DEPARTMENT OF
COMPUTER SCIENCE

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DANIEL PEREIRA E SILVA
Bachelor in ⟨Computer Science and Engineering⟩

SYSTEM FOR MONITORING PHYSICAL THERAPY EXERCISES

INTERACTIVE MOBILE APPLICATION FOR MONITORING WRIST DEVICE MEASUREMENTS

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
⟨May⟩, ⟨2023⟩

DEPARTMENT OF
COMPUTER SCIENCE

SYSTEM FOR MONITORING PHYSICAL THERAPY EXERCISES

INTERACTIVE MOBILE APPLICATION FOR MONITORING WRIST DEVICE MEASUREMENTS

DANIEL PEREIRA E SILVA

Bachelor in ⟨Computer Science and Engineering⟩

Adviser: Carmen Morgado
Assistant Professor, NOVA University Lisbon

Co-adviser: Fernanda Barbosa
Assistant Professor, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
⟨May⟩, ⟨2023⟩

System for monitoring physical therapy exercises

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.7.0) [30].

Acknowledgements

I would like to thank my adviser Carmen Morgado and my co-adviser Fernanda Barbosa who gave me the opportunity to work on this project, and for all the guidance and motivation they gave me during the development of this dissertation work. I would also like to thank Physiotherapist Tabatha Vieira for having me at her clinic and for all the knowledge and feedback she shared that was very helpful for the project.

I am very grateful to the FCT NOVA's Computer Science department and to the teachers with whom I have had the privilege to cross paths throughout these academic years, and for the knowledge and experience they have shared.

To all my colleagues with whom I had the pleasure of sharing amazing experiences and staying up all night working on projects, thank you for all the moments we spent and I am very glad to have built and brought many friendships into my life.

To my friends, especially the closest ones, thank you for always supporting me and for always being there to motivate me in the most difficult moments. I am very grateful to all of you.

Lastly, I would like to give a very special thanks to my family for all their support and for always believing in me.

“Once we accept our limits, we go beyond them.”
(Albert Einstein)

Abstract

It is known that a simple repetitive motion is enough to lead to an injury to a joint. It is important to pay proper attention to the problem and seek dedicated treatment to rehabilitate the affected area, as the situation can worsen and cause serious irreversible damage. The wrist is one of the joints that suffers most from these problems, as it is subject to many repetitive tasks during everyday life, such as using a cell phone, or working on a computer. It is recommended that the rehabilitation process is carried out in the best possible way, so that recovery is quick and efficient, and it is usually necessary for physiotherapy exercises to be done daily at regular intervals, and a personal commitment from the patient and monitoring by the health professional is essential. This work aims to develop a physiotherapy exercise monitoring system for the wrist. Through an interactive mobile application, the patient can access a set of exercises proposed by the physiotherapist and benefit from an interface with real-time information regarding the movement that is being performed. The system's wearable motion-sensing prototype is worn on the wrist to collect and send data regarding the movements being performed during the exercises to the patient's mobile application. Thus, the patient is able to perform the proposed wrist rehabilitation exercises in a more controlled and correct way. On the other hand, the physiotherapist can consult the patient's progress and follow up the patient more closely. The system is evaluated for its functionality and usability of the different user interfaces, and also for its efficiency in reading the angles made with the wrist during physiotherapy movements. User stories were used to evaluate the ease of use and intuitiveness of the system, and it is of interest to carry out a questionnaire for both patients and physiotherapists in order to obtain feedback and suggestions to help improve the system.

Keywords: Wrist Movements, Motion Sensors, Wearable sensors, Mobile Computing, Interactive Mobile Application, Physiotherapy, Telemedicine.

Resumo

É de conhecimento que um simples movimento repetitivo é suficiente para causar uma lesão numa articulação. É importante prestar a devida atenção ao problema e procurar um tratamento dedicado para reabilitar a área afetada, pois a situação pode piorar e causar graves danos irreversíveis. O pulso é uma das articulações que sofre bastante com estes problemas, pois está sujeito a muitas tarefas repetitivas durante o dia-a-dia, tais como utilizar um telemóvel, ou trabalhar num computador. É recomendado que o processo de reabilitação seja realizado da melhor forma possível, para que a recuperação seja rápida e eficiente, e normalmente necessário que os exercícios de fisioterapia sejam feitos diariamente a intervalos regulares, sendo essencial um compromisso pessoal do paciente e um acompanhamento por parte do profissional de saúde. Este trabalho tem como objetivo desenvolver um sistema de monitorização de exercícios de fisioterapia para o pulso. Através de uma aplicação móvel interativa, o paciente pode aceder a um conjunto de exercícios propostos pelo fisioterapeuta e beneficiar de uma interface com informação em tempo real relativamente ao movimento que está a ser executado. O protótipo do sistema de sensor de movimento wearable é usado no pulso para recolher e enviar dados relativamente aos movimentos que estão a ser realizados durante os exercícios para a aplicação móvel do paciente. Assim, o paciente consegue realizar os exercícios propostos para a reabilitação do pulso de uma forma mais controlada e correta. Por outro lado, o fisioterapeuta pode consultar o progresso do paciente e fazer um acompanhamento mais atento do paciente. O sistema é avaliado pela sua funcionalidade e usabilidade das diferentes interfaces de utilizador, e também pela sua eficiência na leitura dos ângulos feitos com o pulso durante os movimentos de fisioterapia. Foram utilizadas histórias de utilizador para avaliar a facilidade de utilização e intuitividade do sistema, e é de interesse realizar um questionário tanto para pacientes como para fisioterapeutas, a fim de obter feedback e sugestões para ajudar a melhorar o sistema.

Palavras-chave: Movimentos de Pulso, Sensores de Movimento, Sensores Wearable, Computação Móvel, Aplicação Móvel Interativa, Fisioterapia, Telemedicina.

Contents

List of Figures

List of Tables

Acronyms

ATT Attribute Protocol 32, 33, 54

BLE Bluetooth Low Energy 30, 31, 32, 33, 37, 40, 43, 44, 50, 53, 54, 55, 57, 59, 60, 72, 85, 96

DOF degrees of freedom 7, 23

GAP Generic Access Profile 32, 33

GATT Generic Attribute Profile 32, 33, 54, 55, 56, 59

I2C Inter-Integrated Circuit 53, 56, 82

IMU inertial measurement unit 3, 6, 10, 14, 15, 19, 21, 22, 24, 27, 40, 43, 46, 69, 82

MEMS microelectromechanical systems 10, 19

MTU maximum transmission unit 33, 34, 54, 55, 88

1

Introduction

This chapter presents the motivation and context for further understanding the development of this thesis and its main goals and contributions. It will also be presented the structure of the document.

1.1 Motivation and Context

Telemedicine has increasingly become an essential key of today's society. The inovation of technology has allowed the rise of better communication and information technologies that are beginning to replace or extend in-person services by remote ones. These remote services are improving the quality of measurements taken during treatment sessions, by making them less subjective and easier to perform. Thus, an opportunity arises to explore new tools that aim to contribute to this area, in order to promote better convenience and quality of services.

1.1.1 Motivation

The Internet of Things (IoT), a network of physical objects with the characteristic of being able to connect and exchange information with other devices, has become one of the most important technologies of the century. This ability to connect and exchange information arises from the fact that these objects are embedded with sensors, software, and other technologies [34]. This composition makes them intelligent, their application covers all kinds of sectors of society, and their presence is becoming increasingly ubiquitous. Thus, communication between people, processes and things has grown, making everything more accessible and automated.

As the population grows, and consequently the inability to provide healthcare services to all, the use of telemedicine services, that is, remote healthcare services designed to prioritize health and improve quality of life by providing personalized services as needed, becomes essential. Telemedicine has many advantages, such as reducing transportation costs and congestion in medical centers, providing better distribution and management of medical services, allowing for better medical education, and promoting the absence of borders and geographic isolation, which means being able to cover a large geographical scope and share data among several centers [31].

In light of the SARS-CoV-2 pandemic, this improvement and accessibility to healthcare through remote services has been a major advantage since health services have been overcrowded and in some situations it is impossible to reach everyone. This increase, and the demand for health care support, may lead to some appointments and treatments being provided with better quality if remote services are used, where there will be a more careful patient-professional interaction.

These remote health services make the most of communication technologies, which are powerful tools capable of breaking down barriers of time and space [56], allowing patient diagnoses to be made at any location in the world, and information to be accessed, evaluated, and consequently returned via wireless communication services. These communication technologies are increasingly necessary due to their ease of use and efficiency, as they avoid the need to have more than one professional to perform manual measurements and it is also possible for these measurements to be performed automatically and outside the clinical environment.

A large number of health problems, especially within musculoskeletal systems, are caused by bad habits such as a sedentary lifestyle and excessive use of mobile phones, as well as daily accidents. This can lead to inflammatory problems in the joints and even wear and tear of the cushion between the bones, causing pain and in some cases, disability. Therefore, the practice of physiotherapy exercises for the rehabilitation of these joints becomes essential, and the need for exercise routines leads to a demand for applications that evaluate the performance and correction of these movements. Some of these applications incorporate a sensor component into the system, which collects data and monitors the patient's status through close follow-up along the entire process, and these applications may be specialized in physiotherapy movements or more interactive.

1.1.2 Context

The importance of telemedicine in society motivates for a continuos research and creation of new applications that embrace this evolution of communication technologies, new microelectromechanical systems, and even frameworks for code development. All these advances allow the development of applications that can improve or overcome existing ones.

In the scope of this thesis, there are applications that intend to improve the person's physical state, establishing a fast and efficient communication between patient and professional that promotes a greater commitment of the patient in the rehabilitation process. There are many solutions whose focus varies according to the joint they assess, for example knee or wrist, and also differ depending on the technologies used to develop the system.

Some interesting work has been developed by colleagues [29, 8, 6] and supervised by the same supervisors of this thesis that follow the same line of thought as the work done in this thesis, in which systems incorporating sensors were created within the area of physiotherapy. These works propose systems to evaluate and measure certain body movements and postures. Since the human body is quite complex, it was necessary to investigate and make an initial assessment of the possible limitations of the system and deliberate on the best places to integrate the sensor component. It is also important to evaluate the communication technologies that can be used in the system, as well as the construction of interfaces to interact with the system, and the definition and specification of the supporting architecture.

A wearable sensor-based gait analysis system for physiotherapy was developed by Tiago Lopes [29]. The aim of his work was to develop a system to facilitate the diagnosis of gait disorders and the assessment of patients' response to treatment, so it was of interest to make diagnosis and assessment easier and less error-prone for physiotherapists. The movements are evaluated by three inertial measurement unit (IMU) sensors placed on the shank of each leg and another on the pelvis, while data is sent to the application via Bluetooth through microprocessors.

The development of a system that provides diagnostic, management and appointment scheduling tools for physiotherapists and patients was developed by Luís Correia [8]. Through a web and mobile computing platform, a physiotherapist was able to monitor the progress of a patient's health status remotely by analyzing movement data collected by a sensor system. The system was composed mainly by two IMU sensors and a microprocessor, and consists of a prototype designed for elbow rehabilitation exercises.

A posture evaluation system using sensors was developed by Gabriel Botelho [6]. The work focused on a head posture evaluation system with the goal of helping physiotherapists in the diagnosis and evaluation of head anteriorization and in the follow-up of their patients. That work made possible the automatization of head posture measurements, in particular the measurement of craniovertebral and sagittal head angles, during a period of time while the patient is seated. For that the system used two IMU sensors, one placed on the neck and the other on the back, both connected to a single microprocessor that communicates with the application via Bluetooth.

The area that this thesis focuses on, telemedicine, is constantly growing, so the system that was developed in this work seeks to innovate or even optimise other solutions. The goals of this work were defined in order to achieve a correct integration and allow the system to be scalable and easily optimized.

1.2 Thesis Goals

The focus of this work is the development of an interactive mobile application that is connected with sensors to monitor physiotherapy exercises, more specifically, the musculoskeletal rehabilitation of the wrist. The movements being evaluated are four physiotherapy exercises for the wrist: flexion and extension, and also radial and ulnar deviations, which can be performed with and without load. In these movements the forearm is fixed on a surface and only the wrist joint will be rotated.

The system consists of a mobile application that has interfaces for both the patient and physiotherapist, a wearable motion-sensing prototype to collect the data, and also a server and database to manage and store the information. The physiotherapist is able to consult the patient's progress regarding the proposed exercises, and thus carry out a more rigorous professional follow-up of the patient's evolution throughout the treatment. On the other hand, the patient benefits from an interactive interface that ensures the correct practice of the exercises at the right moments. It is of interest that the system is scalable, so this application has been designed and developed in a simple and configurable way so that it can be extended to new exercises. This work aims to have a contribution in the area of physiotherapy, promoting an improvement in the quality of the patient's rehabilitation process through a system that can be used autonomously and outside a clinical environment.

Initially, it was necessary to carry out a brief study in order to know the goniometry of the wrist, that is, the angles, movements and rotations of the wrist joint, and also of other parts of the hand and forearm involved in the exercises. At the same time, an investigation of existing projects and products on the market around this area was made in order to make a comparison and take into consideration different sensors and possibilities to measure and collect the data more efficiently and correctly. There are different types of sensors capable of making these measurements, such as sensors based on resistance, force, or even conductive ink, however, some are limited to certain movements and their use does not cover all the physiotherapy exercises proposed in this work. A study was carried out to understand the advantages and limitations of some sensors, as well as where they can be positioned to provide more accurate measurements. It was also important to survey the various mechanisms for filtering and processing the data collected by the sensors in order to provide a system with greater efficiency.

After a careful research on the sensor component of this project, a functional and practical prototype was built, capable of measuring movements and transmitting the data through a communication technology. A survey of the communication technologies between the wearable motion-sensing prototype component and the mobile application was also carried out in order to implement a communication that brings more advantages to the system. It is important to have some understanding of the requirements, such as the need to send the different data collected by the sensors during the physiotherapy exercises, as well as the fact that it is convenient for the mobile application to work close to the wearable motion-sensing prototype so that the patient can have feedback on the movements being performed.

Therefore, an interactive mobile application was developed which allows the visualisation of the exercise movements measured by the wearable motion-sensing prototype component in real time, and also a server and a database were developed to manage and store all the important information that will be dynamically displayed in the patient and physiotherapist interfaces in the application. Besides helping health professionals to monitor their patients' problems, the application can also be easily used for the patient to perform physiotherapy exercises autonomously and at any time of the day. Thus, we believe that the application can be used for professional use as well as well as independently by a patient.

1.3 Document Structure

This document is divided into six chapters. The first chapter includes an introduction that explains the objectives of this work, the motivation and context that inspired its development.

The second chapter presents an investigation of various projects and applications that contain important information for furthering knowledge in the several components involved in this work. Firstly, the anatomy and goniometry of the hand and wrist is described, with the wrist part being presented in more detail as it is the focus of the study. Then, many different applications are presented with the aim of improving especially the wrist condition through physiotherapy exercises, or implicitly with the involvement of a patient in interactive games. This chapter will also present the existing products in the market that follow the same line of thought as this work, and the measurement infrastructure will be addressed.

The third chapter will present the proposal of the system to be developed, where the architecture of the system and the composition of each component that integrates the system are described.

The fourth chapter explains how the application was developed, as well as all the challenges encountered during development and the decisions made during this process.

The fifth chapter presents the tests and evaluations performed on the system and the results obtained. And finally, the sixth chapter presents the conclusions of the work developed, as well as the future developments planned for this thesis.

2

State of the Art

The evolution of technology has been increasingly allowing faster, efficient and more accessible global communication across the world. Telehealth takes advantage of this growth by making use of better digital information and communication technologies to ensure easy management and remote access to health services.

The study of joint movement in the musculoskeletal system of the human body is possible through manual measurements, such as using a goniometer, however, equipment that incorporates different types of sensors allows a reading with greater efficiency and less effort. This is the case with inertial measurement unit (IMU) sensors, which are electronic sensors that combine accelerometers, gyroscopes, and sometimes magnetometers, to measure a body's specific force, angular rate, and orientation. It is also possible to use resistance-based sensors such as gauge and flex sensors, or infrared and even creative solutions using conductive paint, touch sensor, cameras, among others.

This chapter presents several projects that served as a base study for the preparation of this thesis. The anatomy of the hand and wrist, and their gonometry will be presented and referenced. Different projects aimed at monitoring and improving the patient's condition, predicting and reducing the evolution of joint problems through exercises and interactive games will be discussed for the purpose of comparison and analysis of the technologies used.

2.1 Hand and wrist anatomy

The hand and wrist are complex and active parts of the upper extremity of the musculoskeletal system of the human body, and are vulnerable to injuries of small and large complexity that can cause potential functional difficulties and severe damage, which consequently lead to greater economic costs [61, 15]. A body injury can occur when performing a simple gesture, such as when playing sports, or when interacting with an object in everyday life, for example, when using a smartphone. Therefore, a proper understanding of the anatomy and biomechanics of these body parts is necessary for an effective treatment, and physiotherapy exercises may be strongly recommended or even essential.

Both the hand and wrist include various bones, joints, muscles, tendons, and ligaments to ensure stability and proper functioning between the different parts that are interconnected and depend on each other to allow mobility and sustainability. The point of junction between two or more bones is called a joint, and ligaments are elastic fibres that are attached to the membrane covering the bone help to strengthen and stabilise joints. The wrist is a very differentiated section of the musculoskeletal system, as shown in the figure 2.1, that has two degrees of freedom (DOF) and allows the hand to move in various directions relative to the forearm. The injuries to this joint can arise in the group of muscles involved that control the wrist, which can be extensor muscles and flexor muscles of the wrist, as in the bone structure that forms the joints, or even in the nerve tissue present in this area. These injuries, pathologies and disorders that can affect the functioning of the joints and cartilage in the medium and long term, can occur when excessive or repetitive effort is made, or even while performing simple daily tasks. It is important not to ignore these symptoms and to know how to identify the problem in order to follow an appropriate treatment.

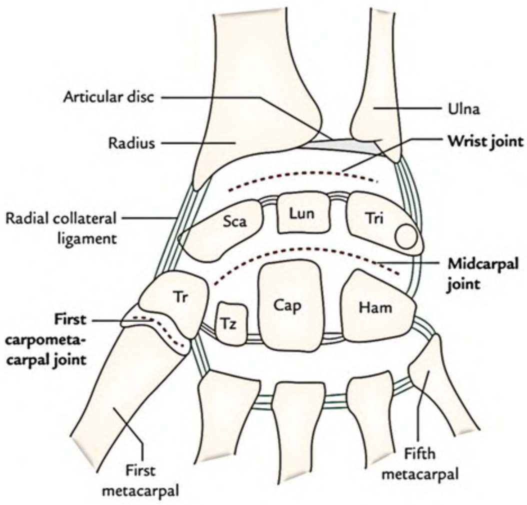


Figure 2.1: Wrist joint (radio-carpal-joint)[1]

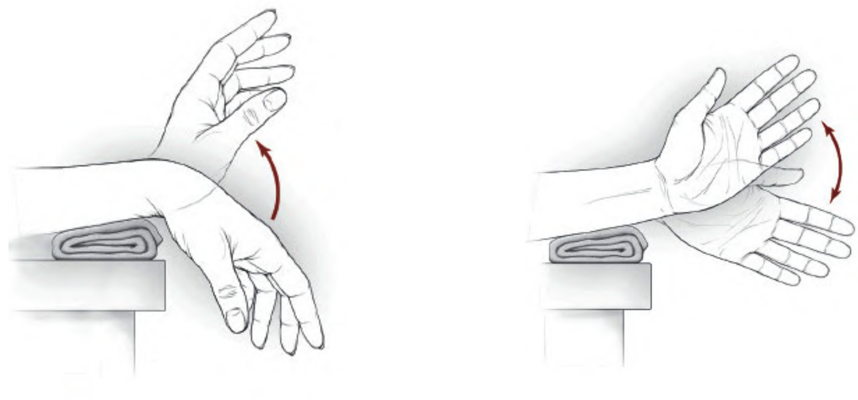
In general, these problems can include inflammation or swelling in the joints, such as arthritis, wearing away of the cushioning between a bone, causing pain in areas where the cartilaginous covering is more reduced, as is the case with arthrosis, or inflammation in the tendons or muscles caused by repetitive strain. An example of a common condition is the carpal tunnel syndrome, which is a neurological disorder that occurs in the carpal tunnel where the median nerve that travels from the forearm to the palm of the hand is pressed into the wrist and causes burning, or numbness. The scapholunate ligament is caused by a high load force that causes trauma or even ligament rupture, and the

[1]Source: https://www.earthslab.com/anatomy/wrist-joint-radio-carpal-joint-surfaces-ligaments-and-movements/

wrist capsulitis or wrist pain syndrome, which arises due to repetitive impact trauma, is quite common in gymnasts and causes diffuse pain in the dorsal wrist with heat and swelling [61].

There have been an increasing number of applications aimed at joint rehabilitation, which seek to prevent or reduce existing injuries. In some cases of patients suffering from osteoarthritis, a wear and tear on joint cartilage that can lead to deformity and disability, exercise to strengthen muscles is strongly recommended. Sometimes these applications are only informative and provide communication between the physiotherapist and the patient, and only a small percentage of these applications incorporate a sensor component that allows for greater efficiency in treatment [42]. Through monitored exercises, or even interactive games, it is possible to improve the patient's condition, as well as predict the evolution and status of the disease.

According to the physiotherapy exercises proposed as part of wrist joint rehabilitation, it is recommended to practice certain wrist movements in order to stimulate and improve the condition of the wrist joint, such as the wrist flexion and extension exercises, shown in figure 2.2(a), and the wrist radial and ulnar deviation, shown in figure 2.2(b). While performing these movements, the forearm must be fixed on a surface. The joint that allows these movements is a biaxial ellipsoid joint called the radiocarpal joint, or wrist, which is articulated with the scaphoid and semilunar joints [60], and connects the forearm and hand.



