos botões de definição de emoções) e realizar todas



- **FirestoreConf** - Permite obter as referências para a base de dados, autenticação e armazenamento do sistema *Firestore*;
- **FirestoreMultiQuery** - Permite efetuar múltiplas *queries* à base de dados *Firestore* e esperar pelo resultado de todas (é útil devido ao facto de as chamadas à base de dados no *Firestore* serem assíncronas, ou seja, o código restante resume mesmo não sendo obtida a informação resultante da *query*, o que por vezes não é o pretendido. Exemplo disto é o caso do carregamento de dados no *Splash Screen* em que é pretendido que todos os dados sejam carregados antes da aplicação transitar para o próximo ecrã);
- **FirestoreStorageManager** - Responsável pela gestão dos ficheiros multimédia presentes no armazenamento do sistema *Firestore*. Permite efetuar o *upload* e *download* de ficheiros.

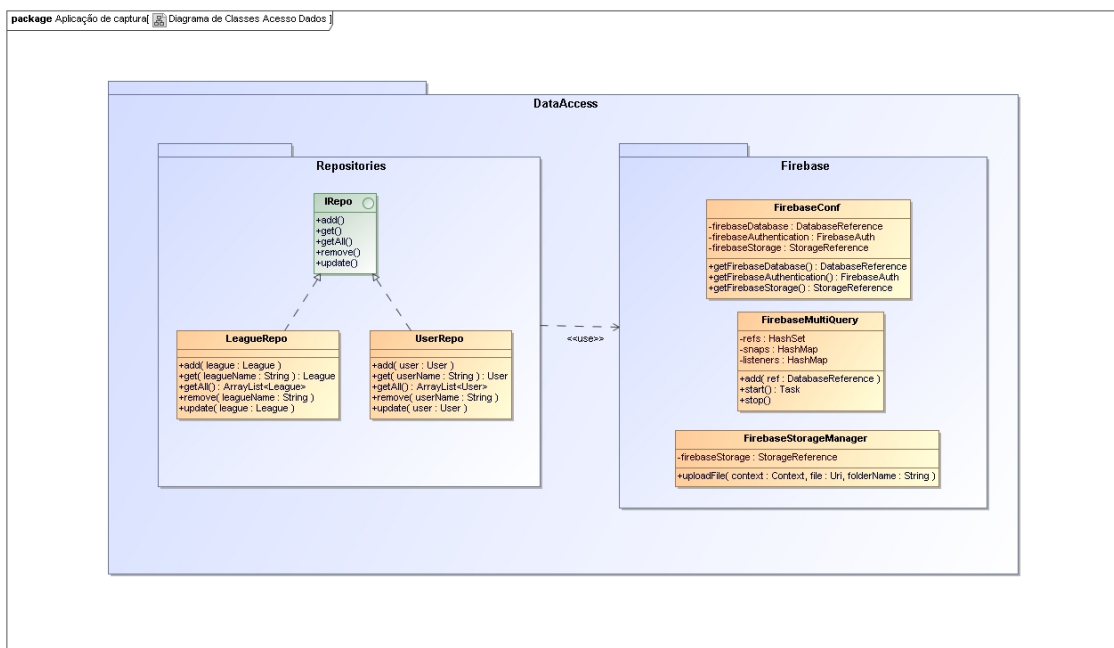


Figura 4.2: Diagrama de Classes Acesso a Dados.

As classes implementadas no contexto do modelo são (Figura 4.3):

- **Team** - Entidade que representa uma equipa;
- **User** - Entidade que representa um utilizador. Um utilizador possui duas equipas, uma local e uma nacional;
- **League** - Entidade que representa uma liga. Uma liga possui várias equipas;

- **LeagueService** - Serviço que oferece várias funcionalidades relacionadas com a entidade Liga permitindo aos controladores comunicar com o acesso a dados;
- **UserService** - Serviço que oferece várias funcionalidades relacionadas com a entidade Utilizador permitindo aos controladores comunicar com o acesso a dados;

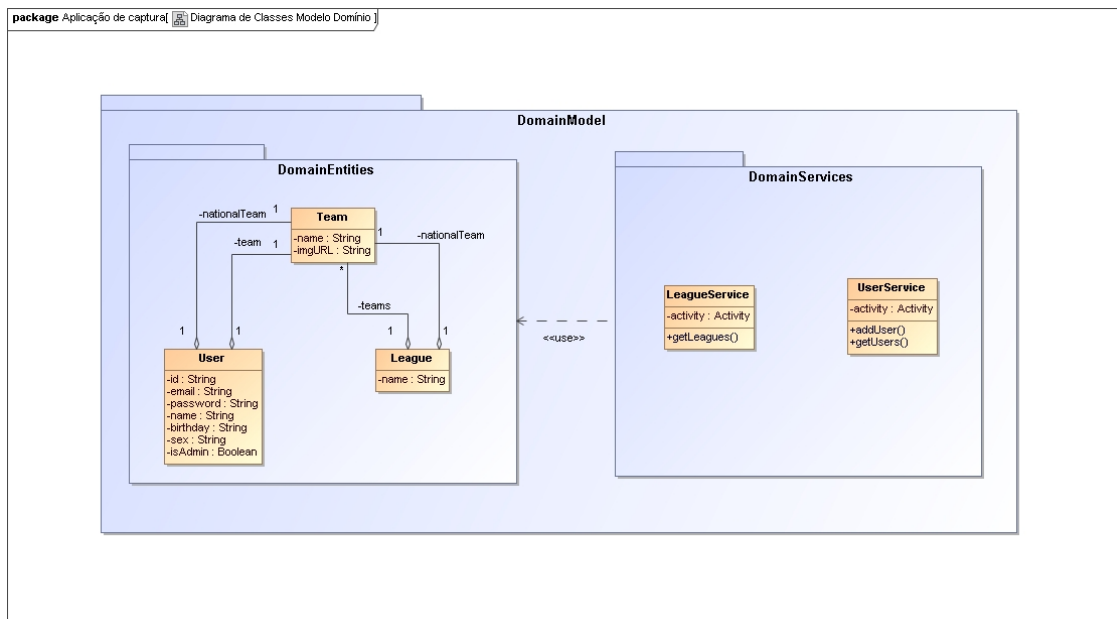


Figura 4.3: Diagrama de Classes Modelo.

### 4.1.2 Serviço de Gravação de Dados Fisiológicos

O objetivo principal da aplicação é a gravação da informação de intervalos RR recebida pelo sensor de medição cardíaca. O processo de gravação é implementado como um serviço na classe "RecordingService" e é lançado pela atividade "RecordActivity".

Na listagem 4.1 encontra-se a implementação da função que permite iniciar o processo de gravação dos dados fisiológicos. Em primeiro lugar é efetuada a subscrição ao sensor por forma a receber a informação enviada periodicamente pelo mesmo atribuindo também um *listener* de notificações que permite executar uma ação sempre que é recebida informação (linhas 2 a 5). Sempre que é recebida uma notificação proveniente do sensor os dados são desserializados (linha 8), é obtido

o tempo corrente (linha 13) e é gravado no ficheiro CSV de dados fisiológicos o tempo seguido do valor de intervalo RR recebido.

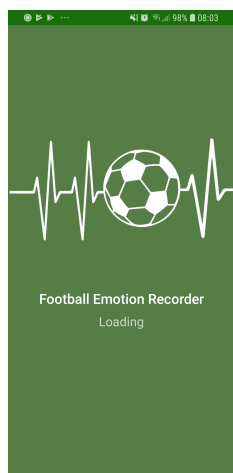
```
1 public void startRecordingProcess() {
2     mdsSubscription = Mds.builder().build(this).subscribe(URI_
3         EVENTLISTENER,
4         formatContractToJson(MovesenseConnectedDevices.getConnectedDevice(0)
5             .getSerial(),
6             HEART_RATE_PATH)
7         , new MdsNotificationListener() {
8             @Override
9             public void onNotification(String data) {
10                HeartRate heartRate = new Gson().fromJson(data, HeartRate.class);
11
12                if (heartRate != null) {
13                    lastRRData = heartRate.body.rrData[0];
14
15                    String time = stopwatch.getElapsedTime();
16
17                    String[] rrData = new String[]{time, Integer.toString(
18                        lastRRData)};
19                    try{
20                        writerRRData.writeNext(rrData);
21                    }
22                    catch(Exception e){
23
24                    }
25                }
26            }
27        });
28 }
```

Listagens 4.1: Início do processo de gravação

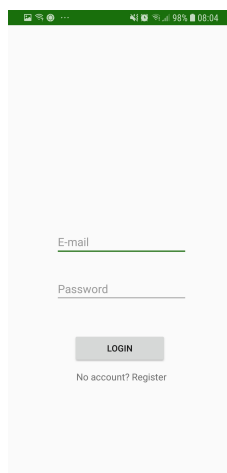
### 4.1.3 Graphical User Interface

A GUI da aplicação de gravação de características fisiológicas é composta por 8 ecrãs independentes. Estes ecrãs são:

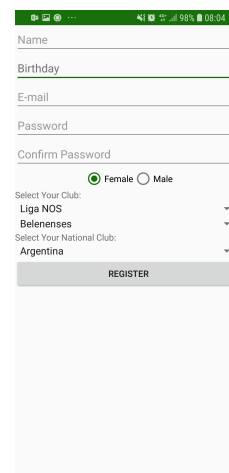
- **Splashscreen** - Ecrã que mostra uma pequena animação no arranque da aplicação. Tem o propósito de mostrar uma animação enquanto os dados presentes no *Firebase* são carregados para a memória do telemóvel (Figura ??);
- **Ecrã de Login** - Ecrã que mostra duas caixas de texto para a introdução de nome de utilizador e *password* e dois botões: um de *login* e outro para passar ao ecrã de registo (Figura ??);
- **Ecrã de Registo** - Ecrã que mostra 5 caixas de texto para a introdução de nome, aniversário, email, *password* e confirmação de *password*. Apresenta também dois *radio buttons* que permitem escolher o género assim como 3 *spinners* que permitem escolher a equipa e a seleção nacional do utilizador. Existe também um botão que permite efetuar o registo do utilizador (Figura ??);
- **Ecrã de Menu Principal** - Ecrã que apresenta vários botões que permitem passar ao ecrã de *scan*, ao ecrã de listagem de gravações e efetuar *logout* (Figura ??);
- **Ecrã de Listagem de Gravações** - Ecrã que apresenta uma listagem das gravações efetuadas (Figura ??);
- **Ecrã de Scan** - Ecrã que apresenta os vários dispositivos detetados após o utilizador premir o botão implementado para o efeito (Figura ??);
- **Ecrã de Seleção de Equipas** - Ecrã que possui 4 *spinners* que permitem a seleção dos clube da casa e do clube visitante (Figura ??);
- **Ecrã de Gravação** - Ecrã principal da aplicação. Permite efetuar a gravação dos dados fisiológicos e a anotação de emoções. Possui 2 botões que permitem controlar o fluxo de gravação e 3 botões que permitem a definição de emoções. Para definir a emoção pretendida o utilizador prime o botão correspondente e arrasta sem soltar até que apareça a intensidade pretendida, soltando em seguida para efetuar a anotação (Figura ??).



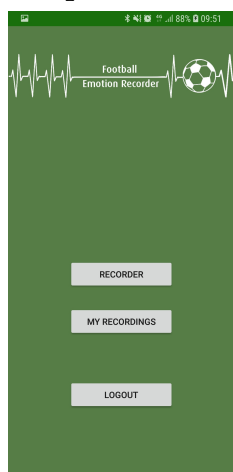
(a) Splashscreen.



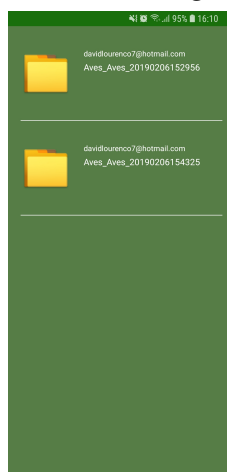
(b) Ecrã de *Login*.



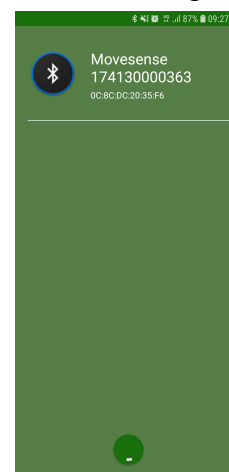
(c) Ecrã de Registo.



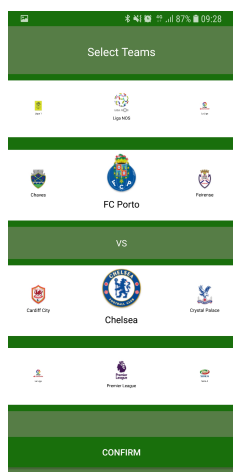
(d) Ecrã de Menu Principal.



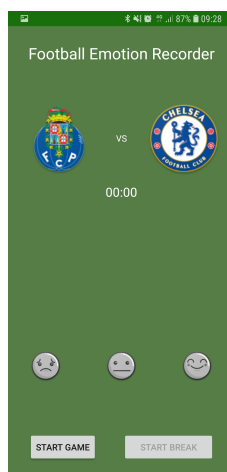
(e) Ecrã de Listagem de Gravações.



(f) Ecrã de *Scan*.



(g) Ecrã de Seleção de Equipas.



