



ISEL – INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
ADEETC – ÁREA DEPARTAMENTAL DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA
UNIDADE CURRICULAR DE PROJETO

Sistema Anotação de Vídeo com dados fisiológicos para análise de emoções

Maria Franco (46320)

Miguel Peixoto (45302)

Orientadores

Professor Doutor André Ribeiro Lourenço

Professor Doutor Pedro Mendes Jorge

setembro, 2021

Resumo

A análise automática de vídeo e a inteligência artificial são ferramentas utilizadas em diversos cenários no quotidiano, e começam a surgir os primeiros sistemas capazes de detetar emoções através da análise de expressões faciais. Como complemento, a utilização de alguns sensores permite medir sinais fisiológicos que também podem revelar mudanças no estado emocional do ser humano.

Estes sistemas de deteção de emoções são requisitados nas mais diversas áreas, como por exemplo, para a criação de sistemas de análise de linguagem corporal, psicologia comportamental ou de suporte a entrevistas ou inquirições.

Este projeto visa a criação de uma aplicação onde o utilizador possa realizar anotações manuais e automáticas em vídeos captados em tempo real ou em vídeos que se encontrem num repositório, alojado num sistema de ficheiros.

Com recurso a dois clientes, um que seja munido de uma câmara e de uma placa de sensores, e outro que pretenda anotar e visualizar o vídeo no serviço *web* em tempo real, a comunicação entre ambos é feita pela ligação a um servidor, com o processamento de dados e o suporte às páginas *web* do sistema como as suas principais funções. O servidor tem também a capacidade de interagir com uma base de dados e com o seu sistema de ficheiros.

A linguagem de programação predominante, escolhida para o processamento e comunicação entre clientes e servidor, utilizando *Websockets*, é *Python*, sendo utilizada a *framework Flask* para a componente do servidor *web*. Para a deteção de emoções, recorreu-se à biblioteca *Deepface* e para captar os sinais fisiológicos, utiliza-se o *wearable EmotiBit*.

O projeto demonstra uma prova de conceito completa e fundamentada e o sistema implementado mostra-se eficiente e robusto.

Índice

Resumo	i
Índice	iii
Lista de Abreviaturas	vi
Lista de Tabelas	vii
Lista de Figuras	ix
Lista de Listagens	xi
1 Introdução	1
2 Trabalho Relacionado	5
2.1 iMotions	5
2.2 HDF5	6
2.3 MPEG-7	7
3 Modelo Proposto	11
3.1 Requisitos	11
3.1.1 Funções do Sistema	11
3.1.2 Atributos do Sistema	12
3.1.3 Casos de Utilização	13
3.2 Diagrama de Blocos	15
3.3 Fundamentos	16
3.3.1 Classificação de Emoções	16
3.3.2 Processamento de Vídeo	19
3.3.3 Redes Neuronais Convolucionais	19

3.3.4	Servidor <i>Web</i>	21
3.3.5	<i>Threads</i>	22
3.3.6	<i>Sockets</i>	22
3.3.7	<i>Websocket</i>	22
4	Implementação do Modelo	25
4.1	Arquitetura do Sistema	25
4.1.1	Cliente “Observado”	26
4.1.2	Cliente “Observador”	26
4.1.3	Servidor	27
4.2	Tecnologias Utilizadas	27
4.2.1	<i>Python</i>	27
4.2.2	<i>Frameworks Flask e Socket.IO</i>	28
4.2.3	<i>DeepFace</i>	29
4.2.4	Bibliotecas <i>Intel RealSense SDK 2.0</i> e <i>PyAudio</i>	30
4.2.5	<i>MySQL</i>	31
4.2.6	<i>EmotiBit</i>	31
4.3	Abordagem	33
4.3.1	Comunicação Cliente “Observado” — Servidor	33
4.3.2	Retransmissão para o Cliente “Observador”	36
4.3.3	Interação com a <i>EmotiBit</i>	38
4.3.4	Processamento do sinal PPG	39
4.3.5	Processamento de Emoções com <i>Deepface</i>	42
4.3.6	Páginas <i>web</i> e suas funcionalidades	44
5	Validação e Testes	51
5.1	Validação do algoritmo com <i>Deepface</i>	51
5.2	Validação do algoritmo com <i>EmotiBit</i>	54
6	Conclusões e Trabalho Futuro	57
A	Gestão de Versões	59
B	Diagrama de Robustez	61
	Bibliografia	63

Listas de Abreviaturas

API *Application Programming Interface.* 7, 28

ASCII *American Standard Code for Information Interchange.* 35

BPM Batimentos por minuto. 39, 41

BVP *Blood Volume Pulse.* 18

CNN *Convolutional Neural Network.* ix, 19, 20, 21

CSV *Comma Separated Values.* 32, 39

ECG Eletrocardiografia. 17

EDA *Electrodermal activity.* 17, 31, 32

EEG Eletroencefalografia. 18

EMG Eletromiografia. 18

EOG Eletrooculografia. 19

FFMPEG *Fast Forward MPEG.* 38

GSR *Galvanic Skin Response.* 18, 31, 32

HDF5 *Hierarchical Data Format.* ix, 6, 7

HTTP *HyperText Transfer Protocol.* 21, 28

IP *Internet Protocol.* 34

IROG *Infrared oculography.* 19

JPEG *Joint Photographic Experts Group.* 38

JSON *JavaScript Object Notation.* 26, 27, 32, 33, 34, 38, 48, 49

LAN *Local Area Network.* 31

MJPG *Motion JPEG.* 38

MPEG-7 *Multimedia Content Description Interface.* ix, 7, 8, 9

PPG *Photoplethysmography.* 18, 31, 32

RESP Respiração. 18

SCR *Skin Conductance Response.* 17

SD *Secure Digital Card.* 31, 39

SDK *Software Development Kit.* 30

SQL *Structured Query Language.* 31

Temp ou SKT Temperatura. 18

UDP *User Datagram Protocol.* 26, 33, 34

XML *eXtensive Markup Language.* 8, 9

Lista de Tabelas

3.1	Funções do Sistema	12
3.2	Atributos do Sistema	12
3.3	Caso de Utilização 1: Gravar	13
3.4	Cenário Principal do Caso de Utilização 1	14
3.5	Caso de Utilização 2: Editar	14
3.6	Cenário Principal do Caso de Utilização 2	14
3.7	Cenário Alternativo do Caso de Utilização 2	15

Listas de Figuras

2.1	Plataforma <i>iMotions</i>	6
2.2	Esquema representativo da organização de um ficheiro HDF5 [National Ecological Observatory Network, 2020]	7
2.3	Esquema representativo da utilização dos componentes da norma MPEG-7 [Bimbo, 2011]	8
2.4	Esquema representativo dos constituintes integrantes da norma MPEG-7 [Dallacosta et al., 2004]	9
3.1	Diagrama de casos de utilização	13
3.2	Diagrama de blocos do sistema	15
3.3	Esquema representativo do modelo de emoções de Plutchik [Bota et al., 2019]	17
3.4	Visão geral da estrutura e o processo de treinar uma CNN . .	21
4.1	Arquitetura do projeto	25
4.2	Exemplo de extração de características faciais para várias ex- pressões, [Serengil e Ozpinar, 2020]	30
4.3	Funcionamento da aplicação <i>EmotiBit Oscilloscope</i>	32
4.4	Diagrama representativo da estrutura de um datagrama	34
4.5	Diagrama de atividades da <i>thread</i> responsável pela receção, gravação e retransmissão dos dados audiovisuais	37
4.6	Sinal PPG com intervalos R-R para o cálculo da variabilidade do batimento cardíaco [Yang et al., 2018]	39
4.7	Sinal PPG proveniente do conjunto de dados [Siam, 2019] . .	40
4.8	Quadrado da derivada do sinal PPG	40
4.9	Quadrado da derivada do sinal PPG após a aplicação de um limiar	41

4.10	Diagrama de atividades da <i>thread</i> responsável pela criação de anotações provenientes das emoções detetadas pela <i>Deepface</i>	43
4.11	Página de Registo	44
4.12	Página de <i>Login</i>	45
4.13	Página de Atualização de Dados do Utilizador	45
4.14	Página de <i>Upload</i> de novos vídeos	46
4.15	Página de Gravação de vídeos	47
4.16	Página de vídeos do utilizador	48
4.17	Página de Revisão de vídeos	49
5.1	Comparação temporal e de resultados da deteção de emoções em imagens entre os modelos de deteção disponíveis na biblioteca <i>Deepface</i>	51
5.2	Gráfico de relação entre emoções classificadas e o grau de confiança obtido	52
5.3	Matriz de confusão dos resultados obtidos da classificação de emoções num vídeo	53
5.4	Sinal PPG captado pelo sensor de luz verde da <i>EmotiBit</i>	54
5.5	Sinal PPG captado pelo sensor infravermelho da <i>EmotiBit</i>	54
5.6	Sinal PPG captado pelo sensor de luz vermelha da <i>EmotiBit</i>	55
5.7	Resposta da Atividade Eletrodérmica captada pela <i>EmotiBit</i>	55
5.8	Temperatura corporal captada pela <i>EmotiBit</i>	56
A.1	Estrutura de acesso ao repositório no <i>Github</i>	59
A.2	Diagrama de fluxo do repositório	60
B.1	Diagrama de robustez	61

Lista de Listagens

Capítulo 1

Introdução

A preocupação em agilizar processos de deteção de ocorrências em grandes volumes de informação é crescente. É necessário integrar, nesses processos, ferramentas digitais que o façam de forma rápida, automática e eficaz, de modo a substituir o trabalho moroso de um analista dedicado. Nestas ferramentas é essencial a incorporação de algoritmos munidos de inteligência artificial e aprendizagem automática.

Nas últimas décadas, tem sido feito um elevado investimento em investigação nos ramos dedicados ao estudo das emoções. Este campo, de elevada complexidade, tem sido requisitado por ciências como a psicologia, sociologia, a medicina e a computação, por exemplo, para tópicos emergentes como a computação afetiva.

Fisiologicamente, o ser humano tende a expressar o seu estado mental para o exterior através da sua postura corporal. A análise da expressão facial esboçada e das alterações no funcionamento autónomo do sistema nervoso simpático e parassimpático permitem a dedução das emoções básicas que um humano possa estar a sentir.

Este projeto surge da necessidade da criação de uma aplicação onde o utilizador possa, através de um serviço *web*, armazenar vídeos com vários dados capturados em tempo-real, e criar, no momento da gravação ou posteriormente, anotações associadas a instantes temporais dos vídeos. Estas anotações poderão conter dados relativos às emoções visíveis na pessoa a ser observada no vídeo, com a possibilidade de serem geradas automaticamente, bem como informação textual livre referente, por exemplo, a acontecimentos ocorridos em determinados momentos do vídeo. Este projeto apresenta um

potencial elevado ao nível de integração noutras aplicações.

As informações a extraír do elemento observado, aquando da captura do vídeo, serão feitas através da câmara *Intel RealSense* e do *wearable EmotiBit*. A câmara será responsável por extraír a imagem e o áudio. A extração dos dados fisiológicos do observado é da responsabilidade do vestível *EmotiBit*.

Toda a metainformação proveniente das anotações dos vídeos será armazenada numa base de dados e as gravações serão armazenadas num repositório de vídeos.

De modo a concretizar este projeto, foram traçados os objetivos para este, as possíveis aplicações do sistema e as metas a alcançar no seu desenvolvimento. Por conseguinte, identificaram-se os casos de utilização, os atores, funções e atributos do sistema para a construção de cenários para cada caso de utilização. Em sequência com o método de desenvolvimento anterior, construiu-se o diagrama de blocos do sistema para o auxílio na implementação do sistema.

No geral, o sistema será composto por uma máquina responsável por receber os dados extraídos e enviá-los para o servidor, pela transmissão do vídeo captado ao cliente (observador) e ao servidor para a gravação no repositório de vídeos, pelo processamento de emoções e pelo registo de anotações na base de dados.

Este trabalho considera-se completo com o bom funcionamento do serviço *web*, da transmissão dos dados adquiridos do observado para o cliente observador e para o servidor de forma eficiente e bem sucedida, e do armazenamento bem estruturado dos dados na respetiva base de dados e no repositório de vídeos. A escolha do algoritmo para análise de emoções baseou-se no resultado proveniente de testes o mais satisfatório possível em relação à sua rapidez e precisão.

Este relatório estará dividido em 6 (seis) capítulos. O primeiro será o presente, referente à introdução. O segundo exporá o trabalho relacionado, destacando as principais funcionalidades. O terceiro apresentará o modelo proposto, com enfoque aos requisitos e fundamentos necessários. Do quarto capítulo constará a fundamentação relativa à arquitetura, tecnologias e abordagem escolhidas para a implementação do modelo. O quinto capítulo conterá os testes e validações realizadas aos componentes integrados no sistema. Por fim, no sexto capítulo encontrar-se-ão as conclusões e trabalho futuro

relativamente ao projeto desenvolvido.

Capítulo 2

Trabalho Relacionado

Neste capítulo são apresentadas tecnologias que se inserem nas áreas de interesse deste projeto. Estas tecnologias serviram de base representativa para o desenvolvimento do projeto corrente.

2.1 iMotions

A *iMotions* é uma plataforma de análise integrada que pretende executar pesquisas de comportamento humano com alta validade. Esta integra e sincroniza vários sensores biométricos que fornecem diferentes percepções humanas, como *eye tracking*, sudação da pele e análise de expressões faciais. A plataforma agrupa todas as tecnologias de *hardware* essenciais e os seus respetivos dados num único curso, [iMotions, 2005].

De momento, esta plataforma cobre uma vasta dimensão de áreas aplicacionais, mais especificamente, nas áreas da psicologia, educação e saúde.

Esta plataforma identifica-se bastante aos requisitos para o projeto corrente, visto que o objetivo principal do projeto é a deteção e identificação de emoções com base nas expressões faciais e nos dados biométricos de um utilizador.



Figura 2.1: Plataforma *iMotions*

A Figura 2.1 representa a *interface* visual da aplicação. Como se pode observar, os gráficos traçam os valores dos dados biométricos no decorrer da experiência e são listados para cada dado biométrico captado.

2.2 HDF5

O ficheiro *Hierarchical Data Format* (HDF5) é uma *toolbox* exclusiva de alto desempenho que consiste num modelo de dados abstratos, biblioteca e formato de arquivo para armazenar e gerir coleções de dados extremamente grandes e/ou complexas. A tecnologia é usada a nível mundial pelo governo, indústria e academia numa ampla variedade de disciplinas de ciências, engenharia e negócios, [The HDF Group, 2006].

A Figura 2.2 apresenta um esquema representativo da organização interna de um ficheiro HDF5. Este ficheiro é constituído por duas categorias:

- Grupos que se comportam como diretórios, onde é possível inserir outros grupos ou conjuntos de dados;
- Conjuntos de dados (*datasets*) que representam a informação a armazenar no ficheiro.

É de se notar que os conjuntos de dados não necessitam de estar dentro de um grupo podendo, assim, serem armazenados na raiz do ficheiro. Não obs-

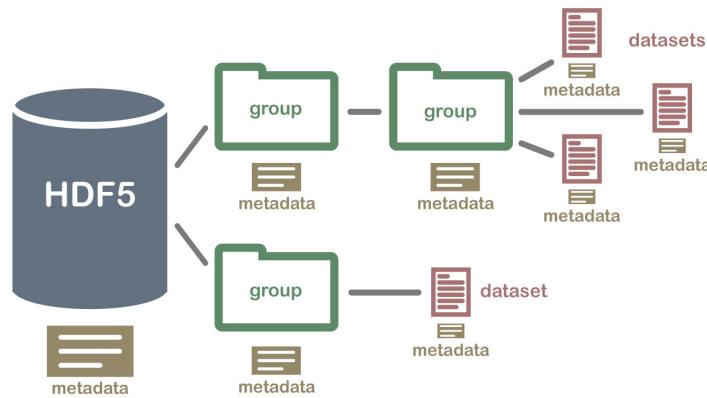


Figura 2.2: Esquema representativo da organização de um ficheiro HDF5 [National Ecological Observatory Network, 2020]

tante, qualquer objeto, incluindo a raiz, pode ter meta informação associado a esse mesmo objeto, como consta na Figura 2.2.

Outras características relevantes deste ficheiro são o facto de suportar dados heterogéneos (dados de naturezas distintas) sem restrições no tamanho da coleção de objetos, e esta tecnologia tem suporte de APIs multiplataforma.

Este ficheiro apresenta funcionalidades úteis ao projeto, dada a necessidade de armazenar anotações (equiparáveis aos meta data) sobre as emoções detetadas/introduzidas, com o armazenamento de vídeos que podem, ao nível de espaço em memória, ocupar bastante.

2.3 MPEG-7

O MPEG-7 é uma norma de descrição de conteúdos multimédia, isto é, define a estrutura, a composição e a descrição de ficheiros deste tipo.