(a) (b)

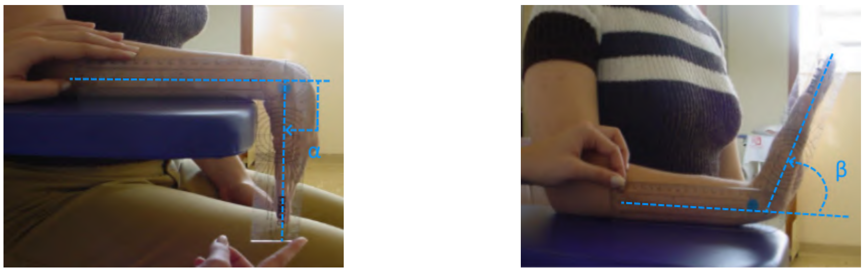
Figure 2.2: Wrist exercise movements. (a) Wrist extension and flexion. (b) Wrist ulnar/radial deviation[2]

Goniometry is an evaluation technique to make measurements diagnosis of joint range of motion, and therefore, wrist range of motion (ROM) measurements are essential when performing a physical evaluation of the wrist. These measurements can be carried out simply using manual instruments such as goniometers, inclinometers and measuring tapes. There are three known techniques for measuring wrist flexion and extension with

[2]Source: https://legacyneuro.com/hand-exercises-daily/

a manual goniometer, referred to as ulnar alignment, radial alignment, and volar/dorsal alignment. For radial and ulnar deviation, it is recommended to place the goniometer on the dorsal surface. These techniques are distinguished according to the placement of the goniometer, which can be placed on the ulnar, radial, or dorsal-volar surface of the wrist [7]. Interesting studies investigate which of the different techniques achieve the best accuracy, and in general, the results indicate that the dorsal-volar technique demonstrates the greatest reliability and consistency [26, 7].

The flexion and extension movements of the wrist occur in the sagittal plane at the radiocarpal and intercapal joints [15], which is a plane of motion that cuts the body into left and right halves, allowing forward and backward movements. In these exercises, the arm must be in pronation and the elbow bent at approximately 90º with the axis on the medial surface of the wrist. Wrist flexion consists of moving the hand downwards, in a joint range between 0º-90º, α angle illustrated in the figure figure 2.3(a). Alternatively, wrist extension will be the opposite movement, in a joint range between 0º-70º [50], β angle represented in the figure 2.3(b).

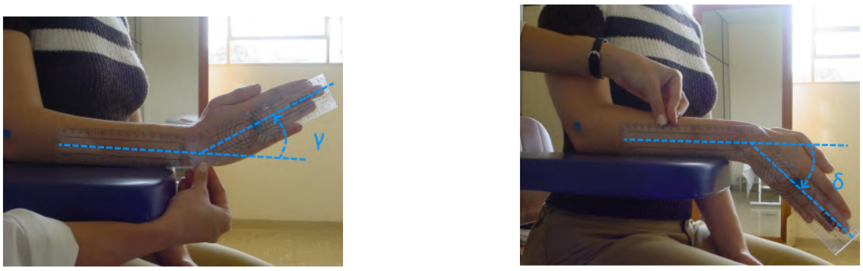


(a) (b)

Figure 2.3: Goniometer positioning for wrist flexion and extension measurement [50]. (a) Measurement of wrist flexion. (b) Measurement of wrist extension

The radial (abduction) and ulnar (adduction) deviation movements of the wrist occur in the frontal plane [15], which is a plane of motion that cuts the body into the front and back halves, allowing side-to-side movements. The elbow should be flexed and the forearm in a neutral position, that is, between supination and pronation with an axis on the radiocarpal joint, where supination of the forearm consists of rotating it upwards, while pronation is the opposite movement, rotating the forearm downwards. The radial deviation occurs when the hand is moved upwards, in a range of motion between 0º-20º, γ angle shown in the figure 2.4(a), while ulnar deviation is the opposite movement, with a range of motion between 0º and 45º angle [50], δ illustrated in the figure 2.4(b).

These are the four movements of the wrist that will be studied and evaluated in this thesis. The usage of sensors allows the reading and control of the angles in the wrist joint during the exercises, thus providing the patient with the knowledge of whether or not the proposed limits are being exceeded. The application that has been developed in this paper focuses on musculoskeletal rehabilitation of the wrist, and the system is developed



(a) (b)

Figure 2.4: Goniometer positioning for wrist ulnar and radial deviations measurement [50]. (a) Measurement of wrist radial deviation. (b) Measurement of wrist ulnar deviation

with the intention of being scalable so that the possibility of including new physiotherapy exercises for the wrist, or even new treatments for other joints, remains open.

Usually, manual measurements are more reliable, however, require more effort, rigor, and sometimes assistance from an expert. Currently, almost all smartphones incorporate accelerometers and magnetometers, and through applications, it is easy to use goniometric functions. A study concluded that using a simple application, Goniometer Pro (G-pro app)[3], it is possible to obtain excellent results compared to manual measurements [39]. However, since the mobile phone must be placed over the hand, instability issues arise and measurements may not be as accurate.

2.2 Applications for phisioteraphy exercises

It is important that a proper rehabilitation process exists so that the patient can make the best possible recovery from their problem or injury. Thus, the existence of rehabilitation systems capable of being used outside of a clinical setting, such as smart home rehabilitation assistants, are key to provide continuity of treatment [63]. This combination of technologies makes it possible to monitor the patient's condition, providing therapeutic instructions as well as information based on the data collected about the effectiveness of the treatment.

The microelectromechanical systems (MEMS) have undergone a great evolution over time, which in turn provides a great advancement in improving and optimizing different types of sensors, such as the IMU. The system developed in the work [21] integrates this type of sensor to evaluate the range of motion of the wrist, with a microcontroller running algorithms to increase the efficiency of these readings, as well as an Android application to visualize the data obtained by the sensors via Bluetooth communication. This work explores different ways to calculate the measurements of the several wrist movements, varying the positioning of the device on the back of the hand. The device is placed on the

[3]Goniometer Pro (G-pro) is an iPhone app for an easy and accurate range of motion (ROM) readings. It is available on ITunes: https://itunes.apple.com/us/app/goniometer-pro/id64692550 3?mt=8

hand at specific points, these being the front, middle, and end positions, represented in the figure 2.5, in order to identify the best location according to the exercise performed. An evaluation and comparison is made between all the readings of the device, and a comparison is also made between these collected values and the manual measurement for the respective angles.

(a) (b) (c)

Figure 2.5: Location placement of the device [21]. (a) Front Spot. (b) Middle Spot. (c) End Spot

The tests performed in the work [21] have been divided into three sections, concerning device stability, measurement tests, and device position. First, the device is placed on a flat surface and is tested at three different angles, 30°, 60° and 90° with different rotation axes, which are clockwise and counterclockwise. In a second phase, the device is placed on the back of the patient's hand, who will make the six proposed movements, while a specialist makes a comparison with manual measurements. Finally, the device will be placed again on the back of the hand, but in three different positions, front, middle and end, with a comparison made in terms of efficiency and stability. The values obtained from the movements of flexion, extension, pronation, supination, ulnar and radial deviation will be compared with manuals made by a physiotherapist, using a goniometer, which can be seen in table 2.1.

Table 2.1: Comparison between manual goniometer and wrist measurement [21]

Measurement	**Manual goniometer**	**Front spot**	**Middle spot**	**End spot**
Extension	40º-50º	59.52º	51.83º	48.86º
Flexion	35º	48.51º	38.54º	50.58º
Pronation	45º-50º	51.45º	44.24º	41.05º
Supination	50º-55º	76.38º	68.44º	63.58º
Radial deviation	10º	18.63º	12.48º	11.00º
Ulnar deviation	15º-30º	40.43º	33.14º	29.35º

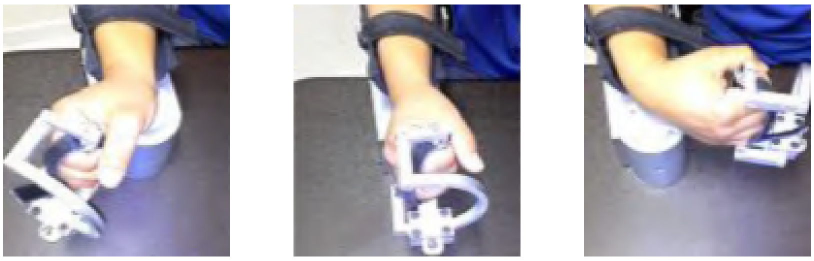
Overall, the device used in the work [21] collected fairly reasonable and acceptable data on the range of motion of the wrist. Although the device accuracy has errors within a range of 1º-5º, the data was not stable and consistent. Values may drop at certain times, even in a fixed position, and some high values appear during measurements. Best results are obtained when the device is placed on the middle and end spot of the hand. However, for radial and ulnar deviation movements, the end hand position is much more suitable, although there is a limitation in movement due to the location of the device. Other problems that occurred were, for example, in the calibration, since it takes a long time to reach zero, and the fact that the device has a somewhat bulky dimension, which ends up affecting its stability and efficiency.

Another interesting study [5] is the use of a combination of tele-rehabilitation technologies with advanced robotic technologies, allowing the development of semi-autonomous physiotherapy training with greater control and complicity between the professional and the patient. The authors of this work have developed a system that motivates the patient to exercise certain wrist movements using a rehabilitation robot controlled by a web application via Internet protocol.

The system [5] is designed with high rigidity, low friction, no backlash, and within the limits imposed by the range of motion of a healthy subject, so it may perform correct abduction-adduction and flexion-extension wrist movements, which are represented in figures 2.6 and 2.7. The electronic component is composed by an arduino board, LCD keypad shield for arduino and three potentiometers to set the rehabilitation parameters. These parameters include angles limit of each derivation, velocity and number of repetition, and also an ethernet shield for arduino that makes it possible to control the device via the web page. Thus, the professional can set the exercise parameters, date and type of exercise remotely for each patient. After the parameters are set, the professional can start the exercise and the control board actuates a servomotor so the rehabilitation robot performs the desired movement. Throughout the exercise, the professional can modify and edit the parameters in real time as well as supervise the time of remaining iterations. The tests performed proves the functionality of the system, and explains that in a local network it would be possible to trigger several synchronized exercises using only one interface, otherwise it benefits from the remote service, and based on cloud technologies, the exercises can be performed and stored for the evolution of the rehabilitation state.

(a) (b) (c)

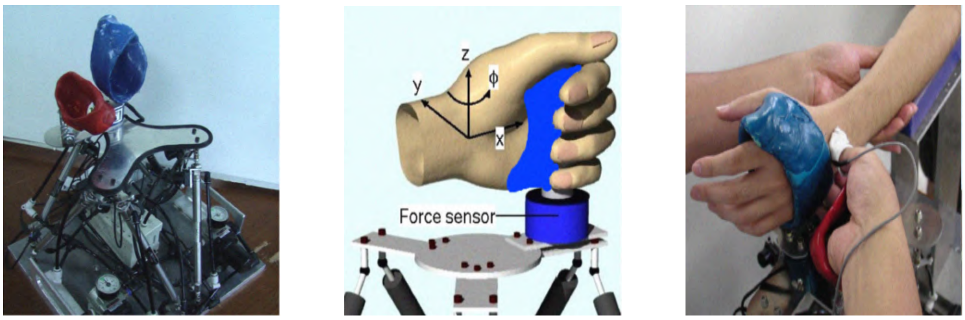
Figure 2.6: Ulnar-Radial derivation [5]. (a) Ulnar. (b) Rest. (c) Radial



(a) (b) (c)

Figure 2.7: Flexion-Extension derivation [5]. (a) Extension. (b) Rest. (c) Flexion

The combination of tele-rehabilitation technologies with advanced robotic technologies is also explored in the work [57], where the authors have developed a mechanical device to support rehabilitation movement of wrist joints. This system consists of a pneumatic parallel manipulator in which the pneumatic actuators allow minimum force control through air compressibility and the parallel manipulator's characteristic of multiple degrees of freedom is quite suitable for the complex movements performed at the wrist joint. A pneumatic parallel manipulator is a pulse rehabilitation device in which six pneumatic cylinders of the low-friction type are used as drive actuators to form the so-called Stewart-type platform, as it can be seen in the figure 2.8. This means that it has six prismatic actuators connected in pairs to three positions on the base plate of the platform, going through to three mounting points on a upper plate. A pressure sensor will detect the pressure in the chamber of each cylinder and the measurement of the displacement of a piston is done by a wire type rotary encoder.



(a) (b) (c)

Figure 2.8: Pneumatic Parallel Manipulator [57]. (a) Overview. (b) Schematic Diagram. (c) Rehabilitation exercise

The exercises carried out in the work [57] consists in the patient placing his forearm above the upper platform and performing the movements while holding a jig mechanically connected to a 6-axis force sensor mounted on an upper platform. The professional can follow the patient's movements by placing his thumb, which has a tactile sensor, on the patient's wrist bones to confirm if they are in the correct position, as well as feel the forearm in order to detect if muscle strain happens or not during a rehabilitation. Through the experiments performed, the validity of the basic concept of these control strategies were confirmed, and thus this ingenious and robust system can correctly evaluate the proposed wrist exercises.

In the projects described earlier, different types of sensors were used to evaluate wrist movements, however, in the following work [20], the focus is on hand finger rehabilitation and it is interesting because resistance-based sensors are explored. In this work, a data glove consisting of three 6-axis IMU sensors and two 2.2-inch flexible bend sensors was developed to detect the displacement of finger joints, as can be seen in figure 2.9. This system includes a graphical interface with a 3D visualization of a rectangle that moved according to the movement made by the finger, which is very helpful for the patient to get a better understanding of how the movement is being performed. In addition, the authors mention that in the future the glove will comprise all fingers of the hand, and an interactive rehabilitation game component will be developed. Although this work does not explore a solution at the wrist joint, the study and comparison between the effectiveness of both types of sensors is quite interesting and was relevant in the deliberation and choice of the wearable motion-sensing prototype of this thesis.

(a) (b)

Figure 2.9: Data Glove [20]. (a) Top view. (b) Sensor positioning

The flexible bend sensors, although more affordable in terms of price, cause the system to show poor accuracy because of the inelasticity of the plastic component of the sensor, since after being flexed for many times, it presents difficulty in returning to the initial position. On the other hand, the IMU sensors, which are also affordable, light and thin sensors, show a low error percentage (0.81% to 5.41%), which indicates a higher precision in the measurements, unlike the flexbile bend sensors that show higher error percentages (11.11% to 19.35%). The study performed with various measurements of the different angles in the finger length, concluded that the IMU sensor is the most suitable due to its higher accuracy and efficiency in the readings.

2.3 Game applications

Games are a way to capture the user's attention and interest for long periods of time, thus there are several works that use different types of sensors and explore their capabilities to build creative solutions to engage the user within the application. The fact that these applications are developed mainly with the integration of interactive games and a strong human-computer interface component, they may have limitations in their use and sometimes the measurements may be less accurate compared to other systems whose focus is especially on the evaluation and correction of movement. However, the exercises become more interesting and the patient's involvement in the process translates into positive results in their rehabilitation.

The work described in the article [45] carried out a comparative study between two commercial sensors, a Leap Motion Controller and Myo armband, according to their performance, efficiency, sensitivity and specificity. This comparison was made through an interactive game, the maze game, which requires five wrist movements to be played, these being the extension, flexion, pronation, supination and neutrality. The sensors collect information regarding the positions of the finger phalanges, electromyography signal and inertial data, and according to the article, they were chosen due to their affordable prices, as they are plug-and-play and compatible with the Unity game engine. The leap motion controller, illustrated in the figure 2.10(a), is an infrared light-based stereoscopic camera, capable of capturing hands and fingers when the space near the camera with infrared light is illuminated. It has two infrared cameras and three LEDS to track hands, fingers and finger-shaped tools. On the other hand, the Myo armband, shown in the figure 2.10(b), is a lightweight elastic armband consisting of a number of metal contacts that measure electrical activity in the muscle of the forearm. This device consists of eight electromyographic electrodes, a nine-axis IMU and a transmission module [59] capable of transmitting the gestures performed, based on the detected electromyography signals.

(a)

(b)

Figure 2.10: Leap Motion Controller and Myo armband [45]. (a) Leap Motion (b) Myo armband

The system presented in the work [45] was evaluated with eight healthy players who participated and played the same proposed game five times with each of the sensors. As said before, the developed game is the maze, which consists of moving a black ball across the map in order to catch colored cubes whose positions are fixed throughout the experience. The balls have different colors, and each color represents a certain type of movement that must be done with the wrist in order to be collected. Thus, it is intended that each participant does various types of wrist exercises until the game ends, that is, when there are no more balls and a red wall is destroyed, so that the player can place the ball in a slot. Note that, the figure 2.11 belongs to the game interface using the Leap motion controller sensor. In the case of the Myo armband, although almost identical, there is no visual feedback from the hand movement.

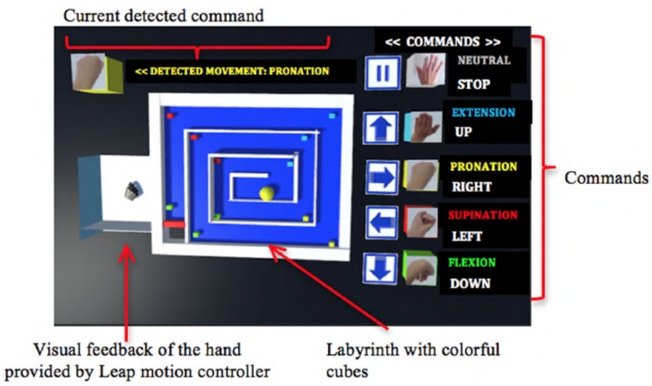


Figure 2.11: Labyrinth game interface [45]

There were performed three types of measurements in the work [45], these being precision, sensitivity and specificity. The sensitivity is the proportion of a wrist movement being correctly detected as positive, specificity is the proportion of a movement being correctly detected as negative, and lastly, accuracy consists of the proportion of movements being detected as positive and negative. At the end, these measurements were averaged for each participant's wrist movement, and in order to perform statistical analyses, the average of all participants for each measurement were computed.

Overall, both sensors effectively recognized 85% of the five proposed wrist movements, and regarding the statistical analysis, there were no differences in terms of accuracy but there are significant differences between the sensors in terms of sensitivity and specificity. The Myo armband has a better average sensitivity in extension and flexion movement, however, using the leap motion controller, better results were registered in pronation and supination movements. In neutral movements, both sensors have almost a similar average.

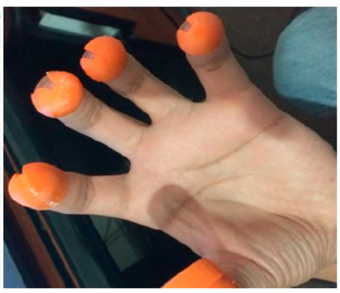
The article [45] presents a very thorough comparison of the advantages and disadvantages of both sensors. Leap motion provides visual feedback of hand movement, requires no calibration, allows identifying specific finger position, covers various types of movements, and detects movement in both hands. However, it has a problem, which is the fact that a simple flexion or extension of the finger can be interpreted as a movement of the wrist. On the other hand, the Myo armband requires calibration, is restricted to some movements, has no visual feedback of hand movement, and some problems have arisen using the left arm, which must be related to the fact that the electrode positions change on each arm. Even so, the Myo armband sensor allows the measurement of the force applied in each movement and grant tactile feedback through vibrations applied to the arm.

Another way to analyze wrist movements with games as a basis can be by using the smartphone and exploiting the features provided by its built-in sensors, as is the case of the study conducted in the work [12], whose application is called DroidGlove, shown in the figure 2.12. According to the authors, games can be used effectively in healthcare to stimulate motivation, reinforce prevention, and provide rehabilitation exercises, referred to as exergames. The system was developed for Android due to easier application development and market penetration, and the user must hold the mobile device in the hand and make the movements proposed by the application. The sensors present in the smartphone, namely accelerometers, compasses and magnetic field detectors, ensure that the motion sequence that identifies rehabilitation movements is tracked. The app automatically records when and how often the exercises were performed, as well as how correct the movements were. During the exercise, the user will have vibration feedback that will inform when movements are incorrect. Since the application stores this information, a professional will be able to consult and evaluate all the information in an appointment. The application verifies the correctness of a performed exercise by checking its compliance to a prestored motion track. In other words, the exercise is divided into stages, and the user must reach these checkpoints to validate the movement. It should be noted that the speed of the movement is calculated with accelerometers, and when the delinead limit is exceeded three times, the exercise is stopped and the user is notified.

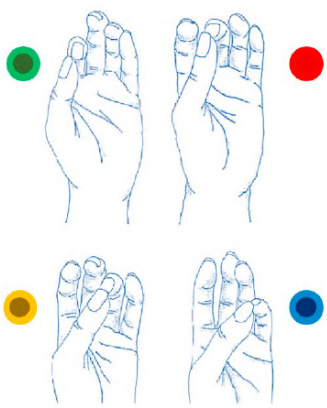
Figure 2.12: DroidGlove: Main menu activity interface [12]

However, the authors who developed the work [12], point out several problems that make this system not very feasible. Power consumption was a problem, since the mobile device has to run the application and collect data with the sensors. Problems also arise with calibration, as the data was sometimes unstable and led to uncertain results. Since ease of use is DroidGlove's strong point, the user could use the application with any rotation of the device, and therefore there was a need to create a procedure on the calibration of the sensors in order to reduce errors due to their sensitivity. Although, DroidGlove [12] is an interesting application and above all quite practical to use.