(h) Ecrã de Gravação.

Figura 4.4: Interface da aplicação.

## 4.2 Módulo de geração de características HRV

### 4.2.1 Diagrama de Classes

As classes implementadas são:

- **HrvGenerator** - Responsável por implementar uma aplicação de consola que permite ao utilizador selecionar a opção de descarregar os ficheiros do *Firebase* ou de processar os ficheiros descarregados por forma a gerar as características HRV;
- **FirebaseTools** - Contém as várias funções relativas à interação com o sistema *Firebase*, tanto de autenticação como de *download* de ficheiros;
- **HRV** - Responsável por receber a informação de intervalos RR e devolver as características HRV;
- **Util** - Classe que contém várias funções genéricas como a leitura de ficheiros CSV.

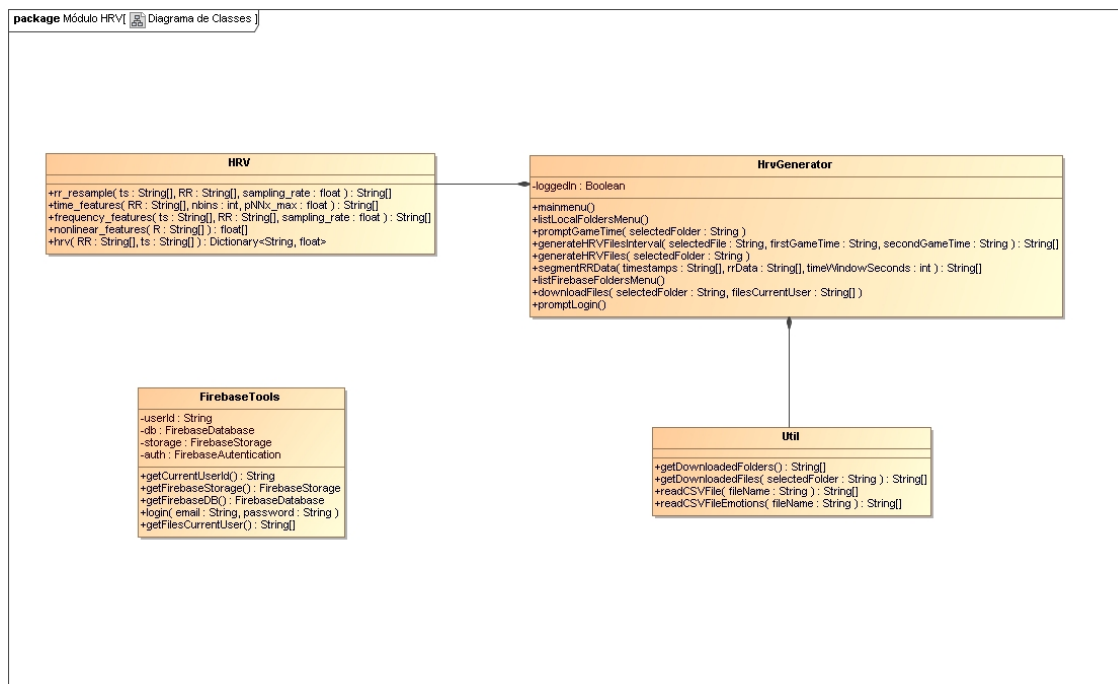


Figura 4.5: Diagrama de Classes HRV.

## 4.3 Módulo de análise de vídeos

### 4.3.1 Diagrama de Classes

As classes implementadas são:

- **InterfaceFinal** - Responsável por implementar a GUI de análise de vídeos onde é possível ao utilizador carregar um vídeo e correr os algoritmos de segmentação e classificação para gerar os segmentos relevantes;
- **InterfaceTeste** - Tem a mesma função que a interface final mas em vez de mostrar como resultado os segmentos relevantes mostra um conjunto de linhas com o resultado dos vários algoritmos envolvidos no processo;
- **SelectorVideo** - Responsável pelo carregamento de um vídeo utilizando o repositório;
- **Repositorio** - Responsável pela leitura de um vídeo e devolução das várias *frames* pertencentes ao vídeo;
- **RepositorioSegmentos** - Responsável pelas operações relacionadas com segmentos como a leitura, a união e o corte de segmentos;
- **Shot** - Entidade que representa um segmento de vídeo gerado pelo algoritmo de segmentação;
- **Segm** - Responsável pela implementação do algoritmo de segmentação;
- **BoxDetection** - Responsável pela implementação do algoritmo de deteção de grande área;
- **CardDetection** - Responsável pela implementação do algoritmo de deteção de cartões;
- **SVMClassification** - Responsável pela implementação do algoritmo SVM de classificação utilizando características HRV e emoções;
- **Audio** - Responsável pela implementação do algoritmo de classificação utilizando características auditivas;
- **Utils** - Classe que contém várias funções genéricas.

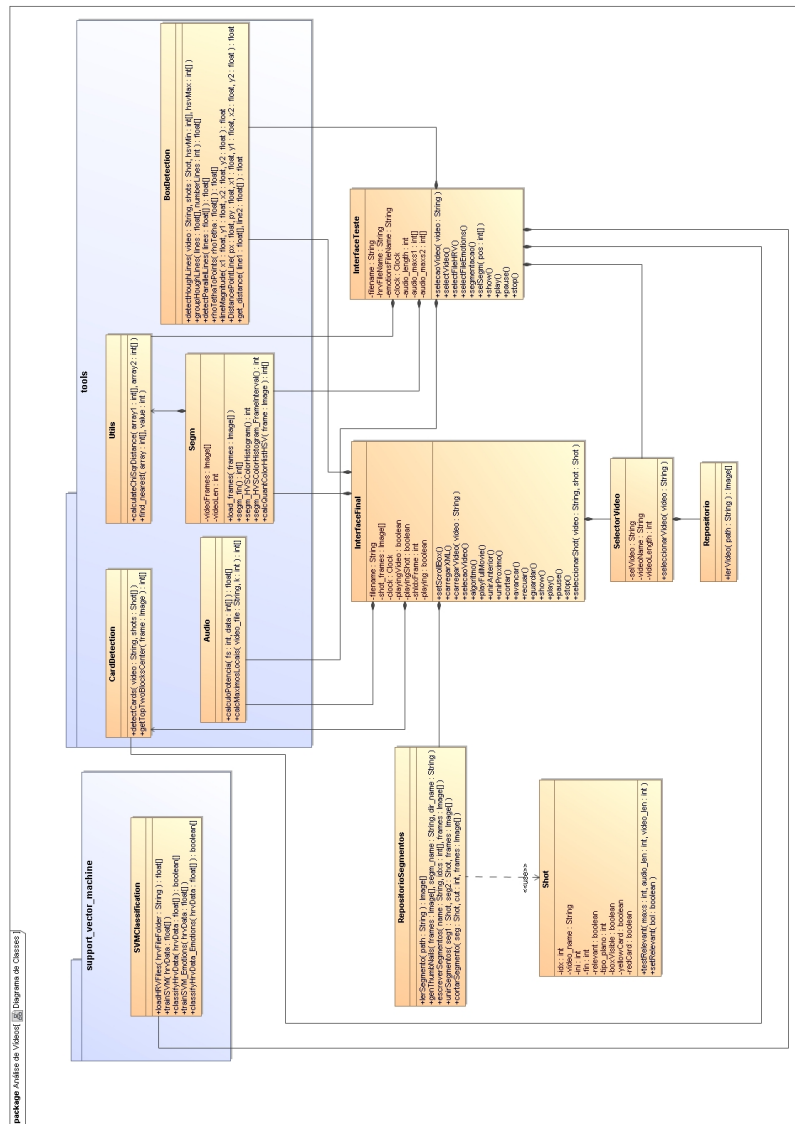


Figura 4.6: Diagrama de Classes Análise de Vídeos.

### 4.3.2 Segmentação

Como já verificado no capítulo 3, a segmentação é executada recorrendo ao cálculo das diferenças do histograma do espaço de cor HSV entre *frames*. Este algoritmo de deteção é executado duas vezes: uma aplicada a todas as *frames* para a deteção de transições abruptas e uma aplicada a um intervalo de 15 em 15 *frames* para deteção de transições graduais. No fim o resultado destas duas segmentações é unido para formar a segmentação final do vídeo.

Na listagem de código 4.2 podemos verificar a função implementada para a deteção de transições abruptas. Como é possível verificar nesta função as *frames*

consecutivas são convertidas para o espaço de cor HSV (linha 8), é calculado o histograma 1D (linha 9) e calculada a diferença *Chi-squared* para o histograma calculado relativo à *frame* anterior (linha 10). No fim deste processamento efetuado a todas as *frames* e obtidas as diferenças entre histogramas de *frames* consecutivas, é calculada a média (linha 16) e o desvio padrão (linha 17) destas diferenças que depois são utilizadas no cálculo do limite adaptativo (linha 20). Por último todas as diferenças calculadas são comparadas com o limite e nas *frames* onde a diferença for maior que este, é definida uma transição (linhas 22 a 25).

```

1 def segm_HVSColorHistogram(self):
2     frame0 = cv2.cvtColor(self.videoFrames[0], cv2.COLOR_BGR2HSV)
3     quantColorHist0 = self.calcQuantColorHistHSV(frame0)
4
5     diffs = [0,]
6     for frame in self.videoFrames[1:]:
7
8         frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
9         quantColorHist = self.calcQuantColorHistHSV(frame)
10        chiSqrDistance = self.utils.calculateChiSqrDistance(quantColorHist0,
11                                                             quantColorHist)
12
13        diffs.append(chiSqrDistance)
14
15        quantColorHist0 = quantColorHist
16
17        mean = sum(diffs) / len(diffs)
18        variance = np.std(diffs)
19
20        S = 4
21        threshold = mean + S * variance
22
23        trans = np.zeros(self.videoLen)
24        for i in range(len(diffs)):
25            if diffs[i] >= threshold:
26                trans[i] = 1
27
28        T_HVSColorHistogram = [i for i in range(0, len(trans)) if trans[i] ==
29                               1]
30
31        return T_HVSColorHistogram

```

Listagens 4.2: Segmentação por diferenças de histograma HSV frame a frame

A detecção de transições graduais é implementada de uma forma semelhante à detecção de transições abruptas (a implementação pode ser verificada na listagem de código 4.3). Mas neste caso em vez de o processamento de cálculo das diferenças ocorrer entre todas as *frames* consecutivas do vídeo, apenas ocorre entre *frames* de um determinado intervalo. Outra diferença é a que o limite neste caso é estático e é definido experimentalmente.

```

1 def segm_HVSColorHistogram_FrameInterval(self):
2     frame0 = cv2.cvtColor(self.videoFrames[0], cv2.COLOR_BGR2HSV)
3     quantColorHist0 = self.calcQuantColorHistHSV(frame0)
4
5     diffs = {0: 0}
6     for i in range(1, len(self.videoFrames)):
7         if i % 15 == 0:
8             frame = cv2.cvtColor(self.videoFrames[i], cv2.COLOR_BGR2HSV)
9             quantColorHist = self.calcQuantColorHistHSV(frame)
10            chiSqrDistance = self.utils.calculateChiSqrDistance(
11                quantColorHist0, quantColorHist)
12
13            diffs[i] = chiSqrDistance
14
15            quantColorHist0 = quantColorHist
16
17            threshold = 45000
18
19            trans = np.zeros(self.videoLen)
20            for i in diffs.keys():
21                if diffs[i] >= threshold:
22                    trans[i-7] = 1
23
24            T_HVSColorHistogram_FrameInterval = [i for i in range(0, len(trans))
25                if trans[i] == 1]
26
27            return T_HVSColorHistogram_FrameInterval

```

Listagens 4.3: Segmentação por diferenças de histograma HSV intervalo frames

Depois de obtidas as transições resultantes dos dois processos de segmentação é necessário remover as transições duplicadas. Para isso é verificado se entre os resultados dos dois processos, existem alguns com uma distância (em *frames*) entre eles inferior a 120 *frames* e em caso afirmativo, uma destas é eliminada. Isto resulta na exclusão de transições demasiado próximas (como é possível verificar na listagem de código 4.4, nas linhas 5 a 9).

```

1 def segm_fin(self):
2     T_HSVColorHistogram = self.segm_HVSCColorHistogram()
3     T_HSVColorHistogramFrameInterval = self.segm_HVSCColorHistogram_
4         FrameInterval()
5
6     for i in T_HSVColorHistogramFrameInterval:
7         nearest = self.utils.find_nearest(T_HSVColorHistogram, i)
8         dif = abs(i - nearest)
9         if dif < 120:
10            T_HSVColorHistogramFrameInterval.remove(i)
11
12     T_Final = sorted(list(set(T_HSVColorHistogram + T_
13         HSVColorHistogramFrameInterval)))
14
15     Tnp = np.array(T_Final)
16
17     for i in Tnp:
18         idxRemove = np.where(np.logical_and(Tnp>i, Tnp<=i+15))[0]
19         if len(idxRemove) != 0:
20             Tnp = [i for j, i in enumerate(Tnp) if j not in idxRemove]
21
22     return Tnp, T_HSVColorHistogram, T_HSVColorHistogramFrameInterval

```

Listagens 4.4: Segmentação Final

### 4.3.3 Classificação do Tipo de Plano

Na listagem de código 4.5 podemos verificar a função implementada para a classificação dos tipos de planos. O processamento é aplicado a todas as *frames* de todos os planos resultantes do algoritmo de segmentação. Para cada *frame* de determinado segmento é em primeiro lugar extraída a região central desta utilizando o método *Golden Region* (linha 21) e em seguida é aplicada uma máscara da cor verde no espaço de cor HSV a esta região (linha 24). Em seguida é aplicada a transformação morfológica de abertura para a remoção de ruído da imagem (linha 27) e calculada a percentagem de pixels brancos presentes na imagem (linha 29). Por fim é efetuada a média destes valores para um plano (linha 31) e comparado com limites estáticos, definindo em seguida o tipo de plano (linhas 32 a 37. Tipo 0 é plano longo, tipo 1 é plano médio e tipo 2 é plano próximo/fora de campo).

