

Thermo Sense – Remote Sensing System based on Thermography

Ana Catarina das Dores Ribeiro

Master Degree in Telecommunications and Computer Engineering,

Supervisor:

PhD Octavian Adrian Postolache, Associate Professor with Habilitation
Iscte – Instituto Universitário de Lisboa

Coordinating Professor:

PhD José Miguel Dias Pereira, Coordinating Professor
Escola Superior de Tecnologia/IPS

November, 2021

Science and Information Technologies

Thermo Sense – Remote Sensing System based on Thermography

Ana Catarina das Dores Ribeiro

Master in Telecommunications and Informatic Engineering

Supervisor:

Prof. Dr. Octavian Adrian Postolache, Associate Professor com Agregação
Iscte-iul

Co-Supervisor:

Prof. Dr. José Miguel Dias Pereira, Professor Coordenador
EST/IPS

November, 2021

Acknowledgments

I would like to express my sincere thanks to my coordinator, Professor Octavian Postolache, for all the availability and support during the realization of this project.

I would like also to acknowledge the institutional support received from Instituto de Telecomunicações and Iscte – Instituto Universitário de Lisboa that provide me hardware as software required for the developments of my research.

My biggest thanks goes to all my family for the unconditional support throughout all of this years, specially to my mother, father and sister.

Last but not least, a huge thanks to all my friends who accompanied me on this journey and colleagues, especially to the ones that also went to the laboratory that enabled some discussions of ideas that really helped me complete this project.

Resumo

A termografia é uma ferramenta de diagnóstico que cada vez mais começa a ganhar reconhecimento pela comunidade científica. Algumas vantagens desta tecnologia são que os resultados são obtidos de uma maneira mais prática pois a camara termográfica é portátil e também, a maior parte das opções que existem no mercado têm um custo inferior a outras ferramentas de diagnóstico. Conjuntamente, e ao contrário de tecnologias como tomografias, raio-X, etc., a termografia é livre de radiação.

No entanto, existe ainda falta de suporte e soluções digitais para levar esta técnica de imagiologia médica para outro nível de automatização no processo de obtenção e processamento de imagens termográficas.

Foi neste sentido que esta dissertação foi desenvolvida, com a criação de uma plataforma móvel Android em que seja permitido a profissionais de saúde realizarem o registo eletrónico de pacientes, visualizarem imagens termográficas diretamente da camara termográfica e associarem as mesmas aos pacientes.

Foi também desenvolvido um software de processamento de imagem com a ajuda do MATLAB. Assim é também possível fazer o processamento das imagens, diretamente na plataforma, o que torna o processo de diagnóstico mais rápido, automático e sem necessidade de extrair as imagens manualmente e introduzi-las num software de processamento de imagem.

A eficácia do software de processamento de imagem bem como a correta comunicação entre aplicação móvel e camara termográfica foram testadas experimentalmente e os vários resultados foram incluídos nesta dissertação.

Palavras-chave: *Termografia; Processamento de imagem; Imagiologia; MATLAB; Aplicação Android.*

Abstract

Thermography is a diagnostic tool that is increasingly gaining recognition in the scientific community. Some advantages of this technology are that the results are obtained in a more practical way because the thermographic camera is portable and also, most options that exist on the market have a lower cost than other diagnostic tools. Together, and unlike technologies such as CT scans, X-rays, etc., thermography is radiation-free.

However, there is still a lack of support and digital solutions to take this medical imaging technique to another level of automation in the process of obtaining and processing thermographic images.

It was in this sense that this dissertation was developed, with the creation of an Android mobile platform that allows healthcare professionals to perform the electronic registration of patients, view thermographic images directly from the thermographic camera and associate them with patients.

Image processing software was also developed with the help of MATLAB. Thus, it is also possible to process the images directly on the platform, which makes the diagnostic process faster, automatic and without the need to manually extract the images and insert them into an image processing software.

The efficiency of image processing software as well as the correct communication between mobile application and thermographic camera were experimentally tested and the various results were included in this dissertation.

Keywords: Thermography; Image processing; MATLAB; Imaging; Android application.

Contents

Acknowledgments	iii
Resumo	v
Abstract.....	vii
List of Figures.....	xii
List of Tables.....	xv
List of Acronyms.....	xvi
Chapter 1	2
1. Introduction	2
1.1. Motivation.....	2
1.2. Objectives.....	2
1.3. Structure of Dissertation	3
Chapter 2	4
2. State of the Art.....	4
2.1. Physics behind Infrared.....	4
2.1.1. Electromagnetic Radiation (EMR)	4
2.1.2. Electromagnetic Spectrum	4
2.1.3. IR and the human body	5
2.2. What's Infrared Thermography (IRT)?	6
2.3. Usage of IRT for assessing skin temperature	6
2.4. Related work	7
2.5. How IRT imaging systems work.....	7
2.6. Teledyne FLIR	9
2.6.1. FLIR Thermography Cameras	9
2.7. FLIR software for image processing	11
2.7.1. FLIR Thermal Studio Suite	11
2.7.2. ResearchIR.....	12
2.8. Mobile applications with thermography usage	13
2.9. Data Storage.....	14
2.10. Mobile application	17
Chapter 3	19
3. System Description	19
3.1. Overview	19
3.2. System Hardware	20
3.2.1. FLIR E60	20
3.2.2. Si7021 I2C Humidity and Temperature Sensor	20
3.2.3. Android smartphone or tablet	21
3.3. Mobile Application	21
3.3.1. Mobile Application Implementation	22
3.3.2. Data Structure in Firebase.....	29
3.3.2.1. Structure in Firebase Realtime Database	30
3.3.2.2. Structure in Firebase Cloud Storage.....	32
3.3.3. Communication with Firebase	33
3.3.3.1. Firebase SDK.....	33
3.3.3.2. REST API.....	37
3.4. MATLAB Processing.....	37
3.4.1. Overview.....	37
3.4.2. MATLAB Script Implementation	38

Chapter 4	42
4. Experimental Description.....	42
4.1. Experimental Work	42
4.1.1. Tailored VR Game.....	42
4.1.2. Experimental Methodology.....	44
4.1.3. Experimental Setup	44
4.1.4. Experimental Protocol.....	45
4.2. Results and Discussions.....	46
4.2.1. Validation Patient.....	47
4.2.1.1. High Angles Sessions Analysis of Validation Patient	47
4.2.1.2. Low Angles Sessions Analysis of Validation Patient	50
4.2.2. All volunteers Analysis.....	52
4.2.2.1. High Angles in all volunteers	54
4.2.2.2. Low Angles Sessions in all volunteers	58
4.2.2.3. All sessions with all volunteers analysis	62
4.2.3. Correlation Value Behavior	65
Chapter 5	68
5. Conclusion and Future Work	68
5.1. Conclusions	68
5.2. Future Work	68
References.....	70

List of Figures

Figure 1 - Electromagnetic Spectrum from [4]	5
Figure 2 - Heat transfer in human body from [7]	5
Figure 3 - Thermograms images: a) before cold provocation, b) post-cooling, c) post-rewarming from [26]	7
Figure 4 - How a thermal imaging system works from [28]	8
Figure 5 - Flir Thermal Studio Suite software from [36]	12
Figure 6 - ResearchIR software from [39]	13
Figure 7 - Flir Tools app, example of a live stream from (1 st image), connecting to a thermal camera (2 nd image), importing images from storage (3 rd image), editing images by adding measurements tools (4 th image) from [40]	13
Figure 8 - Percentage of Android users from [50]	17
Figure 9 – Communication between thermography camera, mobile application, database and MATLAB	19
Figure 10 - FLIR E60 thermal camera	20
Figure 11 - Developed mobile application running on Samsung Galaxy Tab S2 device	21
Figure 12 – Choice of Android version for the project (left) Distribution of android version's percentages used (right)	22
Figure 13 - CameraFragment, discovering camera (left) and connecting to the camera (right)	24
Figure 14 – InfoFragment when pressing the button “Check Camera Info” in Figure 14.....	25
Figure 15 - ImageAssociationFragment when pressing “Associate Image” button in Figure 14	26
Figure 16 - Sharing option when pressing the thermal image in Figure 16.....	27
Figure 17 - PeopleFragment, list of all patients (left) and add a patient by pressing “Add Patient” button (right).....	28
Figure 18 - PatientProfileFragment when pressing a patient in the list in Figure 18	29
Figure 19 - Structure of a patient in JSON format.....	30
Figure 20 - Main nodes in Firebase Realtime Database.....	30
Figure 21 - Structure of node "images" in Firebase Realtime Database.....	31
Figure 22 - Structure of node "patients" in Firebase Realtime Database.....	31
Figure 23 - Structure of node "values" in Firebase Realtime Database.....	31
Figure 24 - Structure of all patients in Cloud Storage	32
Figure 25 - Structure of a patient in Cloud Storage	33
Figure 26 - Firebase dependencies	33
Figure 27 - Implementation of PatientsInfo class	34
Figure 28 - Implementation of writing in Firebase Realtime Database	35
Figure 29 - Implementation of reading in Firebase Realtime Database	35
Figure 30 - Implementation of writing in Cloud Storage	36
Figure 31 - Implementation of reading in Cloud Storage.....	36
Figure 32 - Implementation of image upload with Picasso library	37
Figure 33 - PATCH request in MATLAB.....	37
Figure 34 - Overview of MATLAB and Firebase interaction	38
Figure 35 - Implementation of how list of images is taken from database e putted in the correct format to analyze.....	40
Figure 36 - Implementation of the main aspects of the image analysis.....	41

Figure 37 - Implementation of mean temperature and writing on database	41
Figure 38 - Volunteer performing High Angles session (1 st image) and Low Angles Session (2 nd image).....	43
Figure 39 – Smart Physio app from [56].....	43
Figure 40 - Duration of game sessions	44
Figure 41 - Experimental Setup without volunteer.....	45
Figure 42 - Experimental Setup with volunteer	45
Figure 43 - Average Room Humidity during the experiments in all volunteers	47
Figure 44 - Average Room Temperature during the experiments in all volunteers	47
Figure 45 - VALIDATION PATIENT, High Angles; shoulder to elbow temperature on the right region of the arm	48
Figure 46 - VALIDATION PATIENT, High Angles; shoulder temperature on the right region of the arm.....	49
Figure 47 - VALIDATION PATIENT, High Angles; shoulder to elbow temperature on the left region of the arm	49
Figure 48 - VALIDATION PATIENT, High Angles; shoulder to elbow temperature on the left region of the arm	50
Figure 49 - VALIDATION PATIENT, Low Angles; shoulder to elbow temperature on the right region of the arm	50
Figure 50 - VALIDATION PATIENT, Low Angles; shoulder temperature on the right region of the arm.....	51
Figure 51 - VALIDATION PATIENT, Low Angles; shoulder to elbow temperature on the left region of the arm	51
Figure 52 - VALIDATION PATIENT, Low Angles; shoulder temperature on the left region of the arm	52
Figure 53 – Average Absolute Errors Overall in all sessions of VALIDATION PATIENT	52
Figure 54 - Starting temperatures of the right region of the arm of all volunteers	53
Figure 55 - Starting temperatures of the left region of the arm of all volunteers	54
Figure 56 - 60 seconds high angles game sessions of all volunteers right region of the arm .	54
Figure 57 - 60 seconds high angles game sessions of all volunteers left region of the arm....	55
Figure 58 - 90 seconds high angles game sessions of all volunteers right region of the arm .	56
Figure 59 - 90 seconds high angles game sessions of all volunteers left region of the arm....	56
Figure 60 - 120 seconds high angles game sessions of all volunteers right region of the arm	57
Figure 61 - 120 seconds high angles game sessions of all volunteers left region of the arm..	57
Figure 62 - 150 seconds high angles game sessions of all volunteers right region of the arm	58
Figure 63 - 150 seconds high angles game sessions of all volunteers left region of the arm..	58
Figure 64 - 60 seconds low angles game sessions of all volunteers right region of the arm ..	59
Figure 65 - 60 seconds low angles game sessions of all volunteers left region of the arm.....	59
Figure 66 - 90 seconds low angles game sessions of all volunteers right region of the arm ..	60
Figure 67 - 90 seconds low angles game sessions of all volunteers left region of the arm.....	60
Figure 68 - 120 seconds low angles game sessions of all volunteers right region of the arm.	61
Figure 69 - 120 seconds low angles game sessions of all volunteers left region of the arm...	61
Figure 70 - 150 seconds low angles game sessions of all volunteers right region of the arm.	62
Figure 71 - 150 seconds low angles game sessions of all volunteers left region of the arm...	62
Figure 72 - All patients game sessions values right arm (RA) High Angles(HA)	63
Figure 73 - All patients game sessions values left arm (LA) High Angles (HA).....	63
Figure 74 - All patients game sessions values right arm (RA) Low Angles (LoA).....	64

Figure 75 - All patients game sessions value left arm (LA) Low Angles (LoA)	64
Figure 76 – Cumulative temperature	65
Figure 77 - Correlation Value Evolution in high angles game sessions on the right region of the arm	66
Figure 78 - Correlation Value Evolution in high angles game sessions on the left region of the arm	66
Figure 79 - Correlation Value Evolution in low angles game sessions on the right region of the arm	66
Figure 80 - Correlation Value Evolution in low angles game sessions on the left region of the arm	67

List of Tables

Table 1 - Comparison table of different Flir Series	11
Table 2 - Comparison table of the different databases based on [41]–[45].....	15
Table 3 - Classes and Fragments description	23
Table 4 - Shoulder to Elbow temperature on the right region of the arm values	48
Table 5 – Relationship Strength of correlation value based on [57]	53
Table 6 - Average Absolute and Relative Error, and Correlation Value of all volunteers before game sessions start	54
Table 7 - Example of how Absolute Error, Relative Error, Average Absolute Error, Average Relative Error and Correlation Value are calculated.....	55
Table 8 - Average Absolute and Relative Error, and Correlation Value of all volunteers 60 seconds high angles game sessions	55
Table 9 - Average Absolute and Relative Error, and Correlation Value of all volunteers 90 seconds high angles game sessions	56
Table 10 - Average Absolute and Relative Error, and Correlation Value of all volunteers 120 seconds high angles game sessions	57
Table 11 - Average Absolute and Relative Error, and Correlation Value of all volunteers 150 seconds high angles game sessions	58
Table 12 - Average Absolute and Relative Error, and Correlation Value of all volunteers 60 seconds low angles game sessions.....	59
Table 13 - Average Absolute and Relative Error, and Correlation Value of all volunteers 90 seconds low angles game sessions.....	60
Table 14 - Average Absolute and Relative Error, and Correlation Value of all volunteers 120 seconds low angles game sessions.....	61
Table 15 - Average Absolute and Relative Error, and Correlation Value of all volunteers 150 seconds low angles game sessions.....	62

List of Acronyms

ACID - Atomicity, Consistency, Isolation, Durability
CMOS - Monolithic Complementary Metal Oxide Semiconductor
DFS - Depth-First Search
EMR - Electromagnetic Radiation
HTTPS - Hypertext Transfer Protocol Secure
HVAC/R - Heating, Ventilation, Air Conditioning, Refrigeration
IDE - Integrated Development Environment
IR - Infrared
IRT - Infrared Thermography
JSON - JavaScript Object Notation
NDB - Network Database
NoSQL - Not Structured Query Language
OLAP - Online Analytical Processing
OLTP - Online Transaction Processing
RA - Rheumatoid Arthritis
REST - Representational State Transfer
SDK - Software Development Kit
SQL - Structured Query Language
URI - Uniform Resource Identifier
URL - Uniform Resource Locators
VR - Virtual Reality
XML - Extensible Markup Language
CDN - Content Delivery Network

CHAPTER 1

1. Introduction

1.1. Motivation

Infrared Thermography (IRT) is a non-radiating technology that is used to monitor infra-red rays from objects with a temperature above absolute zero. Also, this technology has a big advantage that is being contact-free which means that is an unobtrusive measurement of skin temperature.

All bodies are constantly emitting electromagnetic radiation, and the amount depends on several factors like its surface temperature, area and characteristics. A warmer object emits more thermal radiation than cooler one, and the human body is not an exception.

The human skin has a particular characteristic when it comes to emitting radiation. It has an emissivity value of 0.98, which means that it behaves almost like a perfect black body. Because of this characteristic, the infrared values measured can be automatically converted to exact values of temperature.

When there is some disfunction in the body (circulatory problems, inflammations, infections, tumors...) sometimes there is also an increasing of skin temperature in the adjacent area and therefore an higher emission of infrared radiation can be measured. Using IRT as a diagnose technique is getting more and more impact due to the fact that is a radiation free method.

This method of diagnose is unobtrusive and can stand out from other ones that expose the human body to ionizing radiation, like X-ray or tomography, in cases where the patient cannot be exposed to radiation or when basic healthcare needs to be maintained with low cost techniques.

Nevertheless, there is still the need for more and more scientific evidence to put this technology in an higher level. For example, in aspects like, which conditions are best for assessing skin temperature, which protocols should be followed and also which pathologies can be in fact diagnosed with only the use of IRT.

1.2. Objectives

The purpose of this dissertation is:

- Development an Android mobile framework that can perform the thermographic image management and primary processing of thermographic images, that includes as main hardware component a thermographic camera;

- Creation of a mobile application with the main goal of facilitating the acquisition of thermal images from the thermography camera and the direct analysis without having to do the process manually as in the existing image process software;
- Development of an image processing software in MATLAB running on the server side;
- Acquisition of thermal images in order to extract the effectiveness of digital physical rehabilitation based on virtual reality serious game that has as main feature and focus the rehabilitation of upper limbs.
- Thermal image analysis to understand the differences between the developed software in Matlab and the software from the thermography camera brand, FLIR.

1.3. Structure of Dissertation

This dissertation includes four chapters, which are:

- Chapter 2 provides an overview of the essential literature concepts that highlight concepts useful for this project, such as infrared technology, thermography, some of its usages in physiotherapy and the comparison between different thermography cameras.
- Chapter 3 describes the architecture of the mobile application, how thermal images will be analyzed and some elements about MATLAB.
- Chapter 4 presents the experimental results of real cases with volunteers according to the implemented protocol.
- Chapter 5 shows a conclusion where the main achievements are discussed for volunteers as well as outlines future work that can be implemented to improve the system.

CHAPTER 2

2. State of the Art

This chapter is based on theoretical research that is fundamental to the development of this project. In the first section is explained the concept of infrared radiations and its connection to the human body. In the second section is explain how thermography cameras work and the specifications to take in concern. Finally, the last section is exposed similar projects that are relevant to the development of this project.

2.1. Physics behind Infrared

2.1.1. Electromagnetic Radiation (EMR)

EMR is a oscillatory phenomenon that transports energy and do not require a medium to propagate. This means that electromagnetic waves can travel not only through air and solid materials, but also through vacuum. Electromagnetic waves are produced by electric fields and magnetic fields oscillating and perpendicular to each other [1].

Infrared (IR), is EMR with wavelengths longer than the visible light, therefore invisible to the human eye [2].

2.1.2. Electromagnetic Spectrum

An electromagnetic spectrum is a scale of several EMRs such as radio waves, microwaves, IR, visible light and ultraviolet, X-rays and gamma rays. Every single one of this waves propagate at the speed of light and all of them are not possible to see at a naked eye, except from visible light [3].

As illustrated in Figure 1, as the frequency of the radiation increases, the wavelength shortens and photon energy (E) increases. This phenomenon is showed by Equation 1:

Equation 1 - Photon Energy Equation

$$E = hf = \frac{hc}{\lambda}$$

Where h is the Planck's constant, f is the frequency of radiation, c is the speed of light and λ is the wavelength of the radiation.

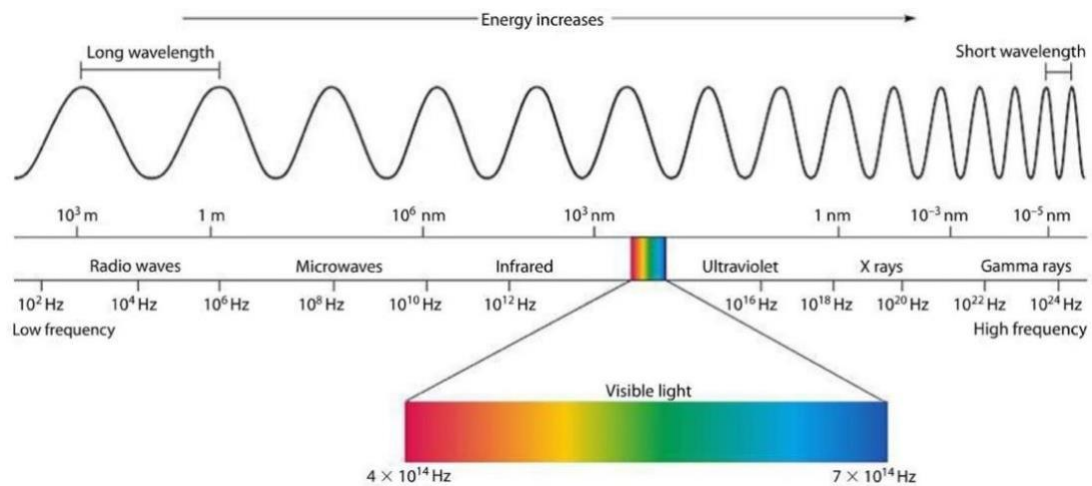


Figure 1 - Electromagnetic Spectrum from [4]

2.1.3. IR and the human body

IR radiation is emitted by all objects that have a temperature above absolute zero, such as the human body and, as stated before, the human skin has a particular characteristic when it comes to the capacity of emitting infrared energy, it has an emissivity value of 0.98, behaving itself almost like a perfect black body.

The temperature regulation of the human body is made by a process called thermoregulation in which the skin plays a major role, as over 30 % of the body's temperature receptive cells are located in it [5]. Under normal conditions, heat is transferred from the body through IR radiation (60%), evaporation (20%), thermal convection (15%), conduction (3%) [6].

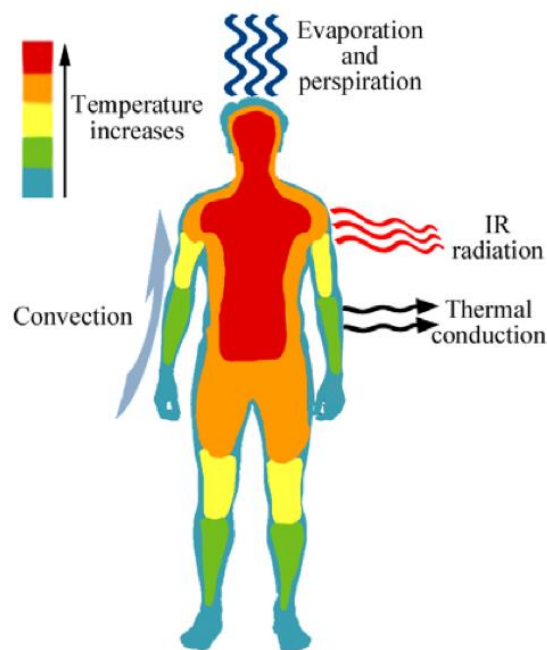


Figure 2 - Heat transfer in human body from [7]

2.2. What's Infrared Thermography (IRT)?

IRT is a non-radiating and contact-free technology that is used to monitor infra-red rays from objects with a temperature above absolute zero and also an unobtrusive measurement of skin temperature related to health problems [8][9].

When there is some disfunction in the body (circulatory problems, inflammations, infections, tumors...) there is also an increasing of skin temperature in the adjacent area and therefore an higher emission of infrared radiation can be measured. Using IRT as a diagnose technique is getting more and more impact due to the fact that is a radiation free method [10]. When compared to other methods of diagnosed like X-rays or a tomography that expose the human body to ionizing radiation it can be a decision factor when the exposure of radiation it's not an option and also with low cost techniques better suited for basic healthcare [11]. However, there is still a need for scientific evidence about which specific conditions can be diagnosed only by evaluating skin surface temperature with IRT [12].

Apart from using IRT in medicine, there is other fields of usage like surveillance, military purposes, gas leaking detections, overheating of electric components, etc [13].