The development of safe, portable and confortable wearable devices with a wide application in different environments has been increasingly explored. The work presented in the article [52] introduces a device that fulfills these functionalities and focuses on hand rehabilitation. It is composed of five thimbles capable of acquiring pressure signals during the execution of movements guided by a video game. Essentially, it consists of an eletrocnic board, which is based on four open circuits that are closed with the finger contacts. The index, middle, ring and little fingers are independent circuits, while the thumb acts as a circuit closure using a copper foil that allows a recognition of when the fingers are pressing on the thumb. All thimbles have LEDs to identify the correct operation of the circuit closure and their color symbolizes the sequence of movements to be performed. In the figure 2.13(a) it is possible to visualize the prototype of the five thimbles placed on the fingers, and in the figure 2.13(b) a preview of how the finger movements can be performed based on colors. The thumb thimble also incorporates a MF01 force sensor to collect information on the pressure applied. Two Arduino-ATmega microcontrollers were used, one to record the number of repetitions that each finger pressed the thumb during the exercise, and the other used to establish communication between the device and the game interface, as well as indicating the sequence of movements to be executed.



(a)



(b)

Figure 2.13: 3D Printed thimbles system [52]. (a) Prototype. (b) Finger movements

The sensor response during the experiments in the work [52] was considered satisfactory, as the device did not generate any noise that would affect the interpretation of the pressure signals. The game played in the experiment was the classic game Guitar Hero, and it was verified that the volunteers who participated in the tests improved their concentration while engaging in the game. Overall, the game experience was positive, as the communication interface worked properly, and therefore the communication between the device and the game worked efficiently, with no failures in finger movement detection.

The work [3] presents a home-made wrist rehabilitation system that incorporates an interactive computer game that can be used without assistance. It consists of a 3D printed three-mouse joystick that will correspond to flexion-extension, adduction-abduction and pronation-supination movements, although adduction-abduction movements are still being implemented. The device contains an IMU, MPU-6500, which consists of a MEMS accelerometer and a MEMS gyroscope on a single chip, however only the accelerometer will be used to measure the acceleration of the wrist movement and send it to a microcontroller, which in turn will process the data and map it into angular data. The values collected by the accelerometer are used to control the mouse cursor for the computer game, and the microcontroller chosen for this system has a board capable of emulating a computer mouse using the USB protocol. The overview of the proposed mouse-like joystick wrist rehabilitation system can be seen in figure 2.14.

Figure 2.14: Overview of the mouse-like joystick wrist rehabilitation system [3]

The purpose of exploring an application based on a mouse-like joystick [3] is to take into consideration the disability of stroke patients who have difficulty in firmly grasping and moving the mouse. The structure of this rehabilitation system allows free movement of the wrist without any restriction in the patient's movement during rehabilitation. The game consists of the patient moving a ball along straight lines until it reaches different targets. These movements are done by moving the mouse cursor using the joystick, while the program saves the exact cursor positions in a text file that will later be interpreted by an Excel file that will produce these movements in graphical images with x and y axes. The different wrist exercises that the patient can do during the game are represented in the figure 2.15. This way, professionals can analyze the performance of the rehabilitation process and also change exercise methods depending on the evaluation of the results.

(a) (b) (c) (d)

Figure 2.15: Mouse-like joystick exercices [3]. (a) Flexion. (b) Extension. (c) Pronation. (d) Supination

Preliminary experimental results in the work [3], shows the usefulness of the proposed system where patients were able to move the ball to every target position. The authors point out that the surface of the system should be improved, since in some cases the trajectory is affected by its roughness. The system is interesting in the way the patient interacts with it, as the joystick-like mouse allows easy interaction, especially for patients who have major wrist limitations.

There were presented different projects that focus on physiotherapy exercises for patients with certain disabilities or joint problems, as well as systems that incorporate interactive games that seek to capture the patient's interest throughout the exercise by presenting goals to be achieved in order to complete the game. It is interesting to note how the various systems were designed and what sensors and technologies were used to conduct studies and evaluations regarding efficiency and user experience. It is important to design a system that is suitable for the functionality to be integrated, and also to ensure that the sensor component is capable of evaluating and making the necessary measurements according to the movements made by the user. Thus, it is necessary to choose sensors that are suitable for the type of movements that are intended to be performed, without disabling or limiting the integrity of the system, as well as being able to get the interest of the patient while performing the physiotherapy movements in order to improve their physical condition. Overall, the studies performed by the various works were interesting and very well done for the purposes for which they were designed, using different sensors such as inertial measurement units, flexible sensors, touch sensors, as well as other types of sensors and robust mechanisms. In the next section, some of these sensors will be discussed in more detail.

2.4 Measurement Infrastructure

Throughout this chapter it was seen that there are several applications that present different types of technologies according to their hardware and software composition, and each one of them brings advantages and disadvantages to the system. In this section a study and evaluation was made of the different technologies that can be chosen to integrate the system that was developed in this work.

2.4.1 Sensors

The use of sensors in the world of applications has greatly changed the quality of human life, as with the help of sensing systems it is possible to reduce the time to study or monitor a particular event. The dynamic use of sensors has led to an improvement and modification of existing sensors, and although its application is wide in various sectors, the monitoring of physiological parameters is one of the most important applications of sensors, as it helps to build a model related to human behaviour [40].

2.4.1.1 Inertial measurement unit

An inertial measurement unit, or IMU, is a set of measurement tools that can collect data and measure the speed, orientation and gravitational force of the device on which it has been placed [2]. It consists of accelerometer, gyroscope, and sometimes magnetometer sensors that capture different types of data, and it is possible to combine all the information using fusion algorithms to get a more precise measurement of the motion. Each sensor that composes the IMU has 3 axis, i.e. three degrees of freedom. Therefore, an IMU with the three sensors presents 9 degrees of freedom, as is the case of the SparkFun 9DoF Sensor Stick shown in the figure 2.16.

Figure 2.16: SparkFun 9DoF Sensor Stick[4]

The accelerometer measures inertial acceleration, while the gyroscope measures angular rotation and the magnetometer measures bearing magnetic direction, which can improve the gyroscope reading. Using an IMU with only one accelerometer and one gyroscope it is possible to benefit from readings without interference from the external magnetic field around the sensor when it is used very close to ferromagnetic material. On the other hand, relying only on these two sensors may not be enough to increase the measurement accuracy due to sensor noise and the gyroscope drift problem [2]. The IMU

[4]The SparkFun 9DoF Sensor Stick is an easy-to-use 9 Degrees of Freedom IMU available in Sparkfun. Source: https://learn.sparkfun.com/tutorials/9dof-sensor-stick-hookup-guide/all

can be applied in any field where motion detection is required, and is generally advantageous when used in navigation devices or as components in navigation equipment such as aviation and GPS systems. Generally, when an IMU is used in combination with a processing computer, it is called an Attitude Heading Reference System (AHRS), which includes the heading information and embedded attitude data solution unit. However, it can be prone to errors that accumulate over time, referred to as drift, so these errors need to be corrected using sensor fusion algorithms [41].

2.4.1.2 Flex sensor

A flex sensor, is a low cost and easy to use sensor that is used to measure bending or flexing [48, 40]. It can be an interesting alternative given its flexibility and because it can be applied in multiple applications in various fields due to its lightness, compactness, robustness, measurement efficiency and low power consumption, as can be seen in the figure 2.17.

Figure 2.17: Three flex sensors with different lengths [48]

The purpose of this sensor is to convert physical energy into electronic energy, that is, to convert the change in bend to an electrical resistance variation. As sensor, flex sensors work on the same principle as strain gauges, that is, bending of the material induces tension along one side of the bend radius and compression on the other side, but with a higher sensitivity. This sensor can be made of plastic and carbon, with the carbon surface being arranged on a plastic strip while the strip is turned sideways in such a way that the resistance of the sensor will change. The flex sensor also presents measurement uncertainties, consisting of a dispersion of values that indeterminates a bending angle, and which may arise from the measurement procedure, manufacturing inaccuracy or sensor placement [48]. Several solutions have been presented to reduce or avoid these uncertainties, such as making the same measurement using two sensors at the same time, or combining the respective readings with a simple multivariate regression model or adopting data fusion filters to reduce problems due to sensor noise. This type of sensors are usually placed on elastic fabric support clothing worn by users in order to allow measurements of certain parts of the body during a given movement, but mainly only for one degree of freedom of movement. It is possible to combine one or more sensors to measure multiple DOF, however it may not be feasible to combine them in the same device or prototype as current sensors may not be able to perform a given bend due to their DOF limitation.

Regarding the exercises proposed in this work, it would be very difficult to integrate this type of sensors, since it is of interest to build only one wearable motion-sensing prototype component for the different exercises, and the bending curve applied to the flex sensors during flexion and extension movements is different from the ulnar and radial deviation of the wrist, so this integration could be impossible to perform. This is due to the fact that the wrist flexion and extension movements occur in the frontal plane and the other movements are lateral, so the sensor cannot be twisted in the same way for all movements and therefore must be placed in different positions to allow for the various types of bends. Thus, inertial measurement units could benefit the prototype that is intended to be built, as they can evaluate all four wrist movements with greater accuracy, without the need for sensor re-location, or the existence of two sensors for separate measurements.

2.4.1.3 Force sensor

Another type of sensor that also has very interesting applications, is the force sensor, which is defined as a transducer designed to convert applied mechanical forces into electrical output signals whose value is measurable and can be used to indicate the magnitude of the force. Thus, the sensor is used for force detection and is usually based on a load cell, a piezoelectric device whose resistance changes under certain deformations, however, in some cases noise and environmental factors, such as temperature, can interfere with performance [62]. When this sensor is applied in microelectromechanical systems (MEMS), which is a process technology used to build small devices or integrated systems that combine mechanical and electrical components and characterized by low power consumption, light, high performance, easy mass production and integration, it is considered micro-force sensing. There are different types of force sensors with certain characteristics and different methods to perform force measurements, for example, strain gauge based force sensor which is widely used in contact force detection, piezoresistive force sensor that is used for micro force measurement and provides a direct transduction between mechanical deformations and electrical signals, piezomagnetic force sensor based on magnetoelastic effect, and also capacitive force sensor, optical force sensor, among others. An example of these sensors is the flat plate sensor, which is a type of strain gage-based force sensor, shown in the figure 2.18, which is designed for measuring force, surface pressure, and displacement

The authors of the work [58] developed rubber-coated cotton glove dipped in latex was equipped with flexion sensors on each finger to measure the position of the fingers and a force sensor made of steel plate substrate attached to the thumb that contains

[5]Source: https://www.futek.com/store/load-cells/force-sensor/flat-plate-FFP350

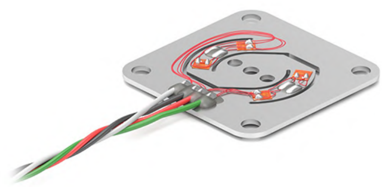


Figure 2.18: Flat plate force sensor[5]

commercial strain gauges, and therefore is designated as a strain gauge force sensor powered by a digital current source. This data glove, shown in figure 2.19, was developed to obtain information about the position of the fingers, as well as the force applied by the fingers on an object, which is classified as a tactile force, and is therefore defined as the continuous detection of variable contact forces. The force sensor consists of a steel blade with two strain gauges placed on either side, which is positioned around the thumb of the rubber glove so that when a force is applied to the steel blade its shape is changed and the voltage proportional to the force is displayed on the Wheatstone bridge that is an electrical bridge circuit used to measure resistance. Basically, the device works by a direct conversion of strain to a change in gauge resistancee, and the authors of this article consider the strain gauge possibly the most important of all mechanical transducers. The measurement principle used in this work consists of a basic strain gauge measurement technique in which a metal foil or wire is placed on the element whose strain is to be measured and, as the strain is applied, both the element and the strain gauge material suffer the same deformation. This glove was used to measure the angle of the fingers and the force applied during different interactions with a ball, from grasping or squeezing the ball at different intensities.

It is interesting to observe how these types of sensors are integrated in useful and functional systems, as well as how they interact with other types of sensors to obtain more complete information. The force sensor could be used in the prototype wearable motion sensor component developed in this thesis to incorporate tactile interaction between the patient and the system, however, the use of IMU sensors is more suitable for this system because it allows for a more correct measurement of wrist angle orientation just by placing the IMU sensor on the wrist without the need for any tactile interaction to validate the motion angles.

Figure 2.19: Data glove with force sensor and flex sensors [58]

2.4.2 Microcontrollers

A microcontroller is an integrated circuit device used to control other parts of an electronic system, and is therefore designed to perform and control specific operations in an embedded system. In general, the basic internal designs of microcontrollers are quite similar, and all components are integrated on a chip and connected via an internal bus [18]. The modules present in the composition of a microcontroller are the processor core that contains the central processing unit (CPU), volatile and non-volatile memories, a number of parallel digital and analog I/O pins, one or more timers that can be used for timestamping events or measurement intervals, various interface controllers, an interrupt controller, as well as other on-chip peripherals and, in certain cases, additional hardware to allow remote debugging. A part of the peripherals of microcontrollers are digital-to-analogue converters (DAC) which transforms a binary signal into an analogue value and analogue-to-digital converters (ADC) which takes an analogue signal and transforms it into a binary signal. The microprocessor, or CPU, performs arithmetic operations, manages the flow of data and produces control signals according to the sequence of program instructions. With regard to memory, this is often divided into program memory and data memory, where program memory is non-volatile and stores the application code and its executions as well as system constants even when there is no power, while data memory is used to store variables, and this memory can be written and read during program execution [18]. The microcontroller shown in the figure 2.20, is an ESP32 WROOM 32D[6] that was used in some initial tests of this thesis.

In order to further develop and expand IoT applications, it is necessary to look for powerful solutions with low cost and low power devices. Another important aspect is regarding the size and weight of the device, and therefore when choosing a microcontroller for a given system it is important to consider the integration of all elements on

[6]Microcontroller developed by Espressif which belongs to the ESP32 Series of Modules that are powerful Wi-Fi+Bluetooth/Bluetooth Low Energy modules. Source: https://www.botnroll.com/pt/ethernet-wi-fi/2452-placa-de-desenvolvimento-esp32-espressif.html

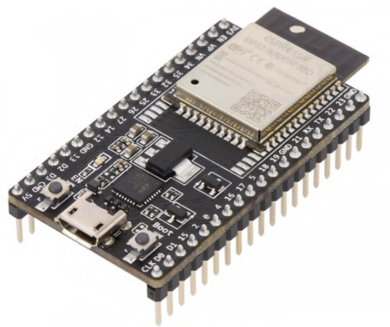


Figure 2.20: ESP32 WROOM 32D

a single chip for better space and cost management [18]. The ESP32 is a type of microcontroller, which can be integrated on different boards to have certain features to suit specific systems. The ESP32-DevKitC breadboard is an example of the ESP32 integration and presents an ideal solution for hobby and educational purposes, as well as the ESP-WROOM-32 module which provides a small solder-friendly footage, and also the ESP32 QFN48 which is the choice for industrial manufacturers and small-scale solutions [36]. The dual core structure and significant extension of operating features make the performance of the ESP32 microcontroller excellent and better than its predecessor microcontrollers.

2.4.3 IMU data

It is important to process the data collected by the sensors in order to provide the best accuracy and correctness of the readings, as well as represent the data in the most suitable format for the system. Therefore, it is possible to apply filters and run algorithms that allow the data processing to satisfy the system requirements.

2.4.3.1 Sensor fusion and filters

As seen earlier in this chapter, the gyroscope and accelerometer measure angular velocity and effective acceleration along the axis respectively, and the magnetometer measures the Earth's magnetic field which can be used to obtain orientation information in the real world, although it is affected by any magnetic or electromagnetic interference. The gyroscope works well for higher frequency movements, which are movements of short duration, as over time the data tends to drift. On the other hand, the accelerometer works well on low frequency movements, which means that the data is more accurate over long periods of time [19]. Thus, the accelerometer and gyroscope have complementary errors since they both produce errors but in different bands of the frequency spectrum and therefore can be merged to get better measurements. Also, the gyroscopoe uses earth gravity to measure the rotation about the three axis but it is prone to drift, which can be corrected using a magnetometer. Therefore, **sensor fusion** is considered a method or process to avoid and overcome the limitations of individual sensors by fusing the data from multiple sensors in order to produce more reliable information. Regarding the data fusion of an IMU sensor, there are algorithms, referred to as filters, that can be used to fuse the readings from the different sensors that compose the IMU and allow for the correction of noise and data bias.

There are several filters that can be used, for example the Complementary, Kalman, Madgwick and Mahony filters, which differ in the way they implement these corrections and estimate the orientation. The Madgwick filter, presented by Sebastian O.H. Madgwick [35] formulates the attitude estimation problem in quaternion space, which will be explained later, and performs the estimation by combining the attitude estimates from the gyroscope and the accelerometer measurements. A study was conducted to compare the Madgwick and Mahony filters [32] using a miniature inertial measurement unit (MIMU) sensor positioned on a person's foot to estimate their orientation. The results indicate that the Mahony filter takes less time to run, however the Madgwick filter performs better because the data has less noise and is closer to reality.

2.4.3.2 Data representation systems

There are different ways to represent the orientation and position of a given object or body, such as using Euler angles, which are a set of three angles, introduced by Leonhard Euler, that are intended to specify the rotations and orientation of an object in three-dimensional space in a way understandable to humans, and are therefore commonly used in user interfaces. These angles are denoted by α, β and γ, or even ψ, θ, ϕ, depending on each author. Euler angles are formed between the fixed coordinate system, denoted xyz, and the coordinate system that is rotated, denominated XYZ [4]. The definition of Euler angles can be seen in the figure 2.21. The alpha angle ($alpha$) corresponds to the angle formed between the x-axis and the N-axis, with the latter representing an orbital node, that is one of the two points at which an orbit intersects a reference plane. On the other hand, the beta angle ($beta$) corresponds to the angle formed between the z axis and the Z axis, and finally, the gamma angle ($gamma$) is the angle between the N axis and the X axis. These angles can be defined by the elementary geometry or by the composition of rotations, since it is possible to achieve any orientation objective of an object by three elementary composite rotations, that is, rotations about the axes of a coordinate system.

The elementary rotations can be extrinsic when they occur around the xyz axes of the original coordinate system, that is, the world coordinate system, which does not change. On the other hand, in an intrinsic system, each elementary rotation is made in the coordinate system as rotated by the previous operations [4]. A representation of Euler angles whose implementation is widely used in computing, is the rotation matrix. This is a 3x3 matrix, as shown in figure 2.22, and does not require an assumption regarding the order of the elementary rotation.

[7] Source: https://en.wikipedia.org/wiki/Rotation_matrix

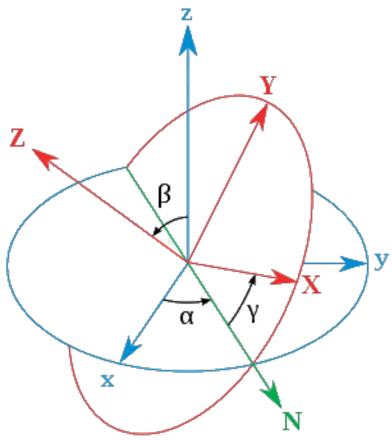


Figure 2.21: Definition of Euler Angles, showing the fixed coordinate system xyz (blue) and the rotated coordinate system XYZ (red) which is the resulting rotation over the axis vector N [4]

$$R = R_z(\alpha)\,R_y(\beta)\,R_x(\gamma) = \overset{\text{yaw}}{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{pitch}}{\begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}} \overset{\text{roll}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}}$$
$$= \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

Figure 2.22: Rotation Matrix[7]

There are other ways of representing Euler angles, such as the Tait-Bryan angle system, presented in the figure 2.23, introduced by Peter Guthrie Tait and George H. Bryan. It is used in aeronautics and engineering and the angles may be denoted by yaw, pitch, and roll, or even by heading (ψ), elevation (θ), and bank (ϕ), which are the rotations about the Z, Y, and X axes, respectively [47]. Usually the X axis is the longitudinal axis pointing to the nose of the aircraft, in turn the Y axis is the lateral axis pointing to the right wing, and finally, the Z axis is the vertical axis pointing downwards.

One of the problems associated with Euler angles is the so-called Gimbal Lock, which happens when there is a loss of one degree of freedom. This situation occurs when two axes are in a parallel configuration, belonging to the same plane. An example of this occurrence is when the pitch angle has positive or negative 90 degrees, causing the yaw and roll axes of rotation to be aligned with each other in a world coordinate system, producing the same effect [4]. Since there is no solution to this problem, an alternative is the use of quaternions.

Quaternions are a fourth dimension extension of complex numbers, invented by

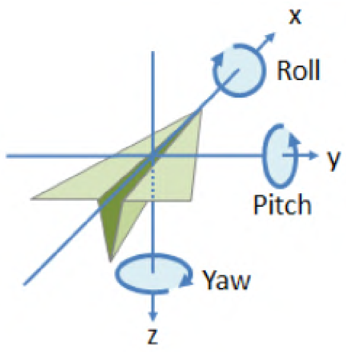


Figure 2.23: Tait–Bryan System [47]

William Hamilton in 1843, consisting of the quotient of two sets of four numbers, which is composed of a scalar part w $\in \mathbb{R}$ and a vector part v = (x,y,z) $\in \mathbb{R}^3$ that is multiplied by a set of quaternion units [i,j,k], such that $i^2 = j^2 = k^2 = ijk = -1$ [9]. Thus, the representation of a quaternion is defined by $q = ix + jy + kz + w$, having a parameterization of the rotation that does not depend on the coordinate system, so it is a very good choice for representing rotations.