O método recebe os parâmetros `hsvMin` e `hsvMax` que representam os limites

(inferior e superior, respetivamente) da cor verde no espaço HSV resultantes da seleção da área pertencente ao relvado (realizada pelo utilizador). Estes limites são utilizados na aplicação da máscara da cor verde para determinar quais os pixels pertencentes ao relvado.

```

1 def calcTipoPlano(video, segmentsIdx, hsvMin, hsvMax):
2     segmentos = []
3     lastIdx = 0
4     for idx in segmentsIdx:
5         segment = video[lastIdx:idx]
6         segmentos.append(segment)
7         lastIdx = idx
8     segmentos.append(video[lastIdx:-1])
9
10    size = (len(segmentos[0][0]), len(segmentos[0][0][0]))
11    x1 = int((1.0/4)*size[1])
12    y1 = int((1.0/4)*size[0])
13    x2 = size[1] - x1
14    y2 = size[0] - y1
15    grSize = (x2-x1, y2-y1)
16    tipo=[]
17    for segmento in segmentos:
18        percs = []
19
20        for frame in segmento:
21            Gr = frame[y1:y2, x1:x2]
22            hsvImg = cv2.cvtColor(Gr, cv2.COLOR_BGR2HSV)
23
24            bwImg = cv2.inRange(hsvImg, hsvMin, hsvMax)
25
26            kern = numpy.ones((3, 3), dtype=numpy.uint8)
27            bwImg = cv2.morphologyEx(bwImg, cv2.MORPH_OPEN, kern, iterations
28                                   =2)
29
30            per = ((cv2.countNonZero(bwImg)*100) / (grSize[0]*grSize[1]))
31            percs.append(per)
32        media = numpy.mean(percs)
33        if media <= 30:
34            tipo.append(2)
35        elif media <=70:
36            tipo.append(1)
37        else:
38            tipo.append(0)

```

```
39 return tipo
```

#### Listagens 4.5: Classificação do tipo de plano

### 4.3.4 Detecção de Grande Área

Como já referido no capítulo 3, a deteção da grande área depende da deteção das linhas da mesma. Na listagem de código 4.6 podemos verificar que o processamento de deteção de grande área é executado em todas as *frames* pertencentes a planos com tipo 0 (ou seja, tipo longo). Para cada *frame* é aplicada uma máscara da cor verde (linhas 13 a 16) e aplicada a operação morfológica de abertura (linha 19), um pouco como acontece com o algoritmo de classificação do tipo de planos. Depois a imagem é convertida numa escala de cinza (linha 21) e é aplicado um limite adaptativo para binarizar a imagem (linha 23). Em seguida é aplicada a transformação de *Hough* para a extração das linhas (linha 27), são agrupadas as linhas próximas umas das outras (linha 30) e por fim é verificado se existem pelo menos 3 linhas paralelas entre si (linha 32).

O método recebe os parâmetros `hsvMin` e `hsvMax` que representam os limites (inferior e superior, respetivamente) da cor verde no espaço HSV resultantes da seleção da área pertencente ao relvado (realizada pelo utilizador). Estes limites são utilizados na aplicação da máscara da cor verde para determinar quais os pixels pertencentes ao relvado.

```
1
2 def detectHoughLines(self, video, shots, hsvMin, hsvMax):
3     for shot in shots:
4         if shot.tipo_plano == 0:
5             segmento = video[shot.ini: shot.fin]
6             framesBoxVisible = 0
7             for frame in segmento:
8
9                 blur = cv2.GaussianBlur(frame, (3, 3), 1)
10
11                 hsvImg = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)
12
13                 mask = cv2.inRange(hsvImg, hsvMin, hsvMax)
14
15                 kern = np.ones((3, 3), dtype=np.uint8)
16                 mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kern, iterations=2)
```

```
17
18     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
19
20     gray = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_
21         C, cv2.THRESH_BINARY_INV, 5, 4)
22
23     gray = cv2.bitwise_and(gray, mask, mask=mask)
24
25     lines = cv2.HoughLines(gray, 1, np.pi / 180, 80)
26
27     if lines is not None:
28         lines = self.groupHoughLines(lines, 7)
29
30         parallelLines = self.detectParallelLines(lines)
31
32         threeParallels = [item for item in parallelLines if len(item)
33             >= 3]
34
35         if len(threeParallels) > 0:
36             framesBoxVisible += 1
37
38     boxVisible = False
39     if framesBoxVisible >= 1:
40         boxVisible = True
41     shot.boxVisible = boxVisible
```

Listagens 4.6: Detecção de grande área

Na figura 4.7 é possível verificar o funcionamento deste algoritmo, em que é efetuada a deteção das 3 linhas de fundo evidenciada pela sobreposição de 3 linhas vermelhas sobre as mesmas.

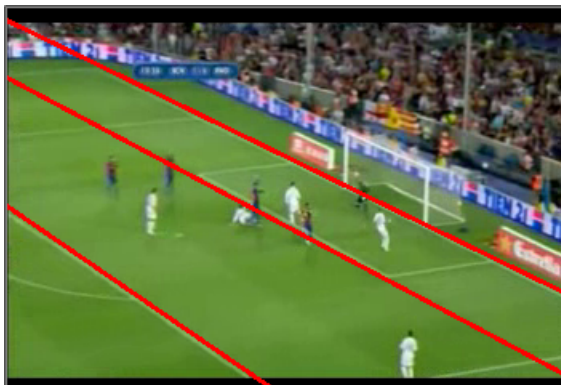


Figura 4.7: Funcionamento da deteção de grande área.

### 4.3.5 Detecção de Cartões

A detecção de cartões é efetuada para todas as *frames* pertencentes a planos do tipo 2 (planos do tipo próximos/fora de campo) como é possível verificar na linha 10. Cada *frame* é dividida em 16 blocos e são extraídos os 2 superiores ao centro (linha 15) e em seguida são convertidos para o espaço de cor HSV (linha 17). Em seguida são aplicadas máscaras da cor amarela e vermelha (linhas 18 e 33) e aplicada a operação morfológica de abertura (linhas 21 e 34). Para a cor vermelha e amarela são obtidos os contornos resultantes do processamento anterior (linhas 23 e 35) e em seguida é verificado se existem contornos com 4 vértices e com uma área compreendida entre dois valores pré-definidos e em caso afirmativo é declarada a existência de cartão de uma das cores nessa *frame* (linhas 24 a 31 e 36 a 42).

```
1 def detectCards(self, video, shots):
2
3     lower_yellow = np.array([13, 233, 199])
4     upper_yellow = np.array([53, 273, 299])
5
6     lower_red = np.array([163, 198, 155])
7     upper_red = np.array([193, 228, 245])
8
9     for shot in shots:
10        if shot.tipo_plano == 2:
11            segmento = video[shot.ini : shot.fin]
12            framesYellowCard = 0
13            framesRedCard = 0
14            for frame in segmento:
15                topTwoBlocksCenter = self.getTopTwoBlocksCenter(frame)
16
17                quarterHSV = cv2.cvtColor(topTwoBlocksCenter, cv2.COLOR_BGR2HSV)
18                bwImgYellow = cv2.inRange(quarterHSV, lower_yellow, upper_yellow)
19
20                kern = np.ones((4, 2), dtype=np.uint8)
21                bwImgYellow = cv2.morphologyEx(bwImgYellow, cv2.MORPH_OPEN, kern,
22                                                iterations=2)
23
24                imageYellow, cntYellow, hierarchyYellow = cv2.findContours(
25                    bwImgYellow, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
26                if len(cntYellow) == 1:
27                    for cnt in cntYellow:
28                        area = cv2.contourArea(cnt)
```

```

27     approx = cv2.approxPolyDP(cnt, 0.01 * cv2.arcLength(cnt, True)
28         , True)
29     if len(approx) == 4:
30         if (area > 34 and area < 37):
31             framesYellowCard += 1
32             break
33
34     bwImgRed = cv2.inRange(quarterHSV, lower_red, upper_red)
35     bwImgRed = cv2.morphologyEx(bwImgRed, cv2.MORPH_OPEN, kern,
36         iterations=2)
37     imageRed, cntRed, hierarchyRed = cv2.findContours(bwImgRed, cv2.
38         RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
39     if len(cntRed) == 1:
40         for cnt in cntRed:
41             area = cv2.contourArea(cnt)
42             approx = cv2.approxPolyDP(cnt, 0.01 * cv2.arcLength(cnt, True)
43                 , True)
44             if len(approx) == 4:
45                 if (area > 34 and area < 37):
46                     framesRedCard += 1
47
48     yellowCardPresent = False
49     if framesYellowCard >= 1:
50         yellowCardPresent = True
51
52     redCardPresent = False
53     if framesRedCard >= 1:
54         redCardPresent = True
55
56     if redCardPresent == True and yellowCardPresent == True:
57         redCardPresent = False
58         yellowCardPresent = False
59
60     shot.yellowCard = yellowCardPresent
61     shot.redCard = redCardPresent

```

Listagens 4.7: Detecção de Cartões

Na figura 4.8 podemos verificar o resultado do processamento do algoritmo de detecção de cartões em que a área onde aparece o cartão amarelo à esquerda se encontra representada por pixels brancos à direita.



Figura 4.8: Funcionamento da detecção de cartão.

### 4.3.6 Support-Vector Machine

No capítulo 1 verifica-se que as melhores características a utilizar no caso da utilização de janelas ultra-curtas são AVNN, RMSSD, pNN50, HF e LF/HF [46]. Na listagem 4.8 está presente um exemplo do ficheiro utilizado para treinar o classificador SVM onde as características utilizadas são HF, AVNN (referida na listagem como mNN) e RMSSD. Nesta listagem é possível também verificar que no ficheiro de treino está presente a classe a que pertencem os dados (classe 'y' implica situação relevante e classe 'n' implica situação não relevante). Estes dados são obtidos ao calcular as características HRV para trechos de um jogo onde ocorrem e onde não ocorrem situações relevantes.

```

1 relevant, HF, mNN, RMSSD
2 y, 0.000712, 0.048993, 0.065823
3 n, 0.000049, 0.002261, 0.023743

```

Listagens 4.8: Exemplo de dados de treino (valores apenas ilustrativos)

Na listagem de código 4.9 encontra-se a implementação da função que permite o treino do classificador SVM. A função recebe os dados de HRV presentes no ficheiro de dados de treino. Em primeiro lugar gera uma matriz onde o número de linhas é igual ao número de linhas do ficheiro e o número de colunas é igual ao número de características utilizadas que neste caso são 3 (linha 2). Em seguida é obtido um *array* (com o número de elementos igual ao número de linhas do ficheiro) onde os valores das várias classes no ficheiro são mapeados para 0 e 1 onde 0 representa 'n' e 1 representa 'y' (linha 3). Por fim o algoritmo é treinado passando a matriz de características e o *array* de classes (linhas 5 e 6).

```

1 def trainSVM(self, hrvData):
2     inputs_model = hrvData[['HF', 'RMSSD', 'mNN']].values
3     class_label = np.where(hrvData['relevant'] == 'n', 0, 1)
4
5     self.model = svm.SVC(kernel='linear')
6     self.model.fit(inputs_model, class_label)

```

## Listagens 4.9: Treino SVM

Depois de treinado o algoritmo é possível passar-lhe novos dados nunca vistos e obter as classes a quais estes novos dados pertencem. Os dados HRV a classificar são lidos dos ficheiros gerados pela aplicação de gravação e passados à função presente na listagem 4.10 e após treinar o algoritmo estes são classificados pelo mesmo (linhas 7 a 11). O resultado é um *array* binário representativo das classes às quais pertencem os dados (este *array* tem o mesmo número de linhas do conjunto de características recebidas).

```

1 def classifyHrvData(self, hrvData):
2     hrvFileFolderTrain = "./support_vector_machine/treino"
3     hrvDataTrain = self.loadHRVFiles(hrvFileFolderTrain)
4     self.trainSVM(hrvDataTrain)
5
6     relevants = []
7     for data in hrvData:
8         if (self.model.predict([data])) == 1:
9             relevants.append(True)
10        else:
11            relevants.append(False)
12    return relevants

```

## Listagens 4.10: Classificação SVM

Depois de implementado o classificador básico que apenas utiliza as características de HRV é também implementado um classificador semelhante mas que utiliza também as emoções anotadas pelo utilizador na aplicação de gravação.