O que distingue este formato dos restantes é a possibilidade do armazenamento de metadados, associados a um instante temporal, para etiquetar, ou anotar, eventos ocorridos em determinados instantes em vídeos e/ou áudio. Para além desta possibilidade, outra principal motivação para a sua criação foi a interoperabilidade entre aplicações, sendo uma norma, pela definição de uma sintaxe comum.

Como o esquema da Figura 2.3 demonstra, o processo de extração de características, cujas categorias são definidas pelos descritores, pode ser feito de

forma manual ou automática, que podem ser diversas, consoante se trate de um ficheiro de texto, de imagem ou de áudio. “Os descritores definem a sintaxe e a semântica dos recursos do conteúdo audiovisual. Há diferentes níveis de abstração a serem tratados pelo MPEG-7. No baixo nível de abstração, os descritores podem incluir forma, movimento, textura, cor e movimento da câmara para imagens/vídeos, energia, harmonia e timbre para áudio. No alto nível de abstração, os descritores podem incluir eventos, — conceitos abstratos, géneros de conteúdo, etc. Os descritores audiovisuais representam recursos específicos relacionados ao conteúdo audiovisual, respetivamente. Os descritores genéricos tratam de recursos genéricos.”, [Chang et al., 2001]. Já os esquemas de descrição permitem a construção de descrições complexas, especificando a estrutura e semântica das relações entre os Ds (*descriptors*) ou DSs (*description schemas*) constituintes, [Chang et al., 2001]. A descrição é composta pelos DSs sendo escrita em formato XML, e é nesse formato armazenada. A transmissão é feita em formato binário. Estes elementos estão representados na Figura 2.4.

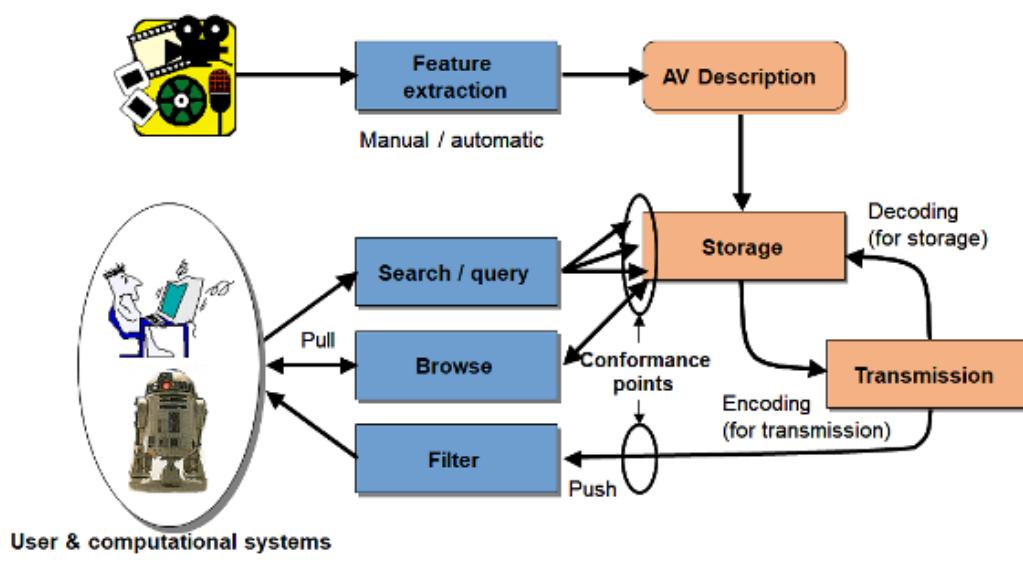


Figura 2.3: Esquema representativo da utilização dos componentes da norma MPEG-7 [Bimbo, 2011]

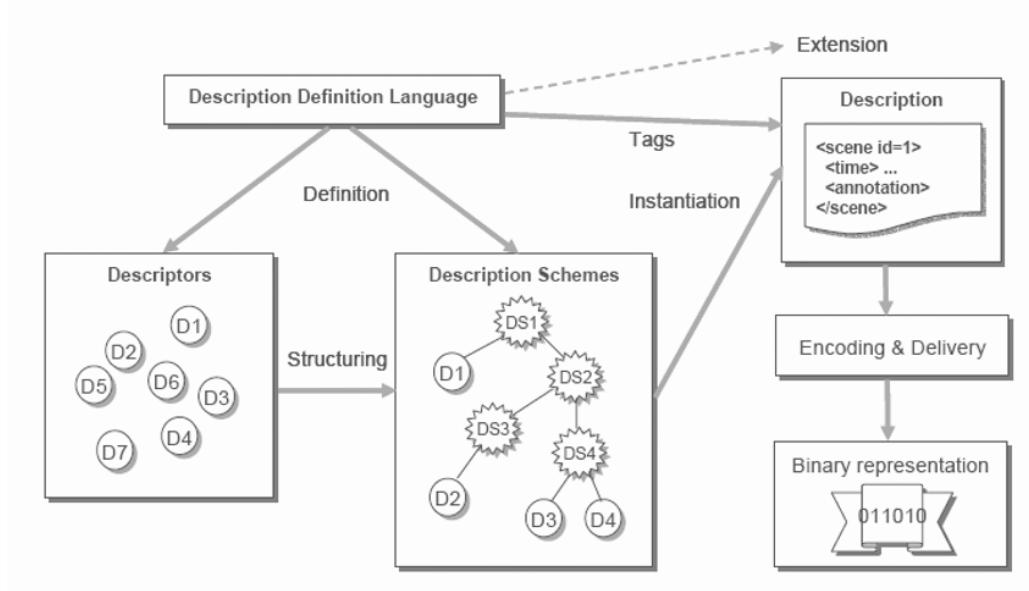


Figura 2.4: Esquema representativo dos constituintes integrantes da norma MPEG-7 [Dallacosta et al., 2004]

Este formato permitiria uma pesquisa mais eficiente, por palavras-chave, que retornasse a parte do ficheiro a que tivessem sido associados. As suas aplicações poderiam ser úteis à educação, medicina, a bibliotecas digitais, jornalismo, indústria audiovisual, entretenimento, cultura, investigação criminal e sistemas de informação geográfica. [Teuhola, 2012]

Embora o XML seja considerado um dos formatos mais compatíveis para comunicação entre diversas linguagens de programação, por exemplo, foi o que ditou o insucesso desta norma. O facto de os dados acessórios serem armazenados em XML implica a sua validação, recorrendo ao uso de *XML Schema* (XSD) em que a DDL (*Description Definition Language*) é escrita.

Esta norma tem características semelhantes às requeridas para o projeto, pois permite associar, a instantes de vídeos, eventos tais como emoções.

Capítulo 3

Modelo Proposto

Neste capítulo, pretende-se mostrar o modelo proposto para a elaboração deste projeto, dos requisitos à descrição e aplicação de fundamentos teóricos e tecnológicos necessários à sua realização.

3.1 Requisitos

Nesta secção, estão incluídos os requisitos funcionais e não funcionais, ou seja, as funções e os atributos do sistema. Seguidamente, serão apresentados e descritos os casos de utilização que deverão ser possíveis de reproduzir no sistema implementado.

3.1.1 Funções do Sistema

As funções do sistema são as funcionalidades que o sistema deve permitir ao utilizador realizar. Neste caso, como se expõe na Tabela 3.1, o sistema deve permitir ao utilizador a captura ou carregamento de novos vídeos, a captura de dados fisiológicos pela ligação ao *wearable* munido de sensores, a deteção de emoções automática, proveniente das expressões faciais observadas nos vídeos, o armazenamento destes dados e das anotações que lhes forem associadas e a revisão posterior dos mesmos para análise ou edição.

Tabela 3.1: Funções do Sistema

Ref. (#)	Função	Categoria
R1.1	Obtenção de vídeo	Visível
R1.2	Obtenção de dados fisiológicos	Visível
R1.3	Extração de características faciais	Invisível
R1.4	Armazenamento de dados extraídos	Invisível
R2.1	Anotações em instantes do vídeo	Visível
R2.2	Análise das emoções registadas manualmente nos vídeos	Invisível
R2.3	Estimação de emoções	Invisível
R3.1	Apresentação ao utilizador do vídeo e das anotações	Visível

3.1.2 Atributos do Sistema

Os atributos do sistema são as características que o sistema deve “ser”, podendo estar relacionadas com a interação homem-máquina, a precisão das suas funções e as plataformas com que é compatível. Como é apresentado na Tabela 3.2, o sistema deve ser *user-friendly*, ou seja, ser adaptado às características e preferências usuais do ser humano e o mais intuitivo possível, permitindo que adicione novos dados ao sistema em várias situações. Deve também ser eficiente, a nível das suas funcionalidades de medição e estimação, e ser compatível com os sistemas operativos mais comuns.

Tabela 3.2: Atributos do Sistema

Atributo	Detalhe / Restrição Fronteira	Categoria
Interação Homem-Máquina	É interativo com o utilizador Permite anotar em tempo real Interface agradável e simples	Obrigatório
Precisão	Ser acessível ao utilizador (não são necessários conhecimentos de programação) Fácil de memorizar	Desejável
Plataformas	É preciso nas medições e características A estimação de emoções tem uma taxa de sucesso elevada Compatível com os Sistemas Operativos: <i>Windows, MacOS e Linux</i>	Desejável

3.1.3 Casos de Utilização

Os casos de utilização permitem compreender melhor as principais funcionalidades e conhecer os vários intervenientes, denominados agentes, do sistema. A Figura 3.1 permite agrupar as principais ações e agentes que se podem envolver no sistema.

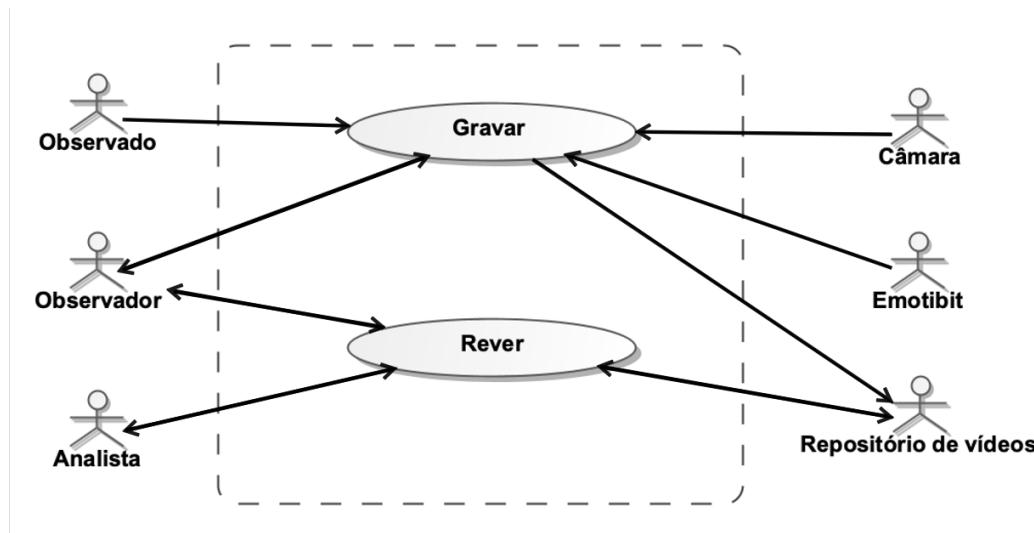


Figura 3.1: Diagrama de casos de utilização

Na Tabela 3.3, encontra-se a descrição do primeiro caso de utilização visível no diagrama que consta na Figura 3.1, neste caso a gravação de um vídeo, como, por exemplo, um interrogatório, com o respetivo resumo do que se trata e a referência às funções do sistema descritas na Subsecção 3.1.1 necessárias para o seu funcionamento. Este caso tem um cenário principal exposto na Tabela 3.4.

ID:	1	Nome:	Gravar interrogatório
Resumo:	Um Observador grava o Observado e pode realizar anotações em instantes temporais concretos do vídeo.		
Refs:	R1.1, R1.2, R1.3, R1.4, R2.1, R2.2		

Tabela 3.3: Caso de Utilização 1: Gravar

Cenário Principal		
Ação do Ator		Resposta do sistema
1 Este Caso de Utilização inicia-se quando o Observado se encontra dentro da visão da câmara		
2 O Observador procede à gravação da entrevista/inquérito	3	Armazena informação facial e fisiológica do Observado
	4	Envia uma <i>stream</i> do vídeo captado pela câmara ao Observador
5 O Observador anota em tempo real	6	Armazena as anotações durante a gravação no servidor
7 O Observador termina a gravação da entrevista/inquérito	8	O sistema envia uma mensagem de <i>feedback</i>
	9	O sistema processa a informação facial e fisiológica do observando
	10	Envia para uma base de dados a gravação em conjunto com a metainformação
11 O Observado sai do campo de visão da câmara		

Tabela 3.4: Cenário Principal do Caso de Utilização 1

Já na Tabela 3.5, encontra-se a descrição do segundo caso de utilização visível no diagrama da Figura 3.1, neste caso a revisão de anotações num vídeo, como, por exemplo, um interrogatório com, novamente, o respetivo resumo do que se trata e a referência às funções do sistema descritas na Subsecção 3.1.1 necessárias para o seu funcionamento. Este caso tem um cenário principal exposto na Tabela 3.6. Tem ainda um caso alternativo para descrever a edição das anotações presentes num vídeo, na Tabela 3.7.

ID:	2	Nome: Rever interrogatório
Resumo:	Um Analista reverá o vídeo e as anotações a ele associadas.	
Refs:	R2.1, R2.2, R2.3, R3.1	

Tabela 3.5: Caso de Utilização 2: Editar

Cenário Principal		
Ação do Ator		Resposta do sistema
1 Este Caso de Utilização inicia-se quando o Analista acede a um recurso criado		
3 O Analista visualiza as anotações e outras marcas no vídeo	2	Devolve o recurso requerido em forma de <i>stream</i>

Tabela 3.6: Cenário Principal do Caso de Utilização 2

Cenário Alternativo		
	Ação do Ator	Resposta do sistema
4	O Analista submete novas anotações ao vídeo ou edita existentes	5 O sistema armazena as novas anotações

Tabela 3.7: Cenário Alternativo do Caso de Utilização 2

3.2 Diagrama de Blocos

Com o objetivo de concretizar os requisitos apresentados na Secção 3.1, a Figura 3.2 apresenta um modelo que responda aos atributos e forneça as funções do sistema.

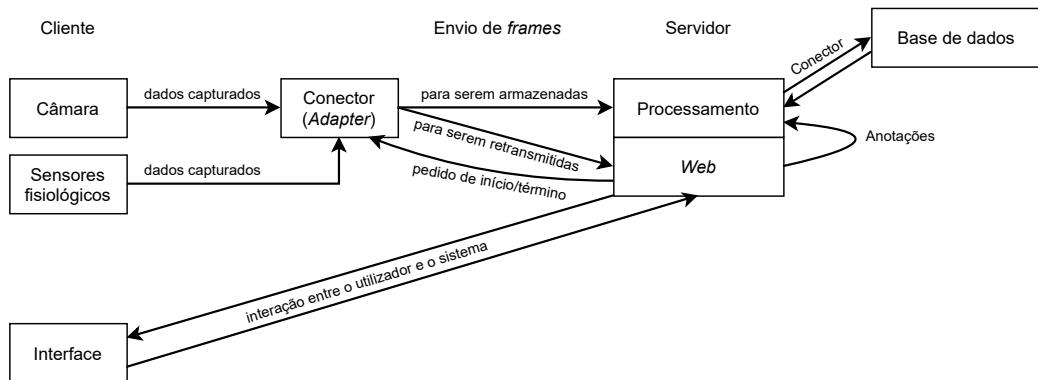


Figura 3.2: Diagrama de blocos do sistema

Como se pode observar na Figura 3.2, o sistema é constituído por um cliente “Câmera” (correspondente ao ator “Observado”, ver Figura 3.1) que apenas transmite as informações captadas por uma câmera 3D (como a câmera Intel RealSense) para o servidor, passando por um adaptador que estabelece a ligação com este. A informação após ser enviada para o servidor, é retransmitida para o cliente “Interface” (correspondente aos atores “Observador” e “Analista”, ver Figura 3.1) que interage com uma *interface* via *browser*.

O cliente “Observador” pode iniciar e terminar gravações, bem como realizar anotações sobre a transmissão ou vídeo gravado. As anotações são armazenadas na base de dados e os vídeos são armazenados no repositório

de vídeos.

3.3 Fundamentos

De modo a desenvolvêrmos o sistema proposto, com as melhores escolhas e tecnologias possíveis, foi necessário adquirir diversas aprendizagens teóricas e a nível de tecnologias a utilizar. Nas secções seguintes, irá ser exposto o conhecimento adquirido acerca dos tópicos que considerámos adequados e essenciais para integração no sistema e para o desenvolvimento do projeto.