2.3. Usage of IRT for assessing skin temperature

After searching some articles about the usages of thermography in medicine, it's possible to conclude that IRT is becoming more and more accepted by the medical community due to the development of the technology and the publication of more medical studies [14-25]. Some examples about the usages of thermography in medicine are:

- Plantar Fasciitis [14]
- Temporomandibular Dysfunctions [15][16]
- Rheumatoid Arthritis (RA) or Osteoarthritis [17][18][19]
- Breast cancer [20][21]
- Enthesiopathies [22]
- Fibromyalgia [23]
- Diabetes [24]
- Fever Screening [25]

In all fourteen analyzed studies, FLIR software is mostly used for image processing and FLIR cameras are used 71% of the times. Also, a main focus of the studies is about the importance of clinical protocols and standardization of thermal image acquiring technique in order to improve the accuracy in thermography measurements.

2.4. Related work

In 2019, it was conducted an investigation in order to conclude if IRT and temperature measurement along the hand fingers could help increasing the accuracy in diagnosing Rheumatoid Arthritis (RA) patients. The investigation was based on two tests, the first one by doing a cold provocation with 0°C for 5 seconds and a rewarming test with 23°C for 180 seconds.

The conclusion of this investigation was that using the cooling process, allowed to see the contrast between maximum and minimum temperature of the hand fingers (Figure 3, b). Also after the cold stressor the results showed a great potential in differentiating RA patients as it was showed higher temperature values in RA patients (Figure 3, c). The results proved that the thumb, index finger, middle finger, and ring finger maybe useful diagnostically.

The camera used was FLIR E60bx (Systems Inc., USA) and the video and image processing was carried out using MATLAB (MathWorks, Natick, MA, USA). When doing the identification of the region of interest (ROI), they modified the iterative algorithm based on the DFS (Depth-First Search), which allowed to define finger profiles that were presented [19].

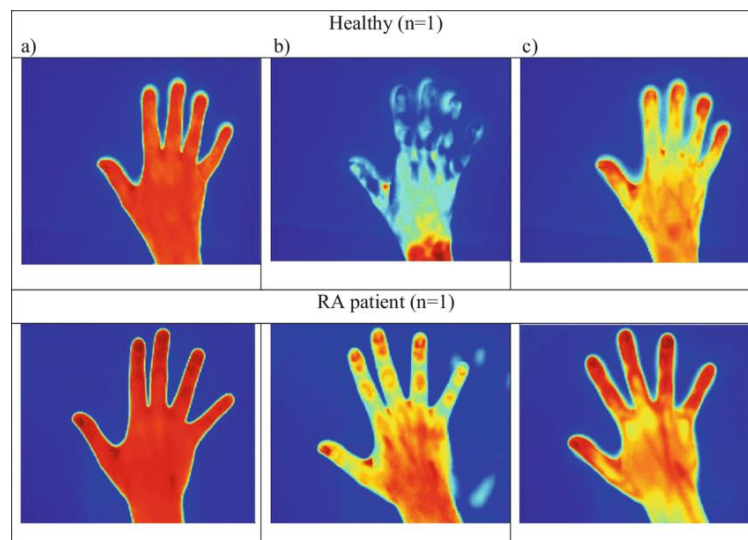


Figure 3 - Thermograms images: a) before cold provocation, b) post-cooling, c) post-rewarming from [26]

2.5. How IRT imaging systems work

A thermography camera is a non-contact device that detects and measures infrared energy (heat) of objects. Also, it converts the infrared data into an image that shows the apparent surface temperature of the object being measured.

In every IR camera there is an optical system that concentrates IR energy onto a special detector chip (sensor array) that contains thousands of detector pixels arranged in a grid, and therefore, each pixel of the sensor array reacts with that IR energy and produces an electronic signal.

Each signal is then taken by the camera processor and it's applied a mathematical calculation in order to generate a color map of the apparent temperature of the object. The resulting matrix has a different color for each temperature. Then the matrix is sent to memory and to the camera's display as a thermal image [27].

There are also IR cameras that have included a visible light camera so that it's possible to blend standard digital images with thermal images in order to be easier to correlate problem areas in IR.

Besides basic thermal imaging capabilities, IR cameras can have other additional features like automate functions, allowance of voice annotations, enhance resolution, record and stream video of the images, and support analysis and reporting.

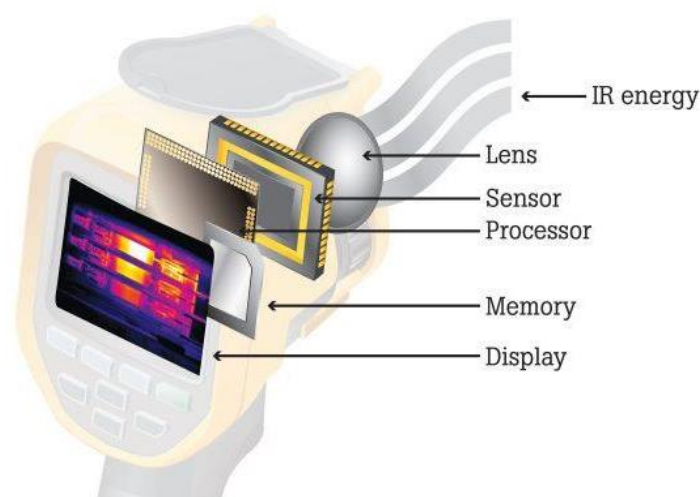


Figure 4 - How a thermal imaging system works from [28]

Below it's showed some important specifications to take in concern in thermography cameras and also in Table 1 it's deepened more specific values of the descriptions [29]:

- **Range:** span of temperatures the camera is calibrated to and capable of measuring.
- **Field of View (FOV):** extent of a scene that the camera will see at any given moment. For close-up work it's advised a lens with an wide angle FOV (45° or higher), and for long distance work, it's advised a lens with a lower angle (12° or 6°).
- **IR Resolution:** how many pixels the camera has on the scene.
- **Thermal Sensitivity:** smallest difference in temperature that the IR sensor is capable of distinguish, as the thermal sensitivity value is lower, the better is the infrared system.

- **Focus:** fixed focus, manual focus or automatic focus.
- **Spectral range:** range of wavelengths that the sensor in the camera detects (μm). Midwave cameras (specially for gas detection), have a spectral range of $3\mu\text{m}$ to $5\mu\text{m}$. Almost all other thermal cameras are longwave, and have a spectral range of $8\mu\text{m}$ to $14\mu\text{m}$.

2.6. Teledyne FLIR

FLIR Systems was established in 1978 and it was a pioneer company in the development of high-performance, low-cost infrared (thermal) imaging systems for airborne applications [30]. Soon it became Teledyne FLIR and it started to leverage its experience in infrared imaging technology to other areas such as transportation, security, industry and health [31].

FLIR cameras are used in a large amount of studies by virtue of high potential detection capacity of FLIR camera models in biomedical applications, medical researchers reach more explanatory and accurate results [32].

FLIR also provides a mobile developer platform where it's possible to have communication with a developer community that gives support. This platform was used throughout the development of the application as it's possible to access numerous software like FLIR Mobile SDK and it's documentation [33].

2.6.1. FLIR Thermography Cameras

FLIR Teledyne has several series of cameras such as:

- FLIR Cx Series
- FLIR Ex Series
- FLIR Exx Series
- FLIR GF Series
- FLIR ONE
- FLIR T Series
- FLIR TG Series

Each series focuses on an specific application, although they can all still be used for the main purpose, assessing temperature. Below is a quick description of the main purposes of each series, along with a table with some specifications, that are explain in Section 2.5, of the main models of each series and the model used in this dissertation, FLIR E60.

Flir Cx Series are pocket-portable cameras and have the advantage of being easily portable due to its size. This cameras are mainly used to find hot fuses, air leaks, plumbing issues, and more [34].

FLIR Ex Series are hand-held cameras and are mainly used for diagnosing electrical, mechanical, building problems and HVAC/R (Heating, Ventilation, Air Conditioning, Refrigeration) applications. It's consider a good solution to confirm repairs easily as it can find hidden problems with high accuracy [34].

FLIR Exx Series are hand-held pistol grip cameras with high thermal resolutions so that the user can diagnose electric and mechanical failures safely. Most of this cameras have lenses with laser distance meter to make sure that a complete coverage of near and distant targets exists [34].

FLIR GF Series are designed for high temperatures scenarios in chemical, petrochemical, and utility industries. Its main focus is on monitoring furnaces, heaters, and boilers with the help of a detachable heat shield to provide protection [34].

FLIR ONE are thermal cameras design for smartphones. It's in a form of an adapter that is connected to the smartphone and uses its camera. It can inspect electrical panels, troubleshoot mechanical systems, look for HVAC problems, or find water damage [34].

FLIR T Series are portable and used to seek out failure signs in a longer time span, both for indoor and outdoor. This series can also diagnose potential faults in industrial, electrical, and mechanical systems [34].

FLIR TG Series are hand-held cameras mainly used for industrial high temperature diagnostic tools, with main usages in measuring the heat of glass furnaces, kilns, and forges allowing to accurately target potential faults, troubleshoot repairs, and monitor processes [34].

To conclude this explanation of all existing series, there were some main factors that led to FLIR E60 to be chosen. First, as the human body was the center of analysis, it was valued the portability and unvalued high temperature ranges as it's more interesting to have a temperature range with smaller values. Having this said, GF, T and TG Series weren't considered good fits for this work. FLIR ONE Series and C Series also weren't considered good fits as most cameras still have big accuracy values which can affect the authenticity of the results.

The cameras in Ex and Exx Series are very similar and in order to break the tie between them, it was taken in concern the specific cameras that were available to use. Nonetheless, FLIR E60 showed better results in terms of spectral range and also good values of accuracy and sensitivity.

Table 1 - Comparison table of different Flir Series

							
	FLIR C5	FLIR E8 XT	FLIR E60	FLIR GF309	FLIR ONE Pro – iOS	FLIR T1020	FLIR TG297
Price (\$)	719.99	3,519.99	4,999.00	107,67 6.55	409.99	41,500.00	999.99
IR resolution	160x120	320x240	320x240	320x2 40	160x120	1024x768	160x120
Sensitivity (°C)	0.07	0.05	0.05	0.015	0.07	<0.02	0.07
Object Temp Range (°C)	-20 to 400	-20 to 550	-20 to 650	-40 to 1500	-20 to 120 and 0 to 400	-40 to 2000	-25 to 1030
Number of color palettes	6	3	6	N/A	8	7	6
Accuracy (range of 0- 100°C)	±3°C	±2°C	±2%	±1°C	±5%	±2°C	±1.5°C
Image Frequency	8.7Hz	9Hz	60Hz	N/A	8.7 Hz	30Hz	8.7Hz
Wi-Fi	Yes	Yes	Yes	No	No	Yes	No
Spectral Range (µm)	8 – 14	7.5 – 13	7.5 – 14	3.8 – 4.05	8 – 14	7.5 – 14	7.5 – 14

2.7. FLIR software for image processing

2.7.1. FLIR Thermal Studio Suite

This software helps users managing thermal images and videos, whether they use handheld cameras, unmanned aircraft systems or optical gas imaging cameras. It also offers advanced processing features

needed for predictive maintenance on critical components, system troubleshooting, and increased productivity.

It's compatible with all cameras that generate images in JPEG format (including A-, B-, C-, K-, T-, E-, GF-, i-, P6- and FLIR One series) and system requirements is Windows 8 or later.

FLIR Thermal Studio software is available in more than 20 languages and it's one of their main programs for producing professional thermal images and reports [35].



Figure 5 - Flir Thermal Studio Suite software from [36]

2.7.2. ResearchIR

FLIR ResearchIR is a thermal measurement software that provides robust camera control, high-speed data recording, real-time or playback image analysis, reporting and data sharing with an intuitive user interface for a variety of research and development applications.

It offers a simplified workflow for displaying, recording, and evaluating data from multiple FLIR cameras simultaneously – allowing you to quickly interpret and understand critical information. FLIR Research Studio also features multi-language (22 languages) and multi-platform support (Windows, MacOS, Linux) to improve collaboration between team members, increase efficiency, and help reduce the potential for misinterpretation due to poor translations [37].

It also provides a valuable feature, it is compatible with MATLAB. It is possible to access MATLAB scripts directly in the software for customized image analysis and processing [38].



Figure 6 - ResearchIR software from [39]

2.8. Mobile applications with thermography usage

When it comes to mobile applications that use thermography as a main feature, it's not that easy to find in the market. Many of the applications that exist are thermography camera simulators that show an image with an effect that mimics how it would be as a thermal image, or apps that need an hardware device to be connected to the smartphone in order to achieve the main goal of the app. For the last situation, there are two applications that belong to two major companies in the thermography world, FLIR Teledyne and Seek Thermal.

Focusing on FLIR's application [40], it's called FLIR Tools and it's available for both Android and iPhone devices. The app allows the user to see live streaming video and capture snapshots from FLIR cameras. With the application, it's possible to import and manage images from the camera by connecting it via USB, Bluetooth or Wi-Fi. It is also possible to edit the images by adding measurement tools. In the following figures there's some examples of the application.

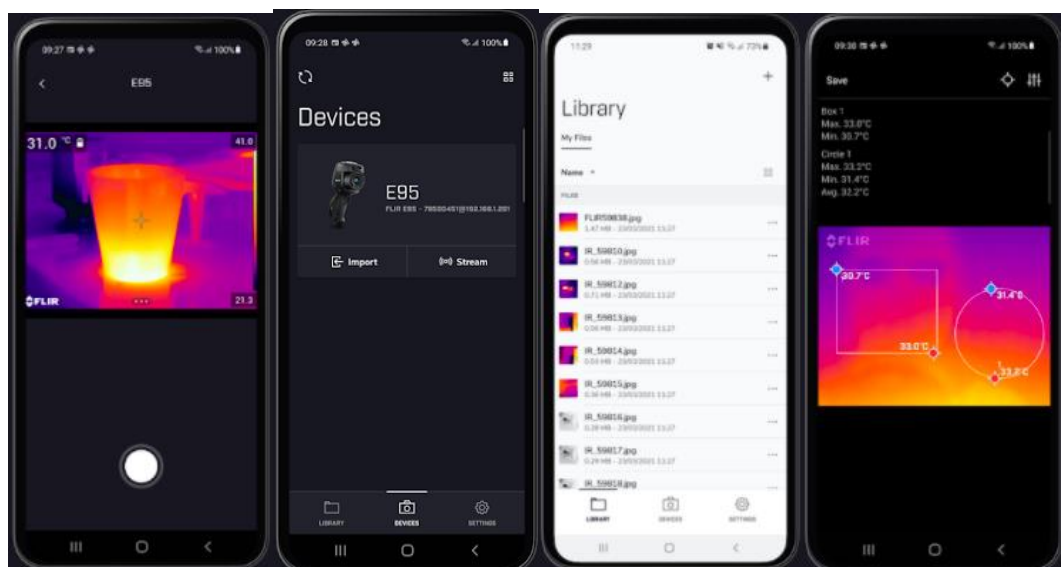


Figure 7 - Flir Tools app, example of a live stream from (1st image), connecting to a thermal camera (2nd image), importing images from storage (3rd image), editing images by adding measurements tools (4th image) from [40]

This application shows some advantages like having the possibility of doing a live streaming of the signal that the camera is detecting and changing some aspects like color palette or temperature range while the streaming is happening. However, the streaming has a poor image quality that can compromise the analysis of it. It was chosen not to use this application due to several reasons.

Taking a perspective of a physiotherapist, if he wanted to use this application to create patients profiles with thermal images associated it was not possible as the feature that allows the user to create profiles with name, sex, age, etc. is not available.

Another reason was in terms of extracting the measurements in a usable format for analysis. It is possible to perform temperature measurements by delimiting a region of interest. However, that result is not associated to any kind of information that can be linked with a patient neither stored in a database or file with JSON, XML, etc format that analyze posteriorly.

With the developed application it will be possible to overcome this disadvantages. It will be possible to create profiles for the patients and associate thermal images to each one of them. Also, the assessment of temperatures will be linked to a specific patient and stored in a database in order to access it easily.

2.9. Data Storage

For this project, there is a need to store multiple information in a database. There were two main options, either to store data on the mobile device or to store it on a cloud service.

For the first option, there are two big disadvantages. The first disadvantage is that a lot of storage of the mobile device will be used and probably will slow it up. The second disadvantage, is that the data will only be available for that specific device. Having this in concern, the smartest option is to store data on a remote location.

To store data on a remote location there were the possibility of using several options like, for example, MySQL, PostgreSQL or Firebase that holds two main databases, Firebase Realtime Database and Cloud Firestore. Below there is a table that compares the main features of these databases.

Table 2 - Comparison table of the different databases based on [41]–[45]

	MySQL	PostgreSQL	Realtime Database	Cloud Firestore
Database Model	RDBMS	ORDBMS	Big JSON tree	Document Oriented
Transaction support	ACID if InnoDB and NDB Cluster Storage engines are used	ACID	Yes	ACID
SQL compliance	SQL only	SQL and NoSQL	Cloud-hosted NoSQL	NoSQL
JSON support	Only JSON, no other NoSQL features	JSON, XML , etc	JSON (JavaScript Object Notation)	JSON
Performance	Good performance in OLAP & OLTP systems when only read speeds are needed	Good performance when executing complex queries	Good performance when real time updates are needed	Good performance when executing complex queries
Best suited projects	Mostly used for web-based projects that straightforward data transactions	Highly used in large systems where read and write speeds are important	Mostly used for mobile application development	Mostly used for web and mobile applications development
Materialized Views	No	Yes	No	No
Main companies using	Airbnb, Uber and Twitter	Netflix, Instagram and Groupon	Twitch, Accenture	Twitch, Accenture

After analyzing the table above, is possible to see that MySQL is used by big companies as it's known to be one of the most secure and reliable database management system, also it requires low hardware resources [46].

However, for this dissertation it was considered a better fit to chose a No SQL compliant database. This choice was made based on how cloud computing is evolving and both Firebase Realtime Database and Cloud Firestore were considered the best option. Both databases are NoSQL and have as main advantages[47]:

- Flexibility
- Scalability
- High- performance
- Functionality
- Ease of development.

In a NoSQL database, the data is stored in JavaScript Object Notation (JSON) or eXtensible Markup Language (XML), which is quite different form relational databases. The JSON format derived from JavaScript, shows a great behavior when there's a need to exchange information between web clients and web servers. Its main advantages are being a schema-less, text-based representation of structured data that is based on key-value pairs and ordered lists [48].

Focusing on the databases chosen, both are non-relational and are incorporated natively into the Firebase architecture powered by Google. Below are explained some of Firebase components [49]:

- **Firebase Cloud Storage**, which is an object stored service that offers security measures in uploads or downloads inside the app. SDK integration is supported for Android, C++, iOS, Unity, and Web applications
- **Firebase Realtime Database** is a NoSQL cloud hosted database which allows to store and synchronize data in real time. The database remains responsive even when offline. Available for mobile and web applications.
- **Cloud Firestore** used as a NoSQL document database, it's possible to store, query, and synchronize the app data. It's also scalable and very flexible service for mobile and web applications.
- **Authentication** it's an SDK that can be used as a backend service to authenticate the users. The authentication can be via password, phone numbers, or through Google, Twitter, and Facebook. It's available for mobile and web applications.
- **Hosting** where it's possible to deploy apps and serve dynamic or static content via a content delivery network (CDN). Only for web applications.
- **Cloud Functions** is used to automate code in backend in response to event triggers. This events can be triggered by HTTPS requests or Firebase features. Supported for Android, C++, Unity, iOS, and web applications.

2.10. Mobile application

In order to complete this project, it was necessary to develop an application that achieves the main goal of having a main user (for example, a physiotherapist) that has several patients in which he/she can visualize their information. It was considered to develop an application specially designed for physiotherapists or professionals that deal with patients doing rehabilitation.

A mobile application was considered the best option for this project as the thermal camera used is portable and a mobile application gives the user more freedom to use it wherever the camera goes. Despite FLIR and other main thermography companies already providing a mobile application, in in Section 2.8. some advantages and disadvantages were exposed comparing to the one being developed.

The operative system chosen for the application was Android, and Android Studio as the Integrated Development Environment (IDE). Android is the most widely used mobile operative system in the world with a percentage of 72.18% in the last 12 months, as seen in Figure 8. In Portugal, there's a percentage of 73.77% in July of 2021 [50]. Another reason is that developing an application in iOS requires for the developer to have an Apple computer able to run XCode which is the IDE for macOS.

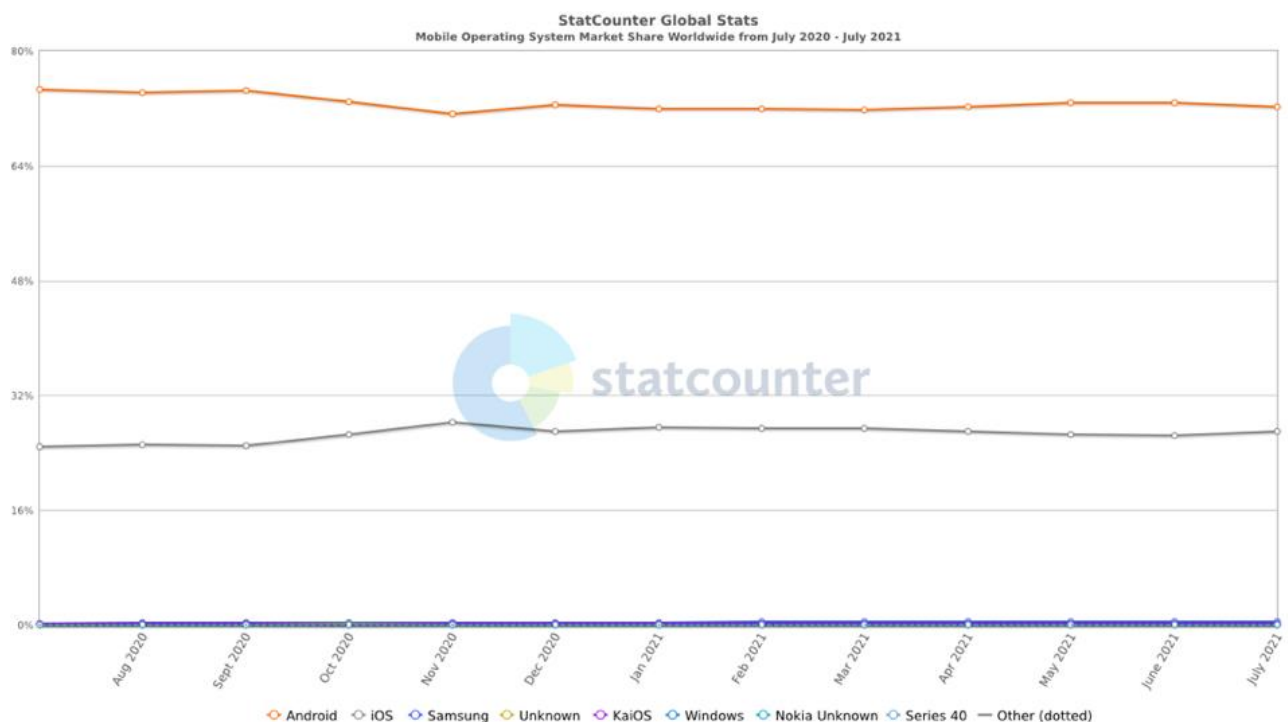


Figure 8 - Percentage of Android users from [50]

In Chapter 3, it's going to be explained in full detail the application architecture as well as all software and hardware used throughout the project.

3. System Description

This chapter describes the architecture and the functionalities of the developed system. More details about the used hardware and developed software are considered.

3.1. Overview

This work aims to develop a thermal image processing system using a thermal camera and different technologies in order to make the collection and the processing of thermographic images in automatic mode.

The figure below illustrates the system architecture which is divided into four main parts.