Na listagem 4.11 é possível verificar que ao contrário do que acontece no ficheiro de dados de treino da versão apresentada anteriormente do classificador, aqui são adicionadas duas novas colunas: o tipo de emoção (0 para emoção negativa, 1 para emoção neutra e 2 para emoção positiva) e a intensidade da emoção (que vai de 1 a 5 no caso de emoções negativas e positivas e é sempre 1 no caso de emoção neutra).

```

1 relevant, HF, mNN, RMSSD, emotionType, emotionIntensity
2 y, 0.000712, 0.048993, 0.065823, 2, 5
3 n, 0.000049, 0.002261, 0.023743, 1, 1

```

Listagens 4.11: Exemplo de dados de treino com emoções (valores apenas ilustrativos)

Na implementação da função do treino do classificador com emoções (listagem de código 4.12) a única diferença é o número de colunas utilizadas, passarão de 3 para 5 com a adição de "emotionType" e "emotionIntensity" (linha 2).

```

1 def trainSVM_Emotions(self, hrvData):
2     inputs_model = hrvData[['HF', 'RMSSD', 'mNN', 'emotionType', '
        emotionIntensity']].values
3     class_label = np.where(hrvData['relevant'] == 'n', 0, 1)
4
5     self.model = svm.SVC(kernel='linear')
6     self.model.fit(inputs_model, class_label)

```

Listagens 4.12: Treino SVM com emoções

No caso da classificação com emoções (4.13) a única alteração é a pasta de onde é retirado o ficheiro (ou ficheiros) de treino do classificador (linha 2), tudo o resto é semelhante.

```

1 def classifyHrvData_Emotions(self, hrvData):
2     hrvFileFolderTrain = "./support_vector_machine/treino_emocao"
3     hrvDataTrain = self.loadHRVFiles(hrvFileFolderTrain)
4     self.trainSVM(hrvDataTrain)
5
6     relevants = []
7     for data in hrvData:
8         if (self.model.predict([data])) == 1:
9             relevants.append(True)
10        else:
11            relevants.append(False)
12    return relevants

```

Listagens 4.13: Classificação SVM com emoções

### 4.3.7 Graphical User Interface

Como já referido anteriormente a GUI já estava previamente implementada, apenas foram feitos alguns ajustes relativos à adição das novas características de classificação. Existe um ecrã normal (Figura 4.9) e um ecrã de testes (Figura 4.10). Ao ecrã normal são adicionados os vários botões que permitem a escolha do ficheiro de intervalos RR e o ficheiro de emoções. Também são adicionadas duas caixas de texto que permitem a determinação do tempo de jogo entre os quais queremos executar o processamento das características biométricas bem como dois *radio buttons* que permitem definir se estes tempos se referem à primeira parte do jogo ou à segunda.

Na interface de teste são também adicionados estes elementos e são adicionadas mais linhas que permitem verificar o funcionamento dos novos algoritmos.

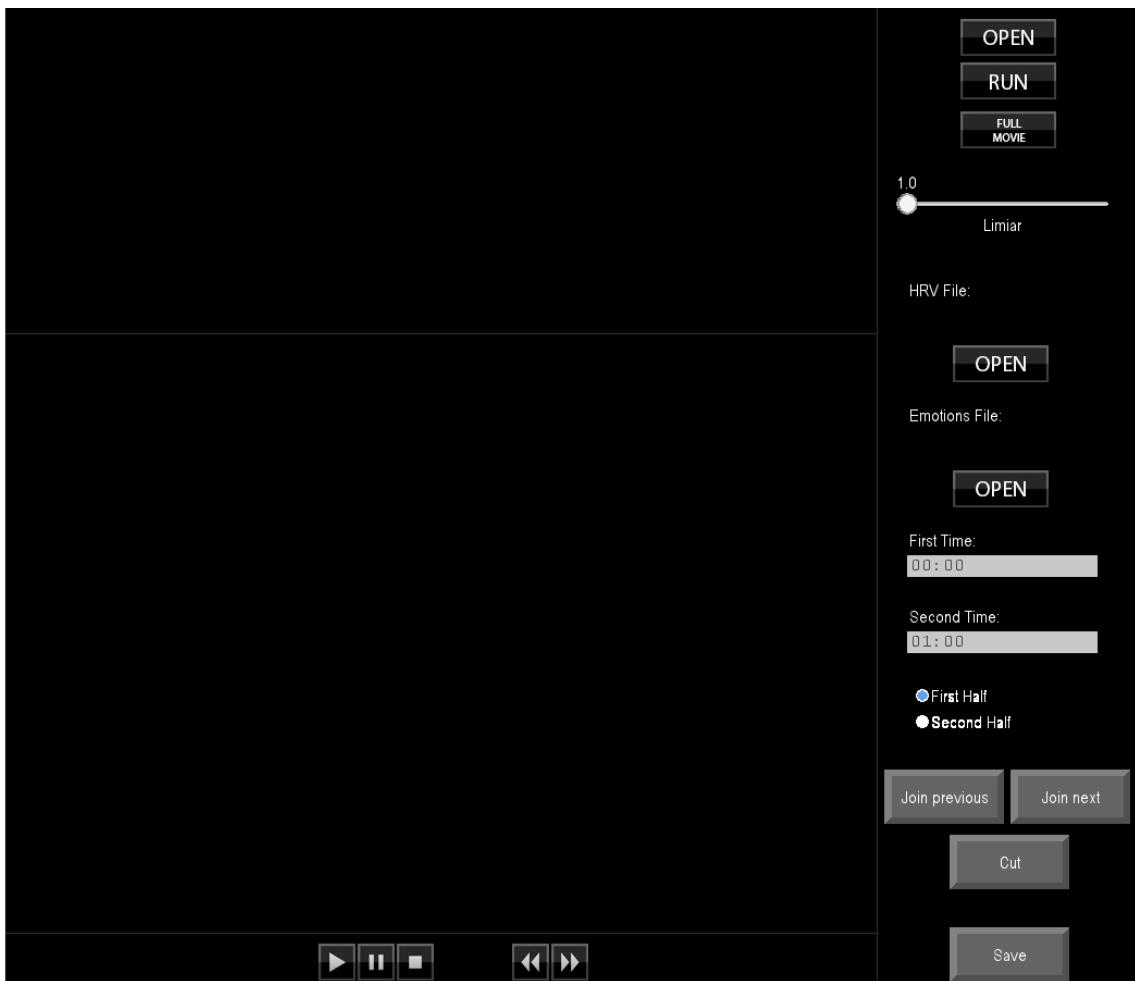


Figura 4.9: GUI da interface.

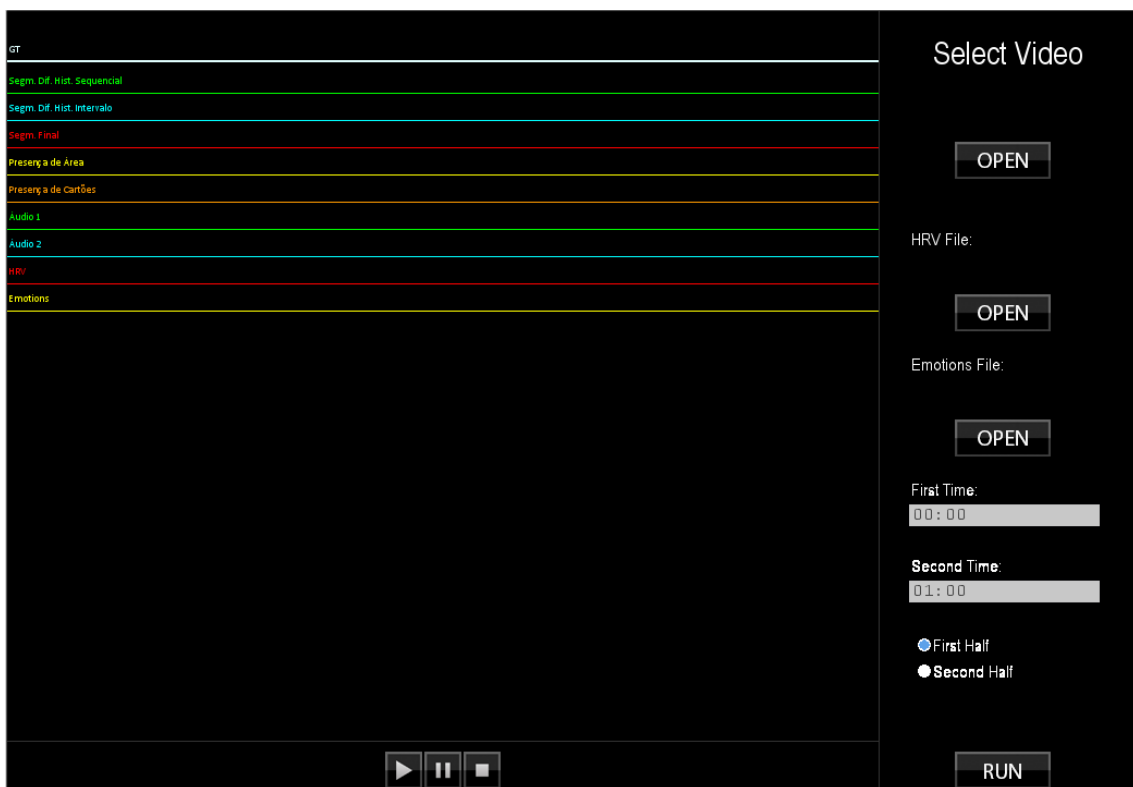


Figura 4.10: GUI da interface de Teste.

# 5

## Recolha de Dados e Avaliação do Sistema

Neste capítulo é efetuada uma descrição da metodologia implementada na realização da captura de dados e uma análise aos resultados obtidos resultantes da implementação dos vários módulos do sistema.

### 5.1 Recolha de Dados

#### 5.1.1 Participantes e *Setup*

O processo de recolha de dados foi apenas aplicado a um participante de 25 anos de idade pertencente ao sexo masculino. O participante encontra-se em boas condições físicas e não possui quaisquer impedimentos visuais ou auditivos. O participante está familiarizado com o processo de assistir a jogos de futebol.

As sessões de gravação ocorreram sempre no mesmo espaço e sob condições idênticas. O ambiente de gravação consiste num espaço fechado e com o mínimo de ruídos externos que possam perturbar o participante impedindo assim a correta captação dos dados. Durante o decorrer do processo não foi permitido o acesso ao espaço a pessoas exteriores ao mesmo. Os estímulos visuais foram aplicados ao participante utilizando um monitor *Asus* de 24 polegadas e os auditivos foram aplicados utilizando uns auscultadores da marca *Sony*.

### 5.1.2 Processo de Gravação dos Dados

O participante foi submetido a um total de quatro sessões de gravação que envolveram a exposição de estímulos visuais e auditivos. Simultaneamente foram recolhidas as informações cardíacas e anotadas as emoções. Cinco minutos antes de cada sessão de gravação o participante não executou qualquer atividade ficando apenas em repouso para estabilizar o seu batimento cardíaco.

As quatro sessões de gravação produziram 16 ficheiros: oito ficheiros que contêm os intervalos RR capturados e oito ficheiros que contêm as emoções anotadas (para cada jogo existem dois ficheiros de intervalos RR e dois de emoções, um para cada parte. Caso o jogo tenha prolongamento e penalties são gerados mais dois ficheiros com intervalos RR e dois ficheiros com emoções).

## 5.2 Avaliação do Sistema

Para aferir o correto funcionamento do sistema é necessário efetuar vários testes a todos os algoritmos implementados. É levada a cabo uma análise de resultados aos algoritmos de segmentação e de deteção de planos relevantes.

O algoritmo de deteção de planos relevantes consiste, na verdade, num conjunto de algoritmos:

- Algoritmo de classificação áudio;
- Algoritmo de deteção de cartões;
- Algoritmo de deteção de grande área;
- Classificador de características HRV.

A correta capacidade de classificação destes algoritmos é testada individualmente e em conjunto, o que permite ter uma ideia do peso de cada um nos resultados produzidos pelo sistema.

Para testar os algoritmos serão utilizados excertos dos quatro jogos utilizados nas sessões de gravação de características fisiológicas e emoções (em direto, sem o utilizador conhecer o resultado final). Os jogos utilizados (gravados na sua totalidade) são:

- Vitória de Guimarães vs Futebol Clube do Porto (a 12/05/2018, terminou com o resultado de 0-1);
- Desportivo das Aves vs Sporting Clube de Portugal (a 20/05/2018, terminou com o resultado de 2-1);
- Real Madrid vs Liverpool (a 26/05/2018, terminou com o resultado de 3-1);
- Portugal vs Espanha (a 15/06/2018, terminou com o resultado de 3-3).