3.3.1 Classificação de Emoções

A definição de emoção não é consensual, podendo ser descrita como uma reação a um estímulo ambiental e cognitivo que produz tanto experiências subjetivas, quanto alterações neurobiológicas significativas. Assim, expressões emocionais podem ser provocadas por emoções, tais como as definidas por Paul Ekman, durante mais de 40 anos, como sendo as 6 emoções básicas: a felicidade, a tristeza, a raiva, o nojo, o medo e a surpresa [Shiota, 2016]. Ao longo dos anos, teoricamente, o espetro de emoções básicas foi sendo alargado por outros cientistas, como Plutchik ([Bota et al., 2019]), que propôs um modelo com 8 emoções básicas e mais as que, pela a variação de intensidade, dessas derivam. Pode-se observar, na Figura 3.3, as emoções básicas, que estão no centro horizontal e, acima, estão as que se verificam quando há uma intensidade maior e as abaixo quando há uma menor intensidade.

Ambos os cientistas referidos redefiniram a classificação discreta de emoções. No entanto, posteriormente, já foram propostos outros modelos contínuos em espaços bi e tridimensionais, embora apenas alguns permaneçam como os modelos dominantes atualmente aceites, [Rubin e Talarico, 2009].

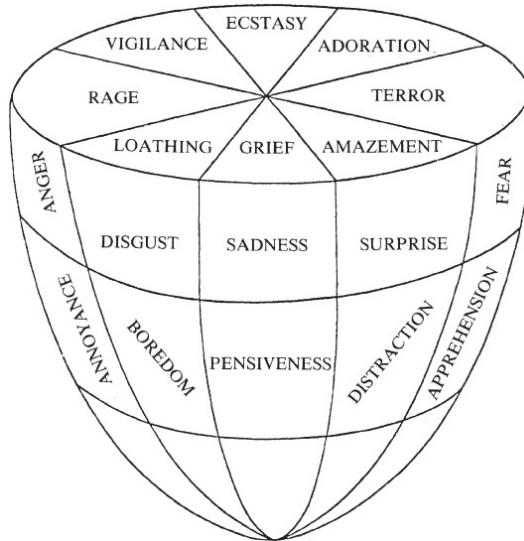


Figura 3.3: Esquema representativo do modelo de emoções de Plutchik [Bota et al., 2019]

Normalmente, as emoções tendem a alterar rapidamente detalhes na fisiologia das pessoas, quando as sentem. Isto é, por ordem das estimulações neurológicas ao cérebro, a expressão facial sofre alterações, despoletadas pela contração espontânea dos músculos faciais e o comportamento dos órgãos também se altera momentaneamente, por ação do sistema nervoso. O sistema nervoso simpático permite acionar as reações de estímulo desencadeadas pelas emoções, levando a um estado natural de alerta, e o parassimpático atenua as reações quando a emoção já não é sentida, para que o funcionamento dos sistemas volte ao estado homeostático.

De acordo com [Bota et al., 2019], estas alterações ao sistema nervoso refletem-se aquando da análise à leitura de alguns sinais fisiológicos através de sensores corporais, tais como:

- A Eletrocardiografia (ECG), que são os dados numéricos relativos às diferenças de potencial propagadas à superfície da pele e que resultam da atividade elétrica da contração e do relaxamento do miocárdio, o músculo cardíaco, que reveste o coração;
- A atividade eletrodérmica (*Electrodermal activity (EDA)*), condutância elétrica (*Skin Conductance Response (SCR)*), ou a resposta galvânica

da pele (*Galvanic Skin Response* (GSR)), que permite medir a resistência da pele pela variação da voltagem entre dois pontos após a passagem de uma corrente imperceptível ao corpo do utilizador. Esta medida é útil para detetar momentos de *stress* ou de surpresa, bem como de deceção, ansiedade e frustração;

- A fotopletismografia (*Photoplethysmography* (PPG)) ou o pulso do volume sanguíneo (*Blood Volume Pulse* (BVP)), em que um fotodíodo mede a quantidade de luz retroespelhada na pele, permitindo também medir o batimento cardíaco. Quando se sente relaxamento, o sistema parassimpático provoca a vasodilatação, aumentando o volume de sangue, e o bombeamento dele não vai provocar tanta pressão. Já quando o sistema nervoso simpático aumenta a pressão sanguínea, por ansiedade ou medo, os vasos estarão menos dilatados e a amplitude do BVP irá aumentar;
- A Respiração (RESP) permite medir o padrão respiratório. Aquando de emoções mais intensas, seja raiva, medo ou felicidade, a respiração será mais rápida e profunda. Respiração pouco profunda indica emoções de maior tensão, como pânico, medo ou concentração. Por fim, uma respiração lenta e mais profunda indicará um estado emocional de relaxamento e calma, enquanto que se for menos profunda pode indicar depressão ou ligeira alegria;
- A Temperatura (Temp ou SKT) é influenciada pela vasoconstrição, que acontece se a pessoa estiver mais fria, ou pela vasodilatação, caso esteja com a temperatura corporal mais elevada, e permite detetar e classificar emoções de forma semelhante à PPG/BVP). Acaba por ser uma medida pouco exata porque há diversos outros fatores externos que a podem alterar;
- A Eletromiografia (EMG), cujos sensores são colocados na face, de modo a detetar quais músculos estão mais ou menos contraídos;
- A Eletroencefalografia (EEG) permite estudar o campo elétrico do córtex cerebral, permitindo analisar as regiões do cérebro que estejam a ser mais ou menos estimuladas;

- E análise ocular, por Eletrooculografia (EOG), oculografia por reflexão de infravermelhos (*Infrared oculography* (IROG)) ou técnicas fotoelétricas. O EOG permite monitorizar os movimentos oculares, que quando excessivos podem indicar falta de atenção ou desconforto, e o potencial de repouso ocular. Os restantes permitem monitorizar a dilatação das pupilas, que é alterado pelo sistema nervoso autónomo. Estes podem também indicar, conjuntamente com frequência de piscar, o nível de sonolência, fadiga ou ansiedade.

3.3.2 Processamento de Vídeo

Um vídeo consiste na apresentação de uma sucessão de imagens diferentes a uma dada frequência (*framerate*). Simultaneamente com as imagens, é usual que se reproduza som. Processar um vídeo, de forma digital, consiste no processamento de cada *frame* (imagem), com a particularidade da existência adicional de uma componente temporal, o que implica uma atenção especial à movimentação de determinados elementos que tenham sido detetados, na transição entre cenas. A deteção de elementos nas cenas pode ser feita através de cores, texturas, formas ou padrões, com o objetivo de identificar objetos, por meio de um computador, de forma semelhante à que o sistema visual humano é capaz de percecionar.

Neste projeto, para detetar a face do indivíduo presente no vídeo e a expressão que esteja a esboçar, será necessário aplicar algoritmos capazes de realizar a sua deteção e percepção.

3.3.3 Redes Neuronais Convolucionais

As redes neurais são parte constituinte da área de inteligência artificial, onde tem sido desenvolvida para responder a problemas como o reconhecimento de voz, reconhecimento de faces, deteção de objetos em imagens, entre outros. Em particular, a necessidade de detetar emoções através de expressões faciais é um problema que se insere neste ramo.

Para tal, é necessário escolher a técnica de aprendizagem que permite uma resposta mais eficiente. A técnica mais recorrida para a análise de imagens são as *Convolutional Neural Network* (CNN).

Esta rede neuronal é adequada para o processamento de dados que se

apresenta no formato de uma grelha, tal como imagens, inspirada na organização do córtex visual de um animal. A rede está projetada para aprender de forma adaptativa e automática hierarquias no domínio do espaço dos recursos, para padrões de baixo a alto nível.

A construção matemática de uma CNN é tipicamente composta por três camadas: convolução, *pooling* e camadas totalmente conectadas (*FCs*). Os primeiros dois blocos são responsáveis pela extração de características, enquanto o último bloco é responsável por mapear as características extraídas no resultado como, por exemplo, a classificação de um objeto.

A camada de convolução desempenha um papel fundamental na CNN, composta por um conjunto de operações matemáticas, como a convolução, um tipo especializado de operação linear. Em imagens digitais, os valores de pixels são armazenados numa matriz de números, e através de um *kernel*, um extrator de recursos otimizado, é aplicado em cada posição da imagem, o que torna as CNNs altamente eficientes para o processamento de imagens, visto que um recurso pode ocorrer em qualquer lugar da imagem, [Yamashita et al., 2018].

Esta camada de convolução pode ser repetida múltiplas vezes, permitindo que sejam extraídos progressivamente padrões mais complexos e, por consequente, sejam estruturados hierarquicamente. A última camada pode também sofrer um processo de repetição, com o objetivo de reduzir pontos de ativação (como neurônios) na *FC* seguinte. Esta pilha permite que a classificação da rede seja mais eficiente.

Como processo de treino da CNN, é utilizado o método denominado *back propagation*. Este método compara o resultado obtido pela rede com os valores reais, através da descida de gradiente, método que calcula o sentido e a intensidade necessária para atingir o valor ótimo dos pesos a aplicar nas camadas totalmente conectadas.

A Figura 3.4 apresenta uma visão geral da estrutura e o processo de treinar uma CNN. É possível observar que a CNN, na sua gênese, é uma pilha de camadas e cada camada, por conseguinte, pode constituir uma pilha de operações. No fim, é calculada a diferença total sendo ajustados os pesos da última camada através do método *back propagation*.

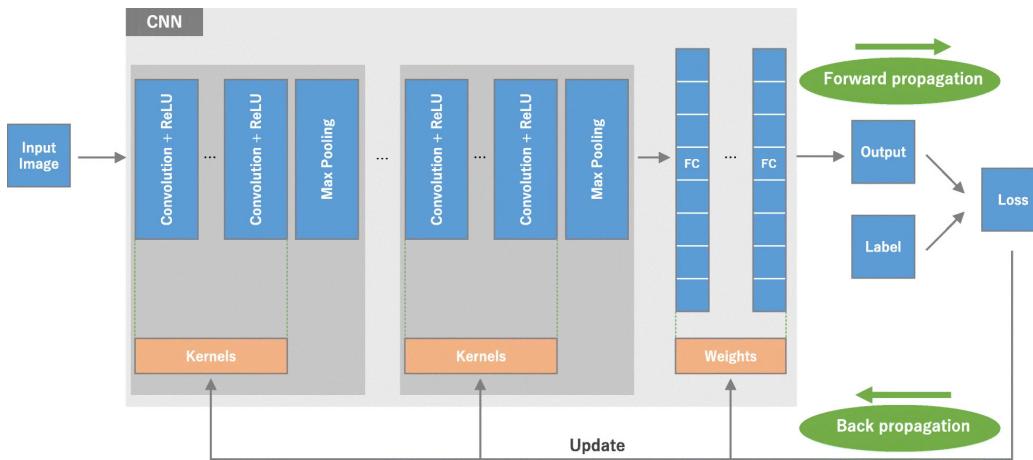


Figura 3.4: Visão geral da estrutura e o processo de treinar uma CNN

Tendo em conta que existem modelos previamente treinados e com taxas de sucesso bastante significativas, no corrente projeto, foram utilizados modelos já treinados especializados na classificação de emoções através de imagens. A biblioteca que fornece estes modelos encontra-se na Subsecção 4.2.3.

3.3.4 Servidor Web

Um servidor *web*, ou *web server* é constituído por, ao nível de *hardware*, um computador, que armazena os ficheiros que compõem os *websites* (por exemplo, documentos HTML, imagens, folhas de estilo, e arquivos JavaScript) e os entrega para o dispositivo do utilizador final. Deve estar conectado a Internet e pode ser acedido através do seu nome de domínio (DNS). Quanto a *software*, deve incluir diversos componentes que controlam como os utilizadores accedem aos ficheiros hospedados (armazenados para disponibilização), ou seja, no mínimo, um servidor *HyperText Transfer Protocol* (HTTP). Um servidor HTTP é um *software* que comprehende URLs (endereços *web* ou *Uniform Resource Locators*) e HTTP (o protocolo que navegador utiliza para visualizar páginas *web*, [Mozilla, 2021b].

Assim, poderemos reproduzir um servidor *web* dinâmico com um computador que disponibilizará, ao cliente que se conectar ao respetivo URL, as páginas *web* da plataforma do sistema desenvolvido no projeto.

3.3.5 *Threads*

Uma *thread*, ou tarefa, ou processo leve, consiste num fluxo de execução separado do processo pai, o fio de execução principal. Criando-se várias *threads*, tem-se a sensação de que estão a ser processadas simultaneamente. Porém, podem estar em execução em processadores diferentes, mas em cada processador, só funcionará uma de cada vez. O sistema operativo é que é encarregado de “escalonar” o tempo de execução das *threads*, [Anderson, 2019].

As *threads* mostram-se bastante úteis quando é necessário correr um ou mais troços de código à parte do principal, ou seja, sem bloquear aquele onde foi chamado.

Um objeto que herda a classe *Thread*, terá 3 métodos essenciais: o *run*, o método principal da *thread*, o *start*, que só pode ser invocado uma vez em cada *thread* e corre o seu método *run* uma vez e termina, num fluxo de execução separado, como normalmente se pretende, e o *join*, utilizado na classe onde a *thread* foi instanciada para que fique à espera que a *thread* termine graciosamente quando puder.

3.3.6 *Sockets*

Um *socket* é um *endpoint* de uma ligação de comunicação bidirecional entre dois programas em execução na rede. Um *socket* é vinculado a um número de porta para que a camada de transporte possa identificar a aplicação para a qual os dados são enviados, [Oracle, 2021].

A utilização de *sockets* irá permitir a comunicação entre os clientes e o servidor do sistema, de modo a serem trocadas informações para, por exemplo, o processo de autenticação, para o envio e receção de vídeos e de anotações e/ou outros dados.

3.3.7 *Websocket*

A API *WebSocket* é uma tecnologia que permite abrir uma sessão de comunicação interativa bidirecional, *full-duplex* e de baixa latência, entre o navegador do cliente e um servidor. Com esta API, pode-se enviar mensagens a um servidor e receber respostas orientadas a eventos sem precisar de consultar o servidor para obter uma resposta, [Mozilla, 2021a]. Foi projetado

para ser executado em *browsers* e servidores *web* que suportem HTML5, mas pode ser usado por qualquer cliente ou servidor aplicacional.

Capítulo 4

Implementação do Modelo

Neste capítulo será apresentada a arquitetura do projeto, bem como a apresentação das várias tecnologias utilizadas e a sua incidência no projeto. Por fim, será descrita detalhadamente a abordagem no projeto, justificando todas as decisões tomadas durante a implementação.

4.1 Arquitetura do Sistema

A arquitetura apresentada na Figura 4.1 é composta por um servidor e dois clientes distintos que comunicam, em paralelo, com o servidor.

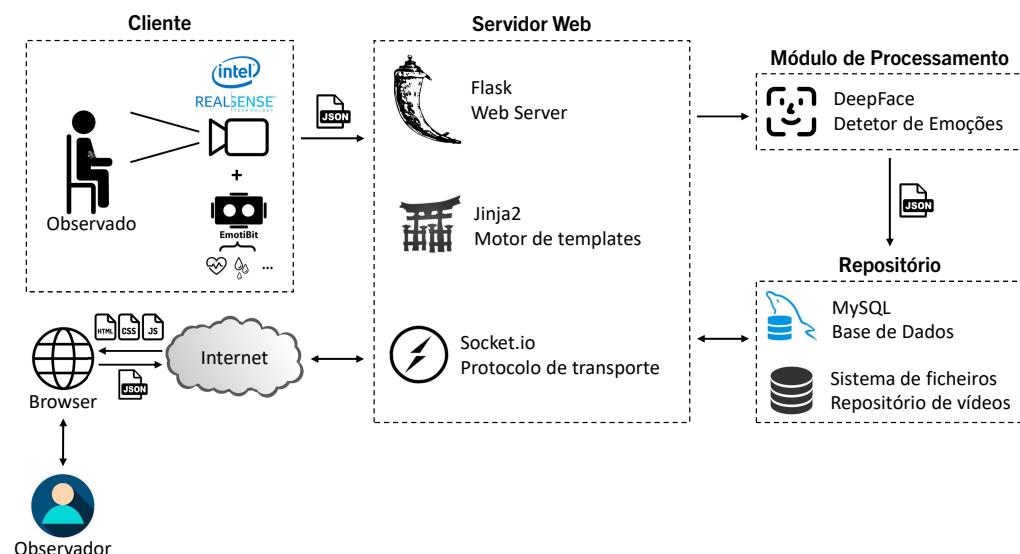


Figura 4.1: Arquitetura do projeto

4.1.1 Cliente “Observado”

O cliente “Observado” (ver Figura 3.1) encontra-se inserido num ambiente onde o *wearable EmotiBit* e a câmara *Intel RealSense* captam informação fisiológica e audiovisual, respetivamente. Para recolher a informação dos dispositivos, é necessário um controlador conectado aos mesmos.

Devido a limitações do *wearable EmotiBit*, para gravar e visualizar os dados fisiológicos, é necessário recorrer à aplicação desenvolvida pela *EmotiBit Team* para interagir com o microcontrolador, sendo este um processo manual. Na subsecção 4.2.6 será explicado ao detalhe o funcionamento e comunicação entre o microcontrolador com o computador.