The average of two quaternions can be easily calculated using a quaternion interpolation, or slerp, with an interpolation factor of 0.5 to find the rotational midpoint of the two quaternions. However, if it is necessary to average multiple quaternion values, a more complex procedure must be used, such as the maximum likelihood method, or Markley's eigendecomposition of the sum of the outer products, which gives a well-defined average [37]. Otherwise, if the set of quaternion values are close to each other, meaning that the quaternions represent very similar rotations, then averaging the quaternions followed by normalization may be sufficient, and is rather faster than the other method. In order to implement this second approach which is an approximation of Markley's method, it is necessary to ensure that the problem sometimes called the "double-coverage problem" of quaternions does not occur, which consists of two quaternions having the same rotation, one being the inverse of the other. There are some ways to "standardize" each quaternion and thus avoid this problem, such as making the dot product of a quaternion value, denoted by q1, with all subsequent quaternions and negating any of the subsequent quaternions whose dot product with the q1 quaternion is negative.

2.4.4 Communication Technologies

There are several types of wireless communication technologies in the IoT network capable of establishing communication between the microprocessor and the application, however it is necessary to understand more about the available technologies in order to choose the most viable technology and make the system more advantageous.

There is no communication protocol that is better than another because they differ in their specifications and may have more benefits depending on the application to be implemented. These protocols may be compared based on certain criteria, such as network, topology, power, range, encryption, propagation, modulation type, coexistence with mechanism and power consumption [49]. Some technologies allow two devices to be connected to each other for direct communication, and only these devices can participate in the connection. Examples of such technologies are Bluetooth, WiFi Direct, Near field communication (NFC).

Bluetooth has a relatively short operating range, is quite low-powered, and thus consumes less energy than WiFi and other types of communication technologies. Alternatively, Bluetooth Low Energy (BLE) technology is designed to transmit/receive small amounts of data while consuming low levels of power, so it has many uses such as transmitting sensor data. Zigbee is another short-range communication technology that consumes very low power and provides mesh networking [49].

It is also possible to establish communication between devices through local area network (LAN) technologies, using WiFi, or via long-distance cellular technologies, such as GSM/GPRS, LTE, among others. WiFi is known as a LAN technology for its moderate coverage area. This technology is cheap, easy to implement, has a good operating range, is widely available, but requires Internet access and must be used near a WiFi access point. Although it has several advantages, the fact that it consumes a lot of power can be a problem when applied to a system that is used for long periods of time.

Nowadays, new generation protocols have emerged, such as Wi-Fi 6, which is much faster than the previous Wi-Fi protocol and performs better in congested areas. The 5G cellular technology also has numerous advantages over previous technologies such as GSM, 3G, 4G and LTE, with these advantages being a higher cellular data transmission speed and support for huge amounts of data transfer, however, the cost is higher.

A very interesting study [33] has been conducted that makes a comparison between three communication technologies, Bluetooth, Wi-Fi and GSM, in terms of transmission delay, throughput, cost, range and infrastructure. In this work, an Internet of Medical Things (IoMT) devices framework has been proposed that consists of a notification and conversation system to monitor patient status with an application with two interfaces, one for doctors and one for patients. The patients will have at their disposal five types of sensors and three types of communication technologies. The Wi-Fi module is used to transmit data from a humidity sensor, the Bluetooth module will be connected to a heart rate sensor, and finally the GSM module also transmits data from the heart rate sensor and two other sensors, which are a temperature sensor and an alcohol sensor.

Bluetooth and Wi-Fi technologies fit better in this application [33], since they are recommended for similar networks, and in this case, it is not necessary to cover a very high distance. However, Bluetooth is much more recommended in a network where there is no Internet access and when the system is located in the same area. Even though Wi-Fi supports high data rates, it suffers from high transmission delays, depends on internet access, and is limited to infrastructure and cost. Finally, the application does not benefit from GSM technology, since there is no need to support long distances. Also, this technology requires the existence of a mobile device infrastructure, has low data rates, and the cost of transmission is high. The table 2.2 presents the results of the study regarding the comparison made between the different technologies.

Table 2.2: A comparative study between the three technologies [33]

	GSM	Wi-Fi	Bluetooth
Transmission Delay	Moderate	High	Low
Throughput	Low	High	Moderate
Cost	High	Moderate	Low
Distance	Very long	Moderate	Limited
Infrastructure	Cellular networks is required	Routing and interned are required	Not Required

It is also important to take into account other types of comparison between communication technologies, such as consumption and the number of devices that can be interconnected. In the work [10] a comparison was made between Zigbee, Z-Wave, Wi-Fi, and Bluetooth wireless technologies based on the constraints of power consumption, usability, range, cost, scalability and interoperability, to identify which is most appropriate for home automation. It is especially interesting to observe the performance of Bluetooth and Wi-Fi, as these are probably the most suitable technologies for the system to be developed in the work of this thesis, and both consumption and the number of devices that can be interconnected are essential considerations to keep in mind.

In the study carried out by the authors of the work [10], Z-Wave technology proved to be the most suitable technology in terms of scalability and interoperability as it allows communication between a huge number of devices from different manufacturers, has low power consumption, yet is expensive. On the other hand, ZigBee technology does not allow communication between devices from different manufacturers. Finally, Bluetooth and Wi-Fi are easy-to-use communication technologies as they allow connecting most devices in the Internet of Things. Wi-Fi is advantageous over Bluetooth as it allows a longer communication distance, and works better when more devices are used in the communication system. However, Bluetooth is a better solution if a long communication distance is not required and if not so many devices are needed in the communication, as it is also a cheaper technology, consumes less energy and does not require an internet connection.

2.4.4.1 Bluetooth Low Energy

The BLE was released in 2010 as part of version 4.0 of the Bluetooth specification and is a separate protocol from Bluetooth, known as Bluetooth Classic, and both protocols are not compatible. BLE provides functionalities in its specification to offer different topologies, such as connection-oriented communication, broadcast communication and large-scale device or mesh networks [25]. In connection-oriented (one-to-one) communication, one device scans for other BLE devices to connect to, which is called the central device, while another device, referred to as the peripheral device, attempts to establish a connection with the central device. In broadcast communication mode (one-to-many), the devices that are transmitters will send connection-less advertisements and the devices referred to as observers will scan these advertisements. On the other hand, the mesh communication mode (many-to-many) is a flood-based network that interconnects all the devices in the network

The BLE protocol stack, presented in figure 2.24, is divided into the application, host and controller categories. The controller protocol layers are at the bottom of the stack, and communicate with the host protocol layers, which are at the top, through the host controller interface [13]. This interface is an optional part of the stack and provides a standardised communication interface between the host and the controller. The application is simply software that acts as an interface for the user to access the functionality of the BLE protocol.

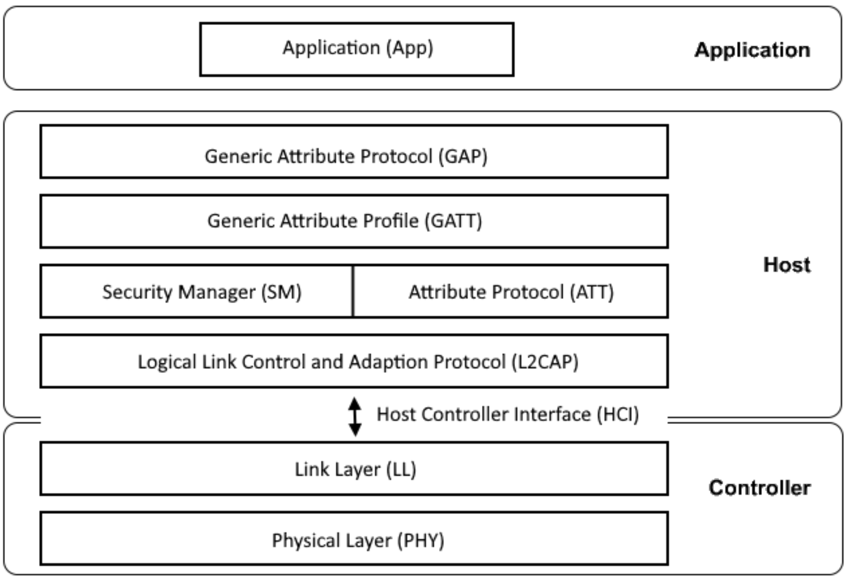


Figure 2.24: BLE protocol stack[8]

The information flowing through a BLE communication is sent using Link Layer (LL) packets and the layer above is the L2CAP. There are only a few L2CAP channels used for BLE, and one of them is Attribute Protocol (ATT) [25]. For a better understanding of how the communication between the microprocessor and the application is done through this communication protocol, it is necessary to understand what the BLE protocol layers are and how they work, these being the Attribute Protocol (ATT), the Generic Attribute Profile (GATT) and the Generic Access Profile (GAP).

[8]Adapted from "https://punchthrough.com/ble-throughput-part-4/"and [13]

The GAP defines the overall topology of the BLE network stack and specifies how BLE devices can make themselves available and communicate directly with each other, using broadcasting and connecting mechanisms [13]. The concepts of one device announcing its presence and the other initiating the connection are part of these GAP definitions. Once the connection is established, the devices have a persistent and synchronized bidirectional data transfer.

The ATT establishes roles based on a client-server architecture, these being the server and the client [11]. This protocol defines a data structure called attribute as a standard method to effectively organise the data that is stored, accessed and updated on the server. It has a hierarchical structure of services which contain characteristics where data are stored, and it is also responsible for the transfer of attributes through methods of reading and writing the data. In general, the ATT is responsible for transmitting data between a peripheral and a central device and also for storing data in a structured attribute table [55].

The GATT protocol layer uses and acts on top of the Attribute Protocol (ATT), and describes in detail how attributes, i.e. data, are formatted, packaged and transferred when BLE devices are connected. It is also responsible for defining a hierarchical data structure that shows the relation or connection between the data stored in an ATT server [55].

It is important to note some terminology in the GATT structure as shown in the figure 2.25, such as services, features and descriptors [55]. The profile contains services for a particular use that the server supports. Services are mainly used for organisational purposes, and contain characteristics. A characteristic is the basic unit of storage and is defined by its value, which is the data that will be stored, and a descriptor which is optional and contains information about the value of the characteristic. GATT uses ATT's client-server architecture, and defines the roles of GATT Server and GATT Client, which can be implemented individually or simultaneously by BLE devices.

The GATT Server contains one or more GATT Services, and a service encapsulates zero or more characteristics. In turn, a characteristic encapsulates at least two attributes, these being the characteristic declaration which contains metadata about the characteristic itself and the value attribute which contains the user data [13]. The characteristic metadata can be the properties of the characteristic which are the GATT operations allowed to execute on the characteristic value, such as notify, write, and read operations. On the other hand, the GATT Client is interested in accessing the data on the GATT server, so it will have to discover the available services, read and write to the characteristic, as well as negotiate the ATT MTU, which is the maximum length of an ATT packet.

The BLE was introduced with the data field limited to a maximum of 27 bytes, with 4 bytes for the L2CAP header, which corresponds to the channel ID and packet length, and 3 bytes for the ATT header, leaving 20 bytes for the attribute data. The maximum transmission unit (MTU) parameter in the Bluetooth stack defines the size of the attribute

[9] Adapted from "https://novelbits.io/bluetooth-le-att-gatt-explained-connection-oriented-communication/".

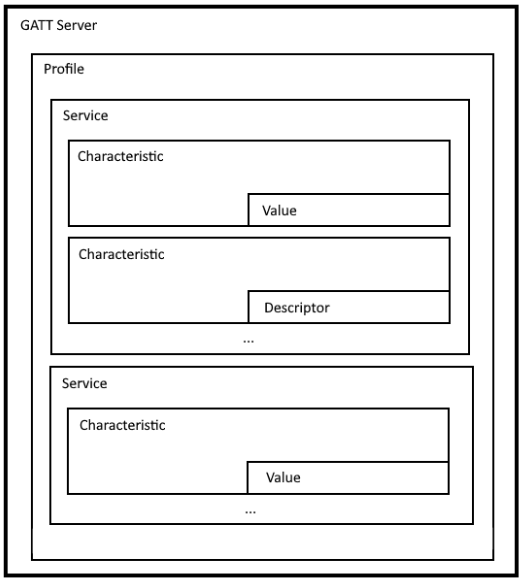


Figure 2.25: Hierarchical relationship of GATT data[9]

data, which by default the MTU size is 23 bytes and that means a payload length of 27 bytes [55]. If the data to be sent is larger in size than it can fit in the packet, then fragmentation is performed at the L2CAP level and the data is divided in different packets. Alternately, there is a feature designed as Data Length Extension (DLE), introduced in Bluetooth 4.2, which allows the payload to be increased from 27 to 251 bytes, and then the MTU will decide the actual data size of the attribute data. Therefore, if it is required a higher MTU value than the default one, it will be necessary to have an agreement between both devices, i.e. negotiate the MTU value, which will be the lowest negotiated value. The composition of the packet structure can be seen in figure 2.26.

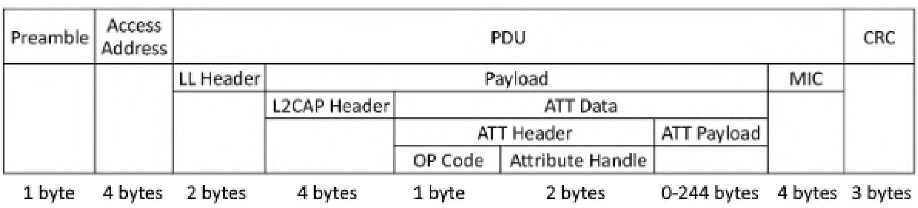


Figure 2.26: BLE Packet Structure[10]

[10]Adapted from [55]

2.4.5 Interactive Mobile Application

It is important to consider some aspects when developing mobile applications, such as the design of the user interface, as it has to be understandable and easy for the user to interact with the system, as well as the efficient use of the device capabilities, such as several sensors and network interfaces [28]. There are two types of mobile application development, which are native or hybrid application, and these can differ in performance cost and development cost.

2.4.5.1 Native and hybrid applications

The development of mobile applications needs dedicated integrated development environments, and if the application is native then it is built for a specific platform and does not work on other platforms as it depends on the execution of the operating system and its structure [28]. Thus, an application developed with the Android framework cannot work on iOS systems, and vice versa. Native applications have a good performance, since there is an ease of access to the resources of the mobile system, however, the development time is longer, since it is necessary to develop differently the same application for the different platforms.

In case the application is hybrid, it can run on different platforms with only a single development of the code with browser supported language, such as HTML, CSS and JavaScript, wrapped as a native application, and this is possible because the application uses the device's browser engine to render and process the code [28]. The use of this type of application makes the design and implementation of the user interface relatively easy and fast, in the sense that hybrid applications have cross-platform characteristics of the web application, and this way developers can mainly focus on the performance of the application [43]. On the other hand, hybrid applications have limitations in accessing mobile device features, such as sensor devices and local storage, but that through the web-to-native abstraction layer it is possible to access these features, although there is an additional overhead that compromises performance. Another problem is that the code used in the implementation is heavier than in native development.

In the comparative study between these two types of mobile application development performed by the authors of some works [28, 43], it was identified that native applications currently have longer interface rendering times than native applications, as there is also network overhead under certain conditions, and consume much more mobile phone resources during execution. Thus, the native application paradigm is preferable when application performance is a high priority, while hybrid applications are great for fast development and code reuse across different platform. It should be noted that the performance of hybrid applications can be improved as web technologies and device capabilities are enhanced over time.

There are different ways to develop hybrid applications, such as cross-platform frameworks that allow for a fast and efficient application development process. There are several frameworks for the development of hybrid applications, such as React Native[11], Flutter[12], Cordova[13], Ionic[14], among others, where the choice criteria focuses on the maturity of the framework itself, in other words, whether it is sufficient to meet the development requirements, as well as the developer's knowledge regarding the programming languages and mechanisms used in these frameworks.

2.4.5.2 React Native

React Native is a JavaScript framework created by Facebook for cross-platform application development and uses the same methodology and concepts as React, which is also a Facebook JavaScript library but targets the browser rather than mobile platforms since it is designed to build user interfaces on web pages [14]. Therefore, the React Native framework allows the development of the same application and an efficient code reuse for the different Android, iOS and Web platforms, and this way it is possible to develop robust applications fast and without compromising the quality of the application and the user experience.

React uses a Virtual Document Object Model, or Virtual DOM, which has enormous power in its abstraction layer and significant performance optimization because this in-memory version of the DOM allows only the rendering of the required elements and interactions on a page rather than loading the entire page. This same abstraction logic provided by the Virtual DOM is used in React Native, where state differences and component changes are performed in this layer, and then bridges are created to perform the rendering process that act as translators of JavaScript calls and invoke APIs and UI elements of the respective platform being used, such as invoking Objective-C APIs to render iOS components, or Java APIs to render Android components [14]. Both React and React Native use JSX which is JavaScript Syntax Extension or JavaScript XML that allows to efficiently structure and produce components that are written in HTML elements in JavaScript. This way, it is possible to combine markup, control logic and styling into one language, and although the logic used is very similar between React and React Native, there are significant differences in how component elements are structured and manipulated. Since it is intended to reuse code across different platforms, it is important to understand how to separate platform-specific components and when logic encapsulated in certain components can be shared.

React Native promotes easy integration of several features of mobile devices with a given operating system, however, sometimes there may not be support for some features of a platform so it is necessary to make or search for existing support implementations in the community. Thus, it is important to make decisions regarding the structuring and

[11] React Native is a Javascript library created by Facebook

[12] Flutter is an open source user interface development kit created by Google based on the Dart programming language

[13] Apache Cordova is a mobile application development framework created by Nitobi that was later purchased by Adobe Systems

[14] Ionic is an open source SDK for developing hybrid mobile applications

development of the project according to the needs and functionalities to be implemented. A React Native project can be started with or without the support of the Expo tool, which consists of a framework that runs on top of React Native, and allows for the fast development of Android, iOS and web apps without the need to write native code [44]. However, one disadvantage of using this Expo tool is that the developer is limited to the existing libraries from the Expo framework community and Expo does not support all iOS and Android APIs, for example there is no features to support BLE communication. On the other hand, Expo has the best 3D animation libraries, such as, expo-gl and expo-three libraries, which are used by other great 3D animation support libraries in React Native, such as React Three Fiber, that indirectly depends on these libraries in most of its available versions. There are different ways to create and develop a project in React Native, which differ in the development workflow. A React Native project with a bare workflow is a project that has access to the Expo libraries but does not use this framework for project management and allows the developer to have full management control alongside the complexity that comes with it. This bare workflow can be achieved by starting a React Native project with managed workflow, consisting of a project with the Expo framework that will initially be responsible for managing much of the complexity associated with the project, and then eject application that will separate the different native components in the project and allow the developer to have full control of the project management. This way, it is possible to have a project structure that allows the development of applications using libraries present in the Expo tool and other native libraries for React Native.

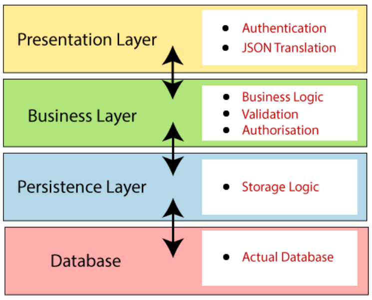
2.4.6 Server and Database

It is intended to store and access different data, including values collected by sensors, so it is necessary to choose a suitable infrastructure to support this application.

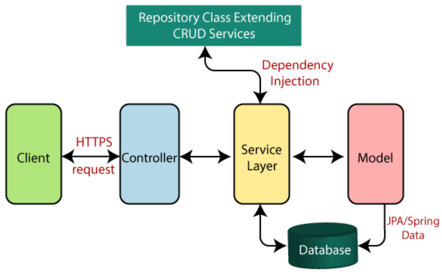
2.4.6.1 Server

There are different ways to host a server, either on a local machine or using cloud computing services. Cloud hosting has many advantages, especially as the hardware and some configurations are supported by the providers. The services offered by the cloud have benefits such as scalability, reliability, efficiency, and also storage space can be purchased when needed, so the cost of hardware is much lower in the short term compared to local system storage [1]. Data can be accessed anywhere, however, application availability may be constrained by the provider's infrastructure, that is, the hardware and software management of cloud services. In addition, cloud storage is more secure than local storage as it has several levels of security and in case of system failure it is still possible to access files through recovery mechanisms. On the other hand, in local hosting, if there are no backups on other machines, the data is lost if the server is damaged. Local hosting allows full control of backups and greater control over who accesses system data, and data storage can be faster than cloud storage if configured and set up correctly. In cloud computing data files can still be accessed without an Internet connection, unlike cloud services which all server information and data is only accessed with an Internet connection.

There are several technologies to develop a server, and it is important to choose one that is the most suitable to implement the system. Spring is a framework used for creating web applications, and may be developed using Spring Boot or Spring MVC modules, which extend the functionality of this framework [53]. Spring Boot is a framework used for RESP API development, built on Spring, and allows a quick bootstrap and start of the development of Spring-based applications, avoiding a large configuration complexity. Unlike Spring MVC, Spring Boot features default configurations and makes it easier to manage a Spring project, reducing development time and increasing productivity.



(a)



(b)

Figure 2.27: Spring Boot Architecture [22]. (a) Spring Boot Layers. (b) Spring Boot Flow

The architecture of Spring Boot can be divided into four layers, as presented in figure 2.27. The Presentation layer consists of the front end, where HTTP requests are handled, JSON parameters are translated to objects, requests are authenticated and moved to the Business layer. This next layer, besides performing authorizations and validations, is essentially for business logic, in other words, it consists of service classes and uses the services provided through the data access layers. The Persistence layer presents all the storage logic, being responsible for translating objects coming from the Business layer to database rows, and vice versa. Finally, the Database layer performs CRUD operations, that is, create, retrieve, update and delete operations.

It is important to note two types of data representation, which exist to distinguish between abstractions for manipulating and handling data, and the underlying storage system. A data transfer object, or the abbreviation DTO, is a structured data object used to transfer data between application classes and models. On the other hand, data access object, or DAO, is a class that has access to CRUD operations to manipulate persistent data values in the database.