Os algoritmos de segmentação e classificação são pesados computacionalmente. Este facto torna necessária a utilização de excertos dos vídeos resultantes das sessões de gravação. Os excertos utilizados para os testes de segmentação e classificação são:

- Vídeo 1 - Excerto do jogo Vitória de Guimarães vs Futebol Clube do Porto com duração de 2 minutos e 18 segundos (ocorre um golo do Futebol Clube do Porto);
- Vídeo 2 - Excerto do jogo Desportivo das Aves vs Sporting Clube de Portugal com duração de 2 minutos e 55 segundos (ocorre um golo do Desportivo das Aves);
- Vídeo 3 - Excerto do jogo Real Madrid vs Liverpool com duração de 2 minutos e 38 segundos (ocorre um golo do Real Madrid);
- Vídeo 4 - Excerto do jogo Real Madrid vs Liverpool com duração de 2 minutos e 23 segundos (ocorre um golo do Liverpool);
- Vídeo 5 - Excerto do jogo Real Madrid vs Liverpool com duração de 2 minutos e 29 segundos (ocorre um golo do Real Madrid);
- Vídeo 6 - Excerto do jogo Espanha vs Portugal com duração de 1 minuto e 30 segundos (ocorre um golo da Espanha);
- Vídeo 7 - Excerto do jogo Desportivo das Aves vs Sporting Clube de Portugal com duração de 2 minutos (ocorre a amostragem de um cartão amarelo a um jogador do Desportivo das Aves);
- Vídeo 8 - Excerto do jogo Real Madrid vs Liverpool com duração de 1 minuto e 30 segundos (ocorre a amostragem de um cartão amarelo a um jogador do Liverpool);

- Vídeo 9 - Excerto do jogo Espanha vs Portugal com duração de 2 minutos (ocorre a amostragem de um cartão amarelo a um jogador de Portugal);

A avaliação dos resultados será efetuada recorrendo à interface de teste que permite ter uma ideia do funcionamento de todos os algoritmos de forma separada. Isto é possível devido à existência de uma linha para cada algoritmo (a extensão da linha coincide com a extensão do excerto de vídeo processado) onde são desenhados traços verticais que indicam as *frames* que foram classificadas positivamente (Figura 5.1).

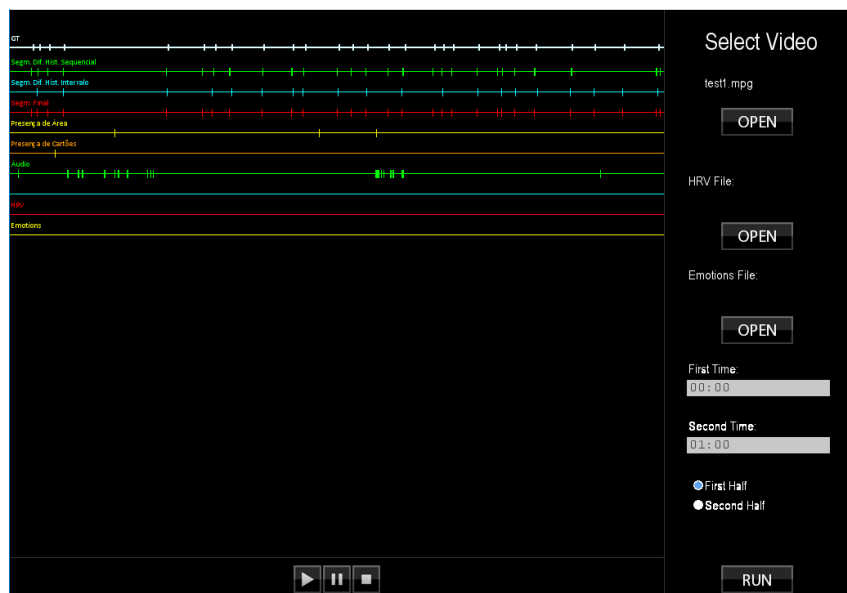


Figura 5.1: Representação gráfica dos algoritmos na Interface de Teste.

Para avaliar os resultados dos algoritmos é necessário visualizar os vídeos (ou pelo menos os excertos utilizados) e anotar as *frames* onde existe segmentação e os planos que são relevantes. Este processo gera dados de referência (também conhecidos como *Ground Truth*) com os quais é possível comparar os resultados dos algoritmos verificando se estes estão corretos ou não. O *Ground Truth* relativo à segmentação foi conseguido adicionando uma funcionalidade à interface final do módulo de análise de vídeos que permite ao utilizador criar um ficheiro XML (com o mesmo nome do vídeo) contendo *frames* onde ocorre transição. O utilizador ao visualizar o jogo completo pode carregar na tecla "S" do seu teclado o que faz com que a *frame* corrente seja gravada no ficheiro XML. Na comparação do *Ground Truth* com os resultados do sistema no que toca à segmentação será tido em conta que as *frames* de transição podem sempre ter algum desfasamento devido ao tempo de reação de utilizador (cerca de 60 *frames*).

### 5.2.1 Avaliação de algoritmos que utilizam HRV

Uma vez que o algoritmo de deteção de golos que utiliza HRV consiste num classificador SVM, é necessário estabelecer as condições utilizadas para o seu treino e consequentes testes.

Neste processo é utilizada uma técnica de validação cruzada [45] onde são utilizados os dados HRV resultantes da gravação de um jogo para treinar o SVM e dados HRV resultantes da gravação de outros jogos para validar o seu funcionamento nos testes em combinação com os restantes algoritmos de classificação.

Neste caso o jogo utilizado para treinar o SVM é o jogo Espanha vs Portugal, uma vez que possui 6 golos e é possível assim retirar várias amostras HRV de várias situações de golo. Os restantes jogos são utilizados para testar a viabilidade do sistema. O jogo utilizado no treino não é utilizado nos testes para evitar que ocorra *over fitting* [2].

Os dados HRV retirados do jogo Espanha vs Portugal utilizados para o treino do HRV referem-se aos seguintes tempos de jogo:

- 3:15 - 4:05 - Ocorre um golo de Portugal. Utilizado como amostra de treino positiva;
- 23:00 - 24:05 - Ocorre um golo de Espanha. Utilizado como amostra de treino positiva;
- 43:28 - 44:50 - Ocorre um golo de Portugal. Utilizado como amostra de treino positiva;
- 53:40 - 54:40 - Ocorre um golo de Espanha. Utilizado como amostra de treino positiva;
- 57:30 - 58:30 - Ocorre um golo de Espanha. Utilizado como amostra de treino positiva;
- 87:00 - 88:00 - Ocorre um golo de Portugal. Utilizado como amostra de treino positiva;
- 27:00 - 30:00 - Não ocorrem eventos relevantes. Várias amostras utilizadas como amostras de treino negativas.

Uma vez que o jogo de onde são retiradas as características HRV de treino é o Espanha vs Portugal, o vídeo 6 não é utilizado nos testes das deteções de golos que envolvam a utilização do algoritmo SVM.

### 5.2.2 Métricas de avaliação

Para verificar os resultados dos algoritmos é preciso utilizar métricas que permitam quantificar o grau de desempenho dos mesmos. Para isso são utilizadas as probabilidades de erro de cada classe, a Precisão e a Cobertura.

A probabilidade de erro de uma classe é calculada utilizando a equação 5.1 onde  $p_{ij}$  representa a probabilidade dos dados pertencentes à classe  $\varpi_i$  serem classificados noutra classe e  $p_{ii}$  representa a probabilidade dos dados dessa mesma classe serem classificados corretamente.

$$P_{Erro}(\varpi_i) = \sum_{j \neq i}^c p_{ij} = 1 - p_{ii} \quad (5.1)$$

A métrica Precisão tenta responder à seguinte questão: "Daqueles que foram classificados como corretos, quantos efetivamente eram?". Esta métrica é calculada utilizando a equação 5.2.

$$Precisão = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Positivos\ (FP)} \quad (5.2)$$

A métrica Cobertura tenta responder à seguinte questão: "Quando realmente é da classe X, o quão frequente é classificado como X?". Esta métrica é calculada utilizando a equação 5.3.

$$Cobertura = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Negativos\ (FN)} \quad (5.3)$$

### 5.2.3 Segmentação

O *Ground Truth* gerado previamente (através do processo referido anteriormente) é utilizado para aferir a qualidade do processo de segmentação. Na tabela 5.1 é possível verificar os resultados obtidos pelo algoritmo. O conceito "Falso Positivo" representa uma situação em que o algoritmo deteta uma transição onde esta não existe e o conceito "Falso Negativo" representa uma situação em que existe uma transição e esta não é detetada pelo algoritmo.

Vídeo	Nº de segmen- tações	Deteções Cor- retas	Falsos Positi- vos	Falsos Negati- vos
Vídeo 1	28	28	1	0
Vídeo 2	26	25	2	1
Vídeo 3	26	25	5	1
Vídeo 4	21	21	1	0
Vídeo 5	19	19	2	0
Vídeo 6	13	13	2	0
Vídeo 7	15	14	2	1
Vídeo 8	5	5	0	0
Vídeo 9	16	16	4	0
Total	169	166	19	3

Tabela 5.1: Resultados da Segmentação.

Os valores de Precisão e Cobertura (calculados a partir da informação presente na tabela 5.1) são apresentados na tabela 5.2:

Vídeo	Precisão	Cobertura
Vídeo 1	0.966	1
Vídeo 2	0.926	0.962
Vídeo 3	0.833	0.962
Vídeo 4	0.955	1
Vídeo 5	0.905	1
Vídeo 6	0.867	1
Vídeo 7	0.875	0.933
Vídeo 8	1	1
Vídeo 9	0.8	1
Conjunto	0.897	0.982

Tabela 5.2: Precisão e Cobertura da segmentação.

Como é possível verificar, foram obtidos valores de Precisão e de Cobertura bastante elevados para o conjunto de vídeos testados. Um valor de Precisão elevado é indicativo da existência de poucas deteções erradas e um valor de Cobertura elevado é indicativo de que a maior parte das transições foram detetadas.

Os erros existentes em relação às falsas deteções devem-se ao facto de, por vezes, existirem jogadores que se atravessam à frente da câmara, o que faz com que na

prática o algoritmo declare a existência de uma transição quando na verdade o plano é sempre o mesmo. Os erros existentes em relação às falhas na detecção devem-se ao facto de, por vezes, a transição ocorrer entre planos que possuem histogramas com valores semelhantes (transições graduais entre dois planos do tipo longo normalmente apresentam este problema).

Também é possível verificar que as transições graduais são corretamente detetadas, o que representa uma melhoria em relação ao sistema implementado por Cunha J. e Graça H. em [13], contribuindo isso para um aumento significativo nos valores de Precisão e de Cobertura.

#### 5.2.4 Detecção de Grande Área

Antes de se proceder à avaliação dos testes relacionados com o algoritmo de detecção de planos relevantes é necessário avaliar a execução da detecção de grande área, que funciona como complemento à classificação, apesar de não servir como classificador de lances relevantes por si só. O detetor de grande área utilizado sozinho não tem grande utilidade visto que deteta apenas quando a grande área está presente, mesmo que ocorra num lance não relevante. No entanto pode ser utilizado para complementar o resultado de outros algoritmos (se o classificador HRV detetar um lance relevante é possível utilizar o algoritmo de detecção de grande área para verificar em que plano este lance ocorre uma vez que o algoritmo HRV não é executado por plano mas sim por intervalo temporal).

Para obter o *Ground Truth* é necessário definir algumas regras em relação ao que é considerada *frame* que contém grande área e *frame* que não contém. A *frame* tem que ser proveniente da câmara normal de jogo e não de câmaras de repetições uma vez que o algoritmo não está preparado para lidar com essa situação (o que nesta situação é desejável uma vez que não são pretendidos os planos ocorridos durante uma repetição, apenas o plano onde ocorre o golo). As *frames* provenientes de câmaras de repetição normalmente apresentam as três linhas de fundo da área mais na vertical (Figura 5.3), ao contrário das *frames* provenientes da câmara normal de jogo que apresentam estas linhas na diagonal num ângulo de cerca de 45 graus (Figura 5.4). A grande área tem que estar totalmente enquadrada na *frame* uma vez que não basta a grande área aparecer totalmente para existir momento de perigo (por exemplo num pontapé de baliza não há perigo e a grande área está totalmente visível).

As figuras 5.2 e 5.3 representam situações em que apesar de aparecer a grande área, não são consideradas para o *Ground Truth* uma vez que não se enquadram

nas regras definidas anteriormente (a primeira não está enquadrada na *frame* e a segunda resulta de uma câmara de repetição).



Figura 5.2: Grande área não totalmente enquadrada na *frame*.



Figura 5.3: Grande área filmada por câmara de repetição.

Na figura 5.4 a grande área encontra-se enquadrada e está a ser filmada pela câmara normal de jogo o que faz com que este plano seja considerado no *Ground Truth*.



Figura 5.4: Grande área enquadrada corretamente com câmara normal de jogo.

São apresentados na tabela 5.3 os resultados do algoritmo de deteção de grande área aplicado aos vários vídeos.

Vídeo	Nº Segmentos Área Vísivel	Deteções Cor- retas	Falsos Positi- vos	Falsos Negati- vos
Vídeo 1	3	3	0	0
Vídeo 2	2	2	0	0
Vídeo 3	1	1	0	0
Vídeo 4	2	2	1	0
Vídeo 5	1	1	1	0
Vídeo 6	1	1	0	0
Vídeo 7	3	2	2	1
Vídeo 8	1	1	0	0
Vídeo 9	4	4	1	0
Total	18	17	5	1

Tabela 5.3: Resultados da Deteção de Grande Área.

Os valores de Precisão e Cobertura (calculados a partir da informação presente na tabela 5.3) são apresentados na tabela 5.4:

Vídeo	Precisão	Cobertura
Vídeo 1	1	1
Vídeo 2	1	1
Vídeo 3	1	1
Vídeo 4	0.667	1
Vídeo 5	0.5	1
Vídeo 6	1	1
Vídeo 7	0.5	0.667
Vídeo 8	1	1
Vídeo 9	0.8	1
Conjunto	0.773	0.944

Tabela 5.4: Precisão e Cobertura da detecção de grande área.