De modo a extrair a informação proveniente da câmara, é necessário o recurso à biblioteca *PyRealSense* para obter as *frames* e o recurso à biblioteca *PyAudio* para extraír o áudio captado. Estes dados são organizados num documento *JavaScript Object Notation (JSON)* e, posteriormente, são incorporados no *payload* das mensagens que seguem o protocolo *User Datagram Protocol (UDP)*, que serão transmitidos ao servidor via *socket*.

4.1.2 Cliente “Observador”

Em paralelo, o cliente “Observador” (ver Figura 3.1) estabelece uma conexão com o servidor através do seu *browser* de eleição, onde a *framework Flask* é responsável por criar as várias páginas e gerir os seus pedidos. Após a conexão estabelecida, é apresentada uma página de autenticação onde dispõe de um *link* para o utilizador efetuar o seu registo na aplicação. Todas as palavras-passe dos utilizadores são encriptadas do lado do servidor e todos os dados de autenticação são armazenados na base de dados *MySQL*.

O cliente, tendo sido autenticado com sucesso, pode selecionar um vídeo da lista de vídeos que gravou anteriormente. Sobre o vídeo selecionado, o utilizador pode editar ou adicionar anotações, sendo emoções ou texto e visualizar os dados fisiológicos captados ao longo da gravação. Pode também alterar alguns dados pessoais relativos ao seu perfil e fazer *upload* de novos vídeos provenientes do seu sistema de ficheiros, ou ainda iniciar uma nova gravação, realizar anotações em tempo real e terminá-la e guardá-la.

4.1.3 Servidor

O servidor, tal como o explicitado na Subseção 3.3.4, é responsável por gerir os pedidos do cliente “Observador” (ver Figura 3.1) e por retransmitir a *stream* recebida pelo cliente “Observado”. De modo a iniciar a retransmissão, é iniciada uma *thread*, cujas características foram mencionadas na Subseção 3.3.5, responsável por receber, via *socket*, os dados audiovisuais e enviar para o cliente “Observador” com o recurso à *framework Socket.IO*. Quando o cliente “Observador” terminar a gravação, a *thread* de retransmissão termina a sua execução.

Ao rever um vídeo, pode ser iniciada outra *thread* responsável por percorrer as *frames* e detetar, automaticamente, qual a emoção que o cliente *Observado* demonstra sentir nesse instante através da biblioteca *DeepFace*. Todas as emoções detetadas são armazenadas na base de dados e, por ordem do utilizador, num documento JSON.

Por fim, o vídeo é armazenado num sistema de ficheiros e toda a informação relativa ao vídeo é armazenada na base de dados *MySQL*.

4.2 Tecnologias Utilizadas

Esta secção pretende apresentar as tecnologias utilizadas na implementação, bem como a elaboração de um *Hello World* para mostrar as funcionalidades das tecnologias em específico.

4.2.1 Python

A linguagem de programação *Python* é a linguagem escolhida para o desenvolvimento deste projeto, incluindo o processamento dos vídeos e o servidor *Web*. Esta linguagem é uma linguagem de alto nível, fácil de aprender e suportada pelos sistemas operativos mais comuns, tais como, *Windows*, *Mac* e *Linux*. Esta linguagem é *open-source* e dispõe de uma comunidade vasta que contribuiativamente para o seu desenvolvimento.

A linguagem *Python* destaca-se pelo facto de ser, geralmente, uma das linguagens mais utilizadas para a construção de algoritmos de inteligência artificial, aprendizagem automática e, em particular, processamento de imagem. Este facto motivou a escolha desta linguagem para o desenvolvimento

do projeto.

4.2.2 Frameworks *Flask* e *Socket.IO*

A *framework Flask* é uma micro *web framework*, escrita em *Python*, que permite o desenvolvimento de aplicações *Web* e, em particular, um servidor *Web*, [Kumar, 2021]. Considera-se uma micro *framework* por não ter uma camada de abstração da base de dados, validação de formulários ou outras funcionalidades que possam ser obtidas através de funções de disponíveis noutras bibliotecas, [Kumar, 2021]. Esta *framework* permite uma interoperabilidade com outras *frameworks*, o que torna a implementação mais focada na resolução do problema. Utiliza o motor de criação de *templates* *Jinja* e o *toolkit* *Werkzeug WSGI*, [Flask, 2021].

A Listagem 4.1 apresenta a implementação necessária para a construção de um servidor *Web* com o recurso à *framework Flask*.

Listagem 4.1: Implementação de um servidor *Web* com o recurso a *Flask*

```

1 from flask import Flask
2
3 app = Flask(__name__)
4
5
6 @app.route('/')
7 def hello():
8     return 'Hello, World!'
9
10 if __name__ == "__main__":
11     app.run()

```

A *framework Socket.IO* é considerado um *wrapper* da API *WebSockets*, referenciada na Subseção 3.3.7, baseada em eventos e em tempo real entre clientes e um servidor, [Socket.IO, 2010]. A implementação oficial desta *framework* está escrita na linguagem *JavaScript*, sendo que existe um *package* que fornece implementações para a linguagem *Python*.

Como o projeto recorre à *framework Flask*, é utilizada a *framework Flask-SocketIO* que permite combinar as funcionalidades do servidor *Web* com as funcionalidades de *WebSockets*. Assim, é possível retransmitir a informação audiovisual proveniente do cliente “Observado” (ver Figura 3.1) e as novas anotações criadas, alteradas ou removidas, evitando-se os pedidos HTTP completos.

A Listagem 4.2 apresenta a implementação de um servidor *Web* com a *framework Flask* em conjunto com a *framework Flask-SocketIO* que implementa o protocolo de transporte.

Listagem 4.2: Implementação de um servidor *Web* com o recurso às *frameworks Flask* e *Flask-SocketIO*

```

1 from flask import Flask, render_template
2 from flask_socketio import SocketIO
3
4 app = Flask(__name__)
5 app.config['SECRET_KEY'] = 'secret!'
6 socketio = SocketIO(app)
7
8 if __name__ == '__main__':
9     socketio.run(app)

```

Com o recurso à tecnologia *WebSockets*, é mais eficaz a retransmissão da informação audiovisual para o cliente observador, pois a comunicação servidor — cliente só acontece quando há dados no servidor a retransmitir, tal como foi abordado na Subsecção 3.3.7.

4.2.3 DeepFace

A *framework DeepFace* é uma *toolbox* de reconhecimento e análise de atributos faciais desenvolvida para *Python*. O reconhecimento facial é constituído por uma *framework* híbrida contendo modelos *state-of-the-art* como *VGG-Face*, *Google FaceNet*, *OpenFace*, *Facebook DeepFace*, *DeepID*, *ArcFace* e *Dlib*. Estes modelos encontram-se pré-treinados e os valores de precisão que atingem são considerados bastante elevados, [Serengil e Ozpinar, 2020].

Esta *framework* permite realizar:

- Reconhecimento e verificação facial;
- Extração de características faciais;
- Análise de características faciais em vídeo em "tempo real".

Em particular, a funcionalidade benéfica para o projeto é a extração de características faciais.

A extração de características faciais permite detetar a idade, o género, a expressão facial e a etnia do indivíduo na imagem. Como o principal foco

do projeto é a deteção de emoções, a característica relevante é a emoção dominante. A *DeepFace* permite detetar as expressões faciais e deduzir as 6 emoções básicas definidas por Ekman, explicitadas na Subseção 3.3.1, e a de neutro, ou seja, deduzir se o utilizador se sente zangado, amedrontado, neutro, triste, enojado, feliz ou surpreso.

A Figura 4.2 apresenta a metainformação obtida pela extração de características para várias expressões faciais.

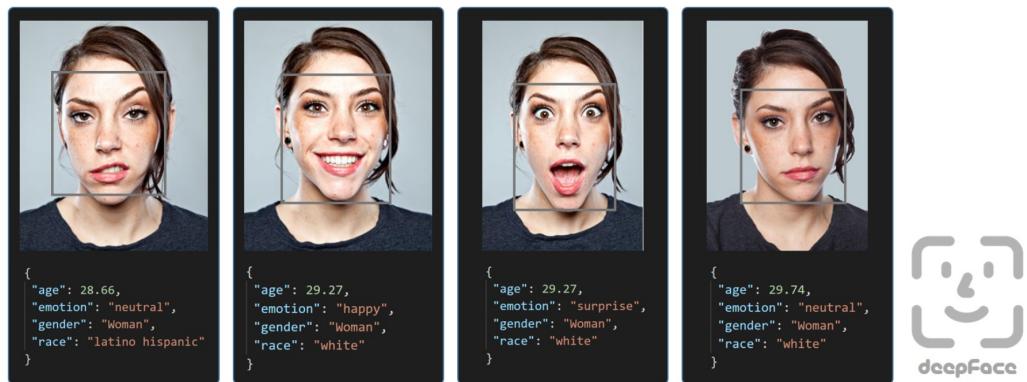


Figura 4.2: Exemplo de extração de características faciais para várias expressões, [Serengil e Ozpinar, 2020]

4.2.4 Bibliotecas *Intel RealSense SDK 2.0* e *PyAudio*

O *Intel RealSense Software Development Kit* (SDK) 2.0 é uma biblioteca multiplataforma destinada a interagir com as câmaras desenvolvidas pela *Intel*. Esta biblioteca permite adquirir uma *stream* da câmara a cores e da câmara de profundidade, fornecendo métodos de calibração dos parâmetros intrínsecos e extrínsecos, [Intel, 2019].

De modo a utilizar a câmara, recorreu-se à biblioteca *pyrealsense2* que estabelece a ligação entre o SDK escrito na linguagem C++ e a linguagem *Python*. Assim, é possível comunicar entre um cliente em *Python* e a câmara.

Como a biblioteca *pyrealsense2* só permite extrair informação visual, é necessário o recurso à biblioteca *pyaudio* para extrair a componente auditiva da câmara.

Esta é constituída por um conjunto de ligações com a biblioteca *PortAudio*, responsável por estabelecer *interfaces* com o controlador de áudio.

4.2.5 *MySQL*

O *MySQL* é um sistema de gestão de bases de dados *open-source* que, de momento, é distribuída pela *Oracle Corporation*, [Oracle, 2019]. Neste sistema, as bases de dados *Structured Query Language* (SQL) são coleções de dados estruturados, onde é possível estruturar e organizar a informação em tabelas relacionadas entre si.

As características de uma base de dados relacional é benéfica ao projeto devido à necessidade de armazenar informação referente aos utilizadores, aos vídeos gravados e às anotações aos vídeos. A escolha deste sistema para o projeto deveu-se à maior familiaridade e preferência por este.

4.2.6 *EmotiBit*

A *EmotiBit* é um módulo de sensores *wearable* para a captação de informação emocional, fisiológica e de movimento de alta precisão, o que torna a *EmotiBit* uma ferramenta ideal para investigação, [Team, 2019].

A informação captada pode ser transmitida por Wi-Fi, bem como armazenada no cartão SD. Este é responsável por fornecer as configurações do módulo Wi-Fi para estabelecer a ligação *Local Area Network* (LAN). Desta forma, o utilizador pode transmitir os dados captados em tempo real e gravar os dados em memória.

Os dados extraídos pela *EmotiBit* podem ser agrupados em quatro conjuntos:

- PPG (Fotopletismograma) — permite estimar o batimento cardíaco, variação do batimento cardíaco, saturação de oxigénio, entre outros dados;
- EDA / GSR (Atividade Eletrodérmica / Resistência Galvânica da Pele) — reflete as respostas do sistema nervoso simpático impulsionadas pela estimulação cognitiva e emocional;
- IMU de 9 eixos (Unidade de Medida Inercial) — permite estimar movimentos e rotações através dos sensores acelerómetro, giroscópio, magnetómetro;
- Temperatura corporal — permite detetar reações emocionais.

Para o desenvolvimento do projeto, apenas os conjuntos de sensores PPG, EDA / GSR e Temperatura corporal são necessários para complementar a informação extraída pela *framework DeepFace* na deteção de emoções mencionadas na Subsecção 3.3.1.

De modo a visualizar os dados captados pela *EmotiBit*, é necessário recorrer à aplicação *EmotiBit Oscilloscope* que traça os vários dados em gráficos temporalmente. Na Figura 4.3 é possível observar o funcionamento da aplicação *EmotiBit Oscilloscope*.

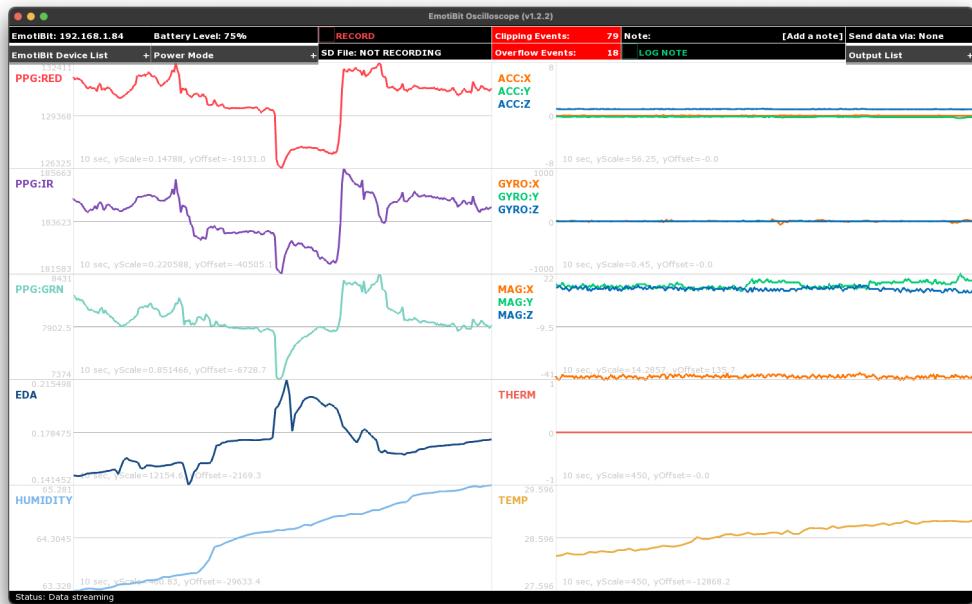


Figura 4.3: Funcionamento da aplicação *EmotiBit Oscilloscope*

Esta aplicação permite controlar qual o dispositivo *EmotiBit* a transmitir, verificar o nível de bateria, alterar o modo de funcionamento, inicializar ou finalizar uma gravação e selecionar qual o protocolo de transporte para a transmissão.

Após o utilizador terminar uma gravação, são criados e armazenados um ficheiro JSON e um ficheiro *Comma Separated Values* (CSV) que contêm, respetivamente, informações sobre os sensores e acerca dos valores captados. De modo que a informação captada seja legível, é necessário recorrer

à aplicação *EmotiBit DataParser*, que organiza os dados pelo instante de tempo, informação sobre o tamanho do valor e, no fim, o valor do sensor específico.

4.3 Abordagem

Nesta secção será abordado o processo de implementação, descrevendo e analisando as várias hipóteses de solução, bem como, a justificação das decisões tomadas.

4.3.1 Comunicação Cliente “Observado” — Servidor

O cliente “Observado”, como referido na Subsecção 4.1.1, está presente num ambiente onde consta a câmara *Intel RealSense* e o *wearable EmotiBit* a captarem informação sobre o “Observado”.

A máquina onde os dois dispositivos estão conectados, é responsável por inicializar a captação e transmissão dos dados do “Observado” para o servidor, onde é necessário executar um *script* em *Python* que inicializa todos os requisitos necessários para a extração dos dados audiovisuais, bem como, a inicialização de um *socket* para a transmissão dos destes dados. De modo a que a transmissão seja independente da captação dos dados, estas ocorrem em *threads* lançadas pelo processo pai.

Como é obrigatório que seja possível anotar em tempo real, como consta na Tabela 3.2, é necessário garantir que a transmissão também seja em tempo real. De modo a concretizar este atributo, cada *frame* captada é codificada no formato JPEG (*Joint Photographic Experts Group*), para não sobrecarregar a rede, e o protocolo recorrido para a transmissão é o protocolo UDP, visto que não cria uma conexão antes do envio do primeiro datagrama e este não redistribui datagramas que não chegaram ao cliente.

Tendo em conta que o *payload* de um segmento UDP tem uma dimensão máxima de 65 507 *bytes*, é fundamental garantir que este tamanho não é excedido. Para tal, todas as mensagens enviadas são organizadas num documento JSON, dado que é um formato de armazenamento e transmissão pouco verboso, isto é, o número de caracteres para representar um conjunto de dados é bastante reduzido, por conseguinte, permite armazenar mais informação. Esta característica é fundamental para garantir que a transmissão

ocorre em tempo real.

Quando a *thread* de transmissão recebe dados, é calculado o número de segmentos necessários para o envio, subtraindo a dimensão da estrutura do documento JSON ao tamanho máximo do *payload*, evitando que se exceda o tamanho do datagrama.

Com o objetivo de otimizar o número de segmentos a enviar e manter a sincronização entre os dados audiovisuais, foi construído um diagrama para representar a estrutura da mensagem a enviar.