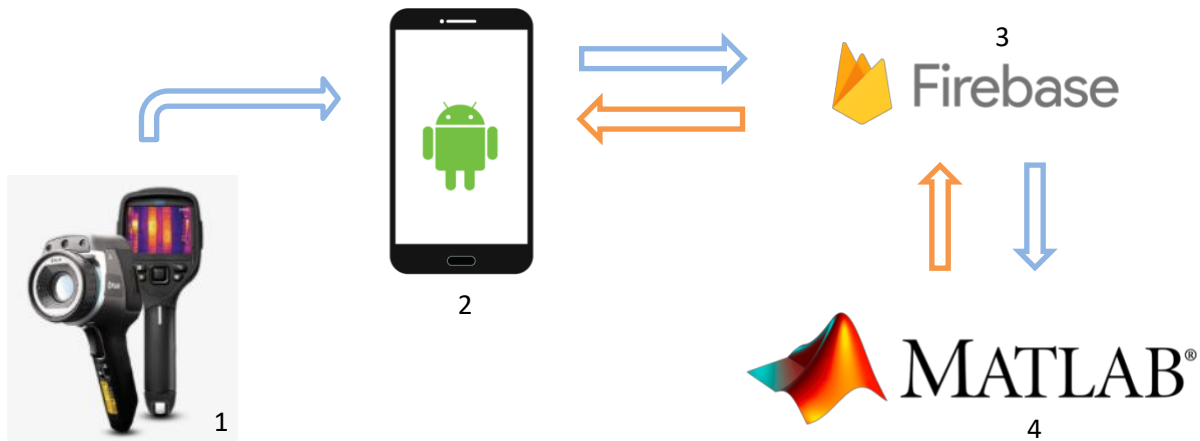


Figure 9 – Communication between thermography camera, mobile application, database and MATLAB

The first part (1) is expressed by FLIR E60 thermography camera. It's where the thermal images are taken, and in this part it's necessary to comply with the correct measurement protocol in order to have quality images and prove the accuracy of the results.

The second part (2) is dedicated to the mobile application, where all the functionalities can be performed. The user can communicate with the camera, visualize the stored images and also create patients and associate the images to them. In this part it's also possible to perform the extraction of the mean temperature of whichever image the user wants.

The third part (3) represents the storage of the thermal images. The chosen technology for this system was Firebase as it is a flexible, scalable and easy-to-use NoSQL database tool.

Finally, in the last part (4), is where the processing of the image happens. It was chosen MATLAB due to the fact that MATLAB offers the Image Acquisition Toolbox that is compatible with FLIR cameras. With this tool is possible to directly capture live video and images into the image processing and computer vision workflow [51].

3.2. System Hardware

3.2.1. FLIR E60

The chosen thermography camera used in this work was FLIR E60, showed in **Error! Reference source not found.** As seen in Table 1 of Subsection 2.6.1, this camera presents more than average value in spectral range, accuracy and sensitivity. It also has an adequate temperature range for skin temperature and its price range is adequate.

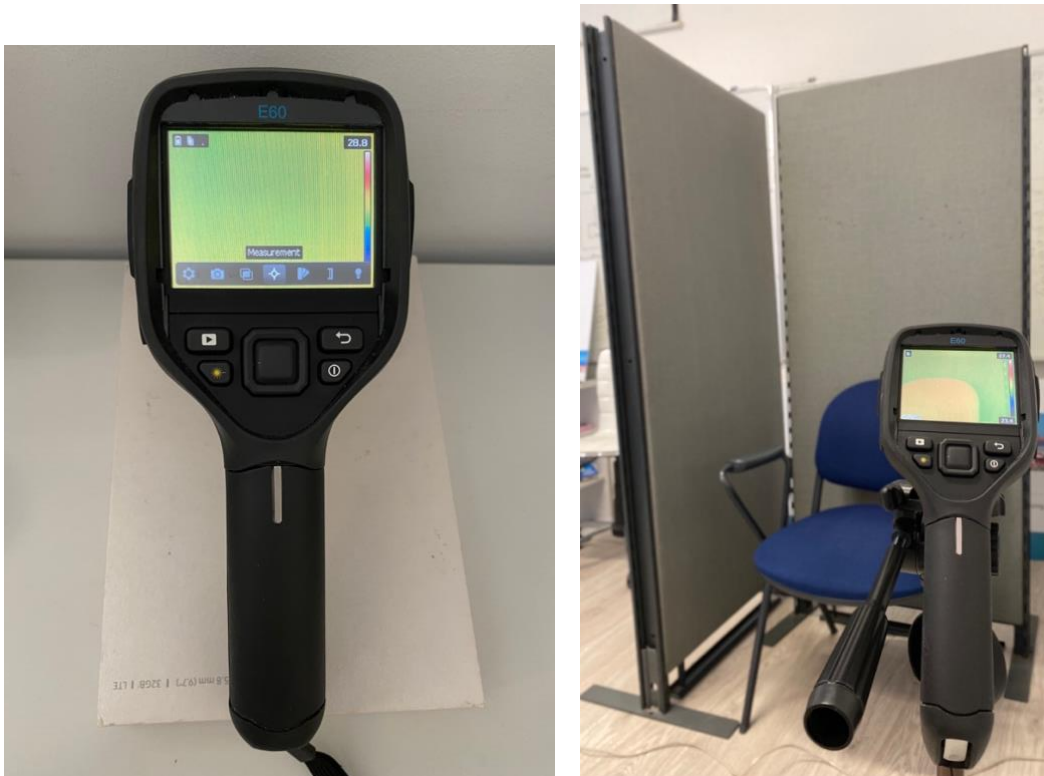


Figure 10 - FLIR E60 thermal camera

3.2.2. Si7021 I2C Humidity and Temperature Sensor

During the experiments, it is important to keep track of room's humidity and temperature. For this purpose, it was used a Si7021 Sensor. This sensor is a monolithic complementary metal oxide semiconductor integrated circuit (CMOS IC) that integrates an humidity sensor with a precision of $\pm 3\%$, and temperature sensor with an accuracy of $\pm 0.4\text{ }^{\circ}\text{C}$.

Its main applications are HVAC/R, thermostats/humidistats, respiratory therapy, indoor weather stations, micro-environments/data centers, and mobile phones and tablets [52].

3.2.3. Android smartphone or tablet

In order to run the application developed it's necessary an Android mobile device. When using the application, it's possible to connect to the thermography camera and check its informations (ex: battery percentage). It's possible to add patients to the database and check the patient's profile where it's possible to visualize their thermal images and personal information. One main device was used to develop and test the application, a Samsung Galaxy Tab S2.

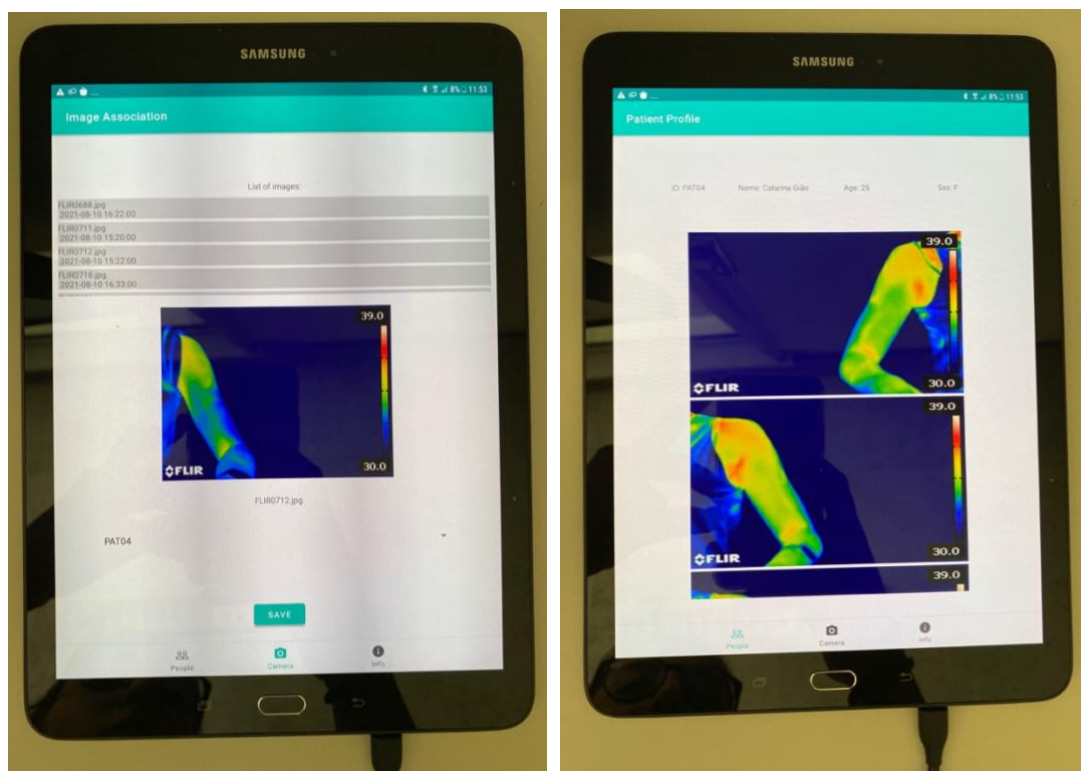


Figure 11 - Developed mobile application running on Samsung Galaxy Tab S2 device

3.3. Mobile Application

The mobile application of this system was developed for Android and aims to help the physiotherapist manage the thermal images of each patient. It is possible to analyze the images directly through the application and extract different metrics. The application was developed in Android Studio IDE as is the official developing tool for Android apps and it's based on IntelliJ IDEA [53].

The chosen version was Android 6.0 (Marshmallow), as highlighted in the Figure 12. This version was chosen because it is quite updated and it has an usage of 84,9%.

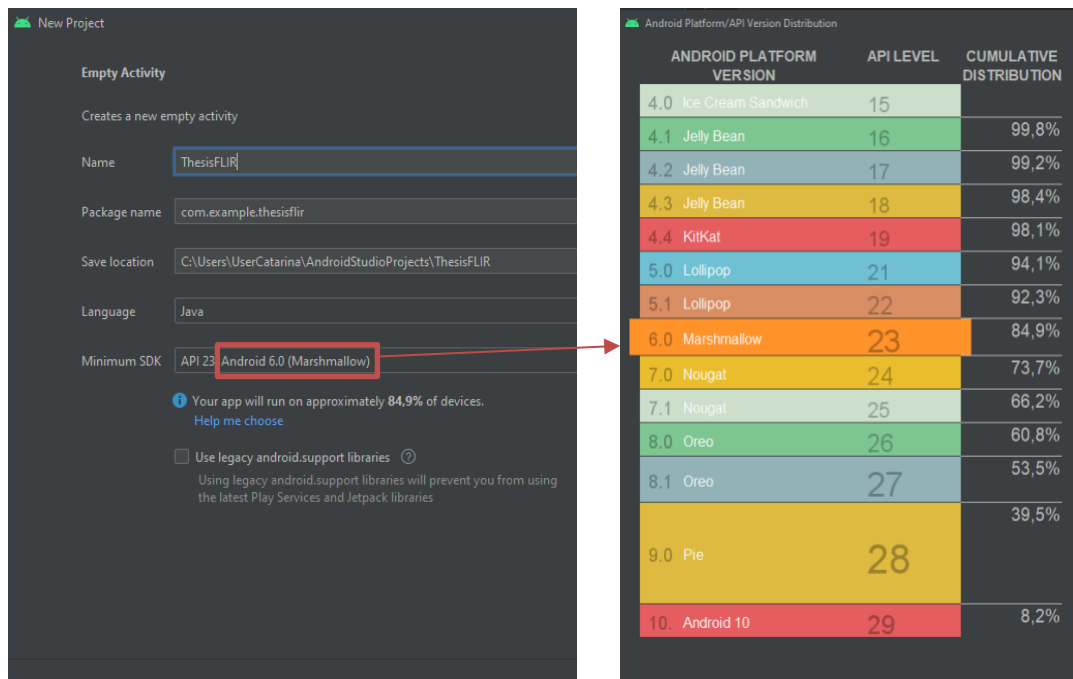


Figure 12 – Choice of Android version for the project (left) Distribution of android version's percentages used (right)

3.3.1. Mobile Application Implementation

The mobile application is composed by 6 fragments and 7 classes that are described in **Error! Reference source not found..** It was decided to use fragments to facilitate the implementation of a bottom bar navigation.

There are some differences between an activity and a fragment. An **activity** is mainly used when there's a single screen of application and it's the main focus of attention on a screen. It also has the ability of hosting multiple fragments at a time.

On the other hand, **fragments** can be seen as sub-activities or pieces of an activity as they are hosted by an activity. They are reusable components that are attached to and displayed within activities. A fragment has its own layout, behavior and lifecycle callbacks which runs in parallel to the lifecycle of the activity.

Table 3 - Classes and Fragments description

AddPatient_Fragment	Allows the user to add a new patient to the database by filling fields of ID, name, sex and age	Fragment
CameraFragment	Allows the user to discover and connect to a thermal camera via wi-fi	Fragment
ImageAssociation Fragment	Allows to associate thermal images to the patients	Fragment
InfoFragment	Information of the camera is displayed (ex: IP address, battery percentage, etc.)	Fragment
PatientProfile Fragment	Allows to see the profile of the patient	Fragment
PeopleFragment	Where the list of patients is displayed	Fragment
CameraHandler	Has all the auxiliary methods related to the camera (ex: start and stop discovery, connect, disconnect, etc.)	Class
PatientsInfo	Represents a patient. It's a simple class with a constructor that ID, name, sex and age as attributes	Class
MainActivity	Class where the layout of the fragments is decided	Class
ListAdapter	Adapter of list of thermal cameras available to connect	Class
ListAdapterPatients	Adapter for list of patients	Class
ListAdapterStorage	Adapter for thermal images on the camera	Class
ImageFirebase Adapter	Adapter to the recycler view where the thermal images are showed	Class

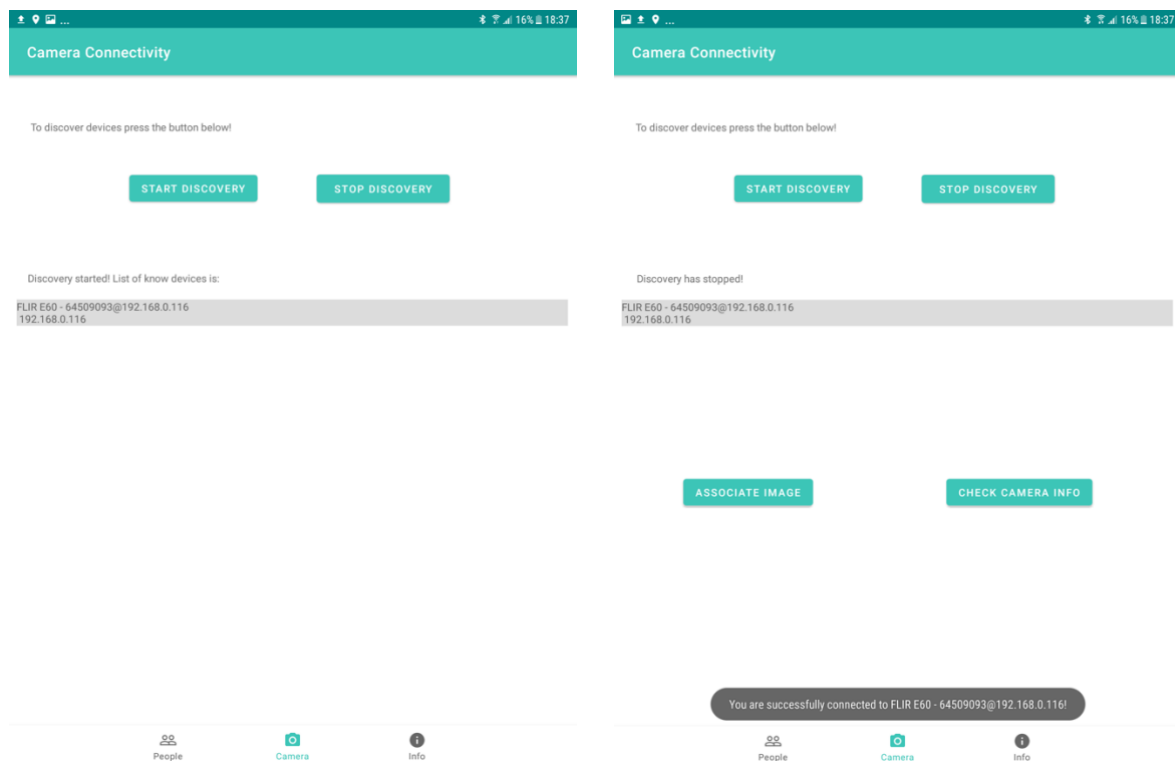


Figure 13 - CameraFragment, discovering camera (left) and connecting to the camera (right)

In Figure 13 is showed the fragment called *CameraFragment*. In this fragment is possible to search for nearby cameras via wireless, see the cameras that are available to connect and perform the connection with the preferred thermography camera. If the connection is successful, two buttons appear, “Associate Image” and “Check camera info”. In this moment, it’s possible to go in two directions:

- Press “Check camera info” and the user is taken to a fragment where it’s possible to visualize information regarding the camera like IP address, battery percentage, etc. (Figure 14)
- Press “Associate Image” and the user is taken to a fragment to associate an image to a patient (Figure 15).

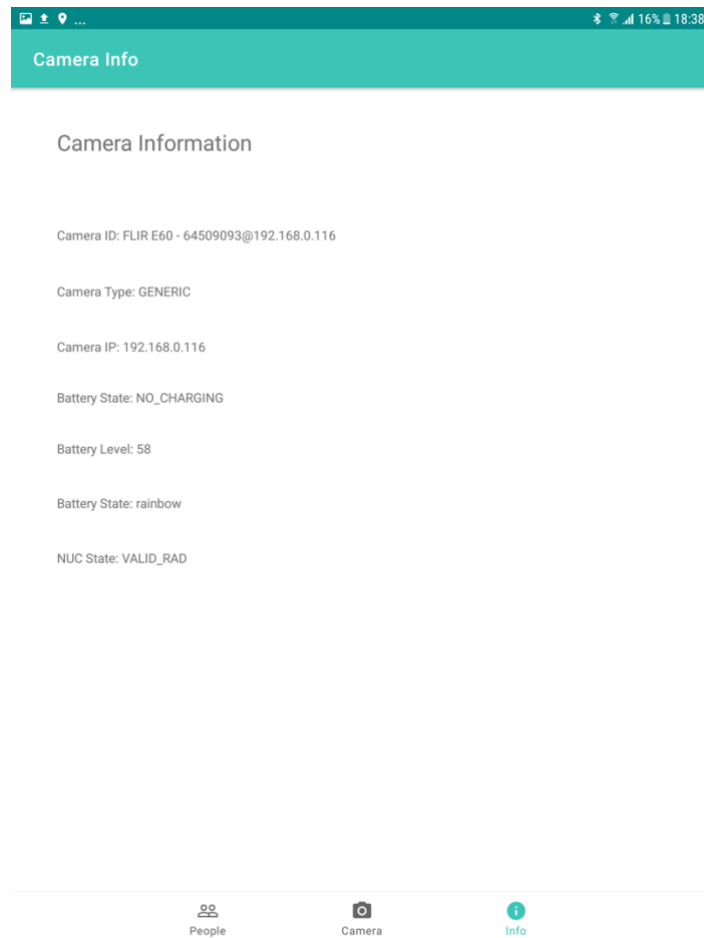


Figure 14 – InfoFragment when pressing the button “Check Camera Info” in Figure 13

As stated before, the Figure 14 shows some information about the camera that is connected. An improvement for this fragment could be to increase the number of information that are displayed.

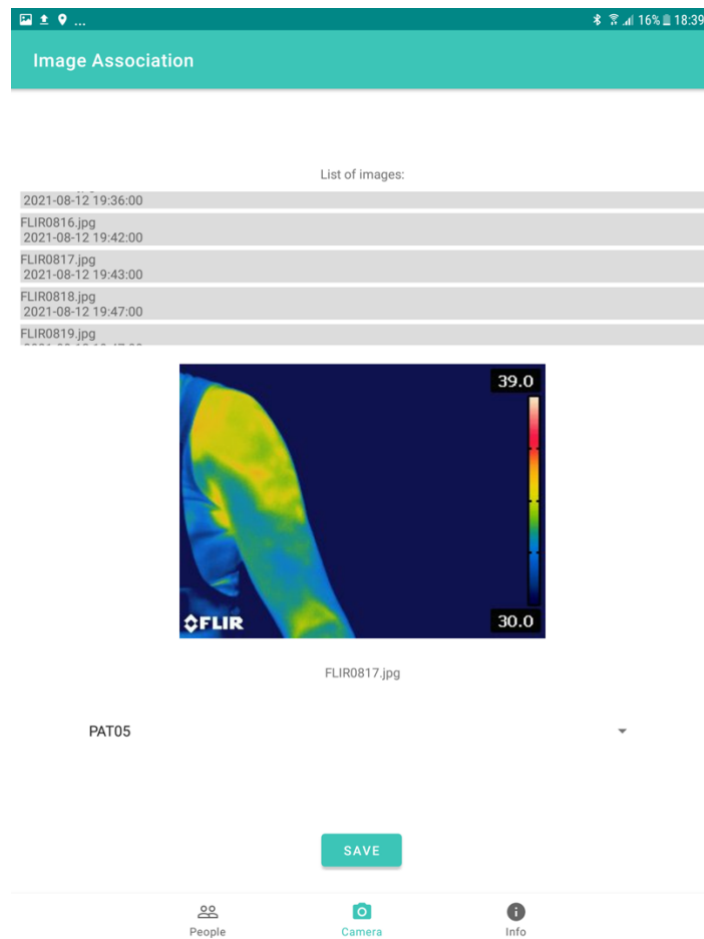


Figure 15 - ImageAssociationFragment when pressing “Associate Image” button in Figure 13

In the Figure 15, it's showed graphical interface that the user sees when he wants to associate an image to an user. A list of available images in the camera appears and if the user presses an image, its pre-visualization appears on the screen. In this list is showed the name of the image and also the date in order for the user to distinguish the images better. After choosing an image, the user needs to select the patient. In the example below, the image “FLIR0817.jpg” will be associate to patient with ID “PAT05”. Practically, this means that in Cloud Storage database there will be the information that this specific image is linked to this specific patient.

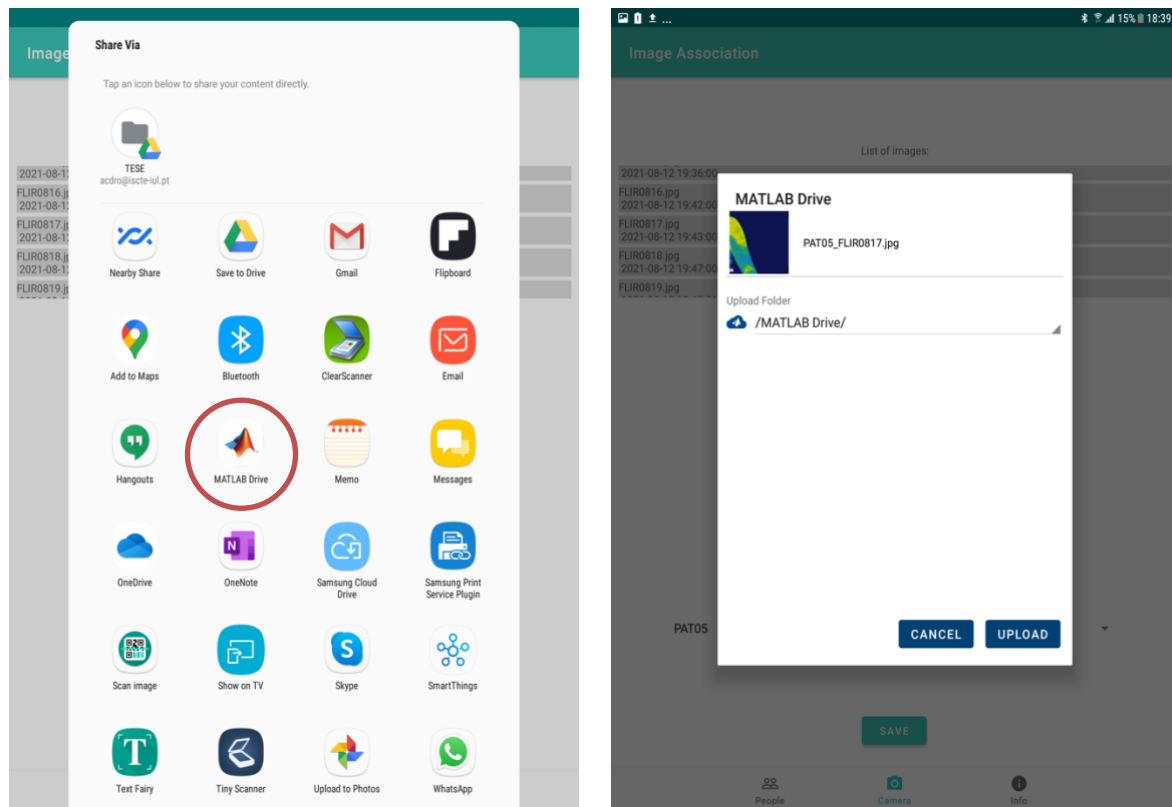


Figure 16 - Sharing option when pressing the thermal image in Figure 15

In Figure 16, it's possible to see how the user can share an image with MATLAB Drive. By pressing the image, the user sees the image on the left and choses the "MATLAB Drive" application. After, the user can either cancel the action by pressing "Cancel" button, or confirm by pressing "Upload" button.