The server's security layer and authentication mechanisms can be implemented in different ways, such as with the use of Json Web Token, abbreviated JWT. JWT is an encoding standard for tokens that contain a JSON data payload which can be signed and encrypted [23], and one of its uses is bearer tokens, which are the predominant type of access tokens used with OAuth 2.0. Bearer tokens are security tokens used in an HTTP authentication scheme called bearer authentication, or also token authentication, that grant access on a service to the bearer of the token, referred to as the token's bearer [24]. This token type is a cryptic string, created by the server in response to a successful authentication request, and can be included in different ways in an HTTP request. One commonly used way, and the one used in this work, is through the Authorization header, but it could also be included in a request parameter, a cookie, or even within the body of the request. A JWT contains claims, which are pieces of information or statements about a particular subject that must be either set to be globally unique or to be locally unique in the context of the emitter. This subject allows the principle of a JWT to be identified, which corresponds to the user for whom the token was created.

2.4.6.2 Database

There are several types of databases, such as relational and non-relational, which have different structures and storage architectures, and therefore differ in data manipulation and access mechanisms, as well as in data availability and consistency. It is necessary to ensure that data can be inserted without any restriction and that reading is consistent and up-to-date. Another important consideration is where to host the database, as this will involve costs, security, availability, and other aspects that could compromise the functionality of the application.

An example of a non-relational database is MongoDB, which is a document database, and incorporates high availability through replication and failover, geographically distributed, as well as horizontal scalability with native fragmentation [38]. This database stores flexible JSON-like documents, so the data structure can change over time, just as fields can change between documents.

Although this type of database is not recommended when complex relations and queries are required in the system, MongoDB allows ad hoc queries that are single queries or requests for a database written in SQL, which are not included in a stored procedure and are not parameterized or prepared. It is also possible to use indexing and real-time aggregation to optimize data access and analysis, and there are management tools for automation, monitoring and backup.

2.5 Summary

As discussed throughout the research done in this chapter, telemedicine is constantly growing due to the great evolution of technologies, which allows solving many health-related problems through remote communication technologies. On the other hand, technological evolution also promotes sedentary lifestyles, which is harmful to health and causes many joint problems, as is the case with the wrist joint.

There are different sensors capable of measuring the orientation of a body, however the IMU sensor seems to be a better choice for the wearable motion-sensing prototype component in this system given the versatility in its use for the different physiotherapy movements proposed in this work, which are flexion, extension, ulnar and radial deviation of the wrist. It is necessary to consider the placement position of these sensors, in order to provide a more correct reading and more comfort during exercises. It is also important to structure and implement algorithms for processing and fusing the data collected by the sensors, which will be executed by the microcontroller. These filters are essential to merge the different readings to provide a more accurate orientation estimate, reducing significant noise, bias and data drift, as well as allowing the use of quaternions. Quaternions represent an orientation in a four-dimensional space, thus avoiding the gimbal-locking problem that occurs in Euler systems, which is the loss of a degree of freedom in a three-dimensional space.

It is important to choose a communication technology that suits the system and can bring the most benefits in order to establish the connection and transfer data between the wearable motion-sensing prototype component and the application. BLE is an excellent choice for the system in this work because it is ideal for transmitting and receiving small amounts of data, consumes low power, and works over short distances. This communication technology satisfies the system requirements, since it is of interest to keep the wearable motion-sensing prototype running during exercise time intervals to send the processed data from the sensors to the application over a relatively short distance.

Finally, regarding the server and database component, the use of a non-relational database is reasonable, since there is no need to perform complex queries and it is intended to store a lot of data regarding the patient's progress during the exercises without a great relation between the data.

3

System Proposal

This chapter presents the proposed solution to clarify the goals and structure of the developed work. The motivation and system requirements to build an useful and easy to use system for the monitoring of physiotherapy wrist exercises will be addressed. Then, the design and structure of the system will also be addressed, by explaining how resource management and communication between the different parts of the system are achieved in order to provide the necessary information that will be used by the distinct types of users in the system. Also, the wearable motion-sensing prototype component will be adressed.

3.1 Overview

The system consists of an interactive mobile application that through the use of simple sensors is capable of measuring and monitoring physiotherapy wrist exercises. The purpose of the system is to provide a better and faster recovery for the patient, since the application provides interfaces to access and manage physiotherapy exercises, as well as to perform each exercise in a controlled way with a visual feedback about the progress. It can be used with the assistance of a physiotherapist, as well as autonomously and independently. Said that, this system can be used on a daily basis and outside of a clinical setting. It is intended to be developed at low cost to be affordable for all patients, comfortable and easy to use so all the exercises can be done without too much difficulty. The application provides a great interactivity between the physiotherapist and the pacient for a better experience in the patient's rehabilitation process.

The sensors used in the wearable motion-sensing prototype component will measure the wrist movements and, after the data is collected and properly processed, it will be stored in the system in such a way that it can be displayed in the application. Therefore, it is necessary to ensure that the system meets the requirements, so that the patient has a good experience when interacting with the system, and also the physiotherapist can do a proper management and evaluation of the patients.

The physiotherapy movements for the wrist that are proposed in this work are the wrist flexion, extension, ulnar deviation and radial deviation movements. The system supports that these movements can be performed by both the right and left wrist.

The physiotherapist can create and manage training plans for the patients in the system. A training plan can have several physiotherapy exercises of these different types of movements, and each exercise is composed of sets with a number of repetitions that must be performed. Each repetition has an objective angle to reach and time intervals to fulfil in order to validate the angle, and thus allowing the exercise and consequently the plan to be successfully completed.

Both the patient and the physiotherapist can see the exercise information through different interfaces. The patient can view these values in real time which are displayed on a graphical and intuitive interface, in order to be aware of the correctness and intervals at which to perform the movements. On the other hand, the physiotherapist accesses the information regarding the exercises performed by the patient once they have been completed, allowing a follow-up of the patient's progress for further evaluation. The displayed data should be clear and concise.

The server and database component, will be running in a local environment during the testing phase, however, it will be left open the possibility of hosting the system on a cloud platform to process and store all the data.

3.2 System Architecture

The system consists of three major components, which are the **server and database component**, the **application component** and the **wearable motion-sensing prototype component**. The architecture of the system and the communication established between the several components can be seen in the figure 3.1.

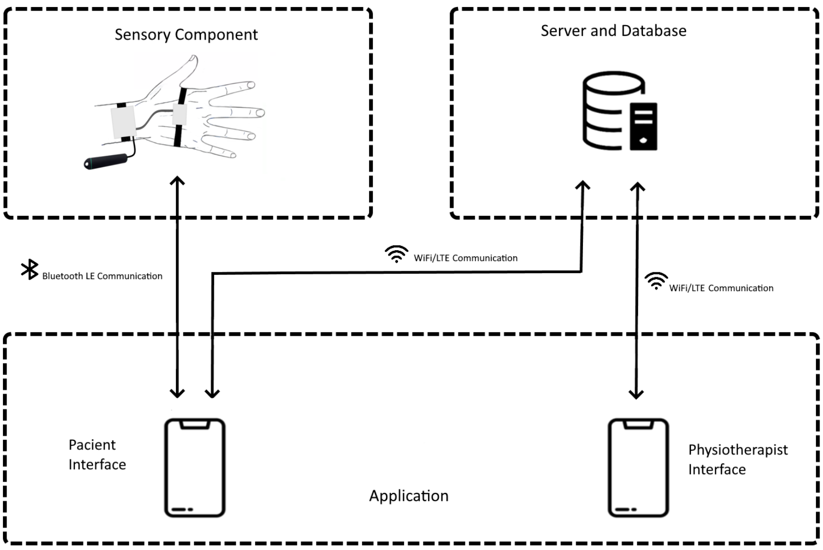


Figure 3.1: System Architecture Overview

The application component has distinct interfaces for different types of users, namely the patient and the physiotherapist. The patient interface allows the application to connect to the wearable motion-sensing prototype component in order to receive the values captured by the sensors during the exercise, as well as establish a communication with the server. The physiotherapist interface allow a communication with the server to access and manage the data related to the patients, training plans, exercises and the patient's rehabilitation process.

In general, the application communicates with both wearable motion-sensing prototype component and the server and database component. The connection stablished between the application and the wearable motion-sensing prototype component is achieved with the BLE. This technology, as seen in the previous chapter, allows a connection with a certain range between two devices, and no internet connection is required. Also, it has the advantage of consuming low power, so it is suitable for systems that use sensors which may run for a long time.

It is important to evaluate and choose the technologies and libraries to implement in the system in order to bring the most advantages and to ensure that the various components that compose the system are compatible and that there are no limitations on the functionalities expected in the system. It was created a simple application with React Native, which is a framework that allows the development of hybrid applications, and some libraries have been integrated to help establish a communication with the microcontroller connected to an IMU. The results were positive, and it was possible to observe the raw data collected by the sensor in the application. A study of potential libraries to integrate a graphical component of 3D motion visualization into the application was also made, so it would be possible to have a better monitoring of the exercises.

Overall the communication between both application component and the server and database component requires an internet connection, such as WiFi. The server and database are hosted in a local machine but in a later stage a cloud integration could be an option.

3.3 User interfaces and requirements

The system can be used in two different ways, from a patient perspective and a health professional perspective, that is, the physiotherapist. This way, it is necessary to ensure some necessary **features and requirements** for the different types of use of each user.

3.3.1 Pacient interface

The patient will use the system with the main objective of performing the physiotherapy exercises proposed by the physiotherapists. Therefore, it is necessary to ensure that the application is easy to use so that the patient has no problems using it and is able to do the routine exercises. The patient's pages in this application are more complex because it includes interfaces to establish connection between the mobile device and the wearable motion-sensing prototype component. The patient's application also has an intuitive interface with a visual feedback to guide the patient through all the exercises. Thus, the patient interface in this application should integrate some **functionalities and requirements** in order to provide the best experience in using the system.

View and start a training plan. The patient should be able to view all the training plans, as well as start one of the plans that is available. When a training plan is initialized, the patient can visualise and perform all the plan's proposed exercises. To complete the training plan all exercises must be performed.

Connect with the wearable motion-sensing prototype component. To perform an exercise, the patient must properly use wearable motion-sensing prototype on the wrist, and through an intuitive interface in the application, a connection via BLE must be established between the microcontroller and the mobile device, as well as the calibration of the sensors. After this procedure, the patient can start the physiotherapy exercise.

Perform a physiotherapy exercise. When the patient is performing the proposed exercise, an interative interface is displayed with coloured bars, messages, timers and sound effects to indicate the correctness and duration of the movements. Two different views are available, these being an interactive circular graph that increases and decreases the size of the bar, as well as changes color accordingly to the angle that is made by the wrist during the movement in real time, and also a 3D model of wrist that is rotated along XYZ axis according to the angle being performed. Once all the physiotherapy exercises, and therefore the training plan, are completed, all data regarding progress during the plan is persistently stored for the physiotherapist to consult it.

The application will provide the patient with better control over the rehabilitation process, allowing the patient to view the plans already completed, as well as others that are available or unavailable to be performed on specific days. Some mockups of patient interfaces in this application can be seen in the figure 3.2.

3.3.2 Physiotherapist interface

The physiotherapist is responsible for registering plans with several physiotherapy exercises for the patient to perform, as well as accessing information on the progress of each exercise after the patient has completed the respective plan. Thus, only physiotherapists have access to a restricted area in the application for all the management of patients, training plans and physiotherapy exercises. The physiotherapist's application interface aims to allow the physiotherapist to make a more complete diagnosis of the patient's evolution status. This way, some **functionalities and requirements** are necessary for all

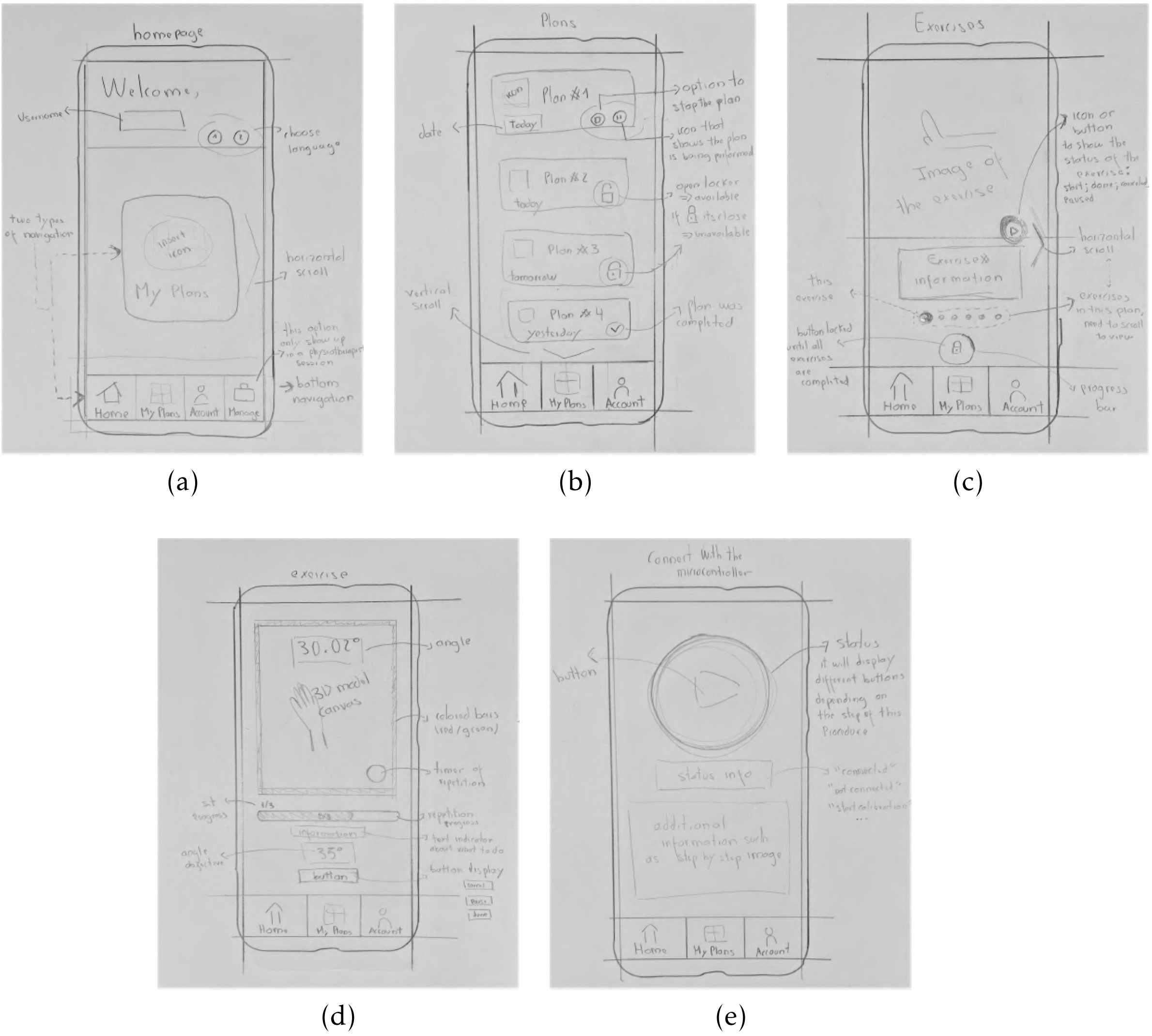


Figure 3.2: Sketches of some important interfaces of the patient interface. (a) Home page. (b) List of plans. (c) List of exercises. (d) Exercise interface. (e) Interface to establish connection with the wearable motion-sensing prototype component and perfom the calibration of sensors

this management to be done properly.

Search and view patient information. The physiotherapist must be able to search for a specific user present in the system. The patient's information must be accessible to the physiotherapists in order to manage the training plans consisting of a set of physiotherapy exercises that the patient will have to complete.

Manage a patient's training plan. The physiotherapists should be able to access the list of all the training plans of a certain patient but are only responsible for the plans they have created. This way, physiotherapists can only edit and delete training plans created by themselves. Once a training plan is created for a given patient, the physiotherapist will be associated to the plan and will be in charge of managing the exercises within the plan.

Manage physiotherapy exercises in a plan. When accessing a training plan, physiotherapists are allowed to create several exercises of the different four types of wrist physiotherapy movements proposed in this work without having to define a certain sequence of execution, and it is also possible to edit and remove any exercise from the plan.

Observe the patient's progress. After the patient has done all the physiotherapy exercises, and therefore has finished the proposed plan, the physiotherapist will be able to view information regarding the history of the patient's progress during the different exercises. The progress information can be viewed in graphs within the physiotherapist's management area, and will help provide personalized follow-up on the analysis of the results of the rehabilitation process. It is also possible to see detailed information on each repetition performed by the patient during the exercise, as well as overall statistics for the entire exercise. An idea of this interface can be visualised in the mockup present in the figure 3.3.

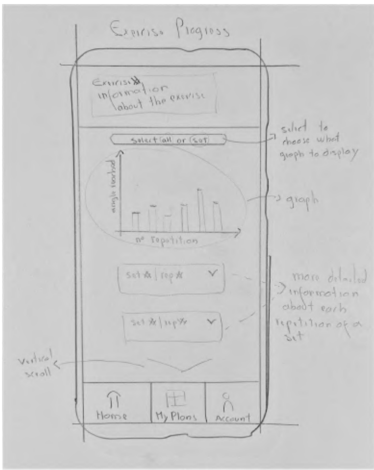


Figure 3.3: Sketch of the interface where the physiotherapist can observe the patient's progress throughout a given training plan

3.4 Wearable motion-sensing prototype component

This component is one of the most important parts of the system, as the wearable motion-sensing prototype component is essential for carrying out the proposed physiotherapy exercises, because it is the sensors of the wearable motion-sensing prototype component that measure and collect data on the orientation of the movements performed with the wrist. Based on the study in the previous chapter, it was used two IMU sensors to read the orientation of the wrist movements. One of the sensors is positioned on the back of the hand, in order to have a more correct reading of the angle made by the wrist, and the other is placed along with the microcontroller between the wrist and forearm to detect if there is any movement of the arm during exercises. The reason of the placement of the microcontroller in this area was to avoid any interference with the movements during the exercise.

Therefore, it is important to ensure some **functionalities and requirements** in this component, in order to provide a good use of the system.

Low cost and affordable. The wearable motion-sensing prototype is intended to be of low cost, so that it is accessible to all patients and to the clinic. As the exercises have to be performed during time intervals throughout specific day, it is important that each patient has access to the equipment to perform the exercises, either in a clinical environment or at home.

Easy to use and comfortable. The wearable motion-sensing prototype is expected to be easy to use and comfortable, so that it can be used outside a clinical environment and independently without any difficulty or movement restrictions. The wearable motion-sensing prototype should be worn on the wrist in a specific position, in order to allow the measurement of all the wrist physiotherapy movements proposed in this work, which are the flexion and extension movements, and also radial and ulnar deviation of the wrist.

Collect, process and send data to the application. The sensors in the wearable motion-sensing prototype component should measure the orientation of the movements and the microcontroller will perform some processing of the collected data and establish a communication with the application to sent these values so it can be displayed and used during the exercises. Part of the processing consists of fusing the values collected by the sensors, and this way it is possible to have a much more accurate orientation estimate, reducing significant noise, bias and data drift. When the training plan is accomplished, all the data about the exercises' progress which also includes the orientation angles read by the sensors, will be stored in the database so the physiotherapist can have access to it.

3.5 Server and database

The server component processes and stores the data transmitted by the application in the system database. The interactions between the server and the other components are performed through a REST API, which consists of an application programming interface (API) that has a REST architectural structure and respects a set of definitions and protocols. It is necessary to make sure that there are certain **features and requirements** so the server and database layer work as expected.

Store and manage data. The system supports different types of requests to create, edit, delete and access information regarding users, training plans, physiotherapy exercises, and evaluations which consist of the exercises' progress. All data is stored persistently in the database and can be accessed and managed through CRUD operations.

Authentication system. The server has an authentication system and all requests except login and registration have filters to check the authenticity of each request in order to allow or deny access to certain endpoints. As there are different types of users, it is necessary to ensure that only users with a certain specification can access particular information, and therefore information is restricted according to existing permissions.

Well structured and easily scalable. It is important that the structure of the API is designed appropriately to facilitate both organisation and to allow the system to be easily scalable. An example of this organization is in the endpoints that are destined only to be used by physiotherapists that have '/doc' prefix in the url path in order to be easily distinguished.

It is interesting to have an overview of how the data is being stored in this layer, and for this it is necessary to have a view of the database structure in the system. It consists of a MongoDB instance that will be implemented on a Spring server. Generally, a MongoDB instance is a running thread or a single copy of the MongoDB software running in the memory of a server. That said, one MongoDB instance can host many databases, in turn, one database can host many collections, and finally, one collection can have many documents.

In general, the database has four data collections, these being the users, training plans, physiotherapy exercises, and also evaluations consisting of the patients' progress during the exercises. The users' collection stores both patients and physiotherapists, the plans collection stores all the training plans created by the physiotherapists and performed by the patients, the exercises collection stores all the physiotherapy exercises from all the plans, and finally, the evaluations collection stores the evolution of the patients' progress throughout the exercises, that is, the essential information regarding the performance of the exercises in order for the physiotherapist to observe and evaluate the state of the patient's wrist rehabilitation process. An overview of the system database structure can be seen in the figure 3.4.