Como consta na Figura 4.4, o datagrama é constituído (lado esquerdo da figura) pelo cabeçalho IP, cabeçalho UDP e pela estrutura do documento JSON que contém os dados audiovisuais e informação sobre a segmentação destes. No lado direito é possível observar a estrutura do documento JSON onde a estrutura é um conjunto de elementos heterogêneos ordenados, que engloba um *bit* para informação dos segmentos da imagem, a própria imagem, outro *bit* para informação dos segmentos do áudio e o próprio áudio. Os valores booleanos permitem indicar ao servidor se todos os segmentos relativos à imagem ou ao áudio já foram enviados na sua totalidade.

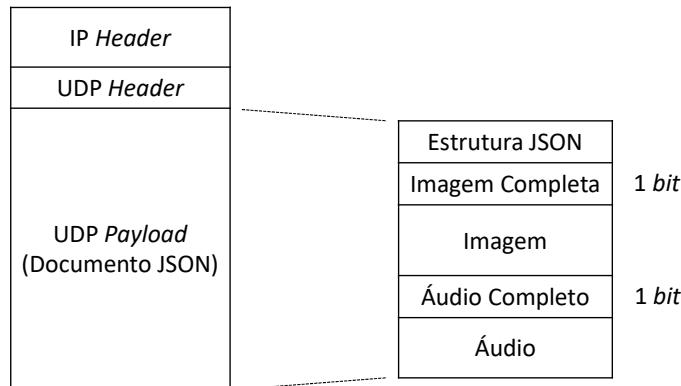


Figura 4.4: Diagrama representativo da estrutura de um datagrama

É de notar que o *payload* do documento JSON é dividido em duas partes iguais para o armazenamento dos dados visuais e auditivos.

Considerando que a imagem possui uma maior dimensão que o áudio, é bastante provável que a imagem tenha de ser dividida por vários datagramas em comparação ao áudio. De modo a otimizar o número total de segmentos enviados por *frame*, é calculado o espaço restante da metade do *payload*

reservado ao áudio, adicionando esse espaço à metade do *payload* reservado à imagem. Assim, num datagrama, a imagem está compreendida entre metade do *payload* e o tamanho máximo deste.

É necessário, também, realçar que toda a informação binária a ser transmitida, tal como imagens e áudio, tem de ser convertida segundo um esquema *binary-to-text encoding*. Para tal, recorreu-se ao esquema *base64* por ser um esquema bastante utilizado na transmissão de dados binários e pela familiaridade com este. O esquema *base64* converte 6 *bits* num carácter ASCII.

4.3.2 Retransmissão para o Cliente “Observador”

De modo a permitir que o cliente “Observador” possa visualizar, anotar e gravar, é fundamental que o servidor retransmita para esse cliente a informação captada.

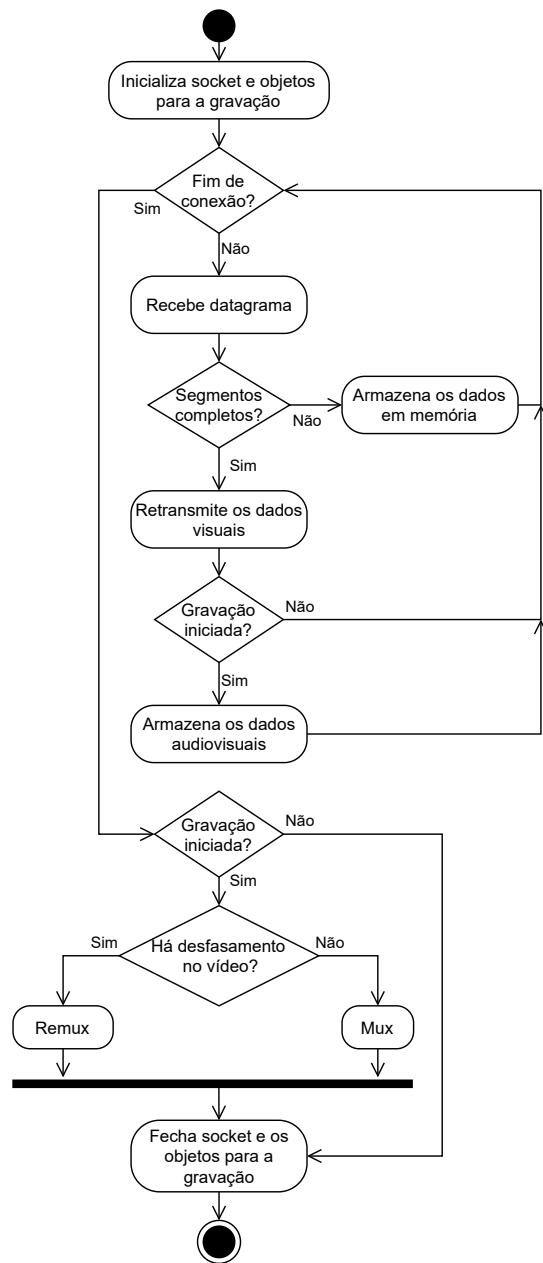


Figura 4.5: Diagrama de atividades da *thread* responsável pela receção, gravação e retransmissão dos dados audiovisuais

Para tal, quando o observador é autenticado, é criada uma *thread* que segue o diagrama de atividades que consta na Figura 4.5.

Quando é iniciado o construtor desta, são inicializadas instâncias de

`socket`, `PyAudio` e `OpenCV`. A instância de `socket` permite receber as mensagens enviadas pelo cliente “Observado” e as restantes permitem gravar os dados enviados em formato de áudio e vídeo, respetivamente.

Como as imagens enviadas encontram-se em formato JPEG, o formato de compressão de vídeo utilizado é *Motion JPEG* (MJPG). Este formato de vídeo é uma compressão *intraframe*, onde cada *frame* é codificada JPEG.

Após a inicialização da *thread*, a `socket` fica a receber mensagens, de forma passiva [Python Software Foundation, 2021]. No momento em que é recebido um datagrama, é extraída e reconstruída a informação contida no documento JSON. Caso a informação esteja completa, é retransmitida para o cliente observador através de *websockets*. Devido a uma política para melhorar a experiência do utilizador, não é possível reproduzir vídeos ou áudio automaticamente, assim, só a imagem é retransmitida para o cliente observador.

Caso o “Observador” tenha iniciado a gravação, a informação é convertida de *base64* para binário e, posteriormente, é escrita nos objetos de gravação. Quando o cliente “Observador” terminar a gravação, é também terminada a conexão, bem como, é iniciado o processo de multiplexagem para a gravação.

Antes de ser procedida a multiplexagem e, de modo a garantir que os dados auditivos estão em sincronismo com os dados visuais, é verificado se o *framerate* do vídeo é de, aproximadamente, 25 *frames* por segundo. Caso o *framerate* seja discrepante, é feita uma recodificação do vídeo para ficar sincronizado com o áudio. Posteriormente, é feita a multiplexagem entre o vídeo e o áudio para ser armazenado no repositório de vídeos.

De modo a realizar a recodificação e a multiplexagem, recorreu-se ao programa FFmpeg que permite gravar, converter e criar *streams* de áudio e vídeo em diversos formatos.

4.3.3 Interação com a *EmotiBit*

Tendo em conta que o objetivo é a captação e envio de dados fisiológicos de forma automática, é fundamental que a *EmotiBit* estabeleça uma comunicação com a máquina responsável por transmitir a informação ao servidor. Para tal, a *EmotiBit* conecta-se à máquina por Wi-Fi e, através de um protocolo de rede, envia a informação captada à máquina. No entanto, esta implementação não é possível, pois a *EmotiBit* apenas estabelece comunicação

com a aplicação *EmotiBit Oscilloscope*.

Outra agravante é os dados da *EmotiBit* só poderem ser armazenados no cartão SD e após o uso do *wearable* é que se pode extrair o ficheiro CSV. Este ainda tem de ser processado pela aplicação *EmotiBit DataParser*, que origina um ficheiro por cada sensor (no total 16 ficheiros) com toda a informação num dado instante. Assim, é necessário construir um algoritmo que agrupe toda a informação originada pela aplicação *EmotiBit DataParser* num único ficheiro e enviar este ficheiro para o servidor, de modo a completar a informação fisiológica em falta no vídeo. O agrupamento de dados, bem como o envio destes, é feito de modo manual.

No entanto, o sistema dispõe de um algoritmo de processamento de sinais PPG para a obtenção dos Batimentos por minuto (BPM) do coração, que se encontra apresentado na Subsecção 4.3.4.

4.3.4 Processamento do sinal PPG

Os dados obtidos pelo sensor PPG da *EmotiBit* necessitam de ser processados de modo a obter o batimento cardíaco do cliente “Observado”. Para tal, foi construído um algoritmo que calcula a variabilidade do batimento cardíaco, ou seja, os BPM são calculados através do intervalo entre impulsos.

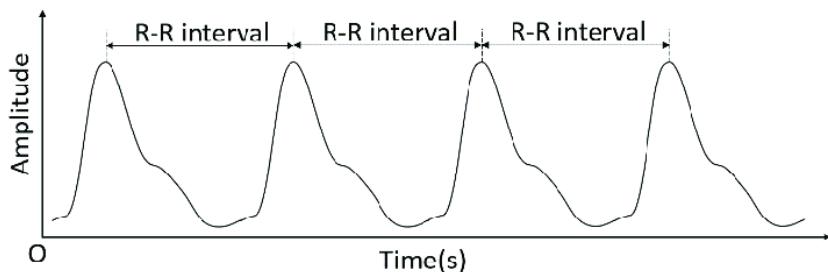


Figura 4.6: Sinal PPG com intervalos R-R para o cálculo da variabilidade do batimento cardíaco [Yang et al., 2018]

Como mostra a Figura 4.6, o BPM é calculado com base no intervalo entre picos do sinal. Para a construção do algoritmo, recorreu-se ao conjunto de dados [Siam, 2019], onde o sinal utilizado para a construção deste encontra-se na Figura 4.7.

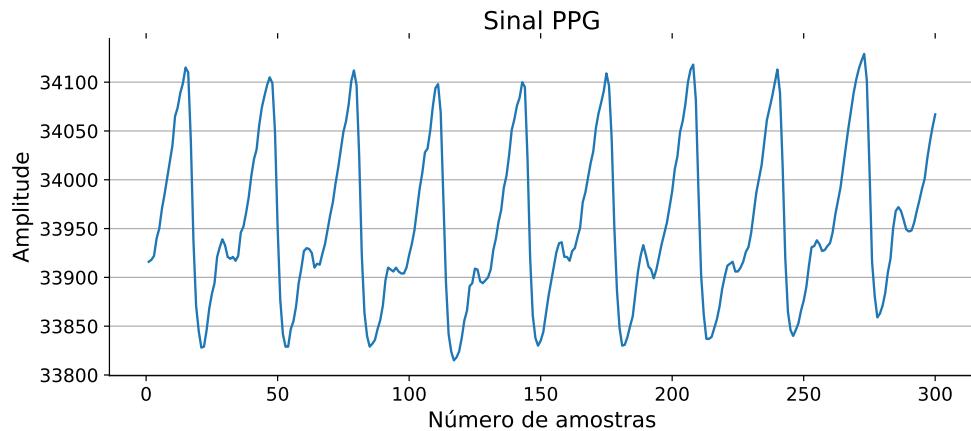


Figura 4.7: Sinal PPG proveniente do conjunto de dados [Siam, 2019]

Na Figura 4.7 é apresentado um sinal PPG filtrado de um indivíduo saudável, com duração de 6 segundos e com uma frequência de amostragem de 50Hz.

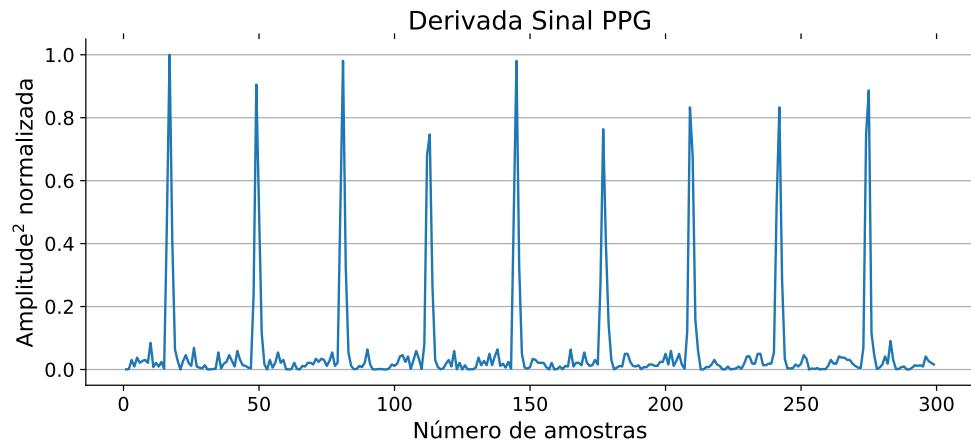


Figura 4.8: Quadrado da derivada do sinal PPG

Com o objetivo de contabilizar os picos do sinal, é calculado o quadrado da derivada do sinal PPG, como consta na Figura 4.8, para isolar os instantes onde o fluxo de sangue é maior, podendo, assim, afirmar-se que ocorreu um batimento.

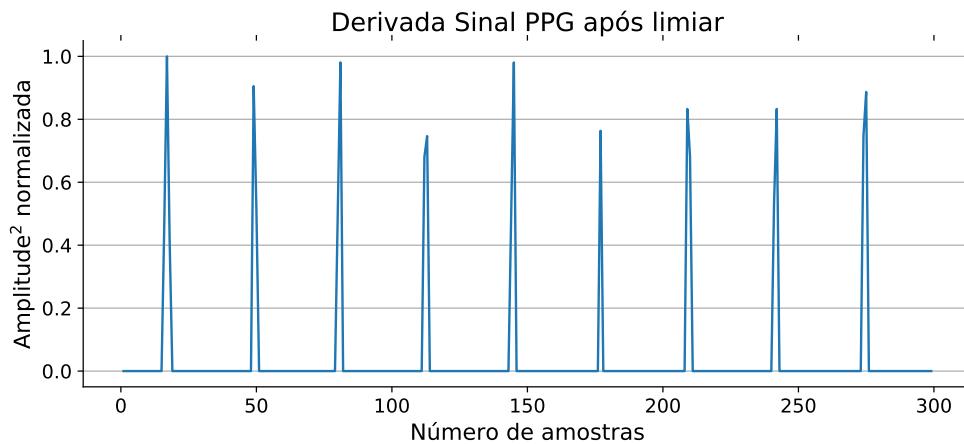


Figura 4.9: Quadrado da derivada do sinal PPG após a aplicação de um limiar

Para a obtenção dos picos do sinal, recorreu-se à função `find_peaks` da biblioteca `scipy`. Esta função procura por todos os máximos locais no sinal comparando com os valores vizinhos, retornando um conjunto de índices dos máximos locais. Um índice traduz-se num ponto do sinal amostrado, que corresponde a um instante no tempo.

Como o sinal na Figura 4.8 apresenta vários máximos locais para uma amplitude quadrada abaixo de 0,2; é necessário eliminar estes valores, pois irão adulterar o valor de BPM. Para tal, é aplicado um *threshold* (limiar) alterando os valores abaixo desse valor para zero. Em particular, o valor limite escolhido é 0,4 de modo que, caso haja algum valor anormal no sinal, não seja considerado um pico e, desta forma, seja considerado uma situação de maior fluxo sanguíneo.

4.3.5 Processamento de Emoções com *Deepface*

O algoritmo *Deepface*, caso consiga detetar uma cara numa imagem, permite obter a emoção dominante. Se se analisassem todas as *frames* que constituem os vídeos, isso iria demorar demasiado tempo em relação ao considerado aceitável e gerar um enorme volume de anotações.

Para não se sobrecarregar a base de dados de anotações relativas a emoções obtidas com um grau de incerteza elevado, ou repetidas em intervalos de tempo contíguos, criou-se o algoritmo representado no diagrama da Figura 4.10.

O algoritmo criado consiste, através da utilização da biblioteca *OpenCV*, em ler as imagens que compõem o vídeo e detetar, com recurso à biblioteca *Deepface*, a emoção dominante, caso seja detetada uma face na imagem. As emoções detetadas são armazenadas num *buffer*, sem tamanho definido, até que um contador de *frames* analisadas atinja um valor definido e a emoção dominante no intervalo de tempo das *frames* analisadas é o resultado da aplicação da medida estatística da moda sobre os elementos existentes no *buffer*. Caso seja a expressão neutra, não será gerada uma anotação.