By sharing the image with MATLAB, the image processing software automatically analyzes it.

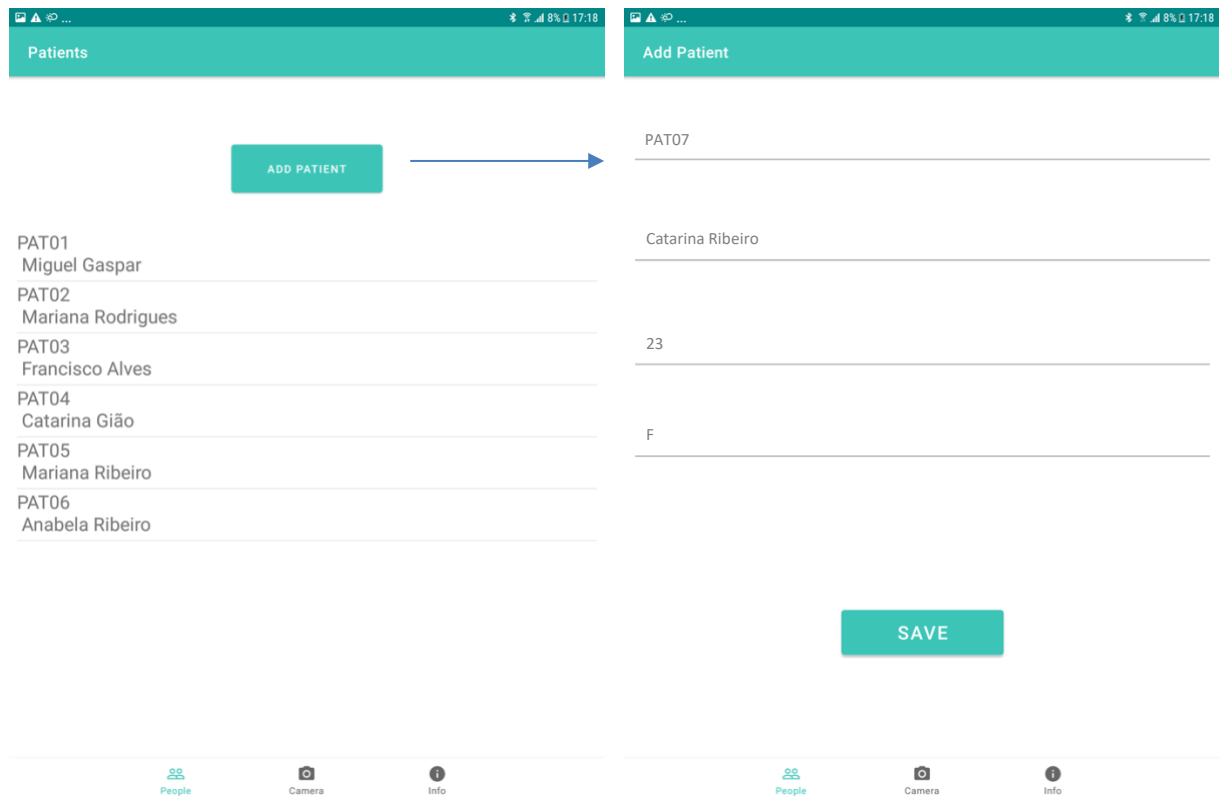


Figure 17 - PeopleFragment, list of all patients (left) and add a patient by pressing “Add Patient” button (right)

In Figure 17, it's showed the fragment “PeopleFragment”, where it's possible to visualize all the patients that the user added to the database. By pressing a patient the user is taken to Figure 18. In this fragment it's also possible to add a new patient, by pressing the button “Add Patient” the user will be redirected to the fragment “Add Patient” where he needs to fill four information about the patient, ID, name, age, and sex.

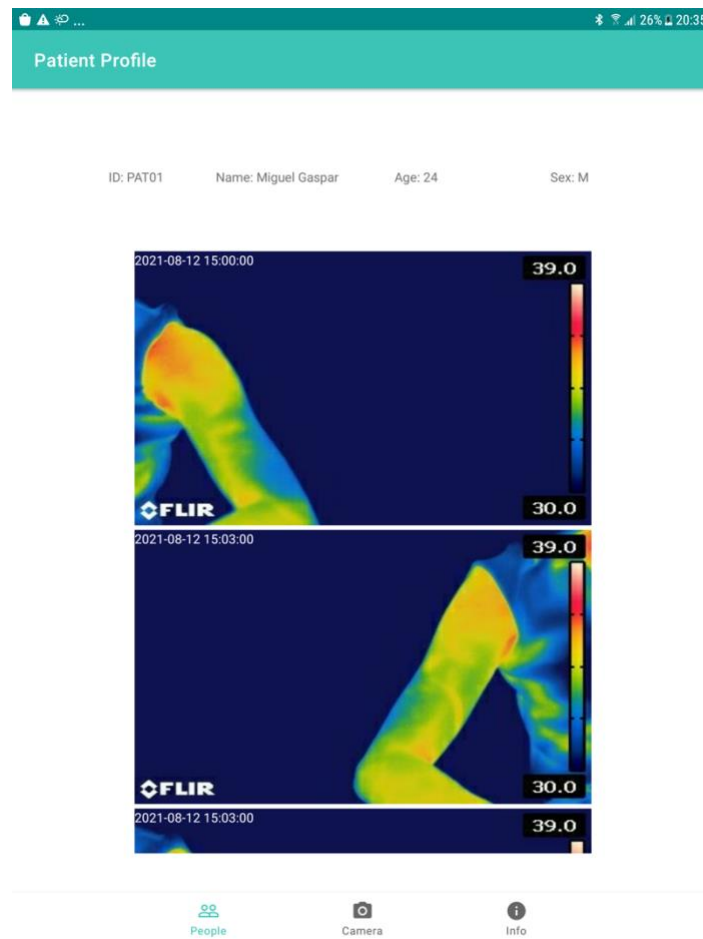


Figure 18 - PatientProfileFragment when pressing a patient in the list in Figure 17

Continuing the logic of the text above, when the user presses the name of a patient it will be showed an interface like the figure above. On the top there's the information given when the patient was added and below all of the images that were associated to this patient. To be noted that the images have in the upper left corner the date of when they were taken. This information facilitates the visualization of the images.

As explained before, every image that's in the database is linked to a specific patient. This allows to organized and group the images by the patient ID. In the Section below it's going to be explain in more detail the structure in Firebase and also how the images are uploaded and downloaded from the database.

3.3.2. Data Structure in Firebase

In order to store all the patient's information (id, name, ex, age) and the thermal images regarding the patients, it was chosen Firebase Realtime Database. In this way, all the information is stored, updated in real time and remains available when the user is not connected to a network [54]. Another reason why this database was chosen is because data is stored as a JSON and we can use any Firebase Realtime

Database URL as a REST endpoint and send requests from an HTTPS client [55]. The advantages of using JSON was explained in Section 2.9 Data Storage. The figure below show an example of how is the structure of a patient in JSON format.

```
{
  "PAT01_FLIR0766" : "https://firebasestorage.googleapis.com/v0/b/testeflir.appspot.com/o/PAT01%2FPAT01_FLIR0766",
  "PAT01_FLIR0767" : "https://firebasestorage.googleapis.com/v0/b/testeflir.appspot.com/o/PAT01%2FPAT01_FLIR0767",
  "PAT01_FLIR0768" : "https://firebasestorage.googleapis.com/v0/b/testeflir.appspot.com/o/PAT01%2FPAT01_FLIR0768",
  "PAT01_FLIR0769" : "https://firebasestorage.googleapis.com/v0/b/testeflir.appspot.com/o/PAT01%2FPAT01_FLIR0769",
  "PAT01_FLIR0770" : "https://firebasestorage.googleapis.com/v0/b/testeflir.appspot.com/o/PAT01%2FPAT01_FLIR0770",
  "PAT01_FLIR0783" : "https://firebasestorage.googleapis.com/v0/b/testeflir.appspot.com/o/PAT01%2FPAT01_FLIR0783",
  "age" : 24,
  "id" : "PAT01",
  "name" : "Miguel Gaspar",
  "sex" : "M",
  "values" : {
    "PAT01_FLIR0767_meanT" : 34.09643836805252,
    "PAT01_FLIR0768_meanT" : 33.97500086805502,
    "PAT01_FLIR0771_meanT" : 34.08078211805608,
    "PAT01_FLIR0772_meanT" : 33.91407682291826,
    "PAT01_FLIR0775_meanT" : 33.96193229166512,
    "PAT01_FLIR0776_meanT" : 33.97800130208307,
    "PAT01_FLIR0779_meanT" : 34.16620269097127,
    "PAT01_FLIR0780_meanT" : 33.8793298611125,
    "PAT01_FLIR0783_meanT" : 34.23523958333098
  }
}
```

Figure 19 - Structure of a patient in JSON format

3.3.2.1. Structure in Firebase Realtime Database

Firebase Realtime Database is dynamically structured as it is organized by nodes. It is possible to have multiple nodes and each node can have multiple children. In this project, it was decided to have two main nodes. In the figure below we can see two nodes, “images” and “patients”.

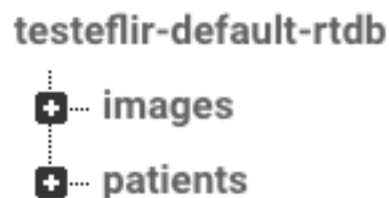


Figure 20 - Main nodes in Firebase Realtime Database

The first node “images” has the names of all images that has been stored in the database. This node was created to help in MATLAB development. It works as an “auxiliary list” to perform a comparison of names between the ones that are already in the database and the ones that are in MATLAB Drive. In Figure 21 it’s possible to see the structure of the children inside the node. The name is structured by the ID of the patient first and then the name of the image.

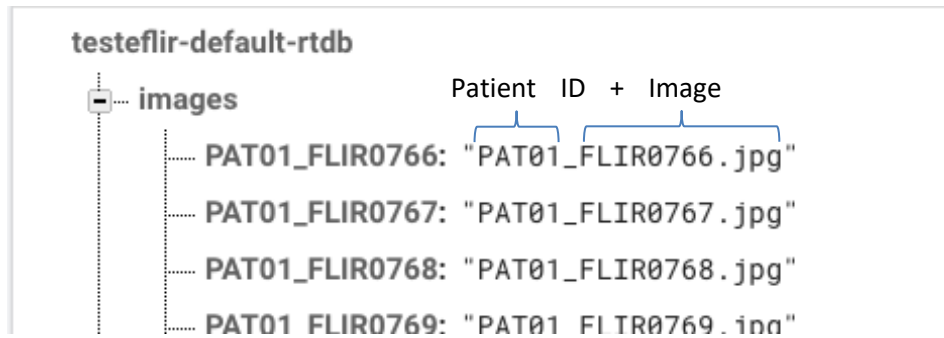


Figure 21 - Structure of node "images" in Firebase Realtime Database

The second node "patients", has multiple nodes that represents the patients. As it is possible to see in the Figure 22, when opening the node PAT01, it appears several the information like the id of the patient, name, sex and all of the thermal images associated to the patient. It also has two more nodes inside, the node "values", with all the metrics extracted from the thermal images and the node "imageDates" with all of the dates of the thermal images.



Figure 22 - Structure of node "patients" in Firebase Realtime Database

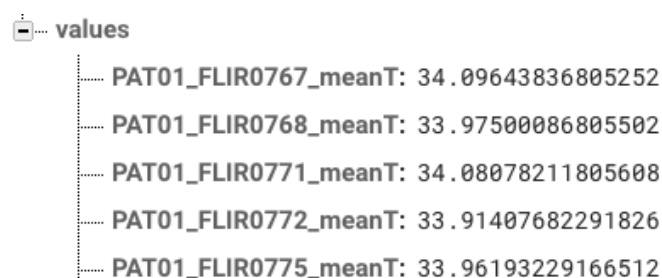


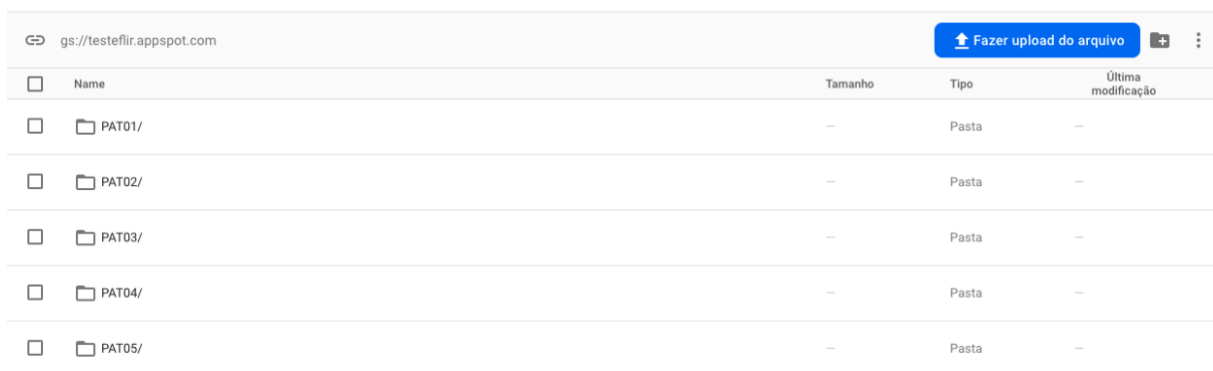
Figure 23 - Structure of node "values" in Firebase Realtime Database

In Figure 22, focusing on node “PAT01”, in child “PAT01_FLIR075”. Its value, is a link that references an image stored in Firebase Cloud Storage.

In Figure 23, it is possible to see the node “values” that is inside the main node “PAT01”. In this case, this node receives the metrics when an thermal image from patient 1 is analyzed in MATLAB.

3.3.2.2. Structure in Firebase Cloud Storage

It was decided to create a folder for each patient in order to organize the images in a better way. In Figure 24 it’s possible to see that organization.



The screenshot shows the Google Cloud Storage interface for the bucket 'gs://testeflir.appspot.com'. At the top right, there is a blue button labeled 'Fazer upload do arquivo' and a menu icon. Below the header, a table lists five folders: PAT01/, PAT02/, PAT03/, PAT04/, and PAT05/. Each row includes a checkbox, the folder name, its size (indicated by a dash), its type (Pasta), and its last modification date (indicated by a dash).

☐	Name	Tamanho	Tipo	Última modificação
☐	PAT01/	—	Pasta	—
☐	PAT02/	—	Pasta	—
☐	PAT03/	—	Pasta	—
☐	PAT04/	—	Pasta	—
☐	PAT05/	—	Pasta	—

Figure 24 - Structure of all patients in Cloud Storage

When opening a folder, it contains all of the images that were associated so far to the patient. Also, due to the format of the images, they are automatically organized by date as the image name “FLIR0xxx” is increasing by one each time. Which means as the number increases, the time that the image was taken also increased. Such organization and structure is possible to see in the figure below.

☐	Name	Tamanho	Tipo	Última modificação
☐	PAT01_FLIR0766.jpg	17.96 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0767.jpg	21.56 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0768.jpg	20.77 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0769.jpg	17.93 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0770.jpg	17.27 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0771.jpg	18.71 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0772.jpg	22.09 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0773.jpg	16.82 KB	image/jpeg	11 de ago. de 2021
☐	PAT01_FLIR0774.jpg	19 KB	image/jpeg	11 de ago. de 2021

PAT01_FLIR0766.jpg	
Nome	PAT01_FLIR0766.jpg
Tamanho	18.395 bytes
Tipo	image/jpeg
Criada	11 de ago. de 2021 16:20:19
Atualizada	11 de ago. de 2021 16:20:19
Local do arquivo	▼
Outros metadados	▼

Figure 25 - Structure of a patient in Cloud Storage

3.3.3. Communication with Firebase

In order exchange of information in Firebase, there are used two different methods. When it comes to the application side and it's needed either to read or write information, it is used Firebase SDK and, on MATLAB side, it is used a REST API method.

3.3.3.1. Firebase SDK

To integrate Firebase in Android it was necessary to install Firebase SDK for both Firebase Realtime Database and for Cloud Storage. The following dependencies were used.

```
//Cloud Firestore
implementation 'com.google.firebase:firebase-firestore:23.0.3'
//Realtime Database
implementation 'com.google.firebase:firebase-database:20.0.1'
```

Figure 26 - Firebase dependencies

When the physiotherapist wants to add a new patient to the database it's necessary to fill four information, ID of the patient, name, sex and age. It was created a custom object called *PatientsInfo* with this four attributes in order to facilitate writing and reading of a patient in the database. Figure 27 shows the code of PatientsInfo class, that is composed by a constructor with four attributtes.

```

package com.example.flirnavigation;

public class PatientsInfo {
    public int age;
    public String id;
    public String name;
    public String sex;

    public PatientsInfo(){}

    public PatientsInfo(int age, String id, String name, String sex) {
        this.age = age;
        this.id = id;
        this.name = name;
        this.sex = sex;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSex() {
        return sex;
    }

    public void setSex(String sex) {
        this.sex = sex;
    }

    @Override
    public String toString() {
        return id;
    }
}

```

Figure 27 - Implementation of PatientsInfo class

- Writing/Reading in Firebase Realtime Database

In Figure 28, it's possible to see the code implemented in the fragment dedicated to add a patient a consequently also adding it to the database. The figure is also highlighted with numbers that represent the order explain in the next paragraph.

When the user wants to add a new patient, it's asked to fill the fields of the patient ID, name, age and sex (1). With this four attributes it's created a *PatientsInfo* object (2) that is added to a local array list in order to have an history of patients (3). After, the object PatientsInfo is added to a specific path in the database with the help of *setValue* function that already exists in Firebase SDK (4). The correct path when adding a new patient is set by taken in concern the data structure of the database (Subsection 3.3.2). The patient ID was used as a main node in order to create an unique entry.

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_add_patient_, container, false);
    editTextUserID = v.findViewById(R.id.editTextUserID);
    editTextName = v.findViewById(R.id.editTextName);
    editTextAge = v.findViewById(R.id.editTextAge);
    editTextSex = v.findViewById(R.id.editTextSex);
    saveButton = v.findViewById(R.id.saveButton);

    // Write a message to the database
    FirebaseDatabase mDatabase = FirebaseDatabase.getInstance();
    DatabaseReference mReferenceDB = mDatabase.getReference();

    saveButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String patientID = editTextUserID.getText().toString();
            String patientName = editTextName.getText().toString();
            int patientAge = Integer.parseInt(editTextAge.getText().toString());
            String patientSex = editTextSex.getText().toString();

            PatientsInfo newPatient = new PatientsInfo(patientAge, patientID, patientName, patientSex);
            listOfPatients.add(newPatient);
            Log.e("TAG", "new patient is: " + newPatient.toString());
            mReferenceDB.child("patients").child(patientID).setValue(newPatient);

            Toast.makeText(getActivity().getApplicationContext(), "Patient was added successfully!", Toast.LENGTH_SHORT).show();
        }
    });

    return v;
}

```

(1) points to the findViewById calls for the form fields.

(2) points to the saveButton.setOnClickListener call.

(3) points to the mReferenceDB.child("patients").child(patientID).setValue(newPatient) line.

(4) points to the return v; line.

Figure 28 - Implementation of writing in Firebase Realtime Database

In order to read the information from the database to show in the application is used the function `getValue` from Firebase SDK. Figure 29 shows how this function was used and is highlighted with numbers that represent the order explain in the next paragraph. It was used a foreach loop to go through all of the patients in the database (1), then add them to a local list (2) and put the list in a recycler view for visualization purposes (3).

```

FirebaseDatabase mDatabase = FirebaseDatabase.getInstance();
DatabaseReference mReferencePatient = mDatabase.getReference("patients");

mReferencePatient.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull @NotNull DataSnapshot snapshot) {
        patientsInfoList.clear();
        for (DataSnapshot keyNode : snapshot.getChildren()) {
            PatientsInfo patientsInfo = keyNode.getValue(PatientsInfo.class);
            patientsInfoList.add(patientsInfo);
            Log.e("TAG", "patients list is" + patientsInfoList.toString());
        }
        viewAdapterPatients = new ListAdapterPatients(patientsInfoList, UIAdapterClickListener);
        recyclerViewPatients.setAdapter(viewAdapterPatients);
    }
    @Override
    public void onCancelled(@NonNull @NotNull DatabaseError error) {
    }
});

```

(1) points to the for loop that iterates through the database snapshot.

(2) points to the recyclerViewPatients.setAdapter(viewAdapterPatients) line.

(3) points to the viewAdapterPatients = new ListAdapterPatients(patientsInfoList, UIAdapterClickListener) line.

Figure 29 - Implementation of reading in Firebase Realtime Database

- Writing/Reading in Firebase Cloud Storage

When the physiotherapist wants to associate a thermal image to a patient, there is data added in to two different places. It is added the image directly to the Cloud Storage and it is added a new entry in the patient with an URI to the image in the Firebase Realtime Database.

The method *putBytes* existing in Firebase SDK is used for Cloud Storage and again, in the Firebase Realtime Database, is used *setValue*. This situation happens like described before in Figure 28.

```
associateButton.setOnClickListener(v12 -> {
    try {
        // Get the data from an ImageView as bytes
        lastStoredImageView.setDrawingCacheEnabled(true);
        lastStoredImageView.buildDrawingCache();
        Bitmap bitmap = ((BitmapDrawable) lastStoredImageView.getDrawable()).getBitmap();
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        byte[] data = baos.toByteArray();

        UploadTask uploadTask = imageRef.putBytes(data);
        uploadTask.addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception exception) {
                // Handle unsuccessful uploads
            }
        }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                // taskSnapshot.getMetadata() contains file metadata such as size, content-type, etc.
                showMessage.show(nameOfFile + " associated successfully to " + stringSpinner);

                Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
                    @Override
                    public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) {
                        if (!task.isSuccessful()) {
                            Log.i("problem", task.getException().toString());
                        }
                        return imageRef.getDownloadUrl();
                    }
                }).addOnCompleteListener(new OnCompleteListener<Uri>() {
                    @Override
                    public void onComplete(@NonNull Task<Uri> task) {
                        if (task.isSuccessful()) {
                            Uri downloadUri = task.getResult();

                            mReferencePatient.child(stringSpinner).child(stringSpinner + "_" + nameOfFile.substring(0,8)).setValue(downloadUri.toString());
                            mReferenceImages.child(stringSpinner + "_" + nameOfFile.substring(0,8)).setValue(stringSpinner + "_" + nameOfFile.substring(0,8) + ".jpg");
                        } else {
                            Log.e("wentWrong", "downloadUri failure");
                        }
                    }
                })
            }
        })
    }
});
```

Cloud Storage
image upload

New entry of URI

Figure 30 - Implementation of writing in Cloud Storage

When there's a need to show the images from the Cloud Storage, instead of accessing the Cloud Storage, it was chosen to get the URI in the Firebase Realtime Database as it's faster to access the URI's of the images instead of the images themselves. It was also used the Picasso library to show the images. First, it is created a local list of all the URI that are associated to the patient and then with the method *load*, the images are placed in an image view.

```
// Here "image" is the child node value we are getting
// child node data in the getImage variable
DatabaseReference getImage = databaseReference.child("patients/" + patientID);
listOfImageLinks.clear();

getImage.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        // getting a DataSnapshot for the location at the specified
        // relative path and getting in the link variable

        for (DataSnapshot keyNode : dataSnapshot.getChildren()) {
            try {
                String link = keyNode.getValue(String.class);
                listOfImageLinks.add(link);
            } catch (DatabaseException e) {
                Log.e("TAG", "list of links is " + listOfImageLinks );
                mAdapter = new ImageFirebaseAdapter(listOfImageLinks, getContext());
                mRecyclerView.setAdapter(mAdapter);
                if (listOfImageLinks.isEmpty()) {
                    Toast.makeText(getContext(), "User doesn't have thermal images to show!", Toast.LENGTH_SHORT).show();
                }
                return;
            }
        }
    }
});
```

Figure 31 - Implementation of reading in Cloud Storage

```

@Override
public void onBindViewHolder(@NonNull @NotNull ImageFirebaseAdapter.MyViewHolder holder, int position) {
    String uploadCurrent = dataset.get(position);
    Picasso.with(mContext)
        .load(uploadCurrent)
        .placeholder(R.drawable.ic_processimage)
        .fit()
        .into(holder.imageView);
}

```

Figure 32 - Implementation of image upload with Picasso library

3.3.3.2. REST API

In order to perform communications between Firebase and MATLAB it was necessary to use Firebase Realtime Database URL as a REST endpoint. With this service, it was possible to exchange data between the two sides easily as it exists already some defined HTTPS requests such as GET, PUT, POST, PATCH and DELETE.

To save data successfully in the database it is possible to use the following requests:

- PUT which writes data, if it doesn't exist, and replaces data if it already exists;
- POST which is used to write lists of data, it's generated a unique key every time a new child is added;
- PATCH which updates a specific children at a location without overwriting existing data.

Every time it's necessary to write data from MATLAB it is used PATCH requests as the purpose is to add data and not replace it.