Na tabela anterior é possível verificar que a Precisão e o Cobertura, apesar de não estarem ao nível dos valores obtidos na segmentação, apresentam valores elevados.

No entanto, visto que parte deste algoritmo consiste na detecção do relvado, o seu funcionamento vai depender da área que o utilizador selecionar como relvado. Neste caso o algoritmo foi executado para os 8 vídeos três vezes com o mesmo resultado, selecionando sempre parte do relvado. Como será de esperar, se o utilizador não selecionar parte do relvado ou selecionar uma parte que contenha outros elementos como um jogador, o algoritmo não funcionará corretamente.

O algoritmo apresenta uma Precisão relativamente reduzida. Isto deve-se ao facto de por vezes existirem muitas linhas que fazem com que a grande área seja detetada de forma errada aumentando assim o número de falsos positivos (linhas referentes às bordas dos *placards* publicitários, das extremidades do campo, etc). Os erros ocorrentes em relação às falhas de detecção (que fazem diminuir o valor de Cobertura) normalmente ocorrem devido ao agrupamento de vários jogadores na área, o que faz com que por vezes a linha da pequena área ou da grande área não estejam totalmente visíveis, impedindo a sua detecção por parte do algoritmo (neste caso esta situação não ocorre com frequência como se pode verificar pelo valor de elevado de Cobertura).

### 5.2.5 Detecção de Planos Relevantes

A deteção de planos relevantes consiste em duas funcionalidades separadas que em conjunto possibilitam a deteção de amostragem de cartões (amarelos e vermelhos) e a deteção de golos.

Na deteção de cartões foi utilizado o algoritmo de deteção de cartões apresentado no capítulo 3 combinado com o algoritmo de características auditivas (o funcionamento do algoritmo de características auditivas não foi aprofundado devido ao facto de ter sido utilizado o desenvolvido por Cunha J. e Graça H. em [13], não tendo sido feitas quaisquer alterações à sua implementação). Esta combinação parte do princípio que normalmente, quando existe a amostragem de um cartão existe um pico sonoro devido a assobios ou animosidade por parte do público.

Para testar o procedimento de deteção de golos serão analisados os vários algoritmos. Para ter uma ideia clara da influência de cada algoritmo no resultado de classificação final serão analisadas várias combinações dos algoritmos:

- Deteção de golos baseada em áudio;
- Deteção de golos baseada em presença de grande área;
- Deteção de golos baseada em áudio e presença de grande área;
- Deteção de golos baseada em áudio e HRV;
- Deteção de golos baseada em áudio e HRV com emoções;
- Deteção de golos baseada em presença de grande área e HRV;
- Deteção de golos baseada em presença de grande área e HRV com emoções;
- Deteção de golos baseada em áudio, presença de grande área e HRV;
- Deteção de golos baseada em áudio, presença de grande área e HRV com emoções.

De notar que não será feita uma análise ao algoritmo de HRV em separado. Isto acontece porque o HRV não é calculado por plano mas sim em intervalos temporais (neste caso de 15 segundos). O HRV é utilizado para confirmar o resultado dos restantes algoritmos, ou seja, se num plano existir pico de áudio e grande área visível mas não existir nenhuma avaliação positiva por parte do algoritmo

HRV nas proximidades (neste caso é definida uma margem de 250 *frames*) o plano é descartado.

Nos vários testes os planos são apenas classificados positivamente (como relevantes) se todos os algoritmos que compõem o classificador determinarem que o plano contém um golo. No caso de um deles determinar a não existência de golo em determinado plano este é automaticamente classificado de forma negativa (como não-relevante).

### 5.2.5.1 Detecção de Cartões utilizando características visuais

Em primeiro lugar é efetuada uma análise ao funcionamento do algoritmo de deteção de cartões, antes de serem adicionadas as características auditivas. São construídas as matrizes de confusão onde os elementos a classificar serão os planos gerados pelo algoritmo de classificação.

De referir que infelizmente, em todos os jogos gravados, não ocorreram quaisquer amostragens de cartões vermelhos o que impossibilita a obtenção de uma correta probabilidade de erro de classificação relativa à amostragem de cartões desta cor.

Nas matrizes de confusão seguintes as letras *s*, *a* e *v* representam as classes esperadas relativas a segmentos sem cartão, com cartão amarelo ou com cartão vermelho, respetivamente. *s'*, *a'* e *v'* representam as classes atribuídas pelo algoritmo relativas a segmentos sem cartão, com cartão amarelo ou com cartão vermelho, respetivamente.

	<i>s'</i>	<i>a'</i>	<i>v'</i>
<i>s</i>	28	1	0
<i>a</i>	0	0	0
<i>v</i>	0	0	0

Tabela 5.5: Detecção de cartões - Matriz de confusão Vídeo 1.

	<i>s'</i>	<i>a'</i>	<i>v'</i>
<i>s</i>	23	1	3
<i>a</i>	0	0	0
<i>v</i>	0	0	0

Tabela 5.6: Detecção de cartões - Matriz de confusão Vídeo 2.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	30	0	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.7: Detecção de cartões - Matriz de confusão Vídeo 3.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	21	0	1
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.8: Detecção de cartões - Matriz de confusão Vídeo 4.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	20	1	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.9: Detecção de cartões - Matriz de confusão Vídeo 5.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	14	1	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.10: Detecção de cartões - Matriz de confusão Vídeo 6.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	16	0	0
<b>a</b>	0	1	0
<b>v</b>	0	0	0

Tabela 5.11: Detecção de cartões - Matriz de confusão Vídeo 7.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	4	0	0
<b>a</b>	0	1	0
<b>v</b>	0	0	0

Tabela 5.12: Detecção de cartões - Matriz de confusão Vídeo 8.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	19	0	0
<b>a</b>	1	0	0
<b>v</b>	0	0	0

Tabela 5.13: Detecção de cartões - Matriz de confusão Vídeo 9.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	175	4	4
<b>a</b>	1	2	0
<b>v</b>	0	0	0

Tabela 5.14: Detecção de cartões - Matriz de confusão Total.

$$P_{Erro}(s) = 4,4\% \quad P_{Erro}(a) = 33,3\% \quad (5.4)$$

Como é possível verificar o erro de classificação da classe  $s$  é reduzido, uma vez que praticamente todos os segmentos onde não ocorre a amostragem de cartões são corretamente classificados. O erro desta classe tem por base o facto de nos vídeos 1, 2, 4, 5 e 6 existirem segmentos sem cartão que são classificados como contendo a amostragem de cartões amarelos ou vermelhos:

- No vídeo 1 um segmento é classificado de forma errada como contendo cartão amarelo. Isto deve-se ao facto de no 4º segmento por breves instantes estar visível no topo da *frame* parte da escadaria do estádio que é da cor amarela (Figura 5.5);



Figura 5.5: Exemplo de erro na deteção de cartão amarelo vídeo 1.

- No vídeo 2 é classificado de forma errada um segmento como contendo cartão amarelo e três como contendo cartão vermelho. As incorretas classificações relativas a cartões vermelhos devem-se ao facto de por vezes estarem visíveis no topo da *frame* os painéis publicitários de cor vermelha (Figura 5.6). A classificação incorreta relativa ao cartão amarelo deve-se ao facto de no plano aparecer por uns instantes no topo da *frame* um segurança do estádio vestindo um colete amarelo (Figura 5.7);



Figura 5.6: Exemplo de erro na deteção de cartão vermelho vídeo 2.



Figura 5.7: Exemplo de erro na deteção de cartão amarelo vídeo 2.

- No vídeo 4 um segmento é classificado como contendo um cartão vermelho de forma errada. Isto deve-se ao facto de um jogador do Liverpool com equipamento vermelho aparecer no topo da *frame* no segmento (Figura 5.8);



Figura 5.8: Exemplo de erro na deteção de cartão vermelho vídeo 4.

- No vídeo 5 um segmento é classificado incorretamente como contendo cartão amarelo. Isto deve-se ao facto de no segmento aparecer por instantes o árbitro na zona superior da *frame* usando equipamento amarelo (Figura 5.9);



Figura 5.9: Exemplo de erro na deteção de cartão amarelo vídeo 5.

- No vídeo 6 um segmento é classificado incorretamente como contendo cartão amarelo. Isto deve-se ao facto de no segmento aparecer por instantes o árbitro na zona superior da *frame* usando equipamento amarelo (Figura 5.10).



Figura 5.10: Exemplo de erro na deteção de cartão amarelo vídeo 6.

A probabilidade de erro da classe  $a$  é de um terço uma vez que em três 3 planos onde ocorrem cartões amarelos, 2 são classificados positivamente de forma correta e 1 é classificado negativamente de forma incorreta. Esta falha na deteção deve-se ao facto de, no vídeo 9, o cartão amarelo aparecer mais ao centro na *frame*. Como a deteção de cartões é aplicada à parte superior da imagem o cartão não é detetado.

### 5.2.5.2 Detecção de Cartões utilizando características visuais e auditivas

Na deteção de cartões utilizando características visuais e auditivas um segmento é apenas classificado como contendo um cartão quando o algoritmo de deteção de cartões e o algoritmo de características auditivas classificam o segmento positivamente.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	29	0	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.15: Deteção de cartões com Áudio - Matriz de confusão Vídeo 1.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	24	1	2
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.16: Deteção de cartões com Áudio - Matriz de confusão Vídeo 2.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	30	0	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.17: Deteção de cartões com Áudio - Matriz de confusão Vídeo 3.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	22	0	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.18: Deteção de cartões com Áudio - Matriz de confusão Vídeo 4.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	21	0	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.19: Detecção de cartões com Áudio - Matriz de confusão Vídeo 5.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	14	1	0
<b>a</b>	0	0	0
<b>v</b>	0	0	0

Tabela 5.20: Detecção de cartões com Áudio - Matriz de confusão Vídeo 6.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	16	0	0
<b>a</b>	0	1	0
<b>v</b>	0	0	0

Tabela 5.21: Detecção de cartões com Áudio - Matriz de confusão Vídeo 7.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	4	0	0
<b>a</b>	0	1	0
<b>v</b>	0	0	0

Tabela 5.22: Detecção de cartões com Áudio - Matriz de confusão Vídeo 8.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	19	0	0
<b>a</b>	1	0	0
<b>v</b>	0	0	0

Tabela 5.23: Detecção de cartões com Áudio - Matriz de confusão Vídeo 9.

	<b>s'</b>	<b>a'</b>	<b>v'</b>
<b>s</b>	179	2	2
<b>a</b>	1	2	0
<b>v</b>	0	0	0

Tabela 5.24: Detecção de cartões com Áudio - Matriz de confusão Total.

$$P_{Erro}(s) = 2,2\% \quad P_{Erro}(a) = 33,3\% \quad (5.5)$$

A adição do áudio como característica de classificação teve o efeito positivo de diminuir para metade o erro de classificação da classe *s*. Isto significa que os falsos positivos sofreram uma redução, ou seja, existiram menos segmentos classificados como contendo cartão quando na verdade isso não se verifica.

Não é verificada uma redução no erro de classificação da classe *a* porque a adição do algoritmo de características auditivas tem apenas o efeito de reduzir os falsos positivos. Mesmo que este algoritmo retornasse positivo, o facto do algoritmo de detecção de cartões retornar negativo impedia o segmento de ser classificado como positivo.

### 5.2.5.3 Detecção de Golos baseada em Áudio

Em primeiro lugar é efetuada a análise à execução do algoritmo de detecção de planos relevantes baseado em características auditivas. Na tabela 5.25 é possível verificar em detalhe os valores de planos detetados corretamente, os falsos positivos e as falhas de detecção. Na tabela 5.26 é possível verificar os valores de Precisão e de Cobertura para cada vídeo e também para o conjunto dos vídeos. Nesta análise serão utilizados todos os vídeos onde ocorrem golos.

Vídeo	Nº de segmentos com golo	Deteções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	5	0
Vídeo 2	1	1	7	0
Vídeo 3	1	1	8	0
Vídeo 4	1	1	5	0
Vídeo 5	1	1	8	0
Vídeo 6	1	1	3	0
Total	6	6	36	0

Tabela 5.25: Resultados da Deteção de golos utilizando Áudio.

Vídeo	Precisão	Cobertura
Vídeo 1	0.167	1
Vídeo 2	0.125	1
Vídeo 3	0.111	1
Vídeo 4	0.167	1
Vídeo 5	0.111	1
Vídeo 6	0.25	1
Conjunto	0.143	1

Tabela 5.26: Precisão e Cobertura da deteção de golos utilizando áudio.

Na tabela 5.26 é possível verificar que apesar do valor de Cobertura ser elevado, o valor de precisão é muito reduzido. O valor de Cobertura é bastante elevado porque quando ocorre um golo normalmente ocorrem sempre vários picos sonoros, o que faz com que os golos sejam sempre corretamente detetados. O valor de Precisão é reduzido porque normalmente ocorrerem vários picos sonoros durante o jogo em situações que não são de golo. Isto deve-se em grande parte ao som produzido pelos comentadores durante o decorrer do jogo, o que leva a que vários planos sejam classificados como planos onde ocorre um golo quando na realidade não ocorre qualquer situação relevante. Para melhorar o desempenho isolado deste algoritmo seria possível remover o som dos comentadores o que faria com que os picos sonoros fossem menos acentuados.