3.6 Summary

The system consists of three main components, which are the wearable motion-sensing prototype component, the application component with interfaces for physiotherapists and patients, and a server and database component. The proposed system is intended to help patients to perform the physiotherapy exercises for the wrist in order to improve

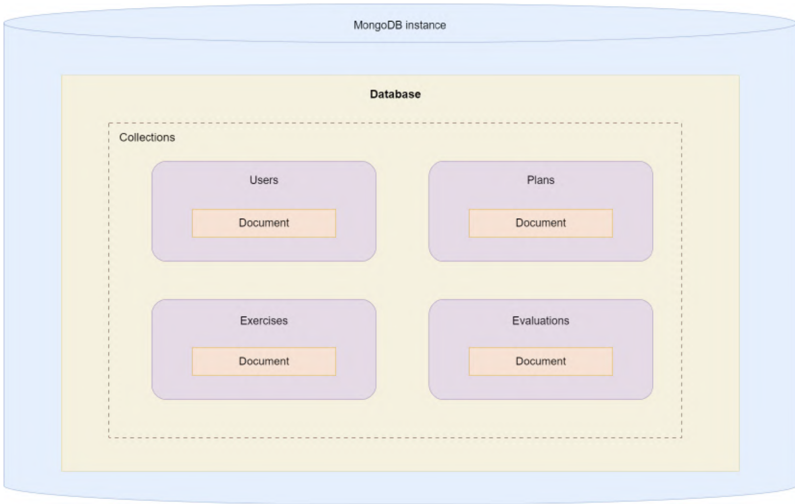


Figure 3.4: Database structure

their medical condition and promote a better rehabilitation process of the wrist. With the measuring data, it is also intended to help the physiotherapists to perform a better evaluation of the patients' condition.

4

System Implementation

This chapter presents how the whole system was built and what decisions were made to develop the wearable motion-sensing prototype component and all the components that compose the system. It is essential that all system components are compatible and communicate correctly with each other. The code running on the wearable prototype was developed in Arduino IDE, the mobile application in React Native, the server in Spring Boot and the database is a MongoDB instance. Some technologies and libraries were used to implement the different functionalities of the system, such as connecting and establishing the communication between the sensors and the microcontroller present in the wearable prototype, and in the communication via BLE between this component and the mobile application in order to transmit the collected and processed data to the application. It was also necessary to consider some system requirements, such as the graphic component of the 3D model present in the application's exercise interface, which requires some graphic support technologies. The specifications and implementations of each component will be presented throughout this chapter.

4.1 Wearable motion-sensing prototype

The wearable motion-sensing prototype component is one of the most important components of the system, since it is responsible for collecting and first processing of the data obtained by the sensors, and all the information from this component is essential and indispensable for performing the physiotherapy exercises in the system.

4.1.1 Prototype development and design

The prototype of this component consists of two boxes that are placed on the area between the wrist and forearm and on the back of the hand, using elastic bands with velcro. A powerbank is used to supply the system with power. The schematic diagram of this component can be seen in the figure 4.1 and all parts are identified with numbers for better understanding.

The box number 1 consists of an ESP32 microcontroller and an MPU9250 sensor, while box number 2 contains only one MPU9250 sensor. Both sensors are connected to the microcontroller via a cable connection, and the connection is made with I2C SCL (IO22), I2C SDA (IO21), GND and 3.3V inputs, which are soldered. The sensor placed in box one has the main function of assessing whether there is movement of the forearm during physiotherapy movements, and the sensor in box two will measure the orientation of the wrist angles during all exercises and provide the data for most of the interactive interface displayed during the performance of the exercise, such as for animating and moving the 3D model in the application, as well as providing data for the interactive graph showing the angle being performed in real time.

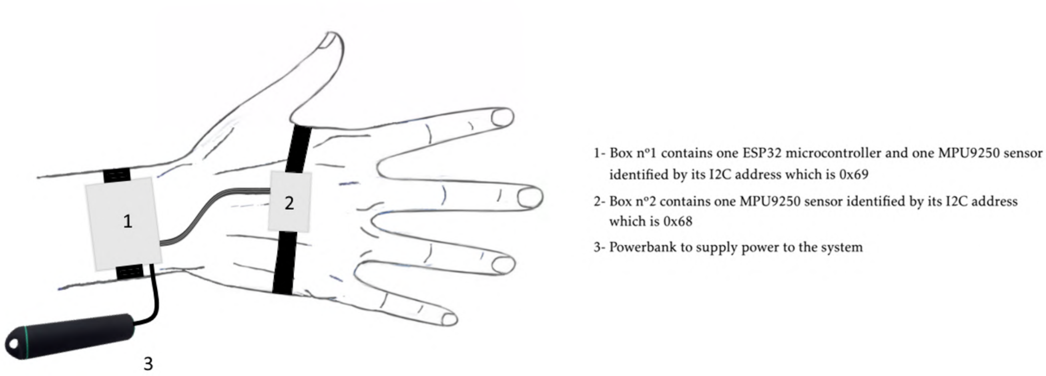


Figure 4.1: Wearable motion-sensing prototype component scheme

At a very early stage, a simplified wearable motion-sensing prototype was developed, consisting of an ESP32 microcontroller fixed on a breadboard that was used to connect the two MPU9250 sensors, as shown in figure 4.2. This way it was possible to test all communications as well as to check that there were no system incompatibilities.

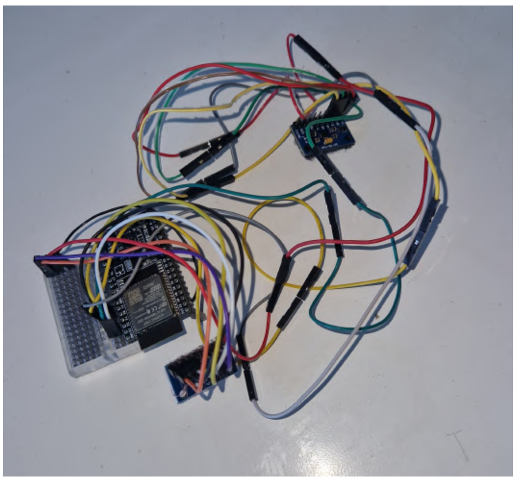


Figure 4.2: Wearable motion-sensing prototype at a very early stage

Then, a new wearable motion-sensing prototype was developed in order to allow testing the usability of this component when worn on the wrist, and this is the most up-to-date prototype developed in this work, as can be seen in figure 4.3. It is important to point out that the wearable motion-sensing prototype component can be used on either the right or left wrist, as the system supports both situations.

Figure 4.3: Wearable motion-sensing prototype in a final phase

There are two different types of communication used in the wearable motion-sensing prototype component, these being the communication established between the microcontroller and the MPU9250 sensors, and the communication made between the microcontroller and the mobile application.

4.1.2 Communication between microcontroller and sensors

A survey of some existing libraries with the functionality to support communication between the MPU9250 sensors and the microcontroller was carried out. The chosen library was created by hideakitai[1] and presents full compatibility with the system hardware, and also allows the application of filters, such as the Madgwick filter, which is used to obtain the sensor values in quaternions. As explained in chapter 2.4.3.1, quaternions are a four-dimensional extension of complex numbers, and are essential to obtain information about sensor orientation without errors associated with the gimbal lock found in Euler systems [17]. This way, it is possible to produce more correct results during the exercises.

[1] Arduino library for MPU9250 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking Device. It is available on the website: https://github.com/hideakitai/MPU9250

In short, this library is used in the code that runs on the microcontroller and corresponds to the execution part of the MPU9250 sensors.

The communication between the ESP32 and the two MPU9250 is done through a simple serial communication protocol called Inter-Integrated Circuit (I2C) which allows a communication link between integrated circuits. The I2C protocol is a two-wire interface composed of serial data and serial clock signals, both bidirectional, referred to as SDA and SCL, respectively [51]. This protocol can handle a large number of slave devices that are connected to the same bus. Basically, the master device puts the slave's address on the bus, and the slave will recognize the master with the corresponding address. Each device connected to the bus has its own address and, depending on the functionality, can behave as a receiver and/or transmitter.

To allow separate communications with the sensors, each MPU9250 has its own I2C address. The sensor placed in box one has the I2C address 0x69 and the sensor in box two, the I2C address 0x68. This difference in addresses is done by activating the AD0 pin. That said, the sensor containing I2C address 0x69 has this pin connected to the ESP32 microcontroller's 3.3V input. Each address has 7 bits and the least significant bit (LSB) is determined by the logic level of the AD0 pin.

4.1.3 Communication between microcontroller and application

The communication between the microcontroller of the wearable motion-sensing prototype and the mobile application is done through the BLE communication protocol, since it was the chosen technology that best suits the system and can bring more advantages regarding the system and the protocol specifications and functionalities. The system will benefit from using this technology as the BLE consumes low power and it is of interest that the wearable motion sensing prototype component is connected and sending the various collected and processed data regarding movement orientation at short intervals and possibly over long periods of time, depending on the duration of the plan and physiotherapy exercises that are performed. Furthermore, the wearable motion sensing prototype and the mobile application components are supposed to be used close to each other to follow the progress of the movement being performed, and the BLE has the feature of only working over short distances. It is important to note that not all mobile devices support BLE, and depending on the mobile phone model, the Bluetooth protocol version may differ, however, the most modern smartphones have system versions and technologies much higher than the requirements of the developed system, as will be addressed in this chapter.

4.1.3.1 Communication establishment and configurations

As seen in chapter 2, there are several communication types presented in the BLE specification that offer different topologies and use cases. The method used in this work is connection-oriented (unicast), and therefore data is only exchanged between devices after the connection has been established. When the connection is being established, an agreement occurs between both devices, and there is a notion of roles which are called peripheral device and central device. The peripheral device announces its presence so that the central device can initiate the connection.

As mentioned before, GATT makes use of a generic data protocol called Attribute Protocol (ATT), and defines how two BLE devices transfer data in both directions. In this system, the GATT server is the ESP32 microcontroller and the GATT client is implemented in the application of the mobile device.

In the code running on the microcontroller a BLE server is started and a service with three characteristics is defined. Two characteristics contain information about the orientation values collected by each of the MPU9250 sensors and the other is used to exchange information during the calibration of the sensors. Both the service and the characteristics are identified by a unique UUID predefined in the code.

A configuration was made in this communication regarding the ATT maximum transmission unit (MTU) which is the maximum length of an ATT packet. It was necessary to increase the maximum length of an ATT packet, since the default size of this packet was not large enough for the amount of data intended to be sent from the reading of each of the sensors.

The method used to increase the default MTU limit was through packet length extension (DLE), introduced in Bluetooth 4.2. The alternative would be to send fragmented messages, and in this system it is preferable that data from each sensor reading is sent at once in order to not overload the application with too many messages, as well as to keep the 3D model and the rest of the information in the exercise interface with consistent data from the current movement. Thus, the implemented system requires at least Bluetooth 4.2, as well as support for BLE.

When negotiating the MTU size, the client has to send an MTU adjustment request, and this way ensure that both sides are using the same MTU value. This definition is made at the mobile application level, since the microcontroller has the role of server in this communication. However, if the ESP32 microcontroller was the client, the arduino default library itself has functions that allow structuring these BLE communication settings, being possible to achieve through the function `BLEDevice::setMTU(value)`. In the mobile application, with the support of the library that was chosen to handle the whole BLE part which will be discussed later, the function requestMTU(peripheralId, mtu) was used, which is available for Android versions with API 21+, i.e. higher than Android version 5.0. There are interesting discussions in some issues found in the github repositories[2], which address decisions and implementations of these definitions. The MTU value proposed by the mobile application was estimated based on the maximum byte size of the character arrays that the microcontroller requires to send the data collected by the

[2]Sample ESP32 snippets and code fragments library that have interesting discussions in the existing issues. It is available on the website: https://github.com/nkolban/esp32-snippets

MPU9250 sensors through the corresponding characteristics. The data sent is previously processed and represents the orientation of the sensors in quaternion values and also in the Euler angle system, which is used in the exercise interface in the application. The sensor orientation represented in a quaternion has four variables, x,y,z and w, and can range from -1.00 to 1.00. Since each character is one byte long, the largest byte value of a quaternion is when it is negative, and therefore the maximum byte value of one of the quaternion variables is 5 bytes, including the negative sign characters and the separation point character between integers and decimals. The Euler angles can have two integers and two decimal places, and as the value can be negative, each of the three Euler angles can have a total of 6 bytes. The format of the message sent to the application by each of the characteristics consists of quaternion value variables, followed by the same value but in the Euler angle system, and each variable is separated by a ";" character, which has a size of 1 byte. Thus, the maximum estimated size of the character array sent by each of the characteristics has a size of 44 bytes. An example of this message could be `-0.15;0.12;0.58;0.79;-5.92;-21.10;-70.51`, and in this case the size is 40 bytes. The quaternion value is an average of several sensor readings, and the Euler angles were calculated by converting the quaternion value to the Euler system. The data processing part, i.e. the averaging of the quaternions, and also the conversion of the values to the Euler system, will be covered ahead.

4.1.3.2 Message exchange and calibration process

The mobile device scans for BLE devices and attempts to establish a connection if it finds the ESP32 microcontroller via its Bluetooth address, which is a 6-byte identifier determined by the manufacturer. Once the connection is established by using the protocols explained in chapter 2, the device discovers the services and the available characteristics, and also negotiates the MTU. After this procedure, the communication between both devices is initiated and it becomes possible to read and write in the characteristics. Before the sensors start collecting values, the ESP32 microcontroller and the mobile device will synchronously calibrate the sensors. After successful calibration, the sensors will start collecting data and the microcontroller will send the processed data to the application to be provided in the exercise interface.

An attempt is made to establish a connection between the mobile device and the microcontroller through a specific interface in the mobile application, and if it is successfully established, the MTU is negotiated and the discovery of the microcontroller's services and characteristics is performed. As previously stated, the mobile device acts as the GATT client and the microcontroller takes the role of GATT server.

The calibration process is initiated with the mobile device sending a message (001) through the specific characteristic for this initial communication. If the message is successfully sent, three notification systems are initiated on the mobile device. Each notification system is intended for each of the three characteristics, which are the one being

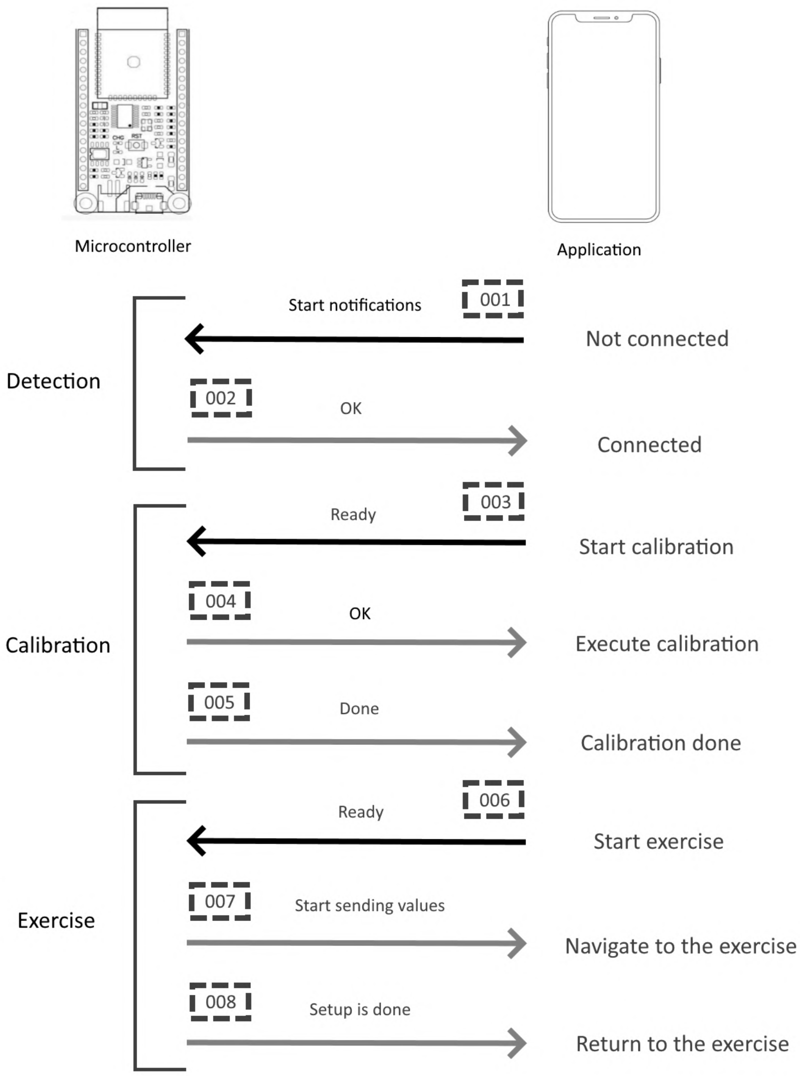


Figure 4.4: Communication schema between the micro and application

used in this calibration process and two others that will be used to receive the values from each of the sensors. This notification system will allow the mobile device to also receive messages from the GATT server, which is the microcontroller. After receiving the first message sent by the mobile device, the microcontroller will then return a message (002) which will ensure that everything is ready for two-way communication. This whole procedure of exchanging synchronised messages between the microcontroller and the mobile device can be seen in the scheme shown in the figure 4.4.

In this phase, called "Detection", the two sensors will be initialised with their respective I2C addresses 0x68 and 0x69. Some sensor settings have been configured, namely the magnetic declination, which must be set depending on the geographical location where the sensor values are to be read, in order to obtain more accurate data. This declination is the result of the Earth’s magnetic field, more specifically, the difference between true geographic and magnetic North. The value of the magnetic declination used in the configuration was calculated based on the geographical location of when the practical tests of the wearable motion-sensing prototype component were carried out, which was in Lisbon, being this value -1° 26’ according to the web application[3] suggested by the creator of the library used in the arduino code executed in the microcontroller.

Another configuration that is important to note is the filter used to calculate the quaternions of the values collected by the sensors. The library used, allows the use of either the Madgwick filter or the Mahony filter to merge the data read by the sensors. As seen in chapter 2, the Mahony filter takes less time to run, however the Madgwick filter performs better because the data has less noise and is closer to reality, therefore the Madgwick filter was chosen to integrate this system.

The "Calibration" phase, as the name suggests, is for the calibration of the MPU9250 sensors. The calibration is done using two functions provided by the library created by hideakitai. One of the functions is used to calibrate the accelerometer and gyroscope and it is necessary to keep both MPU9250 sensors stable on a flat surface for a certain period of time. The other function is used to calibrate the magnetometer and it is necessary to move both MPU9250 sensors in a figure eight. The calibration of the magnetometer was eventually ignored, since after some tests made with and without this calibration, it was noticed that the values collected by the sensors did not change.

4.1.3.3 Data reading and processing

In the last phase of this communication, which is called "Exercise" and takes place right after the calibration of the MPU9250 sensors, the exercise can be initiated and the ESP32 microcontroller starts sending the sensor values to the mobile application, every 508ms average cycle. This time interval is suggested by the library, and after conducting some tests it was confirmed that increasing this time causes the validation of the angles in the application to be inconsistent with the movement being performed in real time and causes stuttering both in the movement of the 3D model in the exercise interface and in the interactive graph that displays the angle being performed. In turn, decreasing this time interval increases the transmission of data packets via BLE, and consequently, the application is flooded with notifications about the characteristics of this service, causing the application to block. As said before, the message sent in the characteristics corresponds to the values collected by each MPU9250 sensor and is composed of the average of the values collected during this time interval in quaternion format and its respective conversion to Euler angles. Each reading cycle takes an average of 8ms, and the longest cycle when the processed values are sent to the application takes about 28ms. This means that the time used to average the quaternion value and convert it to the Euler system for both sensors, as well as sending these values to the application through the characteristics,

[3]Magnetic Declination is an web application to find the magnetic declination at any location. It is available on the website: https://www.magnetic-declination.com/

takes about 20ms. Several cycles of sensor readings were performed during the 508ms time interval, and by making the quotient of this time interval by the average time of each reading cycle it is possible to obtain an estimate of the number of cycles that are executed. Comparing this result with the number of values collected by a sensor during this time interval, which corresponds to the exact number of cycles executed, it is possible to validate that the information about the time intervals is correct. Thus, doing the quotient of 508ms by 8ms gives a result of, on average, 63.5 cycles executed, which corresponds to the number of quaternion values collected in this interval, which is 63 cycles. The figure 4.5 shows the serial monitor log messages regarding the wearable motion-sensing prototype component, where the format of the messages sent to the application can also be verified.

```
-> small cicle time: 8 milliseconds
-> small cicle time: 8 milliseconds
-> small cicle time: 8 milliseconds
-> small cicle time: 7 milliseconds
-> cicle time sending to app: 508 milliseconds
-> nr cicles/ nr values read: 63
-> mpu1(0x68) values:
-> quaternion: 0.08, -0.01, 0.67, 0.73
-> euler convertion: 6.26, 7.32, -83.10
-> mpu1 array char size: 38
-> mpu1 sent: 0.08;-0.01;0.67;0.73;6.26;7.32;-83.10
-> mpu2(0x69) values:
-> quaternion: -0.15, 0.12, 0.58, 0.79
-> euler convertion: -5.92, -21.10, -70.51
-> mpu2 array char size:: 41
-> mpu2 sent: -0.15;0.12;0.58;0.79;-5.92;-21.10;-70.51
-> small cicle time: 36 milliseconds
-> small cicle time: 9 milliseconds
-> small cicle time: 8 milliseconds
-> small cicle time: 8 milliseconds
```

Figure 4.5: Log messages in arduino's serial monitor

The method consisting of an approximation of Markley's method to calculate the average of the quaternions was implemented since it is simpler and faster to use. This algorithm can be used in this system because the average is done in intervals of an average of 508ms, and the values of the quaternions consist of very identical orientations that arise during the movement of the physiotherapy exercise. This quaternion average is then converted into Euler angles and both values are sent to the mobile app. The quaternion value will be used for the movement of the 3D model and the Euler angles to increase and decrease the interactive graph showing the angle being performed by the wrist during the exercise, as well as validating the angle, and it is also displayed on the exercise interface so that the user is aware of the angle being performed with the wrist in real time, as it is more intuitive and easier to understand than a quaternion value.