Para diminuir o tempo de processamento, visto que com o *framerate* dos vídeos, entre *frames* seguidas não é possível modificar-se consideravelmente a expressão facial, consequência da emoção do “Observado”, introduziu-se um contador para limitar o intervalo de *frames* a analisar recorrendo à *Deepface*, de modo que só é analisada uma *frame* em cada *n frames*. O valor quantitativo deste *n* é obtido pela divisão do tamanho do ficheiro do vídeo, em *bytes* por um valor constante, tendo sido considerado um adequado, pois as análises feitas, desta forma, não excediam o tempo da visualização normal do vídeo. O algoritmo ou modelo de deteção de características faciais *backend* escolhido, dos existentes, foi o *OpenCV* pois mostrou-se, segundo [Serengil, 2021], o que mais *frames* por segundo consegue processar, sem prejudicando significativamente a deteção da face e das emoções comparativamente aos restantes.

Para não se gerar duas ou mais anotações iguais em intervalos de tempo seguidos, armazena-se a emoção atual e só é gerada a anotação da emoção anterior guardada se a seguinte for uma emoção diferente, caso contrário, apenas se adiciona a outra lista o tempo final da emoção guardada e continua-se a análise.

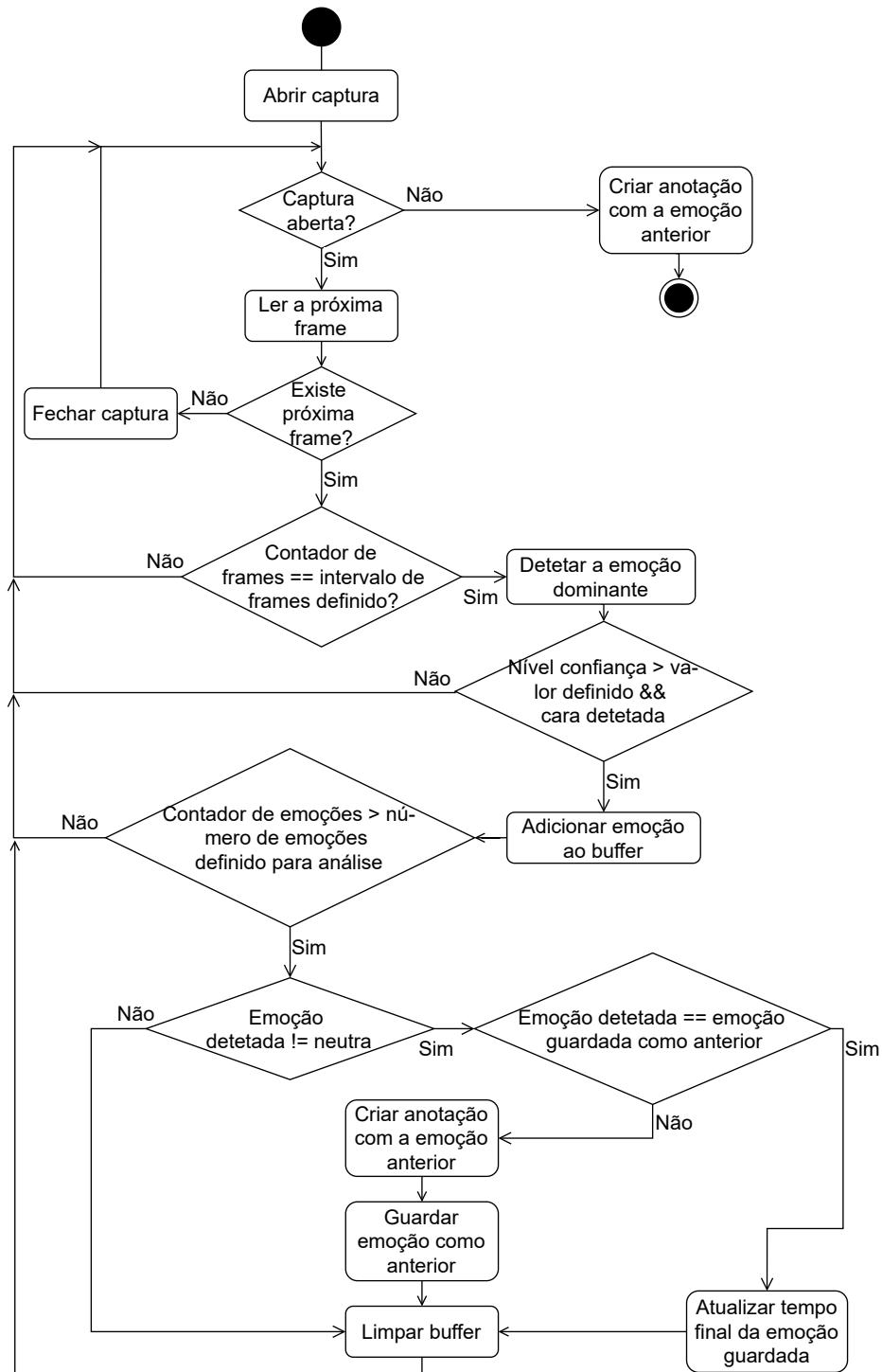


Figura 4.10: Diagrama de atividades da *thread* responsável pela criação de anotações provenientes das emoções detetadas pela *Deepface*

4.3.6 Páginas *web* e suas funcionalidades

O sistema permite registar novos utilizadores na página de registo, representada na Figura 4.11. O utilizador deve introduzir um nome de utilizador e um endereço eletrónico que ainda não esteja a ser utilizado, uma palavra-passe com 6 ou mais caracteres e repeti-la no campo abaixo.

Caso já possua uma conta, pode mudar para a página de *login* no *link* abaixo da ilustração.

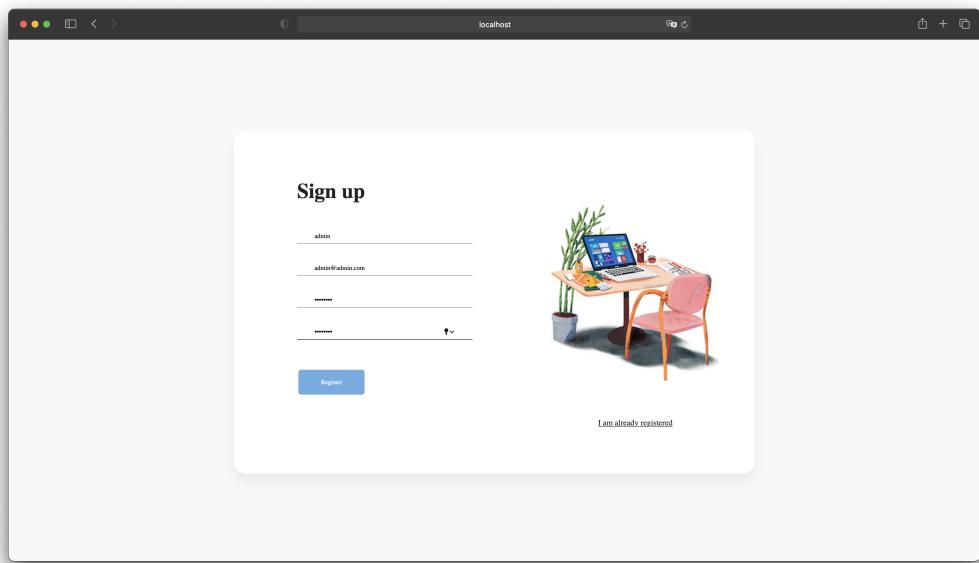
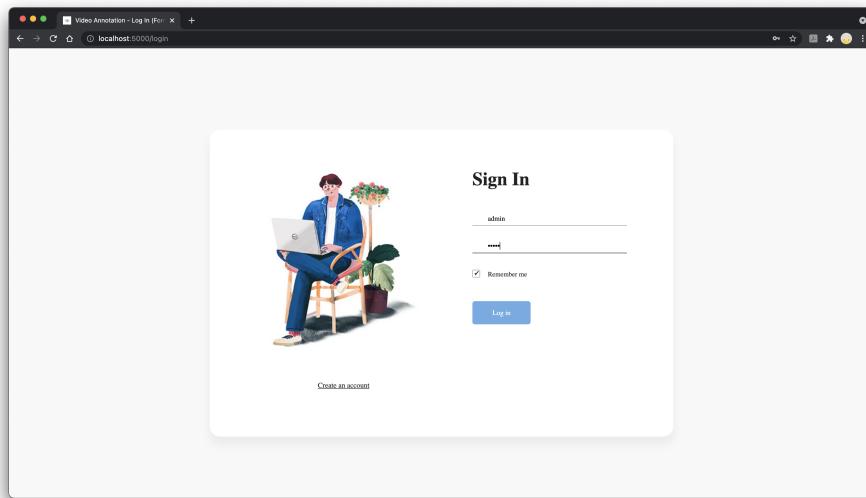


Figura 4.11: Página de Registo

Na página de *login*, que consta na Figura 4.12, o utilizador pode iniciar sessão, providenciando o seu nome de utilizador e palavra-passe, que devem coincidir com as registadas. Se marcar a *checkbox*, serão armazenados *cookies* no *browser* para memorizar e manter o início de sessão do utilizador.

Figura 4.12: Página de *Login*

Se o utilizador pretender alterar algum dos seus dados de registo, à exceção da palavra-passe, pode entrar na página representada na Figura 4.13 através do botão *Update Data*, no menu *dropdown* presente na *navbar* de todas as páginas acessíveis após o início de sessão. Nesta, deve introduzir o nome de utilizador novo e/ou o *e-mail* pretendido e a *password* correta duas vezes.

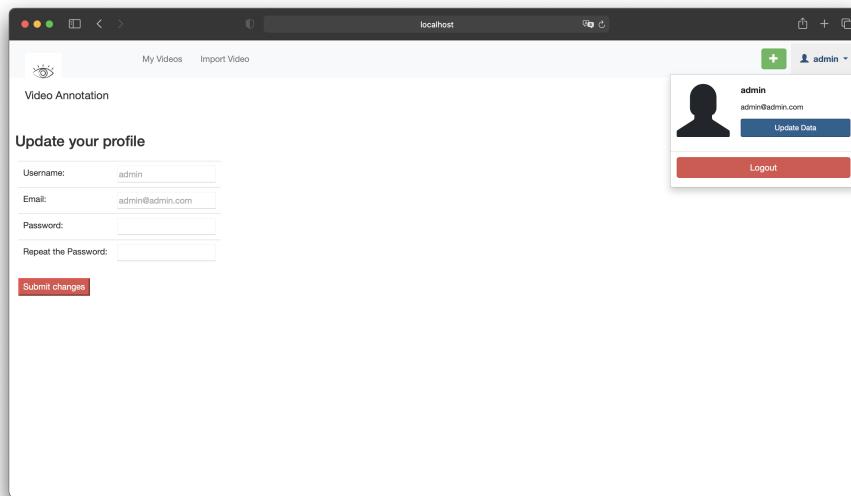


Figura 4.13: Página de Atualização de Dados do Utilizador

O utilizador pode carregar novos vídeos para o sistema, associados à sua conta. Para isso, deve escolher a opção *Import Video* da *navbar*. Nesta página, presente na Figura 4.14, podendo introduzir o título da gravação, sendo que se não o fizer, o título será o nome do ficheiro e pode introduzir os valores de latitude e/ou longitude do local que pretende associar à gravação, sendo que se o *browser* tiver permissão para aceder à sua localização atual, os campos são preenchidos automaticamente. Abaixo, pode escolher o ficheiro de vídeo (em formato MPEG-4) que pretende carregar e, opcionalmente, associar a data e hora da gravação.

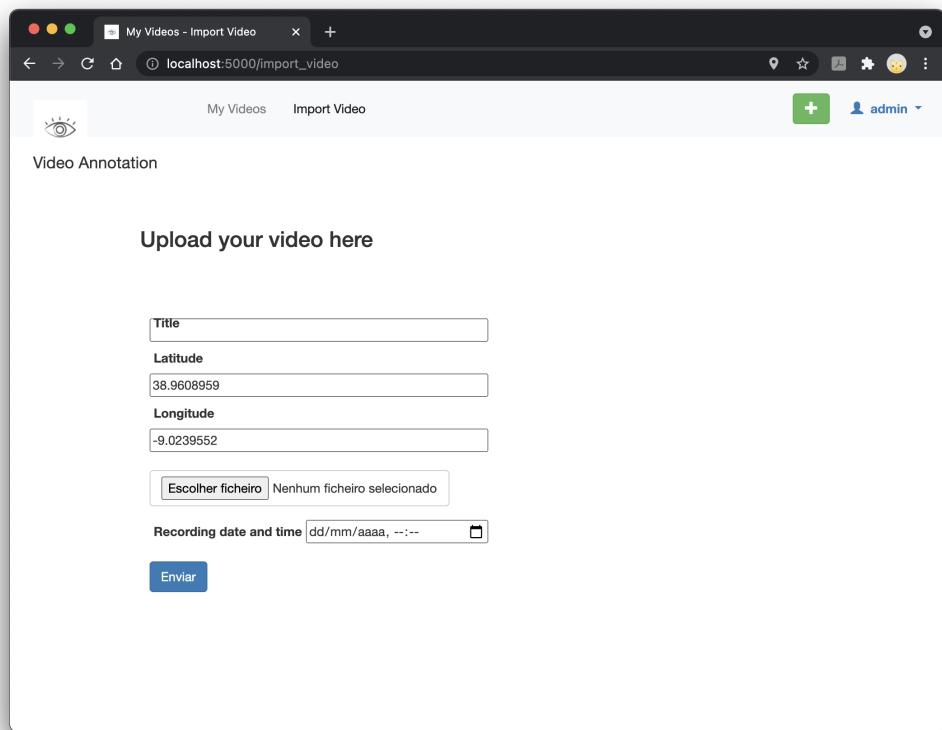


Figura 4.14: Página de *Upload* de novos vídeos

Na página de gravação de vídeos, apresentada na Figura 4.15 e acessível através do botão verde com o símbolo + na *navbar*, é possível dar início à gravação de um novo vídeo, através do botão *Start Recording* e dar um título a esse vídeo, na caixa de texto acima.

O vídeo é exibido conforme está a ser recebido e armazenado. Enquanto o vídeo é gravado, o sinal de gravação a vermelho estará intermitente e, quando o utilizador pretender terminar a gravação do vídeo, deverá pressionar o botão *Stop Recording*, que se encontrará no mesmo local que permitiu iniciar a gravação.

À sua direita, encontram-se os controlos (botões e caixa de texto) para adicionar anotações de emoções e frases personalizadas. Para registar uma emoção, o utilizador deve pressionar o botão da emoção pretendida no quadro com as possíveis emoções representadas por *emojis*, no lado direito da janela da aplicação, ficando com fundo verde a emoção selecionada. Assim, registou o instante inicial dessa emoção e, para registar o instante final, deverá pressionar o mesmo botão, que deixará de ter o fundo colorido. Caso pretenda introduzir uma anotação textual, deve escrever na caixa de texto e pressionar a tecla *Enter*.

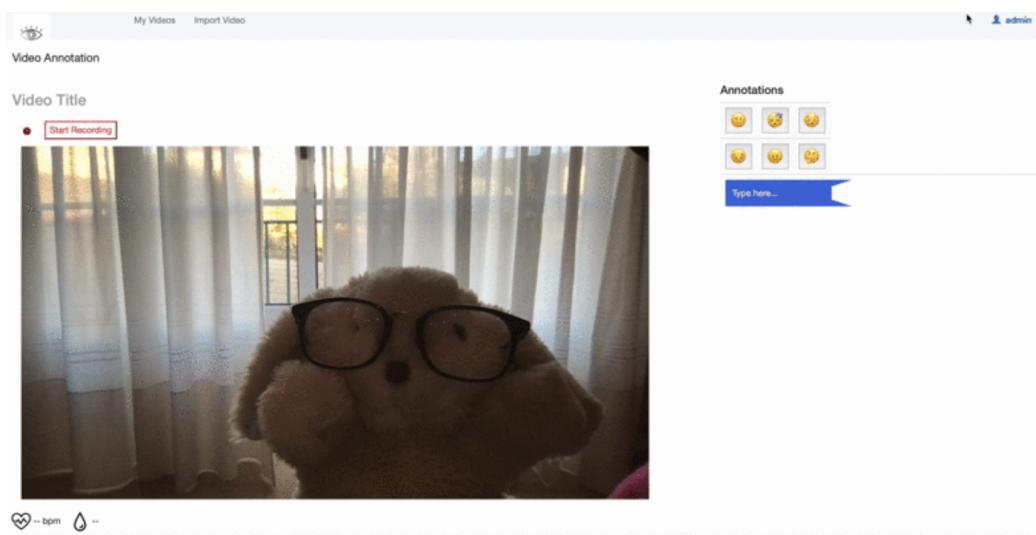


Figura 4.15: Página de Gravação de vídeos

Na página acessível pelo link *My Videos* na navbar, apresentada na Figura 4.16, são dispostos em grelha com, no máximo, 12 itens por página, os vídeos associados ao utilizador com sessão iniciada. Em cada card, que quando pressionado encaminha o utilizador para a página de revisão desse vídeo, são apresentados o *thumbnail* do vídeo, o título e a sua data de

gravação.