### 5.2.5.4 Detecção de Golos baseada em Presença de Grande Área

Como já referido anteriormente, teoricamente a utilização do algoritmo de deteção de grande área não configura uma solução satisfatória quando utilizada isoladamente uma vez que o facto de a grande área estar visível num plano pode não ser indicativo da existência de um golo. Em teoria o algoritmo terá um elevado valor de Cobertura uma vez que em todos os golos a grande área está sempre visível, no entanto irá produzir um valor de Precisão reduzido visto que vários segmentos onde a grande área é visível serão classificados positivamente sem que ocorra qualquer golo durante a sua duração.

No entanto é efetuada uma análise do desempenho deste algoritmo de forma isolada no que toca à deteção de golos.

Vídeo	Nº de segmentos com golo	Deteções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	2	0
Vídeo 2	1	1	1	0
Vídeo 3	1	1	0	0
Vídeo 4	1	1	2	0
Vídeo 5	1	1	1	0
Vídeo 6	1	1	1	0
Total	6	6	7	0

Tabela 5.27: Resultados da Detecção de golos utilizando Áudio.

Vídeo	Precisão	Cobertura
Vídeo 1	0.333	1
Vídeo 2	0.5	1
Vídeo 3	1	1
Vídeo 4	0.333	1
Vídeo 5	0.5	1
Vídeo 6	0.5	1
Conjunto	0.462	1

Tabela 5.28: Precisão e Cobertura da deteção de golos utilizando áudio.

A tabela 5.28 confirma as expectativas em relação ao desempenho do algoritmo.

O valor de Cobertura é elevado porque todos os golos são corretamente detetados, no entanto, existem vários falsos positivos o que resulta num valor de Precisão reduzido. É possível verificar porém, que deste algoritmo resultam menos falsos positivos que do algoritmo de áudio utilizado isoladamente.

#### 5.2.5.5 Detecção de Golos baseada em Áudio e Presença de Grande Área

A combinação dos dois algoritmos anteriores poderá resultar em valores em menos falsos positivos, uma vez que para um segmento ser classificado como golo, terão os dois algoritmos que apresentar resultados positivos.

Vídeo	Nº de segmentos com golo	Detecções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	1	0
Vídeo 2	1	1	2	0
Vídeo 3	1	1	0	0
Vídeo 4	1	1	1	0
Vídeo 5	1	1	1	0
Vídeo 6	1	1	0	0
Total	6	6	5	0

Tabela 5.29: Resultados da Detecção de golos utilizando Áudio e Grande Área.

Vídeo	Precisão	Cobertura
Vídeo 1	0.5	1
Vídeo 2	0.333	1
Vídeo 3	1	1
Vídeo 4	0.5	1
Vídeo 5	0.5	1
Vídeo 6	1	1
Conjunto	0.545	1

Tabela 5.30: Precisão e Cobertura da detecção de golos utilizando Áudio e Grande Área.

A tabela 5.30 evidencia que, combinando os dois algoritmos obtêm-se um valor de Precisão mais elevado que resulta da existência de menos falsos positivos. Os falsos positivos existentes devem-se ao facto de existirem segmentos onde a

grande área é visível e existem picos sonoros mas no entanto não existe qualquer golo. Esta combinação de algoritmos pode, no entanto, ser bastante útil na deteção de lances de perigo uma vez que nestes lances a grande área muitas vezes encontra-se visível e existem picos sonoros devido à excitação do público.

### 5.2.5.6 Deteção de Golos baseada em Áudio e HRV

A partir daqui o algoritmo HRV é combinado com os algoritmos anteriores para verificar qual a influência deste algoritmo nos resultados da classificação.

Vídeo	Nº de segmentos com golo	Deteções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	1	0
Vídeo 2	1	1	2	0
Vídeo 3	1	1	2	0
Vídeo 4	1	0	0	1
Vídeo 5	1	1	1	0
Total	5	4	6	1

Tabela 5.31: Resultados da Deteção de golos utilizando Áudio e HRV.

Vídeo	Precisão	Cobertura
Vídeo 1	0.5	1
Vídeo 2	0.333	1
Vídeo 3	0.333	1
Vídeo 4	1	0
Vídeo 5	0.5	1
Conjunto	0.4	0.8

Tabela 5.32: Precisão e Cobertura da deteção de golos utilizando Áudio e HRV.

Na tabela 5.32 é possível verificar que em relação ao algoritmo de áudio utilizado isoladamente, o valor de Precisão aumentou mas, no entanto o valor de Cobertura diminuiu. Isto deve-se ao facto de no vídeo 4 não existir nenhuma deteção positiva no que toca ao HRV. Esta situação pode resultar do facto de este vídeo se referir ao jogo Liverpool vs Real Madrid, jogo em que o sujeito da gravação não possui afinidade em especial com nenhuma das equipas, o que pode ter levado a que o golo existente nesse vídeo não despoletasse reações mais intensas.

### 5.2.5.7 Detecção de Golos baseada em Áudio e HRV com emoções

Em seguida são adicionadas as emoções anotadas pelo utilizador. Estas emoções são utilizadas no algoritmo SVM de características HRV em conjunto com os dados HRV. O resultado do algoritmo HRV é utilizado em conjunto com o algoritmo de áudio para classificar os dados dos vídeos.

Vídeo	Nº de segmentos com golo	Detecções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	0	0
Vídeo 2	1	1	1	0
Vídeo 3	1	1	0	0
Vídeo 4	1	0	0	1
Vídeo 5	1	1	1	0
Total	5	4	2	1

Tabela 5.33: Resultados da Detecção de golos utilizando Áudio e HRV com emoções.

Vídeo	Precisão	Cobertura
Vídeo 1	1	1
Vídeo 2	0.5	1
Vídeo 3	1	1
Vídeo 4	1	0
Vídeo 5	0.5	1
Conjunto	0.667	0.8

Tabela 5.34: Precisão e Cobertura da deteção de golos utilizando Áudio e HRV com emoções.

A utilização das emoções permite o aumento do valor de Precisão uma vez que diminui a quantidade de falsos positivos produzidos pelo sistema. No entanto o valor de Cobertura continua igual ao obtido anteriormente uma vez que no vídeo 4 continuam a não haver deteções positivas apesar de existir um golo.

### 5.2.5.8 Detecção de Golos baseada em Presença de Grande Área e HRV

Em seguida é testada a combinação do algoritmo de deteção de grande área com o algoritmo SVM de características HRV.

Vídeo	Nº de segmentos com golo	Deteções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	2	0
Vídeo 2	1	1	0	0
Vídeo 3	1	1	0	0
Vídeo 4	1	0	0	1
Vídeo 5	1	1	0	0
Total	5	4	2	1

Tabela 5.35: Resultados da Deteção de golos baseada em Presença de Grande Área e HRV.

Vídeo	Precisão	Cobertura
Vídeo 1	0.333	1
Vídeo 2	1	1
Vídeo 3	1	1
Vídeo 4	1	0
Vídeo 5	1	1
Conjunto	0.667	0.8

Tabela 5.36: Precisão e Cobertura da deteção de golos baseada em Presença de Grande Área e HRV.

Em comparação com a utilização isolada do algoritmo de deteção de grande área é possível verificar que o valor de Precisão aumentou mas o valor de Cobertura diminuiu. Isto deve-se ao problema anteriormente identificado relacionado com a ausência de deteções positivas no vídeo 4 por parte do algoritmo HRV.

### 5.2.5.9 Detecção de Golos baseada em Presença de Grande Área e HRV com emoções

Vídeo	Nº de segmentos com golo	Detecções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	1	0
Vídeo 2	1	1	0	0
Vídeo 3	1	1	0	0
Vídeo 4	1	0	0	1
Vídeo 5	1	1	0	0
Total	5	4	1	1

Tabela 5.37: Resultados da Detecção de golos baseada em Presença de Grande Área e HRV com emoções.

Vídeo	Precisão	Cobertura
Vídeo 1	0.5	1
Vídeo 2	1	1
Vídeo 3	1	1
Vídeo 4	1	0
Vídeo 5	1	1
Conjunto	0.8	0.8

Tabela 5.38: Precisão e Cobertura da detecção de golos baseada em Presença de Grande Área e HRV com emoções.

Na tabela 5.38 é possível verificar que os valores de Precisão sofreram um aumento quando em comparação com esta combinação de algoritmos sem a utilização de emoções.

### 5.2.5.10 Detecção de Golos baseada em Áudio, Presença de Grande Área e HRV

Por último são combinados todos os algoritmos com o objetivo de aumentar os valores de Precisão e de Cobertura obtidos anteriormente.

Vídeo	Nº de segmentos com golo	Deteções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	1	0
Vídeo 2	1	1	0	0
Vídeo 3	1	1	0	0
Vídeo 4	1	0	0	1
Vídeo 5	1	1	0	0
Total	5	4	1	1

Tabela 5.39: Resultados da Deteção de golos utilizando Áudio, Grande Área e HRV.

Vídeo	Precisão	Cobertura
Vídeo 1	0.5	1
Vídeo 2	1	1
Vídeo 3	1	1
Vídeo 4	1	0
Vídeo 5	1	1
Conjunto	0.8	0.8

Tabela 5.40: Precisão e Cobertura da deteção de golos utilizando Áudio, Grande Área e HRV.

Esta combinação de algoritmos apresenta resultados semelhantes à combinação anterior, no entanto, nesta combinação ainda não estão a ser utilizadas as emoções, pelo que ainda poderão existir melhorias no valor de Precisão (no valor de Cobertura não existirá melhoria uma vez que já foi possível visualizar que a adição das emoções nos algoritmos anteriores não possibilitou a deteção do golo no vídeo 4).

#### 5.2.5.11 Deteção de Golos baseada em Áudio, Presença de Grande Área e HRV com Emoções

Ao classificador analisado anteriormente são adicionadas as emoções. Este classificador configura a versão do classificador final objetivo desta dissertação.

Vídeo	Nº de segmentos com golo	Deteções Corretas	Falsos Positivos	Falsos Negativos
Vídeo 1	1	1	0	0
Vídeo 2	1	1	0	0
Vídeo 3	1	1	0	0
Vídeo 4	1	0	0	1
Vídeo 5	1	1	0	0
Total	5	4	0	1

Tabela 5.41: Resultados da Deteção de golos utilizando Áudio, Grande Área e HRV com Emoções.

Vídeo	Precisão	Cobertura
Vídeo 1	1	1
Vídeo 2	1	1
Vídeo 3	1	1
Vídeo 4	1	0
Vídeo 5	1	1
Conjunto	1	0.8

Tabela 5.42: Precisão e Cobertura da deteção de golos utilizando Áudio, Grande Área e HRV com Emoções.

Como podemos verificar pela tabela 5.42 a adição das emoções ao algoritmo SVM fez com que fossem removidos todos os falsos positivos da classificação sem introduzir mais falhas de deteção. Como esperado, no entanto, fica a falha de deteção no vídeo 4, o que leva à conclusão que a utilização deste algoritmo aplicado a jogos onde o utilizador não possui especial afinidade com nenhuma das equipas participantes pode fazer com que surjam eventuais falhas de deteção no que toca à deteção de golos.

#### 5.2.5.12 Comparação dos resultados das várias combinações de algoritmos

A tabela 5.43 apresenta um resumo dos valores de Precisão e de Cobertura para todas as combinações de algoritmos utilizados. Como é possível verificar, existe uma tendência ascendente no que toca à Precisão com a adição sucessiva dos algoritmos e das emoções. No entanto verificamos uma redução aquando da adição do algoritmo de HRV pelos motivos já descritos anteriormente.

Algoritmos	Precisão	Cobertura
Áudio	0.143	1
Presença de Grande Área	0.462	1
Áudio + Presença de Grande Área	0.545	1
Áudio + HRV	0.4	0.8
Áudio + HRV com emoções	0.667	0.8
Presença de Grande Área + HRV	0.667	0.8
Presença de Grande Área + HRV com emoções	0.8	0.8
Áudio + Presença de Grande Área + HRV	0.8	0.8
Áudio + Presença de Grande Área + HRV com emoções	1	0.8

Tabela 5.43: Resumo da Precisão e do Cobertura.



# 6

## Conclusões

A presente dissertação resultou de uma necessidade verificada no mundo real em apresentar soluções que permitam lidar com o crescente número de artefactos audiovisuais que existem nos vários meios de comunicação e de criar uma plataforma capaz de captar as informações cardíacas do utilizador e de guardar as anotações relativas às emoções.

Foi implementada uma aplicação para o sistema operativo *Android* destinado a dispositivos móveis. Esta aplicação provou-se bastante robusta e capaz na captação das informações cardíacas provenientes do sensor *Movesense* e das emoções anotadas, na criação dos ficheiros necessários e do envio para a base de dados *Firebase*. Depois de iniciada a gravação do jogo, a aplicação é robusta a quaisquer ações relacionadas com o ciclo de vida de uma aplicação *Android* (por exemplo, se o utilizador carregar no botão *home* a aplicação é colocada em pausa mas no entanto a gravação continua até que o utilizador a termine manualmente).

Esta aplicação permite então a captura das informações necessárias a serem utilizadas posteriormente na sumarização de vídeos. A captura destas informações é realizada de uma forma simples e sem envolver a utilização de equipamentos intrusivos que induzem no utilizador *stress* ou emoções negativas que possam comprometer de forma negativa os resultados finais do sistema.