```

options = weboptions("RequestMethod", "patch");
webwrite(url_toWrite, meanTempData, options);

```

Figure 33 - PATCH request in MATLAB

3.4. MATLAB Processing

3.4.1. Overview

In order to process thermal images it was developed a MATLAB script that returns a specific metric, in this case, it's the average temperature of the image. In the figure below (Figure 34) is possible to understand how the triggering of the image processing is made through the mobile application and how the connection between MATLAB and the database is processed.

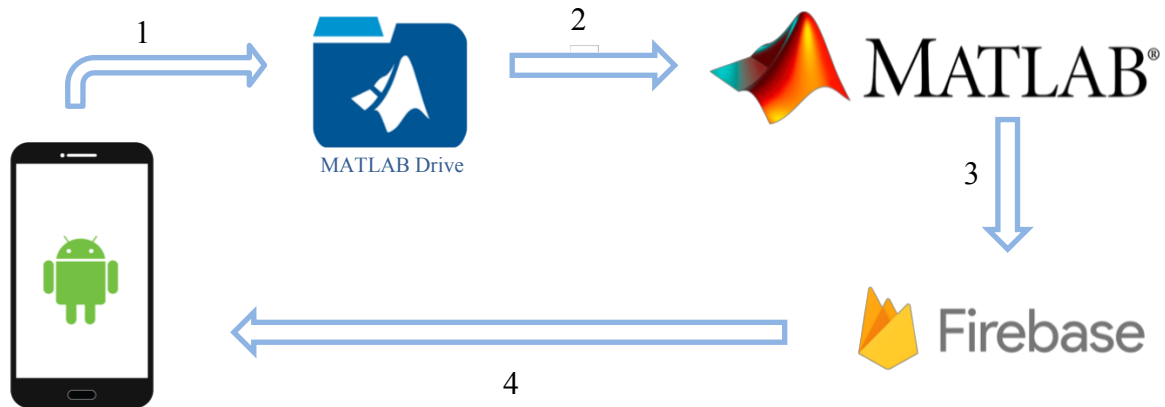


Figure 34 - Overview of MATLAB and Firebase interaction

In order to analyze the image in MATLAB it is necessary for the physiotherapist to share the image with MATLAB Drive (1). This drive provides a common cloud-based storage location for MATLAB files that can be access through the desktop.

The second arrow (2), represents the script that is running in background on a desktop and it is triggered as new images arrive.

In the third arrow (3), the data is sent to Firebase like it was explained in Subsection 3.3.3.2. After this step, it's possible to visualize the results directly in the application.

3.4.2. MATLAB Script Implementation

In this subsection it will be explained the main parts of the implementation of the MATLAB script and the explanation of it. The purpose of this script is to be constantly running on background using the computer as a local server.

In Figure 35, it's possible to see how the images are read from the database and how the preparation of data is made. The preparation of data is centered in three steps:

- How the list of images is accessed from the database
- Comparison between that list and the image to analyze in order to see if that image actually exists
- If it exists, it means that it will be analyzed and the its name is putted in a correct format for later the metric analyzed be putted in the correct path in the database

First, it was used the function *webread* from MATLAB that allows to read from the database the list of existing images (1). As seen in Figure 20, the node “images” is read in this step.

After, on line 19, the structure of the list of images was transform to a cell with function *struct2cell*, in order to facilitate its usage (2) and on line 20, it was saved in a variable the length of that list (3) .

The logic of the loop on line 25 is to compare every element in the list “listimages” by incrementing the value of “x”(4). This comparison will be done on line 29, with function *isfile* from MATLAB. This function is checking if it exist an image locally with the same name in the list “listimages” (5).

It’s important to remember that, as seen in Figure 21, the format of the elements inside node “images” is “PATxx_FLIRxxx.jpg”. If this statement inside the loop is true, there are made the following changes in the name of the image:

- Removal of the “.jpg” by using the *split* function of MATLAB (6)
- Extraction of the patientID by using the *split* function to get “PAT0x” (7)
- Appending “/” to patientID, the result will be “PAT0x/”
- Appending “PAT0x/” to the link of node “patients” in Firebase Realtime Database (9)
- Appending “.json” to the result in the line above to get the final result (10)

Finally, on line 53, it’s used again *webread* function to read the link obtained in step (10). The purpose of this step is to obtain link of the image in JSON format.

```

16 %Reading images list from firebase
17 listimagesStruct = webread('https://testeflir-default-rtdb.firebaseio.com/images.json');
18 %converting from struct to cell
19 listimages = struct2cell(listimagesStruct);
20 cellSize = length(listimages);
21
22 x = 1;
23 n = 2;
24 while n > 1
25     while x <= cellSize
26         pat = listimages(x);
27         patString = string(pat);
28         %check if the image exists locally
29         if isfile(patString)
30             disp('file exists')
31             imageNameSplittedByDot = split(patString, ".");
32             xName = imageNameSplittedByDot(1);
33             %get the name of the image without ".jpg"
34             imageName = string(xName);
35
36             %imageName format is PATxx_FLIRxxx
37             imageNameSplitted = split(imageName, "_");
38             xPos1 = imageNameSplitted(1);
39             %extracting the patient ID
40             patientId = string(xPos1);
41
42             %correct patient with a slash for the url
43             correctPatient = append(patientId, '/');
44             %link from firebase of patients
45             url_database = 'https://testeflir-default-rtdb.firebaseio.com/patients/';
46             url_withcorrectpatient = append(url_database, correctPatient);
47
48             % url to get the correct image from firebase
49             url_withcorrectImage = append(url_withcorrectpatient, imageName);
50             url_final = append(url_withcorrectImage, '.json');
51
52             %link to read
53             image_url = webread(url_final);
54             image = imread(image_url);
55             nameOfImageToProcess = append(imageName, '.jpg');
56             baseFileName = nameOfImageToProcess;

```

(1)

(2)

(3)

Makes the script run in a loop

(4)

Conversion to string

(5)

(6)

(7)

(8)

(9)

(10)

Figure 35 - Implementation of how list of images is taken from database e putted in the correct format to analyze

After the preparation of the data, it's made the analysis. In the figures below, it's showed the main aspects of the process of analysis. On line 82 it's decided to crop the color bar from the thermal image in order to have the original scale to attribute the temperatures to the colors (1). After, on line 121 (2), the color bar is modified to have values between 0 and 1. Finally, on line 139 and 141, it's defined the scale of 33°C to 39°C in order to ignore the temperatures outside of this interval (3) and subsequently, on line 144, the temperatures are re-scaled to the chosen values (4).

```

81      % Crop off the surrounding clutter to get the colorbar.
82      colorBarImage = imcrop(originalRGBImage, [307, 30, 8, 180]);
83      b = colorBarImage(:, :, 3);
120     % Get the color map.
121     storedColorMap = colorBarImage(:, 1, :);
122     % Need to call squeeze to get it from a 3D matrix to a 2-D matrix.
123     % Also need to divide by 255 since colormap values must be between 0 and 1.
124     storedColorMap = double(squeeze(storedColorMap)) / 255;
125     % Need to flip up/down because the low rows are the high temperatures, not the low temperatures.
126     storedColorMap = flipud(storedColorMap);
138     % Highest temperature to take in concern
139     highTemp = 39;
140     % Lowest temperature to take in concern
141     lowTemp = 33;
142
143     % Scale the image so that it's actual temperatures
144     thermalImage = lowTemp + (highTemp - lowTemp) * mat2gray(indexedImage);

```

(1) points to line 82
(2) points to line 124
(3) points to lines 138-141
(4) points to line 144

Figure 36 - Implementation of the main aspects of the image analysis

Lastly, it's calculated the mean of temperatures with function *mean2* (1). As said before, the metrics extracted from the images go to a special node in the database called "values" (2). Also, as explain in Subsection 3.3.3.2 REST API, it's made a PATCH request to write the metrics in the database using *weboptions* and *webwrite* functions from MATLAB (3).