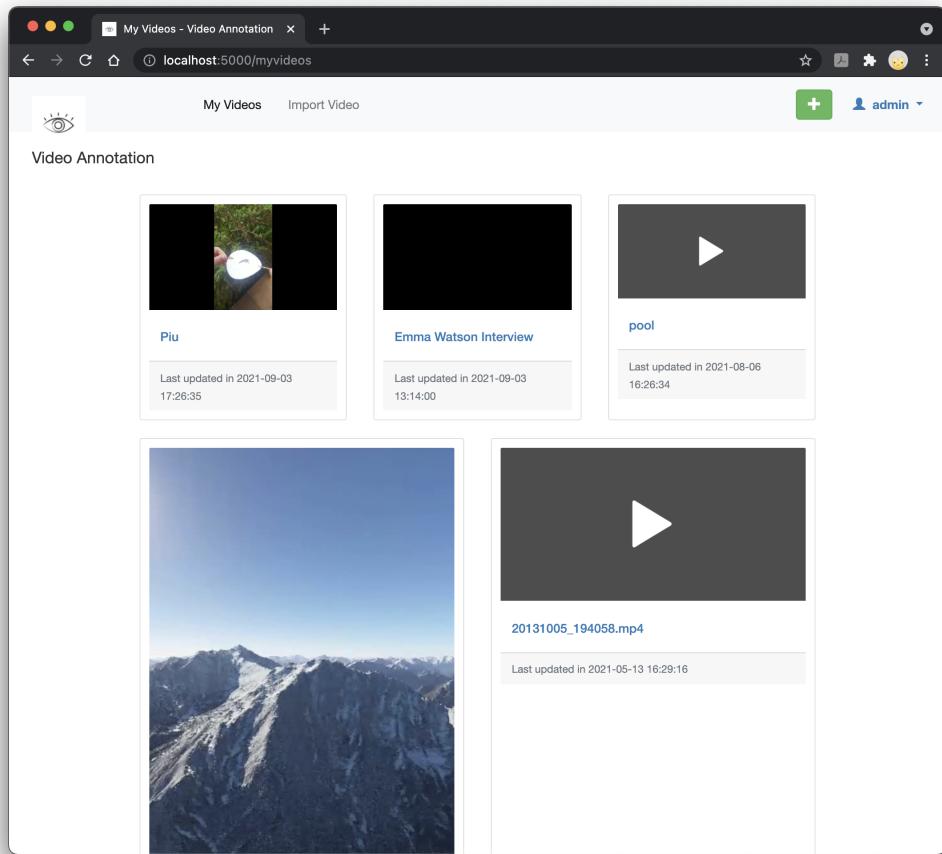


Figura 4.16: Página de vídeos do utilizador

Na página de revisão de vídeos, presente na Figura 4.17, é possível rever o vídeo selecionado, a sua localização e, data e hora de gravação, e as anotações a ele associadas. A lista disposta inicialmente provém da interpretação do ficheiro JSON com as anotações associadas a esse vídeo, se já existir. Conforme o vídeo avança, temporalmente, é automaticamente disposta no início da lista de anotações a anotação que possa estar associada a esse instante.

Em cada item da lista, são providenciadas as opções de eliminar a anotação correspondente ou editar o seu conteúdo e instante temporal.

De forma semelhante à página de gravação de novos vídeos, podem-se

fazer novas anotações de emoções e textuais, manualmente. Adicionalmente, ao pressionar o botão *Auto identify emotions*, é possível executar a deteção automática pelo algoritmo da *Deepface*. O botão *Save changes* permite obter as anotações da base de dados e guardá-las no ficheiro JSON correspondente ao vídeo.

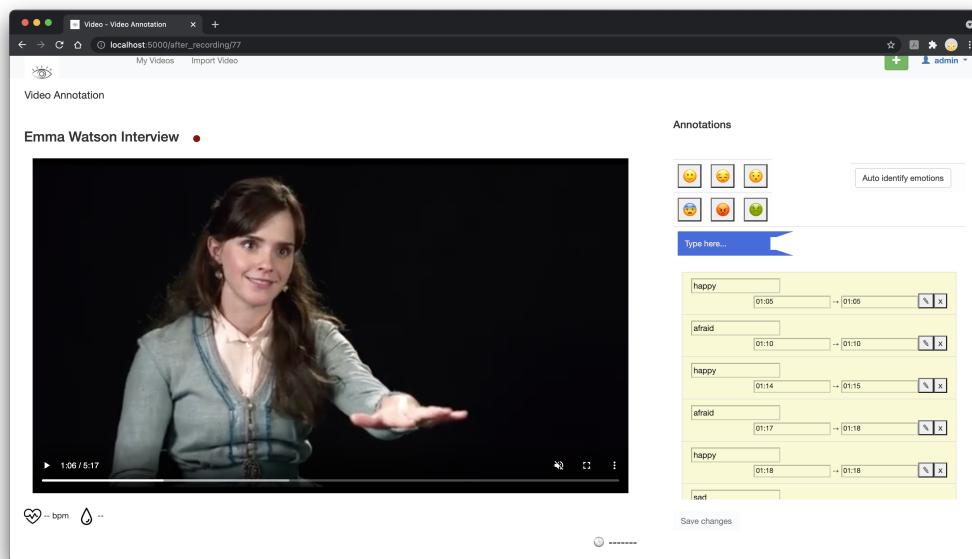


Figura 4.17: Página de Revisão de vídeos

Capítulo 5

Validação e Testes

Neste capítulo serão abordados testes aos métodos apresentados no Capítulo 4. Para a análise dos métodos ter-se-á em conta aspectos como a qualidade, precisão e robustez dos resultados.

5.1 Validação do algoritmo com *Deepface*

O modelo de deteção escolhido para *backend* para a biblioteca *Deepface*, referida na Subsecção 4.2.3, foi o *OpenCV* visto que, na Figura 5.1, este modelo mostrou-se ser o mais rápido e, quando apresenta erros na deteção, são mais próximos ao resultado esperado, refletindo-se também na diminuição ligeira do grau de certeza.



Detetor	Tempo (s)	Emoção	Grau de certeza (%)
<i>OpenCV</i>	0.485	Raiva	99.9995
<i>SSL</i>	2.793	Felicidade	99.9888
<i>Dlib</i>	4.679	Raiva	99.9871
<i>MTCNN</i>	1.446	Felicidade	98.7543

Detetor	Tempo (s)	Emoção	Grau de certeza (%)
<i>OpenCV</i>	0.581	Surpresa	95.6452
<i>SSL</i>	--	Não detetada	--
<i>Dlib</i>	1.391	Felicidade	97.9459
<i>MTCNN</i>	1.953	Felicidade	99.8830

Figura 5.1: Comparação temporal e de resultados da deteção de emoções em imagens entre os modelos de deteção disponíveis na biblioteca *Deepface*

De modo a estipular-se um limite mínimo ao nível de confiança das emoções detetadas a considerar, criou-se um conjunto de imagens a serem analisadas pela *Deepface* e analisou-se os graus de confiança obtidos nas classificações bem sucedidas. Como mostra o gráfico da Figura 5.2, o valor médio do grau de confiança das classificações corretas foi de, aproximadamente, 85. Destaca-se também que, para analisar 19 (dezanove) imagens, o algoritmo com recurso ao modelo de deteção *OpenCV* demorou 2,5 segundos, tal como o *SSD*, apesar de que este segundo acertou apenas na emoção de 5 (cinco) imagens, menos 2 (duas) que o primeiro.

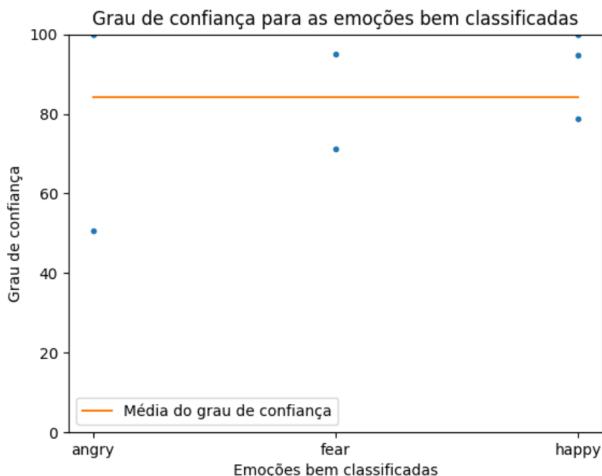


Figura 5.2: Gráfico de relação entre emoções classificadas e o grau de confiança obtido

Com a finalidade de testar o algoritmo de deteção de emoções realizado em vídeos, explicitado na Subsecção 4.3.5, elaborou-se um vídeo composto por partes de outros vídeos com atores a esboçarem alguns tipos de emoções. Depois, utilizou-se o algoritmo criado para se verificarem os resultados de algumas medidas de desempenho, comparando as classificações esperadas com as detetadas automaticamente.

A matriz de confusão resultante da análise com o algoritmo final de deteção de emoções em vídeos, com a anotação de emoções tendo em conta a predominante num conjunto de *frames* analisadas, é a presente na Figura 5.3. A exatidão (*accuracy*) obtida foi de, aproximadamente, 36%. A percentagem não é tão alta como as provenientes de imagens devido à análise menos

exhaustiva que a *frame a frame* e pelos momentos de transição entre emoções, em que se detetam mais expressões consideradas neutras, ou sem emoção. Ainda assim, considera-se uma análise satisfatória, visto que as emoções detetadas incorretamente se traduzem frequentemente na não anotação do que na anotação de outras emoções, em virtude de serem atribuídas à classe neutral que não é anotada.

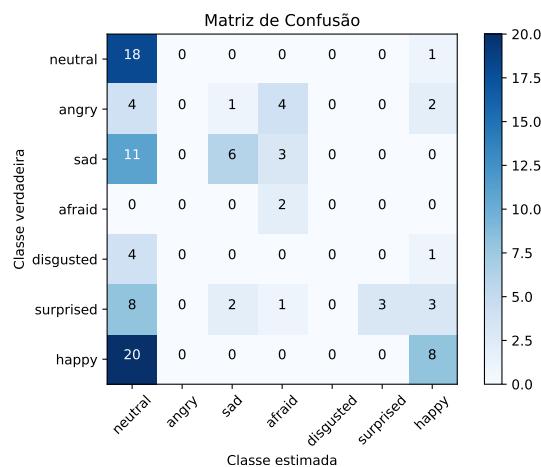


Figura 5.3: Matriz de confusão dos resultados obtidos da classificação de emoções num vídeo

5.2 Validação do algoritmo com *EmotiBit*

Com o objetivo de avaliar a qualidade dos dados captados pela *EmotiBit*, realizou-se uma experiência onde o indivíduo encontra-se sentado com o *wearable* colocado no pulso, dentro de um ambiente fechado com temperatura e humidade controladas. As Figuras 5.4, 5.5, 5.6 apresentam o sinal PPG captado pelos sensores de luz verde, infravermelho e vermelho, respectivamente.

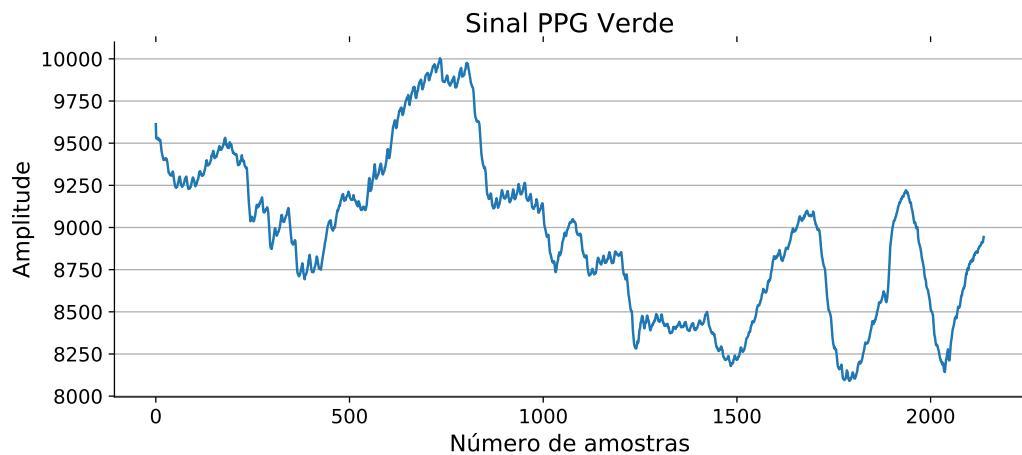


Figura 5.4: Sinal PPG captado pelo sensor de luz verde da *EmotiBit*

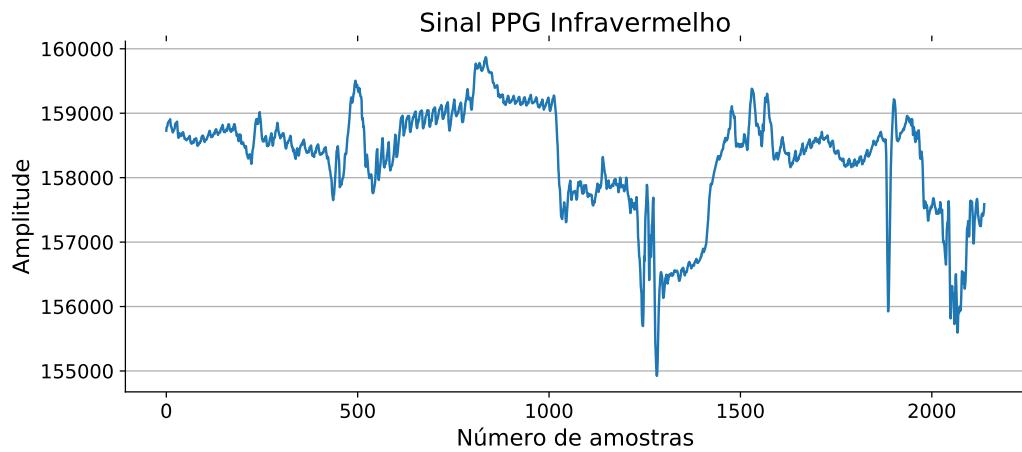


Figura 5.5: Sinal PPG captado pelo sensor infravermelho da *EmotiBit*

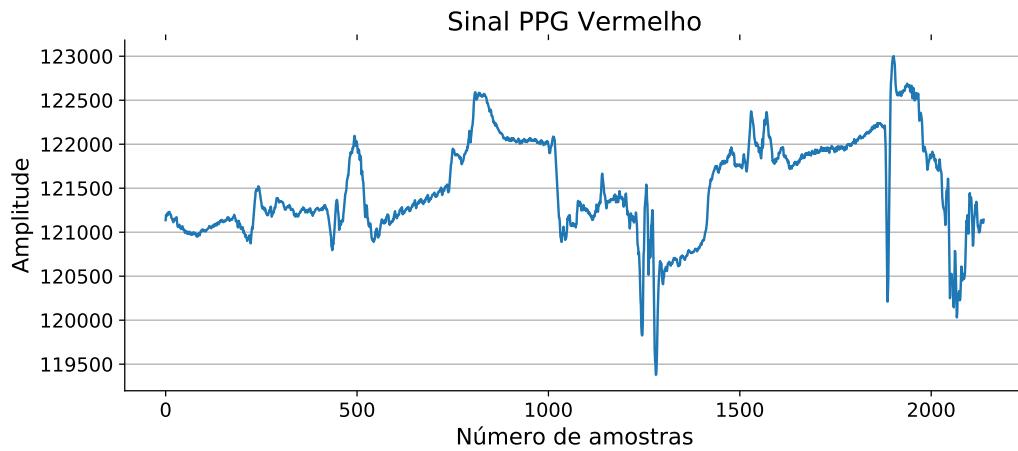


Figura 5.6: Sinal PPG captado pelo sensor de luz vermelha da *EmotiBit*

Comparando os sinais captados com o sinal do conjunto de dados [Siam, 2019] (Figura 4.7), pode-se constatar que os sinais obtidos pela *EmotiBit* não apresentam padrões reconhecíveis de um sinal PPG, bem como apresentam alguma instabilidade nos valores de amplitude. Também é possível observar que o sinal na Figura 5.4 não apresenta nenhuma correlação com os restantes sinais.

As Figuras 5.7 e 5.8 apresentam os valores experimentais da resposta da atividade eletrodérmica e da temperatura corporal do indivíduo observado.