Foi também implementada uma aplicação de análise de vídeos que permite o carregamento de um vídeo, a sua segmentação e a visualização de quais os segmentos que são relevantes. Como parte desta aplicação foi implementada uma

interface de teste que permite verificar também o funcionamento isolado de cada um dos algoritmos que compõem esta aplicação.

Estes algoritmos foram testados de forma exaustiva utilizando vários excertos de jogos de futebol. O peso computacional dos vários algoritmos não permite que o teste ocorra para jogos completos (o algoritmo de segmentação apresenta o maior peso computacional).

Os testes foram divididos em duas partes: a detecção de cartões e a detecção de golos. Na detecção de cartões foram utilizados os algoritmos de detecção de cartões isoladamente e depois em combinação com o algoritmo de características sonoras. Foi possível verificar uma pequena melhoria com a adição do algoritmo de características sonoras. Infelizmente apenas foi possível o teste com cartões amarelos uma vez que nos jogos gravados não ocorreu qualquer cartão vermelho. Não foi utilizado o algoritmo de HRV na detecção de cartões porque, após uma análise aos intervalos RR gerados pela aplicação *Android*, foi possível verificar que não existem alterações significativas aquando da amostragem de cartões (no contexto do sujeito submetido à gravação).

Foram realizados testes à detecção de golos utilizando combinações dos vários algoritmos. Os valores de Precisão e de *Recall* resultantes destes testes sofreram uma trajetória ascendente com a adição sucessiva dos vários algoritmos, principalmente com a adição do algoritmo que utiliza a informação HRV. No entanto, com a realização dos testes foi possível perceber que a natureza subjetiva da informação HRV pode levar a que ocorram falhas na detecção. Esta situação pode ocorrer quando o sujeito não possui especial afinidade por nenhuma das equipas participantes no jogo ou quando o golo não tem grande importância (por exemplo quando a equipa já se encontra a vencer o jogo por muitos golos). A adição do HRV permitiu no entanto eliminar por completo os falsos positivos, ou seja, situações em que o sistema definia a existência de golo mas onde não existia qualquer golo.

## **6.1 Integração da solução proposta em soluções existentes no mercado**

Uma das soluções presentes no mercado para a análise de jogos de futebol é a Hudl (<https://www.hudl.com/sports/soccer>). A Hudl permite que sejam carregados vídeos de jogos completos no sistema para que posteriormente uma equipa

de analistas efetue uma análise completa gerando as estatísticas aprofundadas de jogo (percentagens de posse de bola ou o número de remates de cada equipa). Neste contexto o sistema proposto nesta dissertação poderia facilitar a tarefa dos analistas ao indicar em antemão os lances do jogo onde ocorreriam lances relevantes (golos ou cartões).

Outra integração bastante interessante deste sistema no mundo real seria num contexto de produção de resumos para os jogos de futebol. Um clube poderia, por exemplo implementar sensores cardíacos nas cadeiras do estádio ou utilizando dispositivos *wearables* para obter uma média da informação de HRV de todos os espetadores, o que permitiria utilizar essa informação em conjunto com as informações audiovisuais para depois gerar os resumos do jogos de forma automática (isto se o clube tiver controlo sobre as transmissões televisivas em casa, como é o caso em Portugal do Sport Lisboa e Benfica).

## 6.2 Trabalho Futuro

Apesar de serem efetuados os dois testes separados para cartões e golos, o sistema de sumarização de vídeos apenas faz a classificação dos planos em planos relevantes ou não relevantes (ou seja, os resultados dos dois algoritmos são unidos para gerar o resultado final onde são indicados os planos relevantes). Talvez no futuro seria interessante alargar o sistema para efetuar uma classificação multi-classe em que fosse possível a deteção de faltas ou penaltys utilizando outros algoritmos para o efeito (por exemplo verificando a densidade de transições existentes no excerto).

Ao nível da aplicação *Android* existe espaço para várias melhorias no que toca à interface gráfica uma vez que não foi efetuado um refinamento visual da mesma. A secção de visualização de gravações poderia ser melhorada para, ao invés de permitir apenas a visualização, permitir também a manipulação das gravações (como apagar ou alterar o seu nome).

No que toca à aplicação de geração de características HRV (que foi implementada como uma aplicação de consola) seria interessante a implementação de uma interface gráfica que permita uma experiência mais simples e intuitiva no que toca ao *download* dos ficheiros provenientes do *Firebase* e ao seu conseqüente processamento para a geração dos ficheiros de HRV.

Dada a natureza subjetiva dos dados utilizados na classificação HRV, no futuro seria necessário também incluir mais utilizadores no processo de gravação. Isto

permitiria inferir a validade das conclusões alcançadas com esta dissertação e assim perceber se os resultados obtidos são consistentes entre indivíduos.

# Referências

- [1] Zero crossing detector, 2003. URL <https://homepages.inf.ed.ac.uk/rbf/HIPR2/zeros.htm>. (p. 15)
- [2] Overfitting in machine learning: What it is and how to prevent it, September 2017. URL <https://elitedatascience.com/overfitting-in-machine-learning>. (p. 93)
- [3] Understanding k-means clustering in machine learning, September 2018. URL <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>. (p. 19)
- [4] Gabor filter, February 2019. URL [https://en.wikipedia.org/wiki/Gabor\\_filter](https://en.wikipedia.org/wiki/Gabor_filter). (not cited)
- [5] Erhan Alparslan. Componentwise edge detection by Laplacian operator masks. April 1980. (p. 14)
- [6] I. Boudway. Soccer is the world's most popular sport and still growing, June 2018. URL <https://www.bloomberg.com/news/articles/2018-06-12/soccer-is-the-world-s-most-popular-sport-and-still-growing>. (p. 1)
- [7] Marcelo Campos. Heart rate variability: A new way to track well-being, November 2017. URL <https://www.health.harvard.edu/blog/heart-rate-variability-new-way-track-well-2017112212789>. (pp. 2 e 4)
- [8] Prithwi Raj Chakraborty. Detecting viewer interest in video using facial and heart rate responses (dissertação de doutoramento). 2017. (p. 32)

- [9] Samuel F. de Sousa Júnior et al. An overview of automatic event detection in soccer matches. *IEEE Workshop on Applications of Computer Vision (WACV)*, 2011. (pp. 7 e 11)
- [10] Prithwi Raj Chakraborty e Dian Tjondronegoro. Using viewer's facial expression and heart rate for sports video highlight detection. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015. (pp. xvii, 32, e 33)
- [11] Ganesh I. Rathod e Dipali A. Nikam. Review on event retrieval in soccer video. *(IJCSIT) International Journal of Computer Science and Information Technologies*, 2014. (pp. 25 e 28)
- [12] Sachan P. Rajendra e Dr. Keshaveni N. A survey of automatic video summarization techniques. *International Journal of Electronics, Electrical and Computational System*, 2014. (p. 11)
- [13] João Cunha e Hugo Graça. Detecção de jogadas relevantes em jogos de futebol (projeto final de licenciatura). 2013. (pp. 23, 52, 96, e 100)
- [14] Musfequs Salehin e Manoranjan Paul. Affective video events summarization using EMD decomposed EEG signals (EDES). *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2017. (pp. xvii, 29, e 31)
- [15] Ahmet Ekin e Murat Tekalp. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 2003. (pp. xvii, 11, 14, 15, 17, 18, 24, 25, e 57)
- [16] L. Lucchese e S.K. Mitra. Color Image Segmentation: A State-of-the-Art Survey. January 2001. (p. 1)
- [17] Maheshkumar H. Kolekar e Somnath Sengupta. Bayesian network-based customized highlight generation for broadcast soccer videos. *IEEE Transactions on Broadcasting*, 2015. (pp. xvii, 19, 21, 22, 23, 24, e 59)
- [18] Yasmin S. Khan e Soudamini Pawar. Video summarization: Survey on event detection and summarization in soccer videos. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 2015. (p. 11)
- [19] Glenn E. Krasner e Stephen T. Pope. A description of the model-view-controller user interface paradigm in the smalltalk80 system, January 1988. URL <https://www.researchgate.net/>

- [publication/239452280\\_A\\_Description\\_of\\_the\\_Model-View-Controller\\_User\\_Interface\\_Paradigm\\_in\\_the\\_Smalltalk80\\_System.](#)  
(p. 49)
- [20] Chakshu Puri e Sukhwinder Singh. Image segmentation and classification - a review. *International Journal of Innovative Research in Science, Engineering and Technology*, 2014. (p. 1)
- [21] Amjad Rehman e Tanzila Saba. Features extration for soccer video semantic analysis: current achievements and remaining issues. *Journal Artificial Intelligence Review*, 2012. (p. 11)
- [22] Ahmet Ekin. Sports video processing for description, summarization and search (dissertação de doutoramento). 2003. (pp. 15, 17, 19, 20, e 24)
- [23] Adrián Colomer Granero et al. A comparison of physiological signal analysis techniques and classifiers for automatic emotional evaluation of audiovisual contents. *Frontiers in computational neuroscience*, 2016. (p. 2)
- [24] Angela Zeiler et al. Empirical mode decomposition - an introduction. *International Joint Conference on Neural Networks (IJCNN)*, 2010. (p. 30)
- [25] Barkha Bhavsar et al. Corner event detection based soccer video summarization. *International Journal Of Current Engineering and Scientific Research (IJCESR)*, 2015. (p. 26)
- [26] Chuguang Wei et al. Affective emotion increases heart rate variability and activates left dorsolateral prefrontal cortex in post-traumatic growth. *Scientific reports*, 2017. (p. 2)
- [27] Evgenia Hristova et al. Biosignal based emotion analysis of human-agent interactions. *International Conference on Cross-Modal Analysis of Speech, Gestures, Gaze and Facial Expressions*, 2009. (p. 6)
- [28] Fisher R. et al. Hough transform, 2003. URL <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>. (pp. 14 e 58)
- [29] Hong Shao et al. Shot boundary detection algorithm based on HSV histogram and HOG feature. *International Conference on Advanced Engineering Materials and Technology*, 2015. (pp. 27 e 55)

- [30] Hossam M. Zawbaa et al. Svm-based soccer video summarization system. *Third World Congress on Nature and Biologically Inspired Computing*, 2011. (pp. xvii, 17, 19, 20, 24, 27, e 28)
- [31] Konstantinos Georgiou et al. Can wearable devices accurately measure heart rate variability? a systematic review. *Folia Med (Plovdiv)*, 2018. (p. 4)
- [32] Lizawati Salahuddin et al. Ultra short term analysis of heart rate variability for monitoring mental stress in mobile settings. *IEEE Engineering in Medicine and Biology Society Annual Conference*, 2007. (p. 5)
- [33] Martin Ragot et al. Emotion recognition using physiological signals: Laboratory vs. wearable sensors. *International Conference on Applied Human Factors and Ergonomics*, 2017. (p. 7)
- [34] Mingzhe Jiang et al. Ultra-short-term analysis of heart rate variability for real-time acute pain monitoring with wearable electronics. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017. (pp. 2 e 4)
- [35] Mohamad-Hoseyn Sigari et al. Fast highlight detection and scoring for broadcast soccer video summarization using on-demand feature extraction and fuzzy inference. *International Journal of Computer Graphics*, 2015. (p. 28)
- [36] Rosalind W. Picard et al. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001. (p. 6)
- [37] Said Jai-Andaloussi et al. Soccer events summarization by using sentiment analysis. *International Conference on Computational Science and Computational Intelligence*, 2015. (pp. xvii e 25)
- [38] Sandeep Sripada et al. Summarization of sports videos based on unsupervised deep learning. 2013. (pp. xvii, 24, e 27)
- [39] Yannick Benezeth et al. Remote heart rate variability for emotional state monitoring. *2018 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2018. (p. 6)
- [40] Leandro Pecchia et all. Are ultra-short heart rate variability features good surrogates of short-term ones? state-of-the-art review and recommendations. *Healthcare Technology Letters*, 2018. (p. 4)

- [41] Rui Fortio. Segmentação e sumarização de vídeos de futebol (projeto final de licenciatura). 2005. (pp. xvii, 13, 17, e 24)
- [42] Alan Hanjalic. Generic approach to highlights extraction from a sport video. *International Conference on Image Processing*, 2003. (p. 16)
- [43] Jennifer Healey. Physiological sensing of emotion. *The Oxford Handbook of Affective Computing*, 2014. (pp. 3 e 6)
- [44] Brownlee J. Supervised and unsupervised machine learning algorithms, March 2016. URL <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>. (p. 61)
- [45] Renu Khandelwal. K fold and other cross-validation techniques, November 2018. URL <https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e>. (p. 93)
- [46] Jason Moore. Heart rate variability vs. heart rate, June 2016. URL <https://hrvcourse.com/heart-rate-variability-vs-heart-rate/>. (pp. 4 e 84)
- [47] Gandhi R. Support vector machine - introduction to machine learning algorithms, June 2018. URL <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>. (p. 61)
- [48] Huang-Chia Shih. A survey on content-aware video analysis for sports. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. (p. 11)
- [49] Dharmesh Tank. A survey on sport video summarization. *International Journal for Science and Advance Research in Technology (IJSART)*, 2016. (p. 11)
- [50] Egon L. van den Brock et al. Affective man-machine interface: Unveiling human emotions through biosignals. *International Joint Conference on Biomedical Engineering Systems and Technologies*, 2009. (pp. 4 e 7)