The conversions between quaternions and Euler angles can easily be achieved by matrix operations, but there are also algorithms that implement equations to allow direct conversion. The implemented algorithm was inspired by some works that use an identical formula, such as the study performed by the authors of the paper [16] that show equations they used for such conversions in order to represent an orientation in angles and keep the overall dynamics of the system in the form of quaternions, and also the paper [46] that discusses the use of quaternions in 3D rotation applications. The formula used to perform the conversion from Eulers angles to quaternions is shown in the figure 4.6.

$$roll = u = \tan^{-1}\left(\frac{2(q_0q_1 + q_2q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right) = \text{atan2}[2(q_0q_1 + q_2q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2]$$

$$pitch = v = \sin^{-1}(2(q_0q_2 - q_1q_3)) \quad = \text{asin}\,[2(q_0q_2 - q_1q_3)]$$

$$yaw = w = \tan^{-1}\left(\frac{2(q_0q_3 + q_1q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right) = \text{atan2}[2(q_0q_3 + q_1q_2), q_0^2 + q_1^2 - q_2^2 - q_3^2]$$

Figure 4.6: Formula for converting Euler angles to quaternions [46]

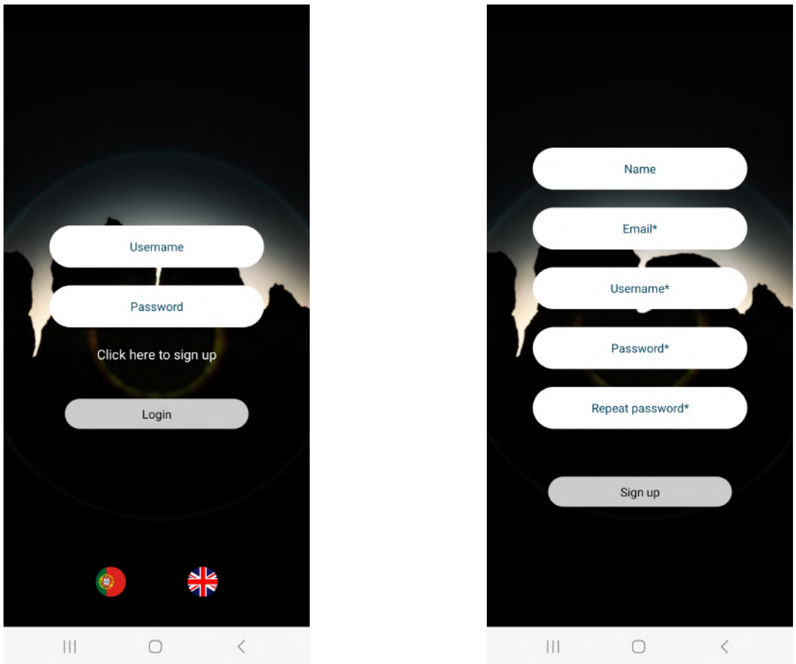
The placement of the wearable motion-sensing prototype component has to be done in a certain way, since the rotation of the boxes where the sensors are located causes the axes to change, and consequently, the angles collected by the wearable motion-sensing prototype component may not correspond to what the application is expecting. This way, the X-axis is in the direction of the arm, the Y-axis perpendicular to the forearm, and the Z-axis pointing upwards. The angle that will be presented in the interface of the exercise with the purpose of showing the angle that is being made by the wrist, is the angle that corresponds to the rotation about the y-axis, which can also be designated as pitch. This is the angle that will be considered from the reading of the sensor placed on the back of the hand to validate the four physiotherapy movements proposed in this work. The angles to be considered from the sensor placed between the wrist and the forearm are the angles corresponding to the rotation on the three axes, as the forearm should remain immobile during the execution of the four physiotherapy exercises, so it is necessary to detect when there is any movement of the forearm.

It is important to note some situations regarding the communication between the microcontroller and the mobile application. The BLE connection can be ended by both sides. In a situation where it is the GATT client, i.e. the mobile device that terminates the connection, either as a result of turning off Bluetooth or closing the application, then the microcontroller will return message (008) when the connection is re-established, as can be seen in the figure 4.4, and will continue to send the sensor values without the need to recalibrate. If the connection is re-established after the disconnection of the GATT server, that is, the ESP32 microcontroller, it is necessary to repeat all the steps represented in the diagram shown in figure 4.4.

4.2 Mobile Application

The mobile application was developed in React Native, a framework that allows hybrid app development. It is important to have a knowledge of which libraries exist to support some functionalities of the application, since there are some conditions and limitations according to the workflow chosen for the development of the project. Since a BLE communication support is needed for the mobile device to communicate with the microcontroller, and also a 3D animation support of the physiotherapeutic movement is necessary, the application was developed following a bare workflow. This way, it was possible to install the Expo animation libraries, and also other native libraries, such as the React Native Ble Manager library[4], that will support the BLE communication, which is fundamental in this application.

In order to use the application it is necessary to register and authenticate first, through the interfaces shown in the figure 4.7, and only then the user is redirected to the home page. The information regarding authentication, that is, the session token, is stored in the persistent memory of the mobile device, as well as the language selected by the user, which can be Portuguese or English. There are two types of users, patients and physiotherapists, and there are different application interfaces depending on the "role" of the session user.



(a) (b)

Figure 4.7: Authentication interfaces. (a) Sign in page. (b) Sign up page

[4]React native ble manager is a React Native library to support BLE communication. It is available on the website: https://github.com/innoveit/react-native-ble-manager

4.2.1 Pacient Application

The main focus in the patient application is the access to the training plans and respective exercises proposed by the physiotherapist, as well as the performance of each exercise.

4.2.1.1 Training Plans

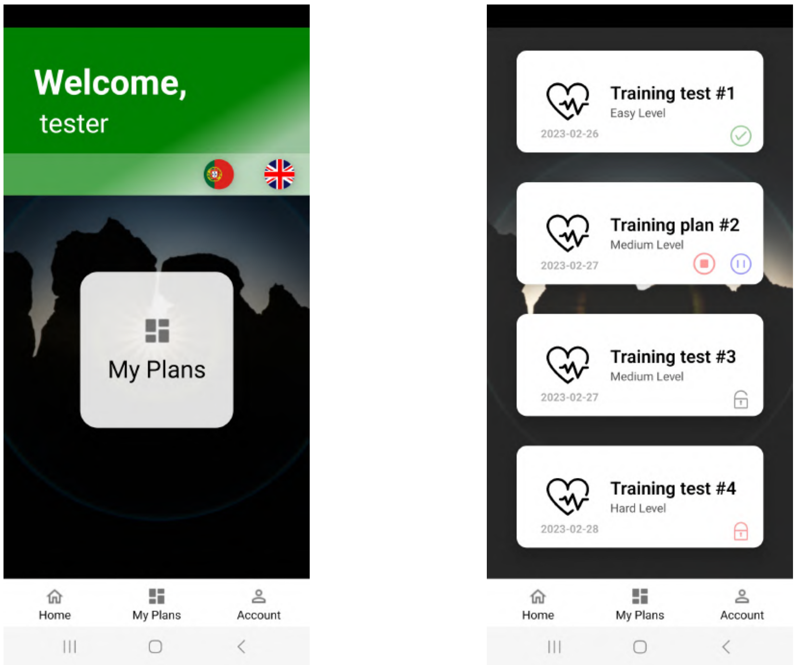
In the plans list interface it is possible to see what plans are available, unavailable, completed, and also, in progress, which means that a training plan has been started and can be resumed. A plan is not available when the date established by the physiotherapist for its realization has not yet arrived. On the other hand, plans which have not yet been completed and the date established for its realization is the current or previous days, are available and can be performed. A completed plan can't be repeated, which means that the patient has completed the proposed physiotherapy exercises and finished the plan. Finally, a plan in progress represents a plan that has been started but has not yet been completed by the patient, and therefore it is possible to resume the plan in order to complete it. This state exists mainly to ensure that in cases where a system problem occurs during the exercise it is possible to resume the plan and that progress is not lost.

It is important to note that it is only allowed to do one training plan at a time, because it would not make sense in a real context for the patient to be doing several plans at the same time. Thus, if a plan is already in progress, it is not possible to start another plan at any other time, so if there is a need to do another plan for any reason, there is the option to stop the plan that is in progress, causing the progress up until that moment to be erased, and the plan becomes available to be started again. Otherwise, it is required to complete all the exercises in the current plan to be able to end the plan in order to start a new one.

The plans page can be accessed from the home page or through the navigation bar located at the bottom of each page after successful authentication, as shown in figure 4.8. When a training plan is started, its identifier is saved in the persistent memory of the mobile device to ensure that there is a system state to resume the exercise if the application is closed for any reason.

4.2.1.2 Exercises

When a plan is accessed, there is an automatic navigation to the interface that displays all physiotherapy exercises associated with the plan. There is no sequence to perform the exercises, so the patient can choose to start any of the exercises in the plan by any order, and there will be information on whether the exercise has been performed successfully or not. When a physiotherapy exercise is initiated, the patient will have three possible scenarios, these being completing all repetitions of the sets proposed in the exercise, as well as pausing or cancelling the exercise. However, the pause and cancel options are mutually exclusive, meaning that they cannot be used simultaneously, so these options



(a) (b)

Figure 4.8: Initial patient application interfaces. (a) Home page where the patient is redirected after a successful authentication. (b) Page of the list of plans proposed by physiotherapists for the patient

are made available in the application according to the patient's current progress. The state diagram of the physiotherapy exercises can be seen in figure 4.9.

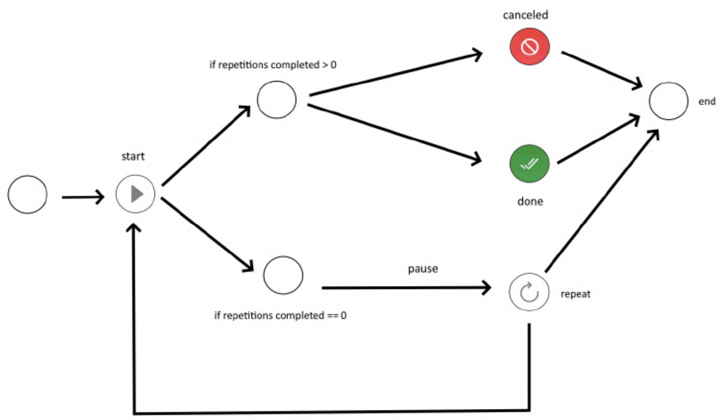
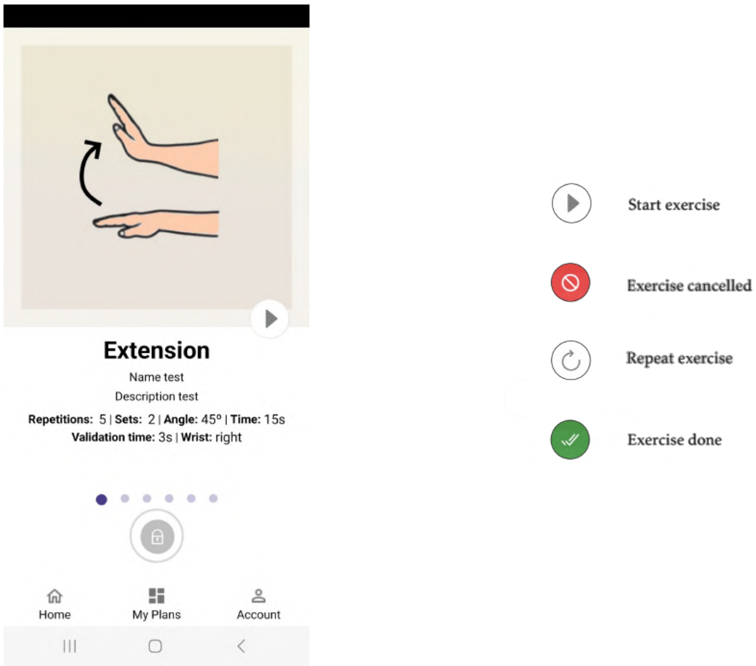


Figure 4.9: Exercise status diagram

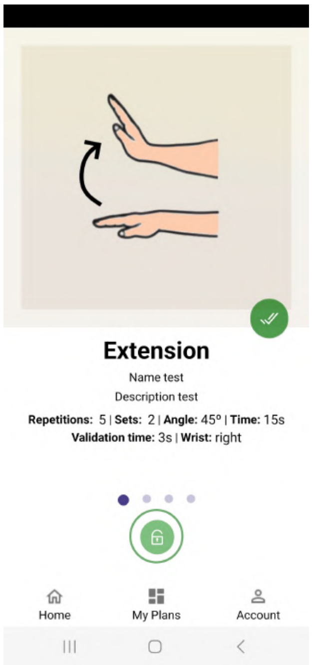
There are options to pause or cancel the exercise in case the patient starts the exercise and does not want to finish it. The pause option appears if no repetitions have been performed yet, and exists for situations where the exercise has been started by mistake. In this case, the patient is redirected to the exercise list interface of the plan that is currently in progress, and the same exercise will appear with the option to "repeat" instead of "start". On the other hand, the cancel option appears to replace the pause option in situations where the first repetition of the exercise has already been performed. All the possible states of an exercise are presented in the figure 4.10.



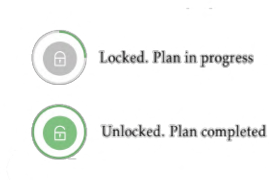
(a) (b)

Figure 4.10: Interface of physiotherapy exercises of a given plan and display of the different exercise states. (a) Example of an exercise that has not yet been performed and therefore the button for completing the plan is still locked. (b) All the possible states of an exercise

In case the exercise is canceled or successfully completed it is not possible to repeat it, and once an exercise has been started, regardless of the outcome, it will be counted as "completed" and the circular progress bar for unlocking the end training plan button will be incremented, as it can be seen in the figure 4.11. A successfully completed exercise has a green arrow indicator, while a cancelled exercise has a red indicator representing that state. All the information regarding the exercise progress is stored in the persistent memory of the mobile device, along with the plan identifier. Also, if the patient tries to access the training plan again while an exercise is in progress, there will be an automatic navigation to the current exercise interface instead of a redirection to the exercise list in the plan to ensure that the current exercise is performed, although there are options to pause or cancel the exercise if necessary. Only an exercise that has not yet been started or has been paused has a status indicator that can be interacted with, and when it is clicked a window appears to confirm that it is intended to start the exercise.



(a)



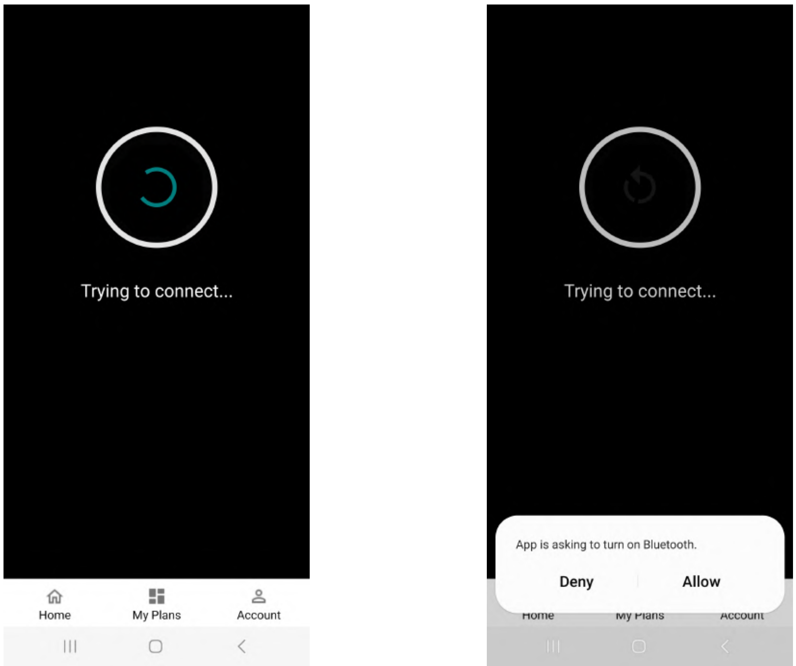
(b)

Figure 4.11: Interface of physiotherapy exercises of a given plan and display of the different plan states. (a) Example of a training plan where all exercises have been completed and as a result the button for completing the plan is unlocked. (b) All the possible states of a plan

4.2.1.3 Calibration

When a patient chooses to perfom a physiotherapy exercise, an interface is presented to establish the connection between the mobile device and the microcontroller, as well as perform the calibration of the sensors. This interface is ignored if the connection already exists and the calibration has also been done, otherwise the patient will have to follow all the steps in this interface before starting the exercise. When this interface is started, an automatic attempt is made to establish a connection with the microcontroller if Bluetooth is enabled on the mobile device, otherwise the connection request must be made manually by clicking the button provided on the interface. On the other hand, if Bluetooth is off, a pop-up window appears requesting its activation, as can be seen in the figure 4.12.

Once the microcontroller has been detected and the connection has been established, the calibration phase starts. In this phase, the patient can start the calibration of the two sensors, reminding that only the accelerometers and gyroscopes of the MPU9250 sensors are calibrated through a method provided by the library used in the arduino code that runs on the microcontroller. At this stage it is necessary for the patient to place the wearable motion-sensing prototype component on a flat surface for a few seconds until the calibration is successfully completed. Some interfaces showing the different phases of the communication will be displayed, as shown in the figure 4.13.



(a) (b)

Figure 4.12: The "Detection" phase on the mobile application consists of interfaces with the goal of trying to establish a successful connection between the wearable motion sensing component and the application. (a) Connection attempt. (b) Bluetooth activation request on the mobile device

As stated earlier, if the connection and calibration have previously been performed successfully, then this interface can be skipped and the exercise interface is resumed, or else all steps in this interface are required and the exercise can be started after all steps have been performed. Both restarting and starting an exercise makes the data collected by the sensors to be sent to the application.

4.2.1.4 Exercise interface

When the exercise is started, the patient is brought to an interface where it is possible to observe both the 3D model of the wrist and an interactive graph showing the angle being performed, and all the information needed to perform the exercise. This information consists of progress bars for repetitions and sets, the value of the angle being made with the wrist, target angles to be reached, time for the execution of each repetition, as well as colored borders to indicate correct and incorrect movements. In addition, temporary messages are displayed to show the stages of exercise progress, such as whether a repetition has been started or completed, or if a movement is being validated or is incorrect, and these messages are followed by different sound effects to help alert the patient to such stages. This interface features two buttons, one to switch the screen between the 3D model of the wrist and the interactive graphic, and the other to disable or

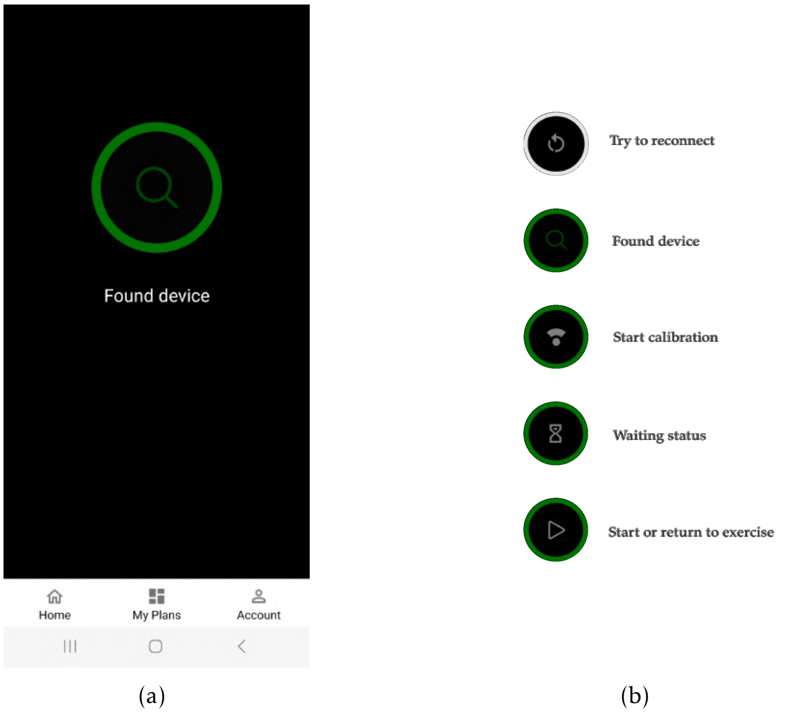
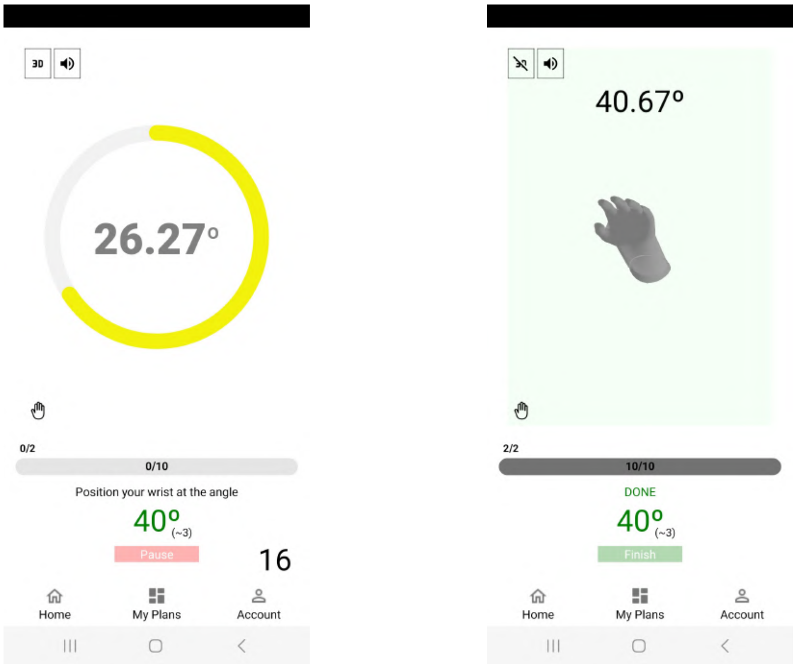


Figure 4.13: Interface of the connection and communication process between the application and the wearable motion sensing component, as well as the calibration of the sensors. (a) Successful detection and connection. (b) All possible states displayed during this procedure until the exercise can be started

enable sound effects during exercise. A hand icon is also displayed, which can be either the left hand or the right hand, representing the wrist that should be used for the exercise.