```

167     % mean temperature
168     meanTemp = mean2(thermalImage);
169     % transform again the format to access the right DB url
170     url_patientNoSlash = append(url_database, patientId);
171     url_patientValues = append(url_patientNoSlash, '/values');
172     url_toWrite = append(url_patientValues, '.json');
173     % write in the database
174     nameforDBMean = append(imageName, '_meanT');
175     meanTempData = struct(nameforDBMean, meanTemp);
176     options = weboptions("RequestMethod", "patch");
177     webwrite(url_toWrite, meanTempData, options);
178     disp(meanTemp);
179     else
180         %File does not exist.
181         disp('file does not exist')
182     end
183     x=x+1;
184 end
185 x=1;
186 end

```

(1) points to line 168
(2) points to line 172
(3) points to lines 176-177

Figure 37 - Implementation of mean temperature and writing on database

CHAPTER 4

4. Experimental Description

In this chapter is explained how the tests were conducted and the experimental results obtained from the developed system.

4.1. Experimental Work

4.1.1. Tailored VR Game

In order to validate the developed system, it was used an already developed serious game that combines two technologies, remote sensing and Virtual Reality with the main application in physiotherapy.

The developed system focus on physical rehabilitation for upper limb rehabilitation, where the user and Virtual Reality scenario is based on Microsoft Kinect. There is also a mobile application that allows for the physiotherapist to setup training plans and visualize the results of the exercise sessions. The system was developed primarily to allow tailoring of the exercises according to the patient's preferences and needs. It was also designed to present metrics that characterize the patient's results.

It's possible to customize each training plan with the type of training (i.e., if exercise will focus on only one of the arms or it will allow both arms to be used), the level of difficulty (i.e., lower angles or higher angles environment), the speed of movement (there are three modes: slow, medium, fast), the quantity of visual elements in the environment (i.e., no elements except the orchard or other elements with animations and sounds with the context of a farm); the duration of the exercise and the minimum points that should be used as a conclusion objective.

Each game session has as main objective to catch fruits associated with angles that might be reached by left hand, right hand or both hands during the game based training session. In Figure 38 it's possible to see the case "Low angles" setting considered by physiotherapist the fruits to harvest are raspberries (right image) while for "High angles" the fruits to harvest are apples (left image) [56].



Figure 38 - Volunteer performing High Angles session (1st image) and Low Angles Session (2nd image)

This system is also composed by a mobile app called Smart Physio (Figure 39) which allows to access, for example, charts representing the variation of the angles of the left and right shoulder, elbow and also the spine and neck angle, in the moment that each fruit.



Figure 39 – Smart Physio app from [56]

4.1.2. Experimental Methodology

The research was carried out in the Smart Sensing Joint International Laboratory, from Iscte – Instituto Universitário de Lisboa. The study considers a group of six volunteers, four females, and two males, with age between 23 and 25 years. Prior to data collection, each volunteer was familiarized with the objectives of the study as well as with all the procedures to be followed. All the volunteers were using a shirt without sleeves and female volunteers have their hair stuck.

Eight game sessions were performed on each volunteer. The first four game sessions started with high angles. To start is performed a 60 seconds session, after 90 seconds session, then 120 seconds and finally 150 seconds. Below it's possible to see a figure with a timeline of how the tests were conducted, the red dots represent when thermal images were taken.

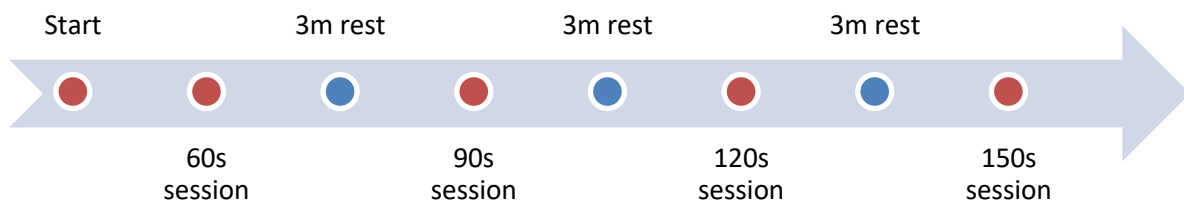


Figure 40 - Duration of game sessions

In the “Start” moment it's taken thermal images in order to have an idea of how the body temperature of the subject was when starting the sessions. Right after each sessions, there are taken also thermal images to understand how the body response was.

When changing from high angles to low angles, the subject waits five minutes with a purpose of reestablishing its body temperature.

4.1.3. Experimental Setup

The Figure 41 shows the experimental configuration used for data acquisition. As can be seen in the Figure 42, a FLIR thermographic camera, model E60 was fixed by a tripod and placed at a distance of 2 meters perpendicular to the volunteer. In order to avoid reflections, and to avoid the effect of radiation sources two grey structures were placed on the back and on the right side of the volunteer.



Figure 41 - Experimental Setup without volunteer

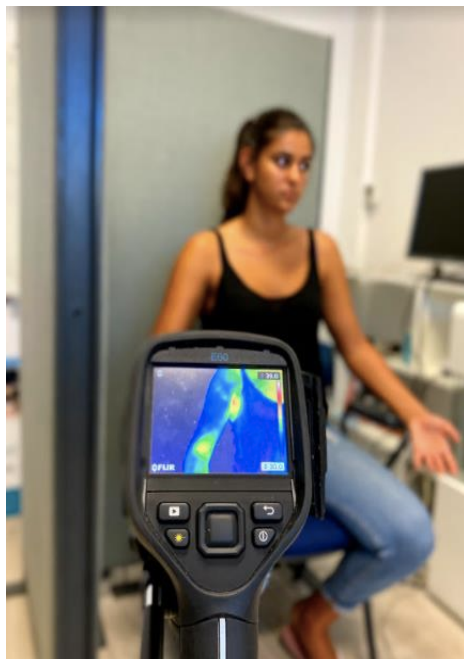


Figure 42 - Experimental Setup with volunteer

During the acquisition of the thermographic images, the temperature range of the camera was setup to a minimal value of 30°C and maximum value of 39°C in order to ignore other temperatures outside of that range.

4.1.4. Experimental Protocol

In order not to affect the credibility of the experiments, there were taken in concern some factors such as the location and preparation of the examination room, the preparation of the equipment as well as

the preparation of the volunteers. Also, the thermographic images were obtained following a defined experimental protocol.

Firstly, the volunteer was instructed to remain for 5 minutes in the room where the data were collected, absence of direct light incident on the volunteers as well as of heat generators, in order to promote the acclimatization at room conditions. Also, measurements of ambient temperature and relative humidity were taken throughout the whole time. After, the volunteer was placed in the seat, trying to find a static comfortably position. For each volunteer, the data is collected in two steps, first the sessions with high angles and later the sessions with low angles. The following protocol was considered:

1. The volunteer plays the game with the specific session duration.
2. After the session the volunteer sits on the chair.
3. A thermography image is taken to right region of the arm and then to the left region of the arm.
4. The volunteer stays in the chair during 3 minutes in order to promote the temperature stabilization;
5. The same procedure is performed until the high sessions are over.
6. When high sessions are over, the volunteer stays in the chair during 5 minutes in order to promote the temperature stabilization to start the low angles sessions.

4.2. Results and Discussions

In this subsection, the results of the experiments will be exposed and for each game session will be presented and analyzed individually. First it will be exposed the results of one volunteer and then all of the results together.

The laboratory conditions were measured with the help of Si7021 sensor as explained in subsection 3.2.2 while the game sessions were performed. The tests were carried out in three different days, having two volunteers per day. The room remained with an average temperature of 27,05°C and 49,38% of humidity.

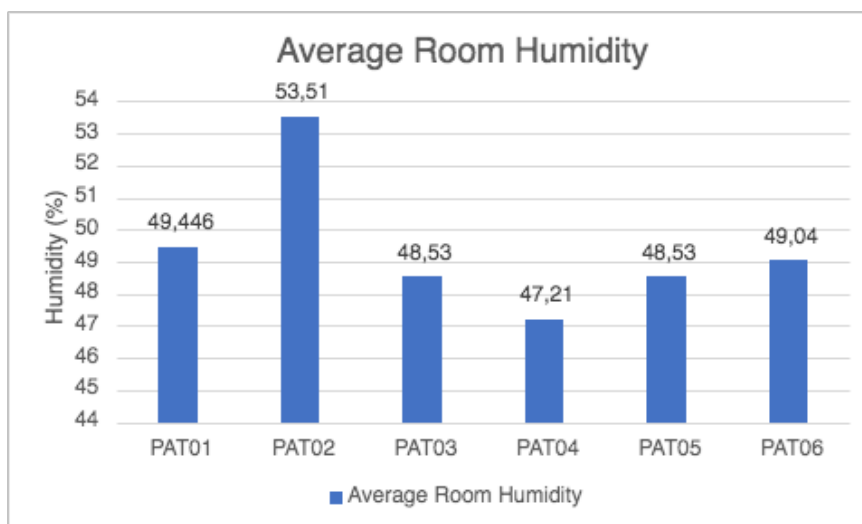


Figure 43 - Average Room Humidity during the experiments in all volunteers

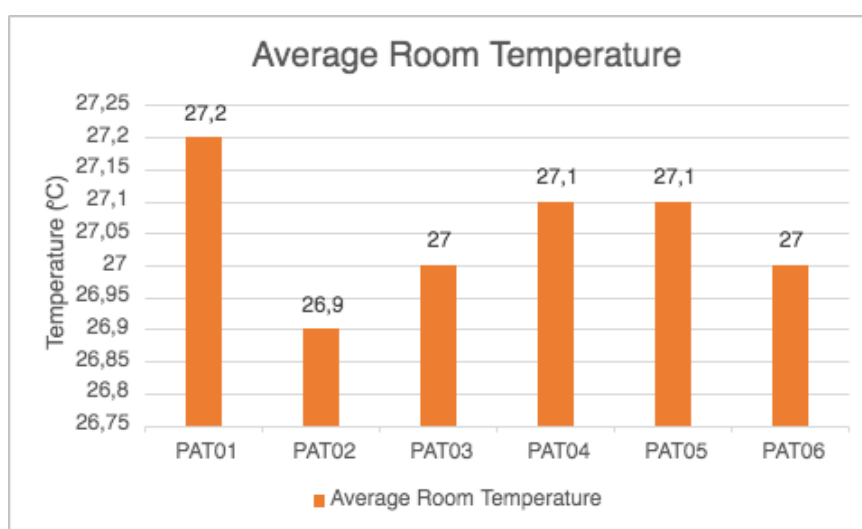


Figure 44 - Average Room Temperature during the experiments in all volunteers

4.2.1. Validation Patient

First will be analyzed the experimental procedure of a validation volunteer. This volunteer was selected randomly and it has as main purpose to understand if there were differences in starting with high angles first and then proceeding with low angles or the opposite using software ResearchIR.

4.2.1.1. High Angles Sessions Analysis of Validation Patient

Analyzing the high angles sessions, we can see that the initial temperature in the region of the arm and shoulder before the game sessions begin are different (Figure 45), as expected. This is due to the fact that, even though we maintain the same experimental protocol, the body temperature is different in each volunteer. This is a factor that is not possible to control.

It's going to be showed figures in which is possible to visualize the differences between doing the high angles game sessions first and then the low angles game session versus doing it the opposite order. Also, it's going to be calculated the absolute error between the temperature values obtained in each situation.

Starting with analyzing the values in the right arm, as the game sessions start, we can see in Figure 45 and 46, that both arm and shoulder temperatures tend to the same value specially after the 90 seconds game session. Focusing in the right region of the arm (Figure 45), the average temperature difference between both variables is 0,55°C (Table 4). In the shoulder to elbow area (Figure 46), the average temperature difference between both variables is 0,35°C, this value is obtained by doing the same calculation in the previous situation and explained in Table 4.

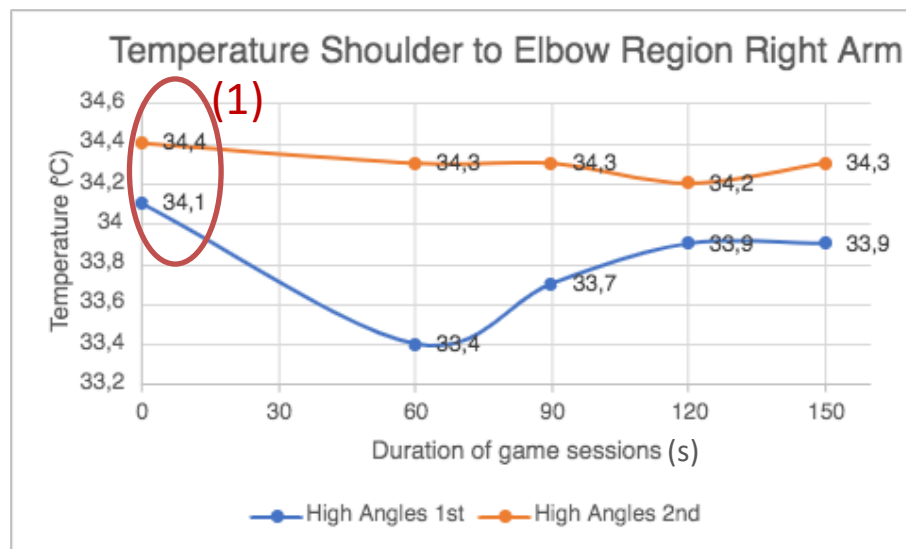


Figure 45 - VALIDATION PATIENT, High Angles; shoulder to elbow temperature on the right region of the arm

The table below shows an example of how the absolute error was calculated, the same method is done in the other game sessions. The values on the table are the exact same values in the figure above.

Table 4 - Shoulder to Elbow temperature on the right region of the arm values

HIGH ANGLES GAME SESSION			
RIGH ARM	Shoulder - Elbow		ABSOLUTE ERROR HighA 1st – HighA 2nd
	High Angles 1st	High Angles 2nd	
0	34,1	34,4	--
60	33,4	34,3	0,9
90	33,7	34,3	0,6
120	33,9	34,2	0,3
150	33,9	34,3	0,4
ABS ERROR AVG=			0,55

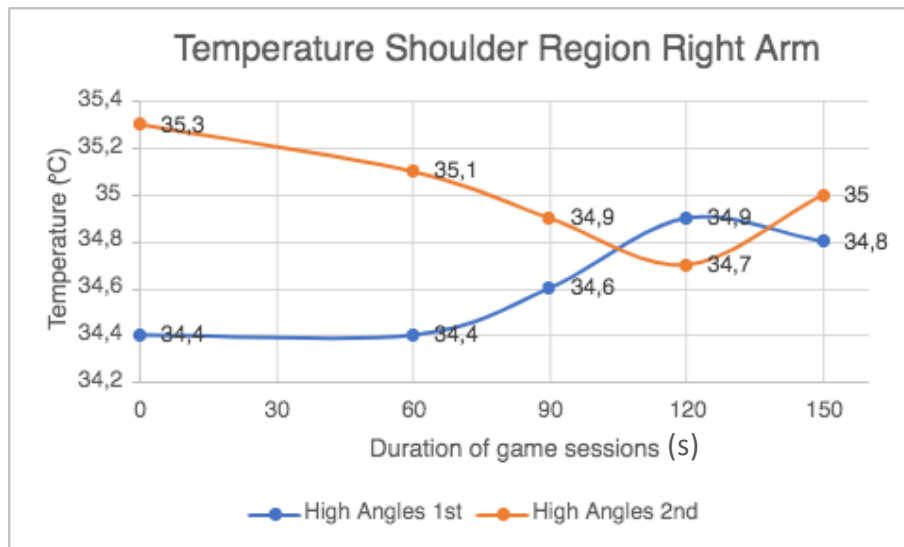


Figure 46 - VALIDATION PATIENT, High Angles; shoulder temperature on the right region of the arm

Analyzing the left arm it's possible to see a similar behavior as stated before, with the difference that the initial temperatures of the arm and should have closer values. The average temperature difference between both variables is 0,75°C in the shoulder to elbow area, and in the shoulder area is 0,53°C.

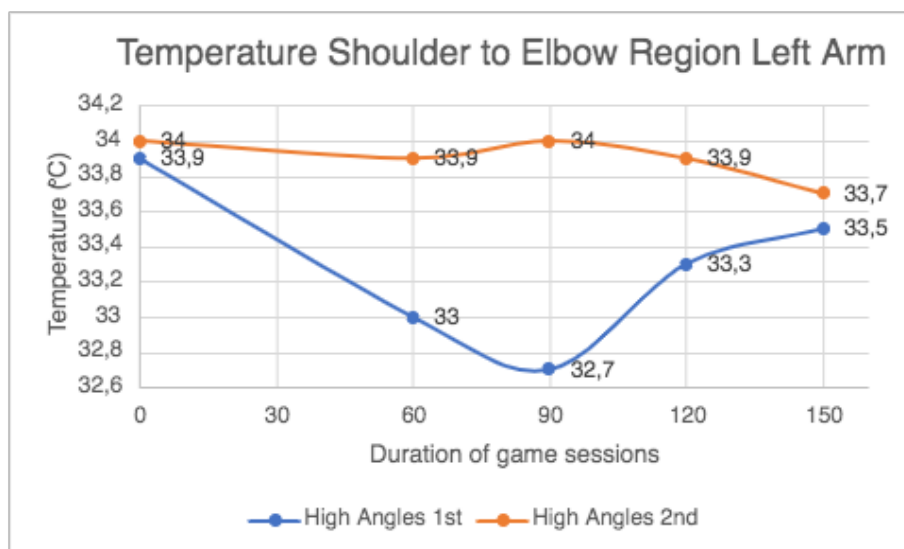


Figure 47 - VALIDATION PATIENT, High Angles; shoulder to elbow temperature on the left region of the arm

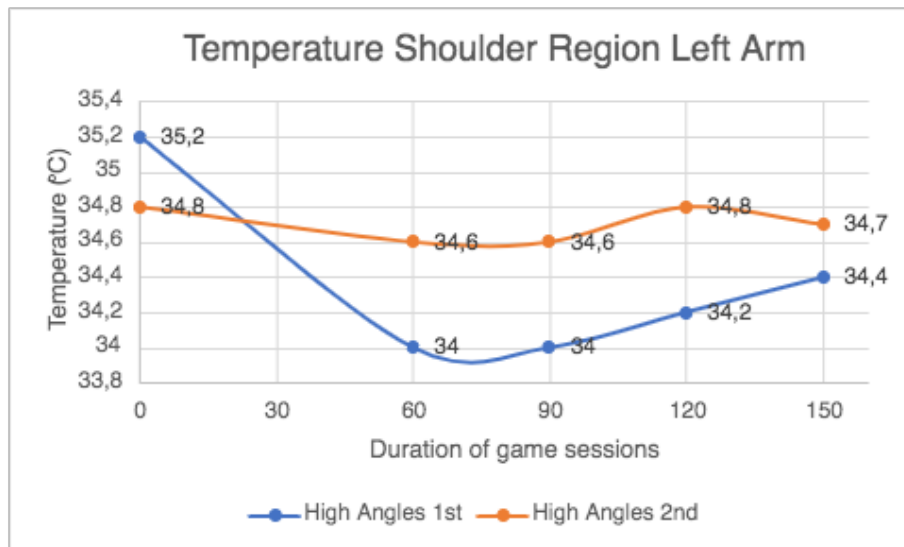


Figure 48 - VALIDATION PATIENT, High Angles; shoulder to elbow temperature on the left region of the arm

4.2.1.2. Low Angles Sessions Analysis of Validation Patient

Analyzing the low angles session on the right region of the arm, it's visible the similarities between each other. As mentioned before, the biggest difference is only the initial temperature as expected. And, again after the 90 seconds game sessions the temperature values start to be closer. In the figure below is possible to see the temperature behavior in the right region of the arm, and the average temperature difference between both variables is 0,30°C in the shoulder to elbow area, and in the shoulder area is 0,13°C.

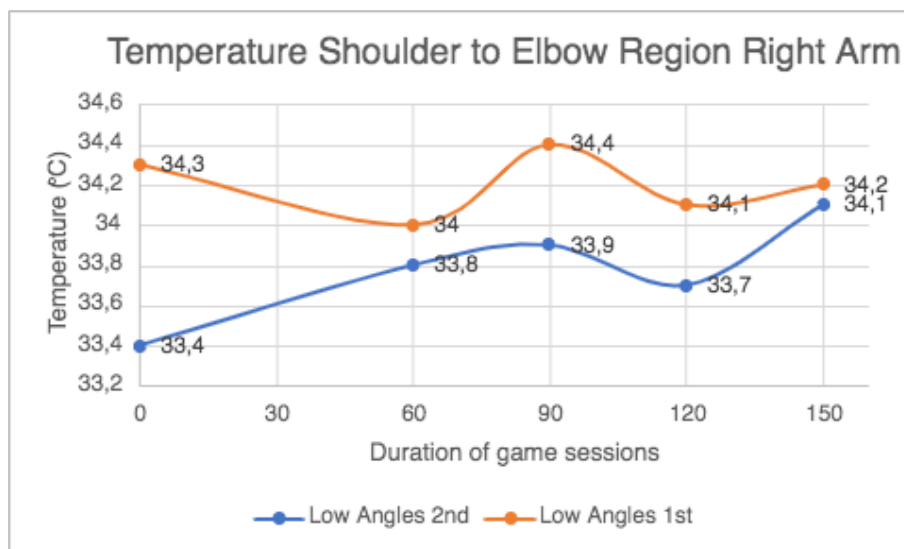


Figure 49 - VALIDATION PATIENT, Low Angles; shoulder to elbow temperature on the right region of the arm

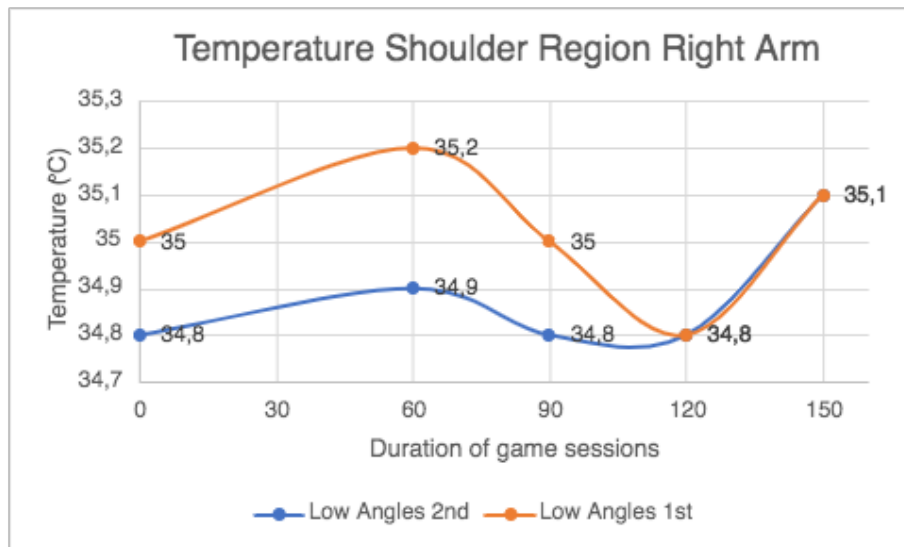


Figure 50 - VALIDATION PATIENT, Low Angles; shoulder temperature on the right region of the arm

In Figure 49 and Figure 50 is possible to see the temperature behavior in the left region of the arm. We can conclude that is very similar as stated before, especially in the shoulder region, where is possible to see almost equivalent temperature values in the last two sessions. The average temperature difference between both variables is 0,53°C in the shoulder to elbow area, and in the shoulder area is 0,40°C.

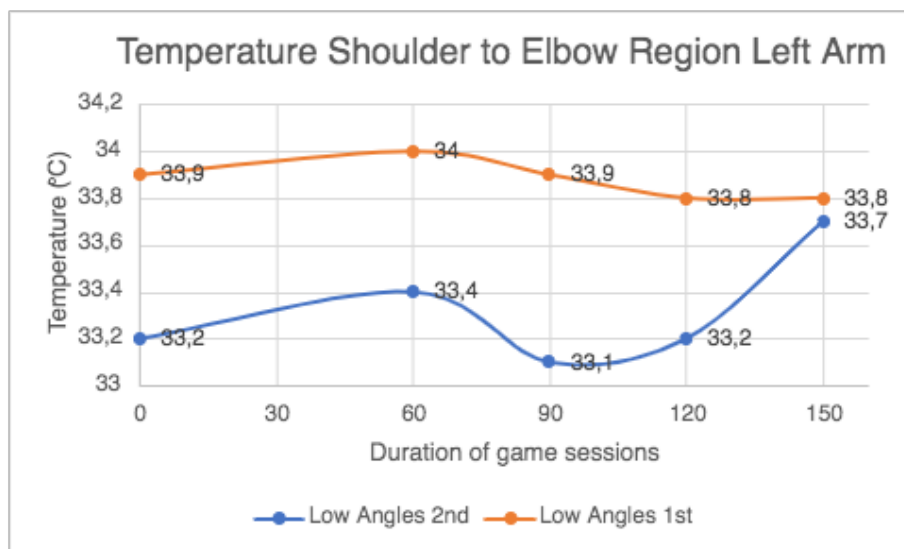


Figure 51 - VALIDATION PATIENT, Low Angles; shoulder to elbow temperature on the left region of the arm

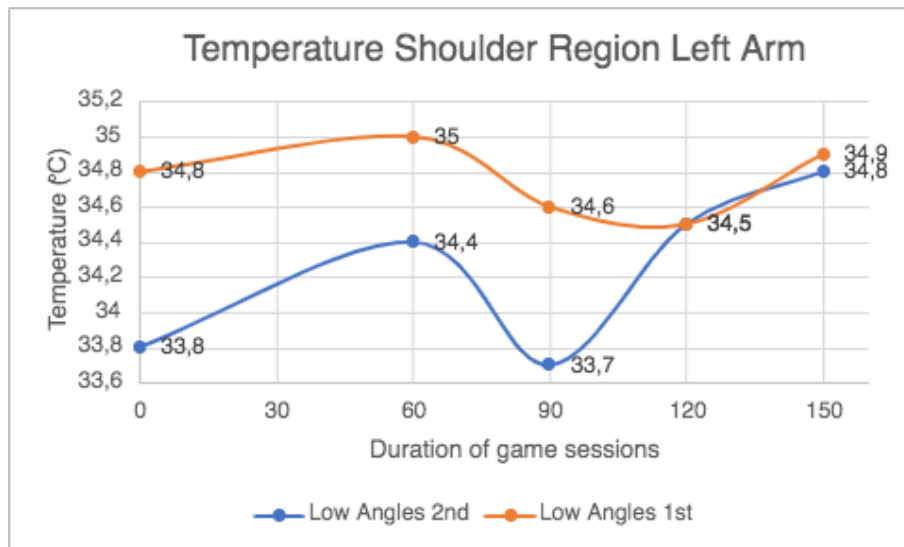


Figure 52 - VALIDATION PATIENT, Low Angles; shoulder temperature on the left region of the arm

The validation patient completed a total of 8 game sessions and in conclusion, the order of the angles in the game sessions it's not a factor that has a big impact on the volunteer's temperature as the values taken don't have a big disparity between them. Having this said, it was decided to perform in all volunteers the high angles game sessions first and then the low angles game sessions.

In the figure below is possible to see all average absolute errors calculated in each game session in a graphic in order to have a better visualization of the difference in every game session. The higher absolute error is 0,75°C, the lowest is 0,125°C and the average is 0,4°C is expressed by the black line.

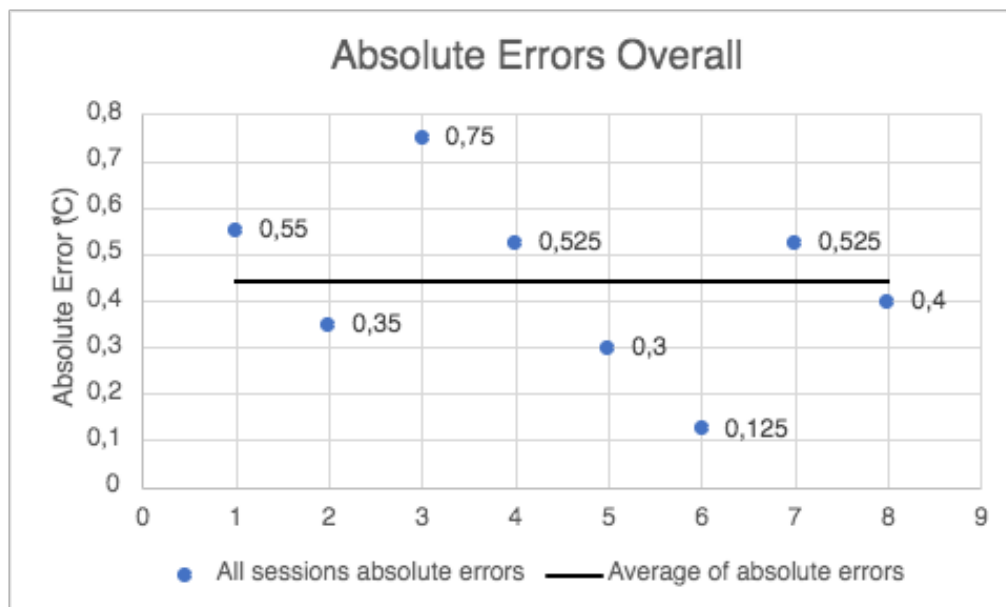


Figure 53 – Average Absolute Errors Overall in all sessions of VALIDATION PATIENT

4.2.2. All volunteers Analysis

In this second analysis, the focus is to understand the main differences between ResearchIR software and the script developed in MATLAB.

It will be exposed three main values for understanding the difference between them:

- Absolute Error Average which is the average of the difference between the measured value and the real value;
- Relative Error Average which is the average of the product between the absolute error and the real value;
- Correlation Value to understand the relationship between the variables, the conclusion of the values will be considered by the following table. This value is going to be calculated with the help of Excel function “CORREL”.

Table 5 – Relationship Strength of correlation value based on [57]

Absolute Value of r	Strength of Relationship
$r < 0.3$	None or very weak
$0.3 < r < 0.5$	Weak
$0.5 < r < 0.7$	Moderate
$r > 0.7$	Strong

Before starting the game sessions, there were taken one picture to each arm in order to understand what was the starting temperature of each volunteer.

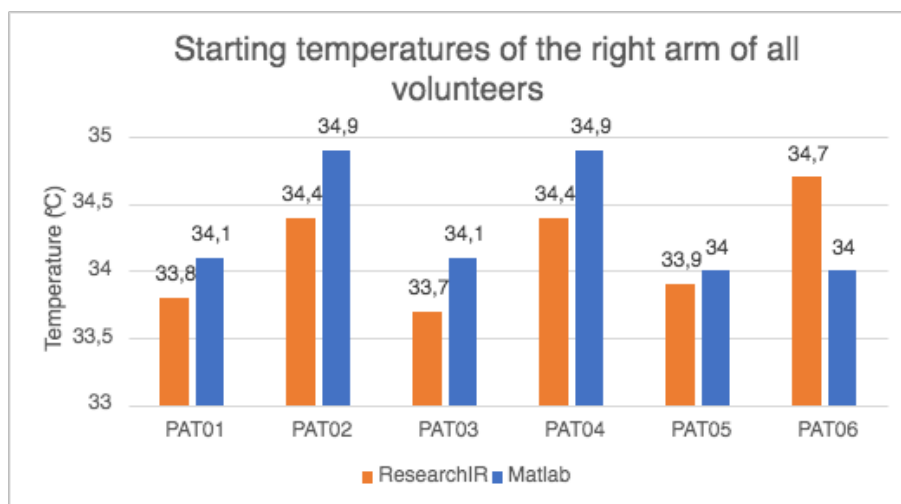


Figure 54 - Starting temperatures of the right region of the arm of all volunteers