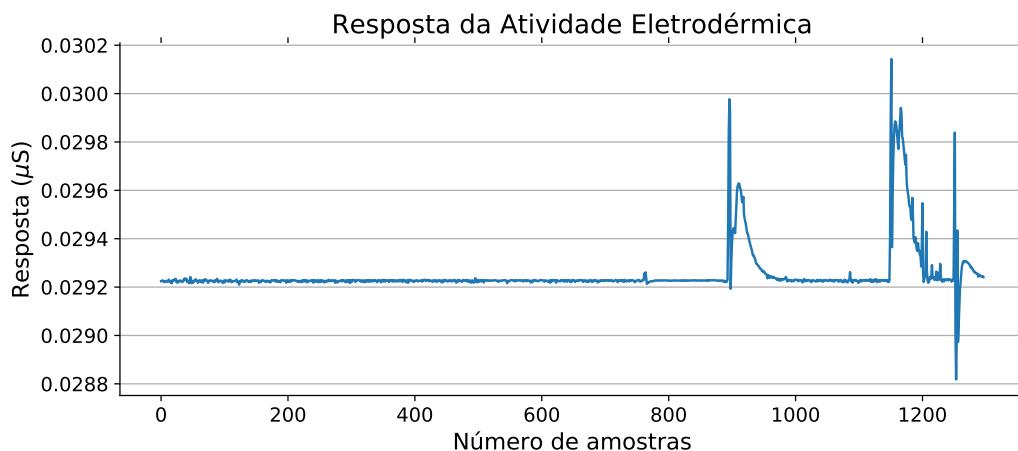


Figura 5.7: Resposta da Atividade Eletrodérmica captada pela *EmotiBit*

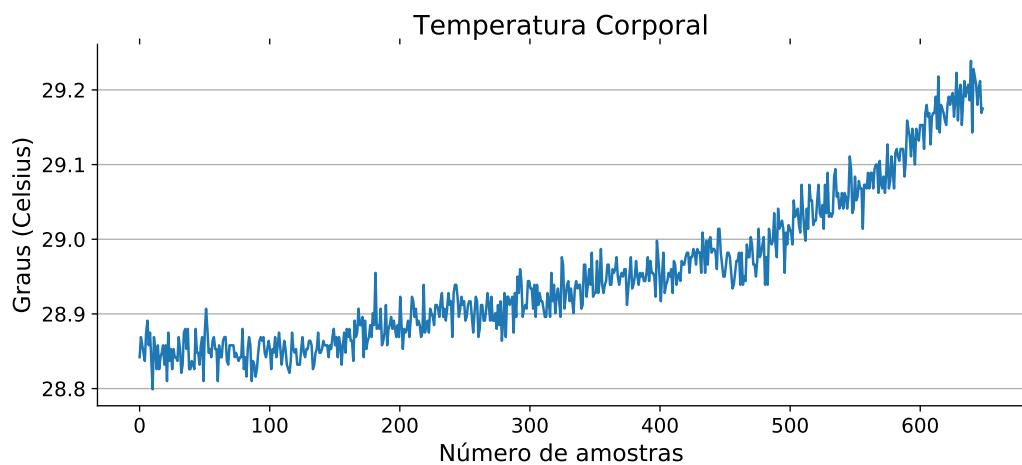


Figura 5.8: Temperatura corporal captada pela *EmotiBit*

Como se pode observar na Figura 5.7, os valores da resposta da atividade eletrodérmica encontram-se na ordem dos nano-Siemens (unidade de medida da EDA), o que indica que o indivíduo apresenta uma condutividade elétrica anormal. Esta situação deve-se ao facto dos elétrodos da *EmotiBit* apresentarem alguma oxidação que, por conseguinte, afeta a captação dos valores.

Com base na publicação [Salim, 2019], é importante ter em consideração quais os valores desejados para o estudo/aplicação a conduzir, antes de se proceder à extração de dados. Assim, não foi possível desenvolver um algoritmo capaz de detetar padrões ou definir quaisquer limiares com os valores de teste obtidos.

Na Figura 5.8 é possível observar um aumento gradual da temperatura corporal do indivíduo. Tendo em conta as características da experiência, este crescimento indica que o sensor não está calibrado, visto que a temperatura máxima detetada pelo sensor encontra-se abaixo da temperatura corporal mínima.

Deste modo, os dados captados pela *EmotiBit* não são fiáveis para complementar as emoções detetadas pela *Deepface*.

Capítulo 6

Conclusões e Trabalho Futuro

Com este projeto conclui-se que a deteção de emoções é uma tarefa bastante complexa, não só a nível humano, como a nível computacional. A anotação de vídeos poderá permitir guardar, não só dados relativos a emoções observadas ou automaticamente detetadas, como também acerca de outros detalhes que o utilizador pretenda associar ao momento do vídeo.

A comunicação em tempo-real foi concretizada com resultados satisfatórios num contexto local, mas crê-se que, com as tecnologias optadas, essa característica não apresentará alterações significativas, como um pior desempenho no *delay* da transmissão, com os clientes e o servidor em redes distintas, por exemplo.

O sistema implementado mostrou-se seguro e robusto, e a utilização de câmaras 3D, como a *Intel RealSense*, unicamente para deteção de emoções, não se revelou essencial, podendo ser utilizada uma câmara comum, exceto se se pretender, futuramente, detetar e armazenar dados acerca de detalhes da expressão facial, como os *facial landmarks*.

Apesar de compreendermos os benefícios da junção de dados fisiológicos, para a classificação de emoções aos restantes dados obtidos aquando da gravação do vídeo, a *EmotiBit* acabou por colocar alguns entraves na implementação da arquitetura e dos objetivos idealizados inicialmente.

Assim, como trabalho futuro, sugere-se a integração dos dados adquiridos pela *EmotiBit* ou outro módulo de extração de dados fisiológicos, em tempo real e a criação ou utilização de um modelo de classificação de emoções com base, não só na imagem, mas também em conjunto com os sinais fisiológicos.

A aplicação desenvolvida permite a integração de outros sinais fisiológicos,

necessitando-se de criar o adaptador para aquisição e inserção desses dados nos outros sinais adquiridos.

Apêndice A

Gestão de Versões

Durante o desenvolvimento do projeto recorreu-se à plataforma *Github* como repositório para o gestor de versões. O *Github* é uma plataforma Web que utiliza o Git, que permite a colaboração em tempo real entre programadores.

A Figura A.1 representa a estrutura de acesso ao repositório para os orientandos e para os orientadores, no *Github*. Os orientandos realizam ações como *push* e *pull* para o repositório enquanto os orientadores apenas têm permissão para visualizarem o código-fonte.

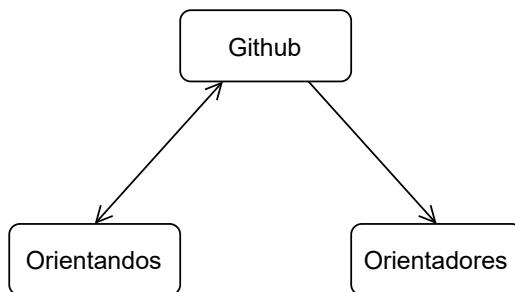


Figura A.1: Estrutura de acesso ao repositório no *Github*

Para simplificar a implementação, dividiu-se o projeto em vários módulos que, ao longo do desenvolvimento, teriam uma maior correlação. Assim, construiu-se três ramos, *Adapter*, *Webpages* e *Db*. O ramo *Full stream* foi adicionado posteriormente.

O ramo *Adapter* tinha como objetivo a extração dos dados audiovisuais e na organização das mensagens para serem enviadas ao servidor. O ramo

Webpages tinha como foco a construção de páginas HTML e pela interação pessoa-máquina. Por fim, o ramo *Db* foi responsável pela estrutura da base de dados para o projeto.

O ramo *Full stream* tinha como foco o envio dos dados audiovisuais para o servidor, a receção dos mesmos dados e a retransmissão.

A Figura A.2 representa uma simplificação do fluxo de versões dos vários ramos no decorrer do projeto. O ramo principal do repositório (*Master*) é o ramo onde os restantes ramos vão unir. Cada união só ocorre quando uma funcionalidade está terminada.

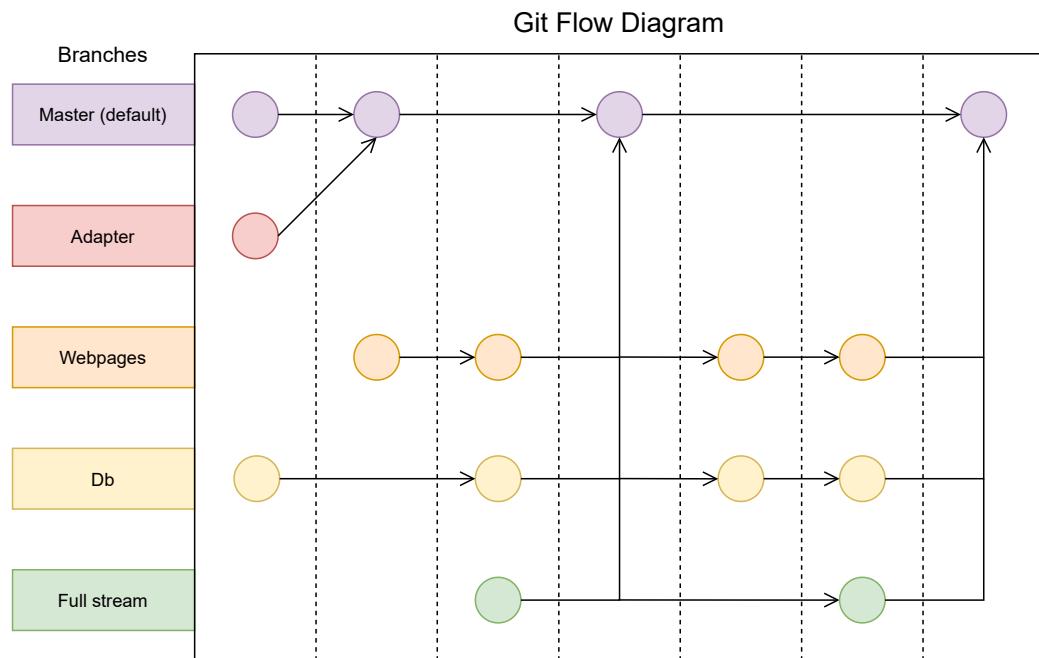


Figura A.2: Diagrama de fluxo do repositório

Tal como se pode observar na Figura A.2, o ramo *Adapter* foi unido ao ramo principal, assim que foi concluído o seu propósito. Também é possível observar uma união em duas versões que coincide com a apresentação FEIM e a apresentação final.

O diagrama é uma simplificação do fluxo de versões, de modo a simplificar apresentação do repositório e a utilização das funcionalidades do Git. O número total de versões não corresponde ao número de colunas apresentado, assim como as uniões em bloco não foram realizadas num único passo.

Apêndice B

Diagrama de Robustez

Na fase de conceção do projeto, foram elaborados os casos de utilização, bem como, o diagrama de robustez, que permite analisar se os casos de utilização são suficientes para os requisitos do sistema, que se encontra na Figura B.1.

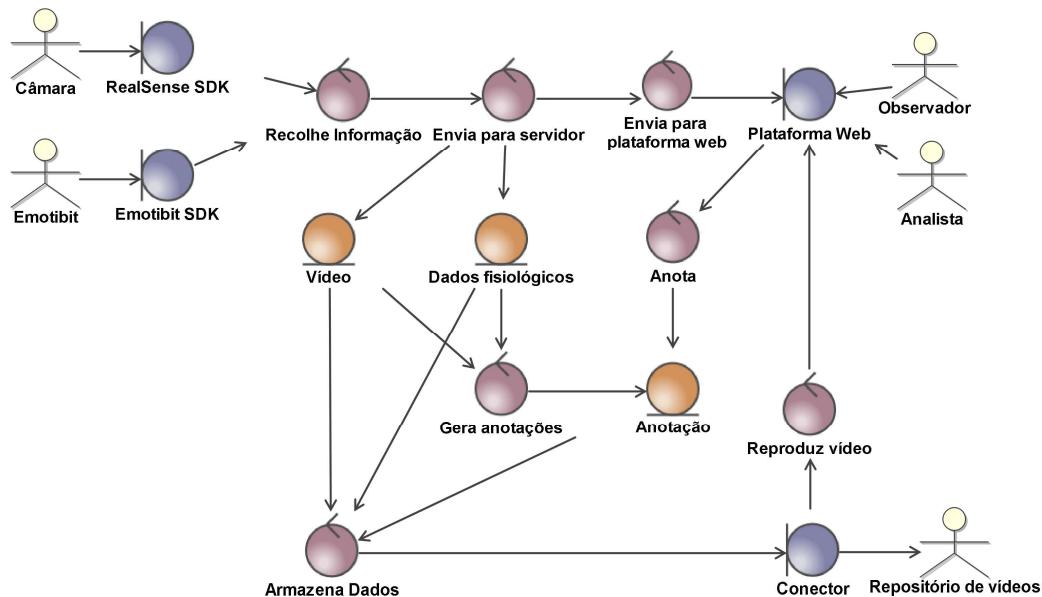


Figura B.1: Diagrama de robustez

Como é possível observar na Figura B.1, os vários atores executam processos através das *interfaces* fornecidas pelo sistema. Os atores Câmara e

Emotibit representam o cliente observado e os atores Observador e Analista pertencem ao cliente observador.

O sistema depende de três entidades categorizadas por: vídeos, dados fisiológicos e anotações. As anotações estão dependentes dos vídeos, visto que é sobre os vídeos que são realizadas anotações.

É de se notar que todas as funcionalidades do sistema estão representados nesta figura.

Bibliografia

- [Anderson, 2019] Anderson, J. (2019). An intro to threading in python. <https://realpython.com/intro-to-python-threading/#what-is-a-thread>.
- [Bimbo, 2011] Bimbo, A. D. (2011). Mpeg-7. http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A13%20MPEG7%20standard.pdf.
- [Bota et al., 2019] Bota, P. J., Wang, C., Fred, A. L. N., e Silva, H. P. D. (2019). A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8849996>.
- [Chang et al., 2001] Chang, S.-F., Sikora, T., e Puri, A. (2001). Overview of the mpeg-7 standard. <https://www.ee.columbia.edu/ln/dvmm/publications/01/csvt-mpeg7.pdf>.
- [Dallacosta et al., 2004] Dallacosta, A., de Souza, D. D., Tarouco, L. M. R., e Franco, S. R. K. (2004). O vídeo digital e a educação. https://www.researchgate.net/publication/265920917_0_Video_Digital_e_a_Educacao.
- [Flask, 2021] Flask (2021). Welcome to flask - flask documentation (2.0.x). <https://flask.palletsprojects.com/en/2.0.x/>.
- [iMotions, 2005] iMotions (2005). Analysis platform for human behaviour research. <https://imotions.com/>.
- [Intel, 2019] Intel (2019). Intel realsense - overview. <https://github.com/IntelRealSense/librealsense>.

- [Kumar, 2021] Kumar, H. (2021). Introduction to micro web framework flask. <https://medium.com/featurepreneur/introduction-to-micro-web-framework-flask-78de9289270b>.
- [Mozilla, 2021a] Mozilla (2021a). The websocket api (websockets). https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API?retiredLocale=pt-PT.
- [Mozilla, 2021b] Mozilla (2021b). What is a web server? https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server.
- [National Ecological Observatory Network, 2020] National Ecological Observatory Network (2020). Hierarchical data formats - what is hdf5? <https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5>.
- [Oracle, 2019] Oracle (2019). What is mysql? <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
- [Oracle, 2021] Oracle (2021). What is a socket? - the javaTM tutorials. <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>.
- [Python Software Foundation, 2021] Python Software Foundation (2021). socket — low-level networking interface. <https://docs.python.org/3/library/socket.html>.
- [Rubin e Talarico, 2009] Rubin, D. C. e Talarico, J. M. (2009). A comparison of dimensional models of emotion: Evidence from emotions, prototypical events, autobiographical memories, and words. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2784275/>.
- [Salim, 2019] Salim, R. (2019). What is eda peak detection and how does it work? <https://imotions.com/blog/eda-peak-detection/>.
- [Serengil, 2021] Serengil, S. (2021). Deepface – the most popular open source facial recognition library. <https://viso.ai/computer-vision/deepface/>.

- [Serengil e Ozpinar, 2020] Serengil, S. I. e Ozpinar, A. (2020). Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, p. 23–27. IEEE.
- [Shiota, 2016] Shiota, M. N. (2016). Ekman’s theory of basic emotions. <https://www.doi.org/10.4135/9781483346274.n85>.
- [Siam, 2019] Siam, Ali; Abd El-Samie, F. A. E. A. E.-B. N. E. G. (2019). Real-world ppg dataset. <https://data.mendeley.com/datasets/yynb8t9x3d/1>.
- [Socket.IO, 2010] Socket.IO (2010). Socket.io home page. <https://socket.io/>.
- [Team, 2019] Team, T. E. (2019). Emotibit wearable biometric sensing for any project. <https://www.emotibit.com/>.
- [Teuhola, 2012] Teuhola, J. (2012). Storage and retrieval of data. http://staff.cs.utu.fi/kurssit/multimedia_databases/autumn_2012/slides/1-Intro.pdf.
- [The HDF Group, 2006] The HDF Group (2006). Hdf5 library and file format. <https://www.hdfgroup.org/solutions/hdf5/>.
- [Yamashita et al., 2018] Yamashita, R., Nishio, M., Do, R. K. G., e Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. <https://doi.org/10.1007/s13244-018-0639-9>.
- [Yang et al., 2018] Yang, D., Cheng, Y., Zhu, J., Xue, D., Abt, G., Ye, H., e Peng, Y. (2018). A novel adaptive spectrum noise cancellation approach for enhancing heartbeat rate monitoring in a wearable device. *IEEE Access*, PP:1–1.