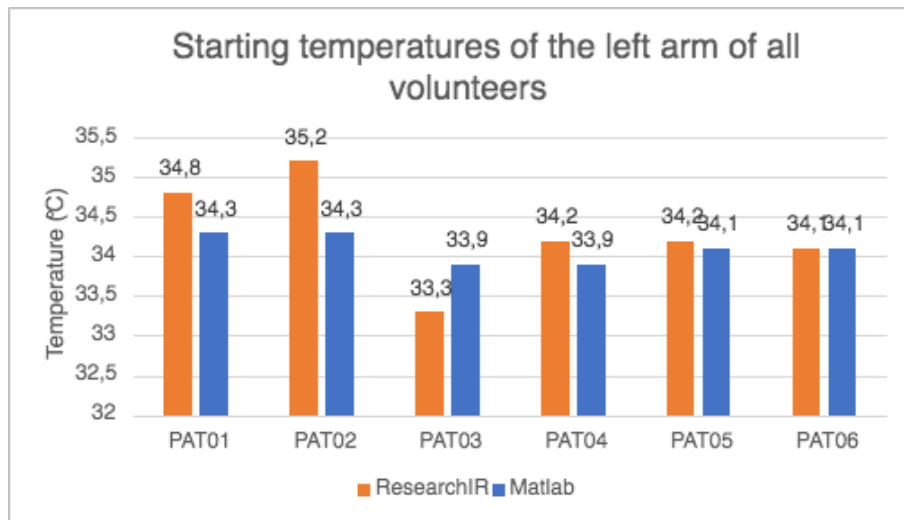


Figure 55 - Starting temperatures of the left region of the arm of all volunteers

As expected and seen in the graphics above, all volunteers started with similar temperatures in both arms. The range of temperatures in the right region of the arm was [33.7; 34.7°C] in ResearchIR and [34; 34.9°C] in MATLAB. In the left region of the arm the range of temperatures was [33.3; 35.2°C] in ResearchIR and [33.9; 34.3°C] in MATLAB. All volunteers show a normal temperature.

When comparing both software of image processing, there's an average difference between them of 0,42°C in the right region of the arm and 0,40°C in the left region of the arm. Analyzing the correlation value, the value on the right region of the arm is considered weak and on the left region of the arm is considered strong, according to Table 5.

Table 6 - Average Absolute and Relative Error, and Correlation Value of all volunteers before game sessions start

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,42°C	0,01	0,42
Left region of the arm	0,40°C	0,01	0,86

4.2.2.1. High Angles in all volunteers

- 60 seconds game session

This is the first game session that the volunteers perform, below it's exposed all of the results.

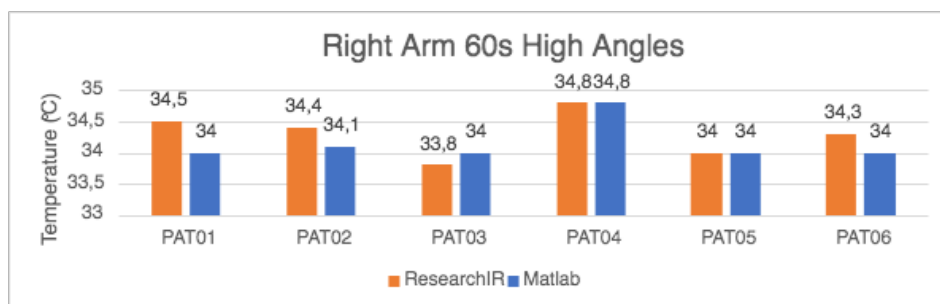


Figure 56 - 60 seconds high angles game sessions of all volunteers right region of the arm

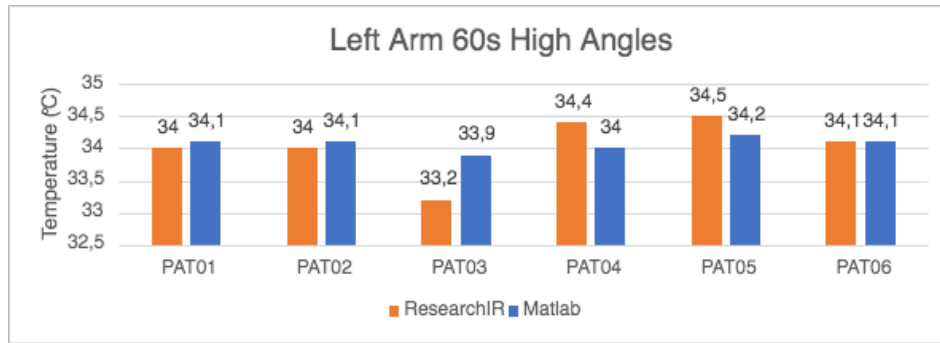


Figure 57 - 60 seconds high angles game sessions of all volunteers left region of the arm

In this game session it's possible to see that both software have similar temperature values. Having in ResearchIR a range of temperature values in the right region of the arm of [33,8;34,8°C] and in MATLAB of [34; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,2;34,5°C] and in MATLAB [34;34,2°C].

In Table 7, it's showed all values in Figure 57 and it's demonstrated how Absolute Error, Relative Error, Average Absolute Error, Average Relative Error and Correlation Value are calculated. This procedure is replicated in all game sessions. Looking into Table 8, there's an average difference between all values of 0,22°C in the right region of the arm and 0,27°C in the left region of the arm. Also, the correlation value of the values in the right region of the arm is 0,71 and in the left region of the arm is 0,75. Both correlation values are positive so it means that both variables move in the same direction and, according to Table 5, both variables have a strong relationship.

Table 7 - Example of how Absolute Error, Relative Error, Average Absolute Error, Average Relative Error and Correlation Value are calculated

	ResearchIR (1)	MATLAB (2)	Absolute Error (1)-(2)	Relative Error ((2)-(1))/(1)	Avg Absolute Error
PAT01	34,5	34	0,5	0,014	0,22
PAT02	34,4	34,1	0,3	0,009	Avg Relative Error
PAT03	33,8	34	0,2	0,006	
PAT04	34,8	34,8	0	0	R
PAT05	34	34	0	0	0,71
PAT06	34,3	34	0,3	0,009	

Table 8 - Average Absolute and Relative Error, and Correlation Value of all volunteers 60 seconds high angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,22°C	0,006	0,71
Left region of the arm	0,27°C	0,008	0,75

- 90 seconds game session

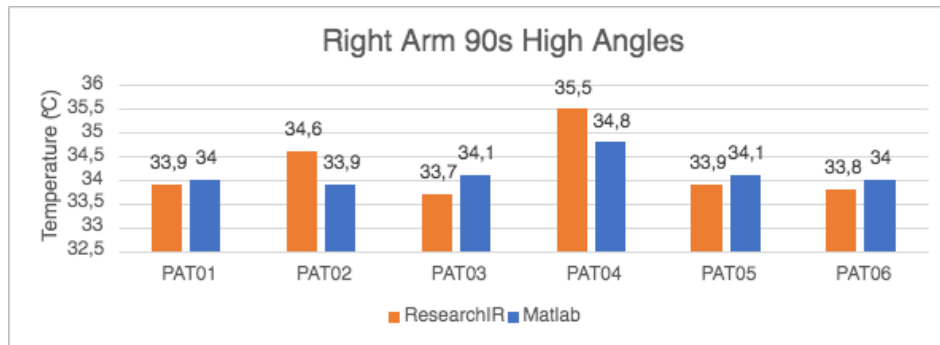


Figure 58 - 90 seconds high angles game sessions of all volunteers right region of the arm

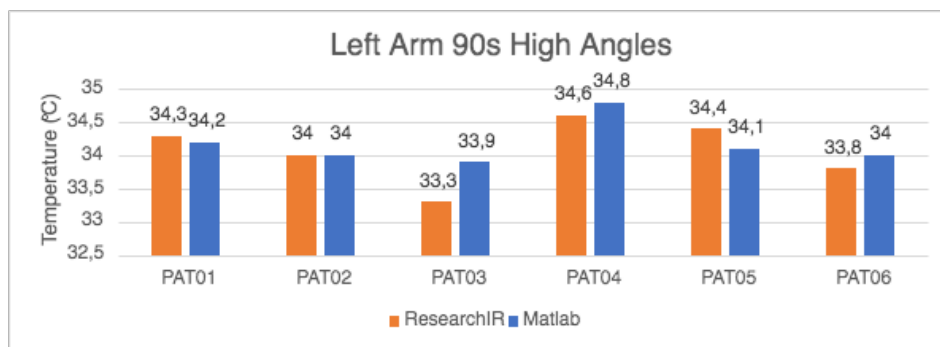


Figure 59 - 90 seconds high angles game sessions of all volunteers left region of the arm

In this 90 seconds game session it's possible to see that both software have again similar temperature values. In ResearchIR there's a range of temperature values in the right region of the arm of [33,7;35,5°C] and in MATLAB of [33,9; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,3;34,6°C] and in MATLAB [33,9;34,2°C].

When doing the same method explained in Table 7, there's an average difference between them of 0,38°C in the right region of the arm and 0,32°C in the left region of the arm. Also, the correlation value of the values in the right region of the arm is 0,78 and in the left region of the arm is 0,76 which shows in both cases, a strong relationship between variables.

Table 9 - Average Absolute and Relative Error, and Correlation Value of all volunteers 90 seconds high angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,38°C	0,011	0,78
Left region of the arm	0,32°C	0,009	0,76

- 120 seconds game session

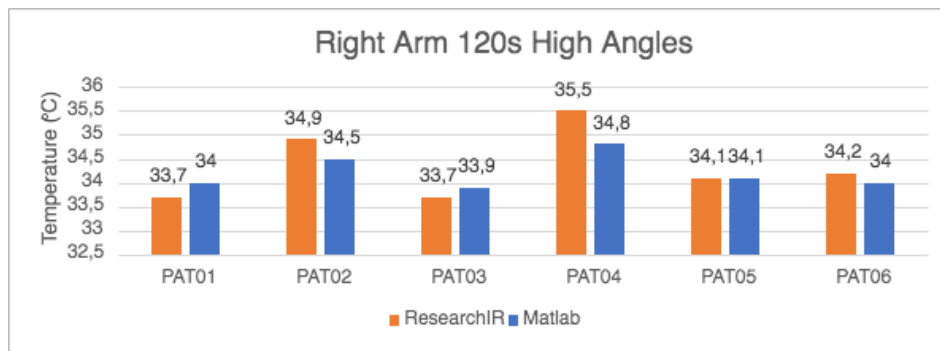


Figure 60 - 120 seconds high angles game sessions of all volunteers right region of the arm

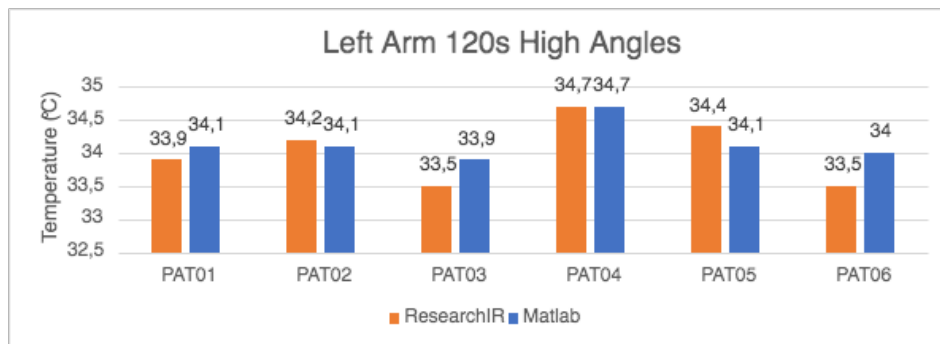


Figure 61 - 120 seconds high angles game sessions of all volunteers left region of the arm

In this game session it's possible to see that the temperature values in both arms are very similar. In ResearchIR there's a range of temperature values in the right region of the arm of [33,7;35,5°C] and in MATLAB of [33,9; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,5;34,7°C] and in MATLAB [33,9;34,2°C].

Also, on the right region of the arm, there's a difference of 0,30°C and a correlation value very close to 1 which shows a very strong relationship. Similarly, on the left region of the arm there's a difference of 0,38°C and the correlation value shows a very strong relationship.

Table 10 - Average Absolute and Relative Error, and Correlation Value of all volunteers 120 seconds high angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,30°C	0,009	0,97
Left region of the arm	0,38°C	0,011	0,82

- 150 seconds game session

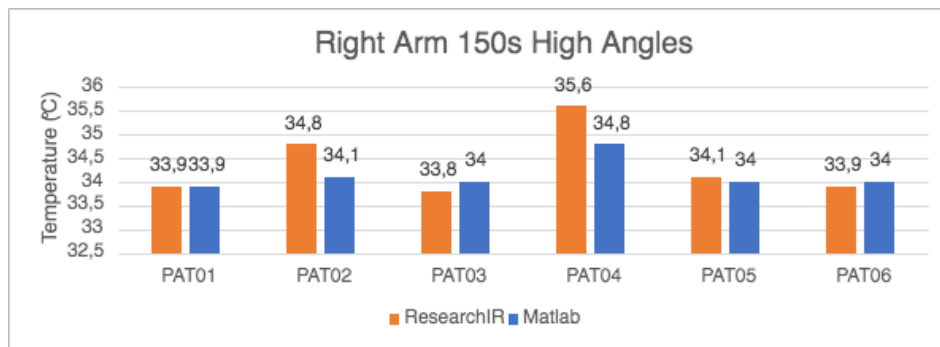


Figure 62 - 150 seconds high angles game sessions of all volunteers right region of the arm

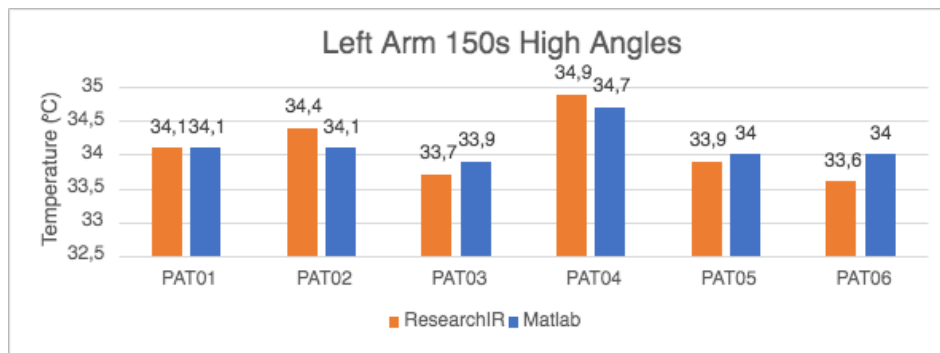


Figure 63 - 150 seconds high angles game sessions of all volunteers left region of the arm

In this game session, there's again, very similar temperature values in both arms. In ResearchIR there's a range of temperature values in the right region of the arm of [33,8;35,6°C] and in MATLAB of [33,9; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,6;34,9°C] and in MATLAB [33,9;34,1°C].

The average temperature difference is 0,32°C on the right region of the arm and 0,20°C on the left region of the arm. Both correlation values are very close to 1, which demonstrates a very good relationship.

Table 11 - Average Absolute and Relative Error, and Correlation Value of all volunteers 150 seconds high angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,32°C	0,009	0,92
Left region of the arm	0,20°C	0,006	0,90

4.2.2.2. Low Angles Sessions in all volunteers

The analysis of this game sessions will be performed as in the subsection before. It will be presented the values of average of absolute and relative error and correlation value.

- 60 seconds game session

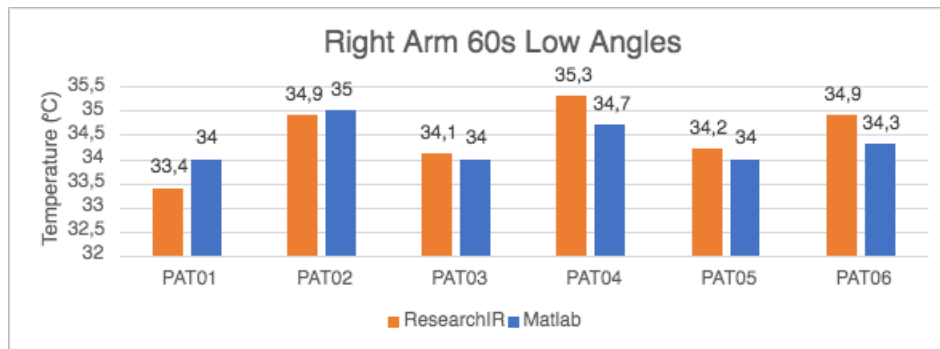


Figure 64 - 60 seconds low angles game sessions of all volunteers right region of the arm

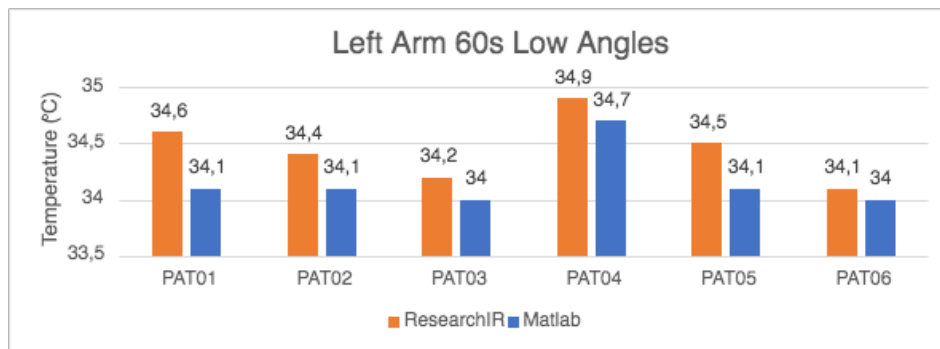


Figure 65 - 60 seconds low angles game sessions of all volunteers left region of the arm

In this game session, it's already visible the similarity of values in both arm region. In ResearchIR there's a range of temperature values in the right region of the arm of [33,4;35,3°C] and in MATLAB of [33,9; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [34,1;34,9°C] and in MATLAB [34;34,1°C].

The average temperature difference is 0,37°C on the right region of the arm and 0,28°C on the left region of the arm. Both correlation values are superior to 0,7, which demonstrates a very good relationship.

Table 12 - Average Absolute and Relative Error, and Correlation Value of all volunteers 60 seconds low angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,37°C	0,011	0,77
Left region of the arm	0,28°C	0,008	0,86

- 90 seconds game session

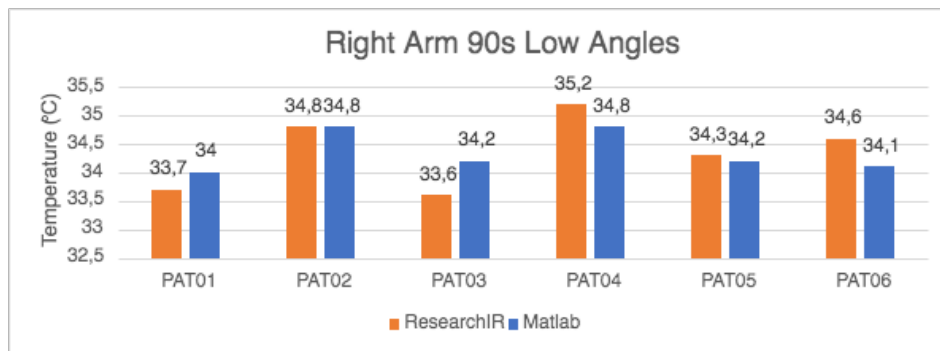


Figure 66 - 90 seconds low angles game sessions of all volunteers right region of the arm

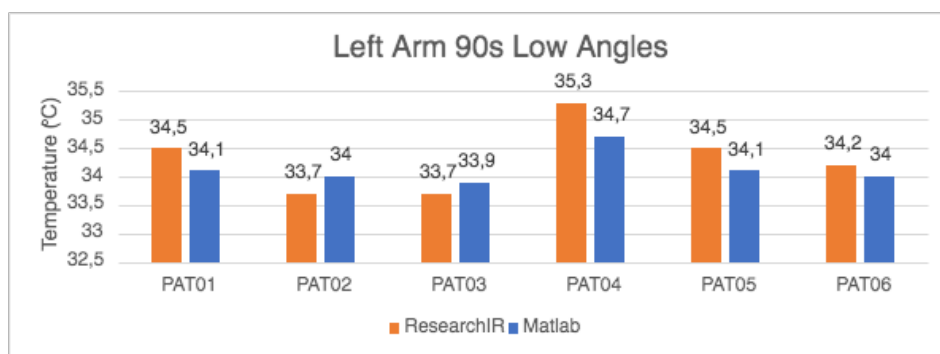


Figure 67 - 90 seconds low angles game sessions of all volunteers left region of the arm

In this game session, the range of temperature values in ResearchIR on the right region of the arm is [33,6;35,2°C] and in MATLAB of [34; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,7;35,3°C] and in MATLAB [33,9;34,1°C].

The average temperature difference is 0,32°C on the right region of the arm and 0,35°C on the left region of the arm. Both correlation values are superior to 0,7, which demonstrates a very good relationship.

Table 13 - Average Absolute and Relative Error, and Correlation Value of all volunteers 90 seconds low angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,32°C	0,009	0,78
Left region of the arm	0,35°C	0,010	0,91

- 120 seconds game session

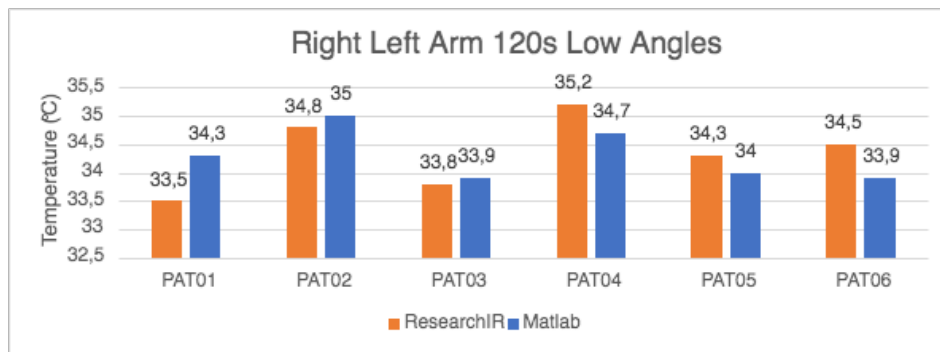


Figure 68 - 120 seconds low angles game sessions of all volunteers right region of the arm

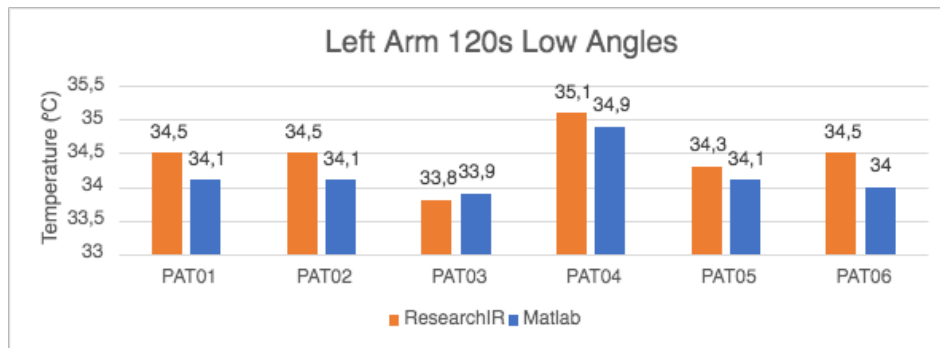


Figure 69 - 120 seconds low angles game sessions of all volunteers left region of the arm

In this game session, the range of temperature values in ResearchIR on the right region of the arm is [33,5;35,2°C] and in MATLAB of [33,9; 34,3°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,8;35,1°C] and in MATLAB [33,9;34,9°C].

The average temperature difference is 0,42°C on the right region of the arm and 0,3°C on the left region of the arm. In this game session the left region of the arm shows a better correlation value, having a strong relationship. In the other hand, the correlation value on the right region of the arm shows a moderate relationship.

Table 14 - Average Absolute and Relative Error, and Correlation Value of all volunteers 120 seconds low angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,42°C	0,012	0,57
Left region of the arm	0,30°C	0,009	0,86

- 150 seconds game session

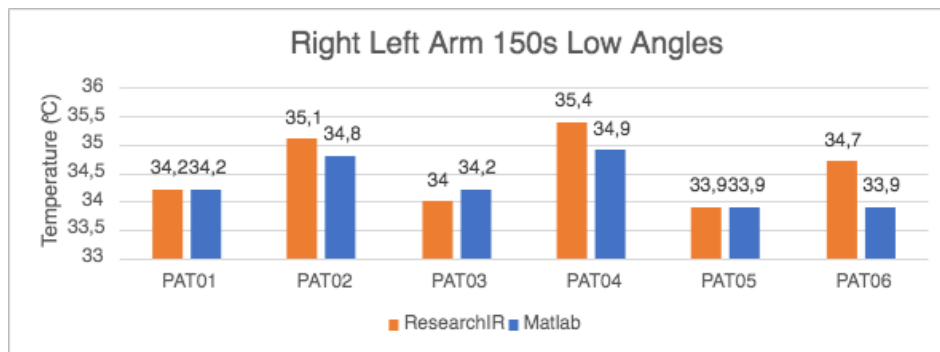


Figure 70 - 150 seconds low angles game sessions of all volunteers right region of the arm

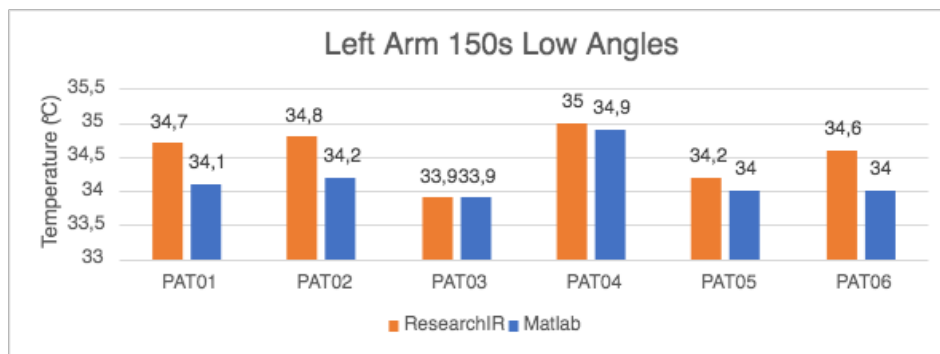


Figure 71 - 150 seconds low angles game sessions of all volunteers left region of the arm

In this game session, the range of temperature values in ResearchIR on the right region of the arm is [33,9;35,4°C] and in MATLAB of [33,9; 34,8°C]. In the left region of the arm there's also a similar range, in ResearchIR [33,9;35°C] and in MATLAB [33,9;34,2°C].

In this game sessions there's better correlation values, with both of them having a strong relationship. The average temperature difference is 0,30°C on the right region of the arm and 0,35°C on the left region of the arm.

Table 15 - Average Absolute and Relative Error, and Correlation Value of all volunteers 150 seconds low angles game sessions

	Average Absolute Error	Average Relative Error	R
Right region of the arm	0,30°C	0,009	0,80
Left region of the arm	0,35°C	0,010	0,74

4.2.2.3. All sessions with all volunteers analysis

In this section, it will be exposed all results combined of all volunteers in order to understand the evolution of each patient throughout the game sessions. Two figures will be showed regarding high angles and other two with low angles sessions.

The Figure 72 shows the six patients and how their temperature of the region of the right arm was in each game session. The value 0 represents before starting the game sessions.

In this figure is visible that PAT04 reached higher temperatures which means that the volunteer putted more effort into the experiment comparing to the other volunteers. In the other hand, the volunteer that had a better performance was PAT03 that reached lower temperatures.

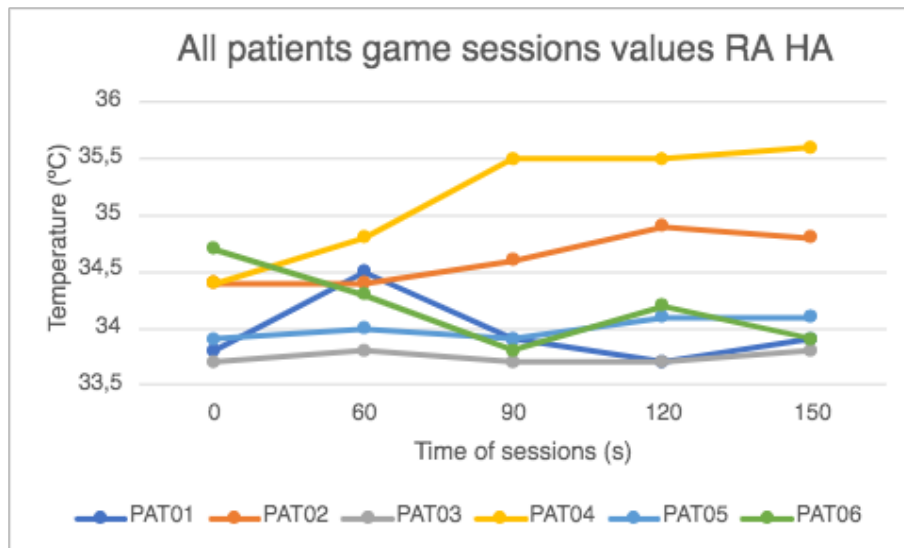


Figure 72 - All patients game sessions values right arm (RA) High Angles(HA)

In Figure 73, similarly to the previous situation, the volunteer PAT04 reached again higher temperatures putting an extra effort comparing to the other volunteers especially volunteer PAT03 which had the lowest values except in game session of 150s.

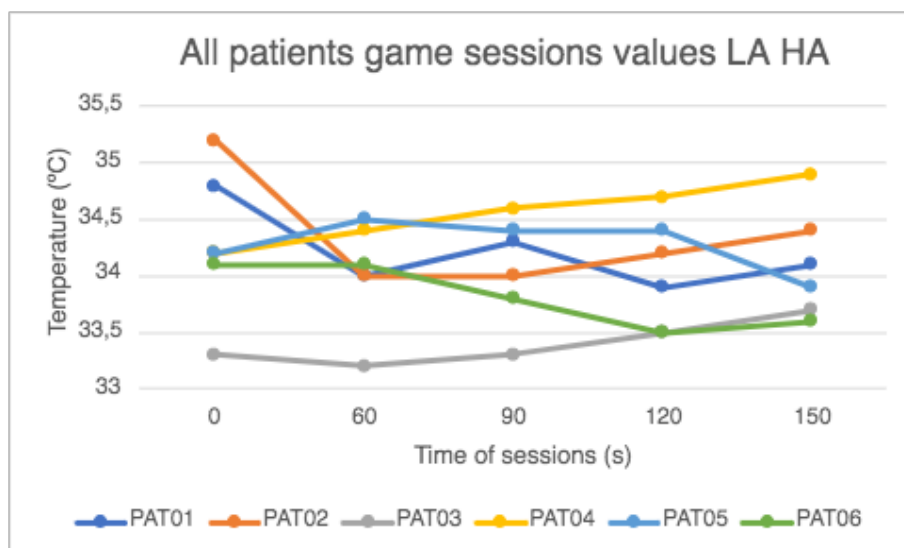


Figure 73 - All patients game sessions values left arm (LA) High Angles (HA)

In Figure 74, are showed the low angles sessions. Again is possible to stand out volunteer PAT04 as it reached again the higher temperatures of all volunteers. In the lower temperatures there's a dispute between two volunteers. Both volunteer PAT03 and PAT01 reached very low temperatures which means a lower effort in performing the game sessions.

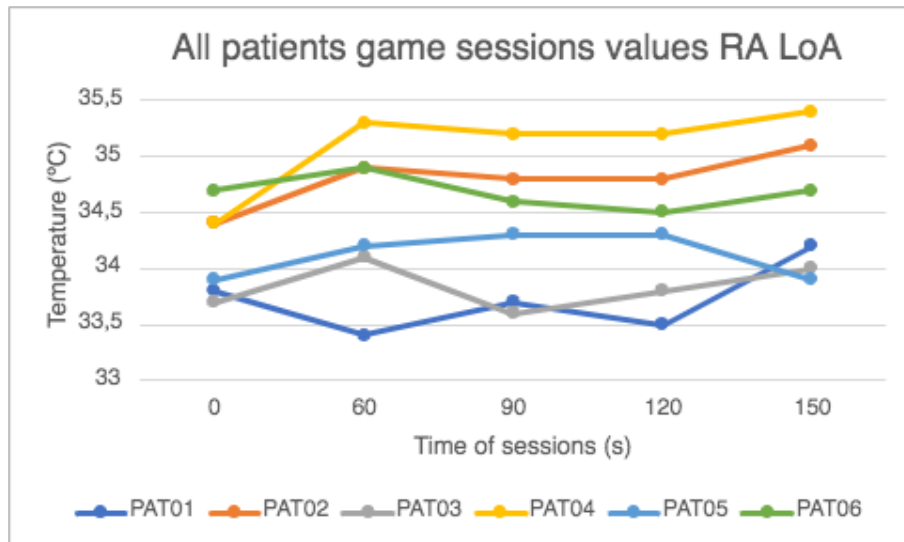


Figure 74 - All patients game sessions values right arm (RA) Low Angles (LoA)

Finally, in Figure 75, it's possible to visualize more discrepancies in all volunteers. However, volunteer PAT04 still stands out from the rest in the high temperatures. Also, volunteer PAT03 still has the lowest temperatures.

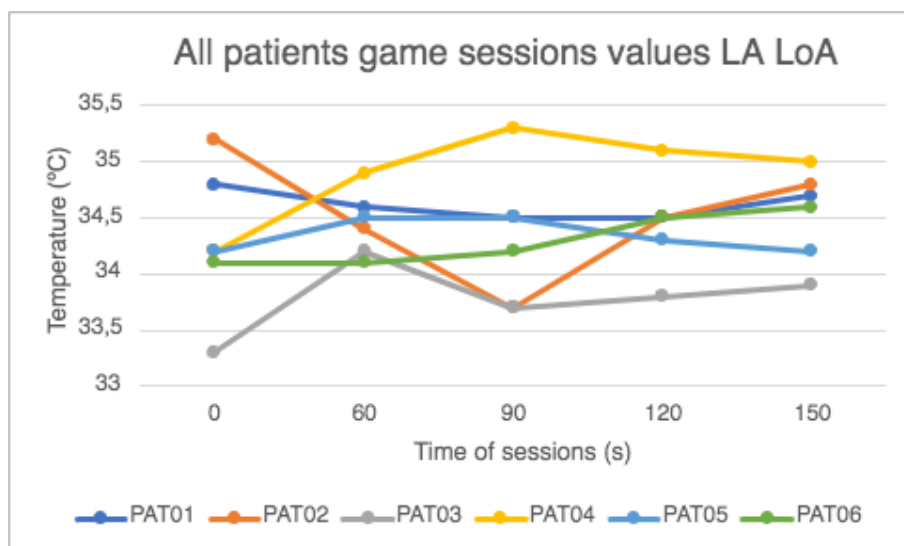


Figure 75 - All patients game sessions value left arm (LA) Low Angles (LoA)

In conclusion, the Figure 76 shows like a ranking of patients in terms of temperature values. It's a cumulative graphic of all temperature values, which means that each temperature value of each session was summed for each patient (total of ten game sessions). Is visually understandable that volunteer PAT04 had higher values than the rest of the volunteers as highlighted before. Concerning PAT03, it can be concluded that was the volunteer that had the best performance in the game as the increase of temperature was the lowest of all volunteers.

The average of the values is 685,85°C which means that the performance of 4 out of 6 volunteers was below average.

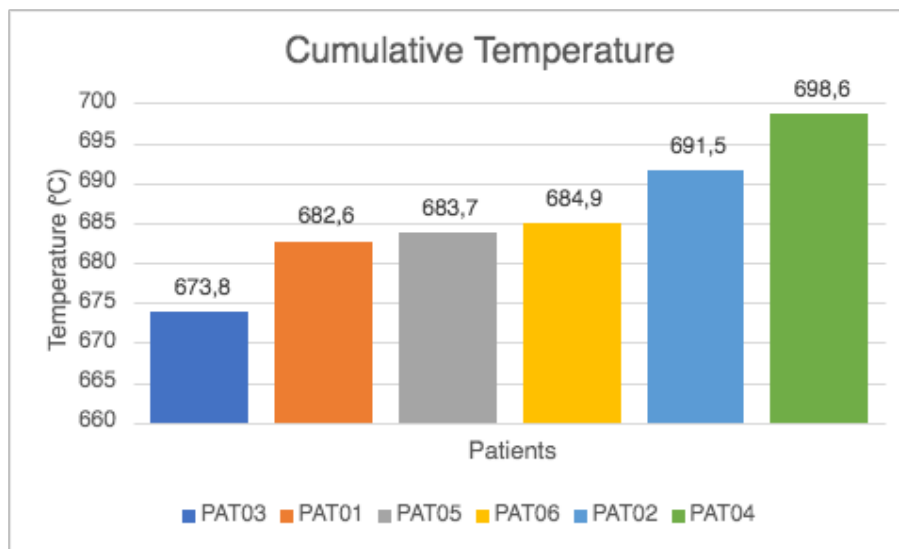


Figure 76 – Cumulative temperature

4.2.3. Correlation Value Behavior

As we could see before, from the analysis in the subsections, the correlation value starts to get closer to 1 as the duration of the game sessions starts to increase. As said before, all values were calculated in Excel with the function CORREL.

In the high angles game sessions, it was calculated a total of ten correlation values. Five related to the right arm region and the other five to the left arm region. They are expressed in Figure 77 and Figure 78 in order to better visualize the tendency of the values.

It's visible that as the duration of the game sessions increases, the correlation values also tends to increase. This behavior was possible to observe in both arms and after the 60 seconds game session all correlation values are higher than 0,7 which means that are consider a strong relationship. In the figures below it's possible to observe what stated before as well as the polynomial tendency lines of the correlation values.

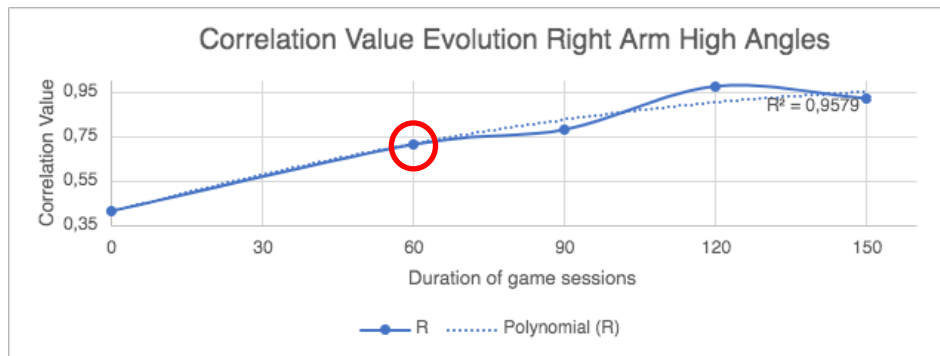


Figure 77 - Correlation Value Evolution in high angles game sessions on the right region of the arm

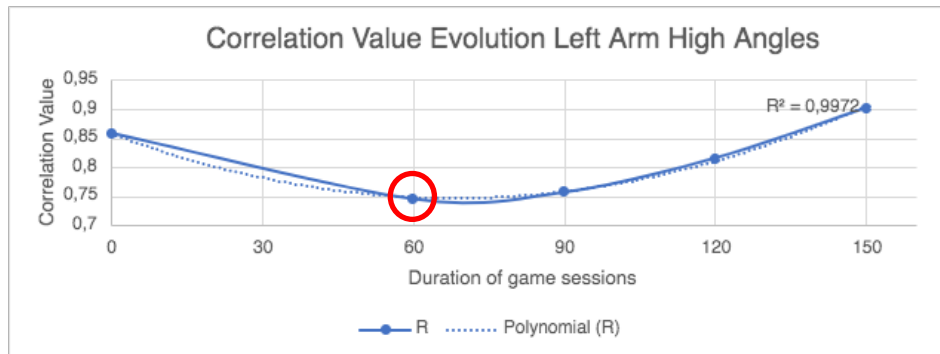


Figure 78 - Correlation Value Evolution in high angles game sessions on the left region of the arm

In the low angles sessions, it was calculated a total of ten correlation values. Five related to the right arm region and the other five to the left arm region. They are expressed in Figure 79 and Figure 80 in order to better visualize the tendency of the values.

Again, in both arms was possible to observe that after the 60 seconds game session all correlation values are higher than 0,7 (except from the 120 seconds game session in the right region of the arm highlighted in red). Regardless of this exception, the majority of the correlation values shows a strong relationship. In the figures below it's possible to observe what stated before as well as the polynomial tendency lines of the correlation values.

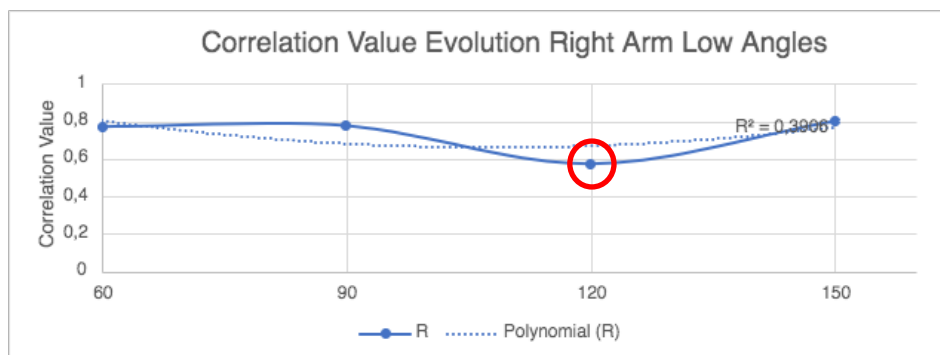


Figure 79 - Correlation Value Evolution in low angles game sessions on the right region of the arm

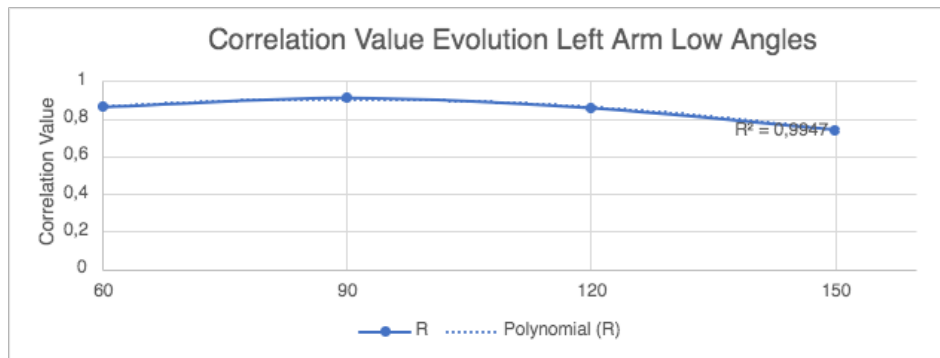


Figure 80 - Correlation Value Evolution in low angles game sessions on the left region of the arm

The correlation value is a very good metric to understand the accuracy of the results as it's a measure of the strength of the relationship between the two variables. The majority of the correlation values were higher than 0,7 and the highest value calculated was 0,97 (Table 10) which is very high value that demonstrated a very strong relationship.

Having this said, the balance of this analysis is positive as the developed software shows very similar values between the already existing software.

5. Conclusions and Future Work

5.1. Conclusions

In conclusion, the main objectives of this dissertation were completed such as the development of an Android mobile platform that enhances the acquisition of thermal images and an image processing software that helps the diagnostic process faster, automatic and without the need to manually extract the images and insert them into an image processing software.

Concerning the database usage, the main goal was to store the thermal images in a way that was possible to associate a patient to an image which it was achieve. Also, the link between MATLAB and Firebase was a difficult step to accomplish as there were not many information about how it could be done a communication between them.

Regarding the developed image processing software, it was proven the accuracy of temperature measurement by direct comparison of the values obtained in ResearchIR software.

To summarize, with this system it's possible to acquire data from the camera and send it to the real-time database. This database keeps the history of the patient's info and thermal images that then, the physiotherapist can access through the mobile application. It's also possible to send the images acquired from the camera to MATLAB in order for them to be analyzed.

With this application, it's possible aid clinics that perform rehabilitation with serious game or other games. For example, to help physiotherapists analyzing the evolution of theirs patients or even helping them giving a better and more accurate diagnose with only their mobile phone/tablet.

5.2. Future Work

Despite the fact that the developed software satisfies the initial requirements, there is always room for improvement. Also, regarding the experimental work, there is also room to test the system in different situations and conditions than those presented in Chapter 4 including extended number of volunteers.

In terms of the mobile application, the aspects that could be improved are, for example, allow for two thermographic cameras to be connected at the same time and also to perform authentication for the patients in order to let them access their own information.

Also, at the level of data processing, some functions could be processed directly in Firebase with the help of Firebase *Cloud Functions* in order to avoid occupying resources of the mobile phone.

Regarding the script developed in MATLAB, one improvement is to add more metrics when the thermal image is analyzed.

References

- [1] 'Anatomy of an Electromagnetic Wave | Science Mission Directorate'. https://science.nasa.gov/ems/02_anatomy (accessed Sep. 20, 2021).
- [2] 'Infrared - Wikipedia'. <https://en.wikipedia.org/wiki/Infrared> (accessed Sep. 20, 2021).
- [3] 'Espectro eletromagnético: definição e exercícios', *Mundo Educação*. <https://mundoeducacao.uol.com.br/fisica/espectro-eletromagnetico.htm> (accessed Sep. 20, 2021).
- [4] 'electromagneticspectrum.jpg (1393×617)'. <https://www.miniphysics.com/wp-content/uploads/2011/07/electromagneticspectrum.jpg> (accessed Sep. 20, 2021).
- [5] J. G. Betts *et al.*, 'Anatomy & Physiology', p. 1420.
- [6] '32431058.pdf'. Accessed: Sep. 20, 2021. [Online]. Available: <https://core.ac.uk/download/pdf/32431058.pdf>
- [7] '32431058.pdf'. Accessed: Sep. 21, 2021. [Online]. Available: <https://core.ac.uk/download/pdf/32431058.pdf>
- [8] 'Thermography', *Wikipedia*. Sep. 18, 2021. Accessed: Sep. 21, 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Thermography&oldid=1044970709>
- [9] O. Postolache, 'Remote sensing technologies for physiotherapy assessment', in *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Mar. 2017, pp. 305–312. doi: 10.1109/ATEE.2017.7905141.
- [10] 'What is Thermography in Engineering?' <https://www.twi-global.com/technical-knowledge/faqs/what-is-thermography-in-engineering.aspx> (accessed Sep. 21, 2021).
- [11] B. Hegedús, 'The Potential Role of Thermography in Determining the Efficacy of Stroke Rehabilitation', *J. Stroke Cerebrovasc. Dis.*, vol. 27, no. 2, pp. 309–314, Feb. 2018, doi: 10.1016/j.jstrokecerebrovasdis.2017.08.045.
- [12] I. Nowak, M. Mraz, and M. Mraz, 'Thermography assessment of spastic lower limb in patients after cerebral stroke undergoing rehabilitation', *J. Therm. Anal. Calorim.*, vol. 140, no. 2, pp. 755–762, Apr. 2020, doi: 10.1007/s10973-019-08844-y.
- [13] '404 Not Found', *Reduction Revolution*. <https://reductionrevolution.com.au/404> (accessed Sep. 21, 2021).
- [14] M. M. P. Leite and M. B. P. Toralles, 'Termografia infravermelha pré e pós-uso da Therapy Taping para controle da dor do paciente com fascite plantar: relato de caso', *Rev. Ciênc. Médicas E Biológicas*, vol. 13, no. 3, Art. no. 3, Mar. 2014, doi: 10.9771/cmbio.v13i3.12952.
- [15] M. P. Clemente, C. Faria, F. A. Coutinho, J. Mendes, J. C. Pinto, and J. M. Amarante, 'The Application of Infrared Thermography as a Quantitative Sensory Measure of the DC/TMD', in *VipIMAGE 2019*, Cham, 2019, pp. 330–340. doi: 10.1007/978-3-030-32040-9_35.
- [16] A. Dionísio, L. Roseiro, J. Fonseca, and P. Nicolau, 'Thermography Evaluation on Facial Temperature Recovery after Elastic Gum', Apr. 2017, doi: 10.5281/zenodo.1130505.
- [17] R. Lasanen *et al.*, 'Thermal imaging in screening of joint inflammation and rheumatoid arthritis in children', *Physiol. Meas.*, vol. 36, no. 2, pp. 273–282, Feb. 2015, doi: 10.1088/0967-3334/36/2/273.
- [18] P. Majumdar, K. Das, N. Nath, and M. K. Bhowmik, 'Detection of Inflammation from Temperature Profile Using Arthritis Knee Joint Datasets', in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, Jun. 2018, pp. 409–411. doi:

10.1109/ICHI.2018.00077.

[19] J. Pauk, M. Ihnatouski, and A. Wasilewska, 'Detection of inflammation from finger temperature profile in rheumatoid arthritis', *Med. Biol. Eng. Comput.*, vol. 57, no. 12, pp. 2629–2639, Dec. 2019, doi: 10.1007/s11517-019-02055-1.

[20] E. M. Amaral, 'Termografia como Meio de Diagnóstico Complementar da Mamografia', p. 145, 2013.

[21] I. A. Avila-Castro *et al.*, 'Thorax thermographic simulator for breast pathologies', *J. Appl. Res. Technol.*, vol. 15, no. 2, pp. 143–151, Apr. 2017, doi: 10.1016/j.jart.2017.01.008.

[22] M. Quintana, P. Carmona, and I. Fernández Cuevas, 'Infrared Thermography as a Means of Monitoring and Preventing Sports Injuries', 2017, pp. 165–198. doi: 10.4018/978-1-5225-2072-6.ch008.

[23] K. Ammer, 'Thermal imaging: A diagnostic aid for fibromyalgia?', *Thermol. Int.*, vol. 18, pp. 45–50, Apr. 2008.

[24] S. Sivanandam, M. Anburajan, B. Venkatraman, M. Menaka, and D. Sharath, 'Medical thermography: a diagnostic approach for type 2 diabetes based on non-contact infrared thermal imaging', *Endocrine*, vol. 42, no. 2, pp. 343–351, Oct. 2012, doi: 10.1007/s12020-012-9645-8.

[25] Y. Zhou *et al.*, 'Clinical evaluation of fever-screening thermography: impact of consensus guidelines and facial measurement location', *J. Biomed. Opt.*, vol. 25, no. 9, p. 097002, Sep. 2020, doi: 10.1117/1.JBO.25.9.097002.

[26] 'Thermograms-images-a-baseline-b-post-cooling-c-post-rewarming.png (850×610)'. <https://www.researchgate.net/publication/336992144/figure/fig5/AS:820950941577221@1572741365489/Thermograms-images-a-baseline-b-post-cooling-c-post-rewarming.png> (accessed Sep. 21, 2021).

[27] 'How Infrared Cameras Work'. <https://www.fluke.com/en-us/learn/blog/thermal-imaging/how-infrared-cameras-work> (accessed Sep. 21, 2021).

[28] '16OTD202_TM2.jpg (1800×1167)'. https://www.ept.ca/wp-content/uploads/2016/10/16OTD202_TM2.jpg (accessed Sep. 21, 2021).

[29] 'Thermal Camera Specs You Should Know Before Buying'. <https://www.flir.com/discover/professional-tools/thermal-camera-specs-you-should-know-before-buying/> (accessed Sep. 21, 2021).

[30] 'Company History | Teledyne FLIR'. <https://www.flir.com/about/company-history/> (accessed Sep. 21, 2021).

[31] 'Research & Development | Teledyne FLIR'. <https://www.flir.com/applications/science/> (accessed Sep. 21, 2021).

[32] A. Kirimtat, O. Krejcar, A. Selamat, and E. Herrera-Viedma, 'FLIR vs SEEK thermal cameras in biomedicine: comparative diagnosis through infrared thermography', *BMC Bioinformatics*, vol. 21, no. 2, p. 88, Mar. 2020, doi: 10.1186/s12859-020-3355-7.

[33] 'Developer Program Main Page | Teledyne FLIR'. <https://www.flir.com/developer/mobile-sdk/> (accessed Sep. 21, 2021).

[34] 'Thermography Cameras | FLIR Professional Tools | Teledyne FLIR'. <https://www.flir.com/browse/professional-tools/thermography-cameras/> (accessed Sep. 21, 2021).

[35] 'Thermal Studio Suite | Teledyne FLIR'. <https://www.flir.com/products/flir-thermal-studio-suite/> (accessed Sep. 21, 2021).

[36] 'thermalstudio-desktop-3qtrfrtft-01.png (600×625)'. <https://www.flir.com/globalassets/imported-assets/image/thermalstudio-desktop-3qtrfrtft-01.png>

01.png (accessed Sep. 21, 2021).

[37] 'ResearchIR 4 | FLIR | Thermal Imaging Software, RESEARCHIR | Acal BFi BE'.
<https://www.acalbfi.com/be/Imaging-for-Science-and-Industry/Radiometric-cameras-uncooled/p/Thermal-Imaging-Software--RESEARCHIR/0000007HAJ> (accessed Sep. 21, 2021).

[38] 'FLIR Research Studio R&D Software | Teledyne FLIR'.
<https://www.flir.com/products/flir-research-studio/> (accessed Sep. 21, 2021).

[39] 'research-studio-software.png (600×625)'.
<https://www.flir.com/globalassets/imported-assets/image/research-studio-software.png>
(accessed Sep. 21, 2021).

[40] 'FLIR Tools Mobile – Apps no Google Play'.
https://play.google.com/store/apps/details?id=com.flir.tools&hl=pt_PT&gl=US (accessed Sep. 21, 2021).

[41] 'Why developers like Firebase', *StackShare*. <https://stackshare.io/firebase> (accessed Sep. 21, 2021).

[42] 'firebase - How can I enforce database schema in Firestore?', *Stack Overflow*.
<https://stackoverflow.com/questions/53819548/how-can-i-enforce-database-schema-in-firestore> (accessed Sep. 21, 2021).

[43] 'Firebase Databases | What is the best option for your app? | Low-code backend to build modern apps', *Back4App Blog*, Oct. 10, 2020. <https://blog.back4app.com/firebase-database/> (accessed Sep. 21, 2021).

[44] 'Modelo de dados do Cloud Firestore | Firebase Documentation', *Firebase*.
<https://firebase.google.com/docs/firestore/data-model?hl=pt> (accessed Sep. 21, 2021).

[45] R. Peterson, 'PostgreSQL vs MySQL: What is the Difference?'
<https://www.guru99.com/postgresql-vs-mysql-difference.html> (accessed Sep. 21, 2021).

[46] '8 Big Advantages of Using MySQL', *Datamation*, Nov. 16, 2016.
<https://www.datamation.com/storage/8-major-advantages-of-using-mysql/> (accessed Oct. 16, 2021).

[47] '10 Advantages of NoSQL over RDBMS', *dummies*.
<https://www.dummies.com/programming/big-data/10-advantages-of-nosql-over-rdbms/>
(accessed Sep. 21, 2021).

[48] J. Freeman, 'What is JSON? A better format for data exchange', *InfoWorld*, Oct. 25, 2019. <https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html> (accessed Oct. 16, 2021).

[49] 'Firebase Storage: What It Is and How It Works | Low-code backend to build modern apps', *Back4App Blog*, Jul. 07, 2020. <https://blog.back4app.com/firebase-storage/> (accessed Oct. 16, 2021).

[50] 'Mobile Operating System Market Share Portugal', *StatCounter Global Stats*.
<https://gs.statcounter.com/os-market-share/mobile/portugal> (accessed Sep. 21, 2021).

[51] 'FLIR Camera Support from Image Acquisition Toolbox'.
<https://www.mathworks.com/hardware-support/flir.html> (accessed Oct. 16, 2021).

[52] 'Si7021-A20 I2C Humidity and Temperature Sensor', p. 36.

[53] 'Meet Android Studio', *Android Developers*.
<https://developer.android.com/studio/intro> (accessed Sep. 21, 2021).

[54] 'Firebase Realtime Database', *Firebase*.
<https://firebase.google.com/docs/database?hl=pt> (accessed Sep. 21, 2021).

[55] 'Instalação e configuração da API REST', *Firebase*.
<https://firebase.google.com/docs/database/rest/start?hl=pt> (accessed Sep. 21, 2021).

- [56] O. Postolache *et al.*, 'Tailored Virtual Reality for Smart Physiotherapy', in *2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Mar. 2019, pp. 1–6. doi: 10.1109/ATEE.2019.8724903.
- [57] D. Mindrila, P. Balentyne, and M. Ed, 'Scatterplots and Correlation', p. 14.