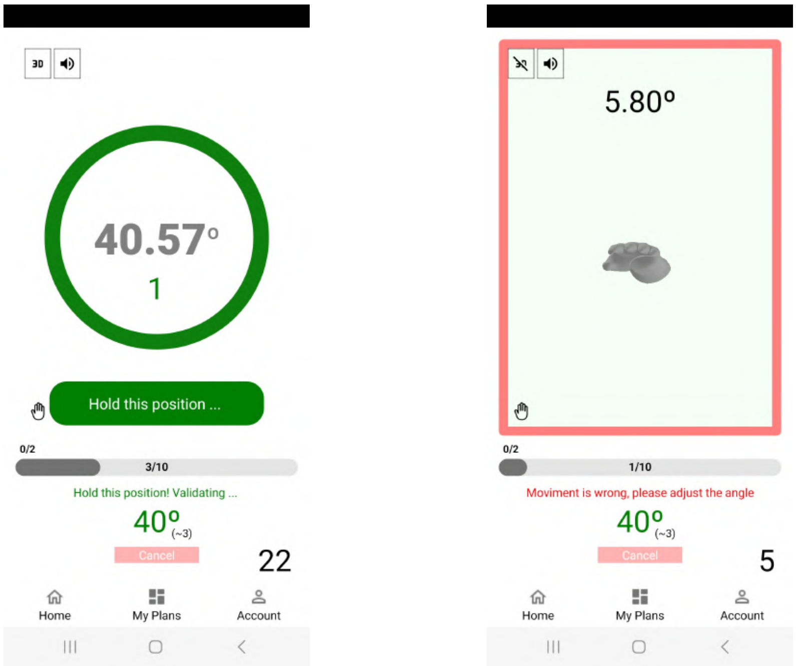
Throughout a physiotherapy exercise, all the repetitions are started by the patient placing the hand that is using the wearable motion-sensing prototype component in a position where the pitch angle makes 0°, therefore a zero inclination between the x-axis of the sensor and the ground plane. After a three-second time interval, the repetition is initiated and the target angle that the patient has to perform with the wrist will be displayed on the exercise interface. There are two time intervals that are variables set by the physiotherapist when creating the exercise, which are the execution time of the repetition, that is only intended to give the patient a sense of how long it should take to reach the target angle from the beginning of the repetition, and the time required to hold the target angle for the repetition to be validated. The time interval set by the physiotherapist for each repetition disappears from the interface when the time is exceeded, however the repetition can be ended by the patient, and the physiotherapist will have access to this value. During the validation time, an average is made of the various angles which eventually differ minimally from each other, since the patient has to keep the wrist at the same exact angle or very close to the target angle. When the repetition is finally validated, this average calculation of the target angle achieved, along with the exercise identifier, the movement execution time, and information regarding the number of repetitions and current set, are stored in the mobile device's persistent memory. Both when validating the target angle and when validating whether the wrist is positioned at a neutral angle before starting the exercise, a threshold of 3° is given, because holding an exact angle is almost impossible due to different factors, such as the sensitivity of the sensors used. Some interfaces of the application in the course of a physiotherapy exercise can be seen in the figure 4.14.



(a) (b)

Figure 4.14: Physiotherapy exercise interfaces at an initial and final stage using different views. (a) First repetition when using interactive graph view. (b) Interface when all repetitions and sets have been completed and 3D model view is being displayed

When there is a rotation of the forearm on any axis of the sensor placed between the wrist and the forearm up to an angle of 10° and after the repetition has been started, the movement is considered incorrect, and therefore the repetition is interrupted until the posture is corrected. The angles of the correct forearm position are set at each repetition start, while the patient has to validate the neutral pitch angle of the wrist. Once an incorrect posture has been fixed, it is necessary to wait three seconds before the exercise is resumed. This detection of forearm movement allows to verify whether the patient is facilitating the exercise by lifting the forearm, which would be incorrect and harmful rehabilitation and treatment process. There is a counter of how many times the patient made an incorrect movement for each repetition during the performance of the physiotherapy movement. When validating an angle, green borders appear around the screen that provide both the 3D model and the interactive graph as positive feedback, and red borders appear when the movement is incorrect as an indicator that something is wrong with the movement. Alongside these colourful borders, there are also sound effects to alert the patient and informative messages for better understanding, as can be seen in figure 4.15.



(a) (b)

Figure 4.15: Some possible scenarios of the physiotherapy exercise interface when a repetition is being performed. (a) Validation of a repetition when using interactive graph view. (b) Wrong movement detected when using the 3D model view

The patient has the options of switching between the 3D model of the wrist and the interactive graph. The graph is the default view, and it can be changed for the model and vice versa using a button provided on this interface. Another button provided on the interface is for enabling or disabling sound effects, as these are merely used to support the informational messages that appear during the exercise. A physiotherapy exercise is defined by the physiotherapist to be performed by the right or left wrist, and therefore there is a hand icon in this interface that changes depending on which wrist should be used.

The interactive graph is a circular graph that is filled as the angle made by the wrist changes during the movement, and changes color between red, orange, yellow and green according to how close the angle being made with the wrist is to the target angle. The angle being performed with the wrist is displayed in the center of the circular graph for easy understanding. This graph has two types of behavior, depending on whether the repetition has been started and there is a target angle to be reached, or if the repetition has not yet been started and it is necessary to validate the neutral angle of 0º. In the case where it is necessary to validate the neutral angle, the graph is completely filled and only changes color as the angle made by the wrist approaches the 0° angle. The graph only turns green when the angle performed is within the defined threshold for the angle to be validated. During the execution of a repetition, the graph is filled according to the angle being performed by the wrist, and is completely filled when the angle is reached. If the angle performed exceeds the defined threshold for the angle to be validated, the graph remains filled and only changes color as the distance between the angle performed and the objective angle increases. In extension and radial deviation exercises where the objective angle is positive, if the angle performed is less or equal to 0°, the graph is not filled in and only shows the angle to be performed within the circle. The same occurs in the case of flexion and ulnar deviation exercises, however, since the objective angles in these cases are negative, if the angle performed is greater or equal to 0°, the graph is not filled in.

The 3D model is loaded in different ways depending on the type of exercise and the side of the wrist being performed to ensure that the model is displayed in the correct axes closest to reality. This detail in the 3D model is necessary because with only the use of two IMU sensors it is difficult to rotate the 3D model realistically, and as will be explained later, it was necessary to cancel the interaction of the sensor placed between the wrist and forearm with the 3D component because it caused distortions in the model. Thus, since only the vertices of the model associated with the sensor placed on the back of the hand will rotate during animation, it is necessary to fix the other part of the model. It is important to remember that the system works correctly with both the left and right wrist, and this detail is only for the visualization of the 3D model component.

During the exercise, it is possible to see the progress of the repetitions and sets increasing, and when the physiotherapy exercise ends, an option to end the exercise appears. When all exercises in a training plan have activity records, meaning that they have been completed, cancelled or paused, it is possible to end the plan, and therefore all information regarding angles, times, faults, repetitions and sets for each exercise is sent to the server and stored in the database. The training plan information in the mobile device's persistent memory is released, and it is possible to see that the plan has been completed in the plan list interface.

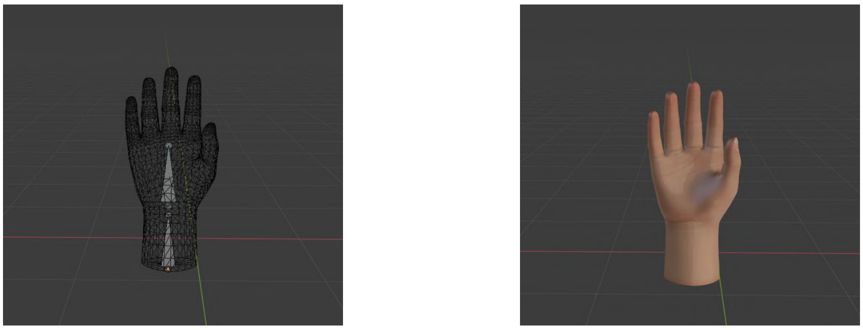
4.2.1.5 Graphic component

The graphical interface of the 3D model of the wrist was a major challenge in the development of the mobile application, and is displayed along with other types of interactions and feedbacks depending on the movement being performed during the physiotherapy exercise.

The Blender software[5] was used to create the 3D model, which is a free and open

[5] Blender is a free and open source 3D creation suite. It is available on the website: https://www.blender.org/

source 3D creation suite and supports the entire 3D pipeline modelling, animation, rendering, among other features. Skeletal animation refers to how two individual parts of an object are represented and coordinated, these being, the surface representation used to draw a model, referred to as a skin or mesh, and a set of hierarchical and interconnected parts, called a skeleton. That said, the skin or mesh used in this work, i.e. the 3D model of the wrist and textures, were made available by Hugo Colauto through the Sketchfab platform[6], under the CC-BY-4.0 licence, which allows use for commercial purposes as long as the author is mentioned. This model was imported into Blender without any associated motion animation, as shown in figure 4.16. As seen previously, two MPU9250 sensors will be placed, one of them placed on the back of the hand in order to measure the orientation of the wrist movement and the other placed in the area between the wrist and forearm to ensure that there is no unwanted movement of the forearm during exercise, so it is necessary to create a skeletal system that allows that only parts of the model are animated as the movements are performed during the physiotherapy exercises.



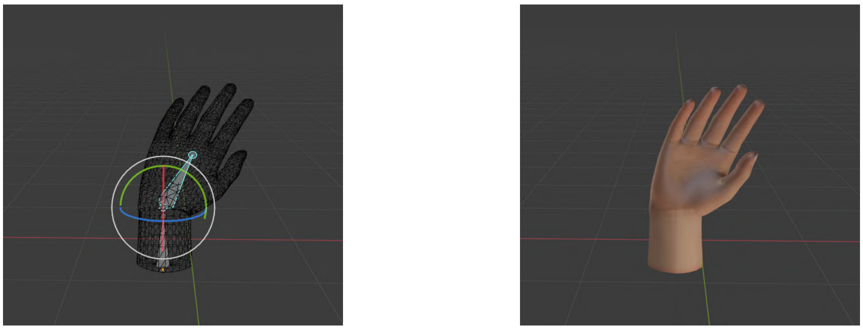
(a) (b)

Figure 4.16: In Blender software it is possible to apply the texture to the model, as well as visualise the model in wireframe mode to observe the structure of the skeleton that was created to perform the rotation of the wrist. (a) Wireframe preview. (b) Material preview

It is important to note that rigging, is a technique used in skeletal animation where a control is added to a model, using a hierarchical set of interconnected bones that allow the model's movement to be handled. A skeleton was created for the 3D model of the wrist, called armature [54], which consists of two interconnected bones, one bone being associated with all the vertices of the hand region of the model, and the other bone with the remaining vertices of the model that are located between the wrist and forearm region, with the purpose to deform the mesh. The initial idea was that the MPU9250 sensor placed on the back of the hand would animate the bone associated with the hand region on the model, and the other MPU9250 sensor would be responsible for animating the bone that manipulates the vertices found between the wrist and forearm of the model. It was observed that the simultaneous animation of the bones of this skeleton caused

[6]This work is based on hands animation by Sketchfab licensed under CC BY 4.0. It is available on the website: https://sketchfab.com/3d-models/hands-animation-9f63e924253b4179adb79b117a26d385

unrealistic distortions at the junction between the two bones. The solution found was to animate only one of the bones, and therefore only the MPU9250 sensor placed on the back of the hand has a relevant role in the animation of the 3D model of the wrist. Thus, it was possible to obtain a slightly more realistic animation, which consists of hand movements conditioned to the union of the two bones of this skeleton, which represents the wrist and can be seen in the figure 4.17. One possible way to animate the whole model without any distortion problems would be to add new sensors to bring the rotations of the 3D model closer to reality.



(a) (b)

Figure 4.17: Wrist model rotation movement with texture and wireframe preview. (a) Wireframe preview. (b) Material preview

The 3D model is exported in the GL Transmission Format extension (gltf), a standard file format for 3D models and scenes, which unlike other formats such as OBJ and FBX, is very efficient in transferring 3D model data and has several features. Some of these features are the support of vertices, basic materials, texture coordinates, hierarchical objects in the 3D scene structure, skeleton structure which is essential in this application, among others [27]. This model will then be loaded into the application through the support of the React Three Fiber library[7] for React Native, developed by Cody Bennett and which has several contributors. It is important to be aware that Reacth Three Fiber works with Three.js[8] and therefore uses the Three.js coordinate system where the x, y and z axes represent the left and right direction, up and down direction, and finally forward and backward perspective, respectively. In turn, Blender uses a different coordinate system, as the x, y and z axes represent left and right direction, forward and backward movement, and up and down direction, respectively. However, when a scene is exported from Blender to the glTF file it is possible to convert the coordinates by swapping the Y and Z axes, and it is also possible to change the coordinate system of Three.js in the application code in order to make the 3D model compatible when exported and made available in the application.

[7]React Three Fiber is a React Native library to support 3D animations. It is available on the website: https://github.com/pmndrs/react-three-fiber

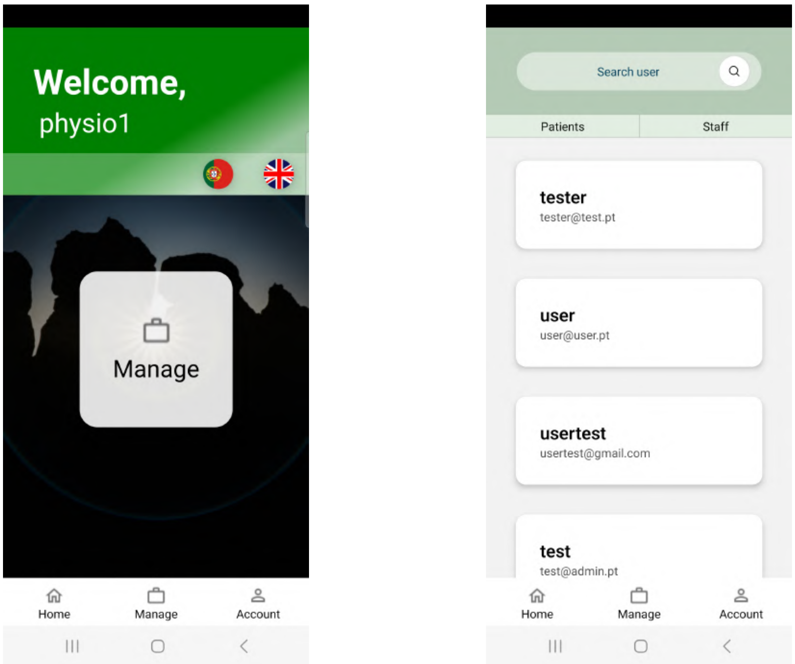
[8]Three.js is a JavaScript-based WebGL engine used to create and display animated 3D computer graphics content on the web. It is available on the website: https://threejs.org/

An error occurred in the physiotherapy exercise interface when the application switched to the background of the mobile device after an exercise was started. It was due to the accumulation of requests to animate the 3D model with the values coming from the communication with the microcontroller via BLE. This way, a monitoring of the application's operating state on the mobile device was introduced in the application code in order to restrict the transmission of values to handle the rotation of the 3D model only when the application is running in the foreground.

4.2.2 Physiotherapist Application

The interfaces of the physiotherapist application can be accessed when authentication is done with a physiotherapist account, and this way a new page is presented where it is possible to manage different system resources, such as users, training plans, physiotherapy exercises, and evaluations of the patient's progress in the exercises performed.

In this management interface it is possible to access the list of all users in the system, as well as search for a specific user by their username or email address. It is also possible to filter the list of users by patient or physiotherapist in order to facilitate the process of finding the desired user, as shown in figure 4.18.



(a) (b)

Figure 4.18: Interfaces of the physiotherapist application. (a) Home page. (b) Page to search for any patient or physiotherapist in the system

Each physiotherapist can access and observe any physiotherapy plans and exercises, regardless of whether it was created by themselves or by other physiotherapists. However, physiotherapists can only create, edit and delete physiotherapy exercises from plans created by themselves, as well as edit and delete plans of their own authorship. This way, physiotherapists have read permission for all plans and exercises, but only management permission for their own plans and exercises. The physiotherapists can also delete a patient's progress on a training plan created by themselves, and this functionality exists in case the exercise has been poorly performed or made by mistake and the record of progress is not necessary for the evaluation. In a situation of absence or problem related to a specific physiotherapist, a substitution can be made, as physiotherapists have access and can observe a patient's progress in the exercises created by another physiotherapist. Therefore, the substitute physiotherapist can make an evaluation of those physiotherapy exercises regarding the patient's rehabilitation process, and if further treatment is needed, new plans with new exercises can be created based on that evaluation.

The physiotherapist can access a patient's information in order to create and manage the patient's training plans, however, when accessing their own information as a physiotherapist there is an option to create a training plan, but it is necessary to select which patient the plan is intended for, because only users who are patients can perform training plans. The option to select and associate a patient to the plan allows the physiotherapist to transfer a plan to another patient by editing this field after creating the plan. If the transfer is performed at the same time that the previous patient on the plan has started the plan and is performing the physiotherapy exercises, it will result in the patient to be notified of this change and that the plan cannot be completed. In the interface concerning a physiotherapist's information, it is possible to see all the plans created by that physiotherapist and which patients are associated to the training plans. On the other hand, if the interface is regarding a patient's information, it is possible to see all the plans that have been created for that patient. By accessing a plan, it is possible to see information about the physiotherapy exercises associated to the plan, as well as manage the exercises in case the physiotherapist of the current session is the author of the plan. Finally, when accessing a physiotherapist exercise, an interface with the patient's progress is displayed. The interfaces for the plans and physiotherapy exercises management can be seen in the figure 4.19.

The patient's progress during the physiotherapy exercises in a given plan can be observed when the physiotherapist accesses the information regarding each exercise. This interface includes a bar graph with the different angles achieved during the various repetitions, general statistics, and a list with detailed information about each repetition.

The bar graph shows the angles achieved by the patient for each well-performed repetition and it is possible to filter the data provided on the graph by selecting only the repetitions of a certain set in order to make a more focused evaluation of the progress in that set, as can be seen in the figure 4.20. This way, the physiotherapist can focus the observation on the general progress of the whole exercise, as well as only on a specific part of the exercise. The angle presented in each repetition consists of an average of the angles obtained from the several readings performed by the wearable motion-sensing prototype component during the objective angle validation time interval.

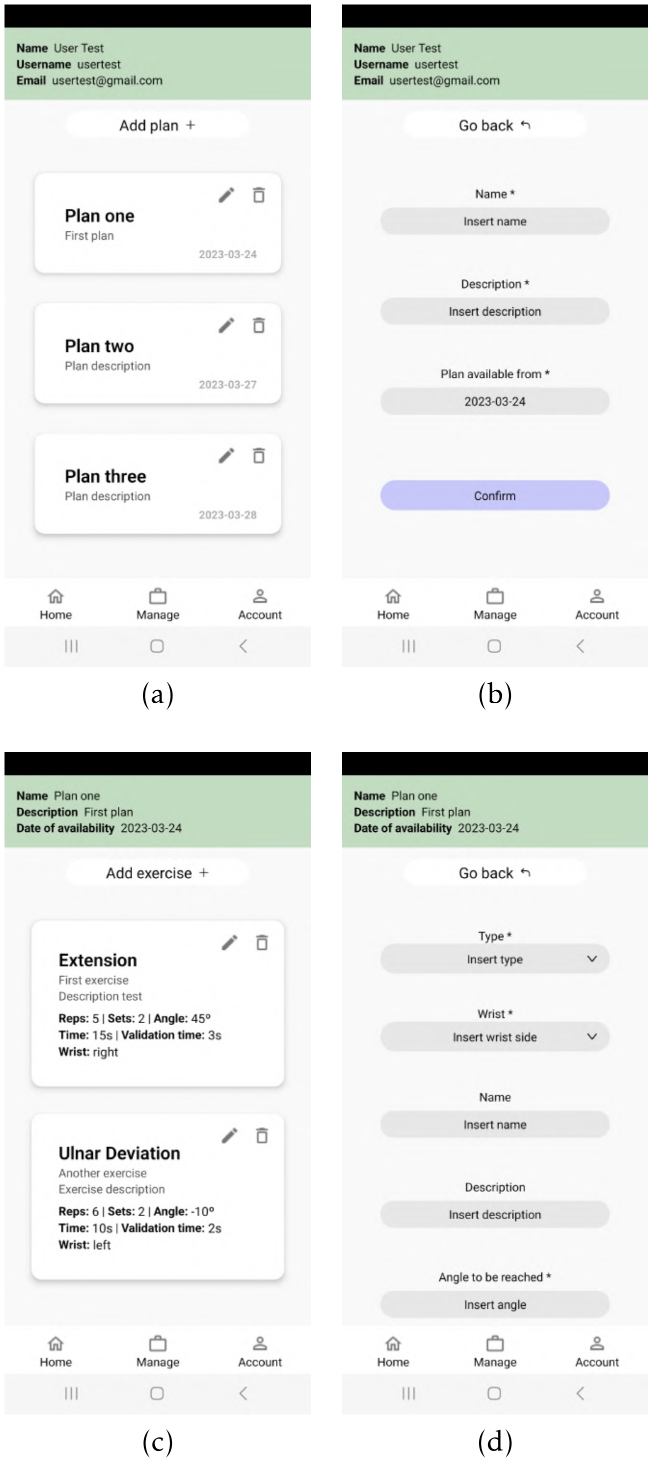
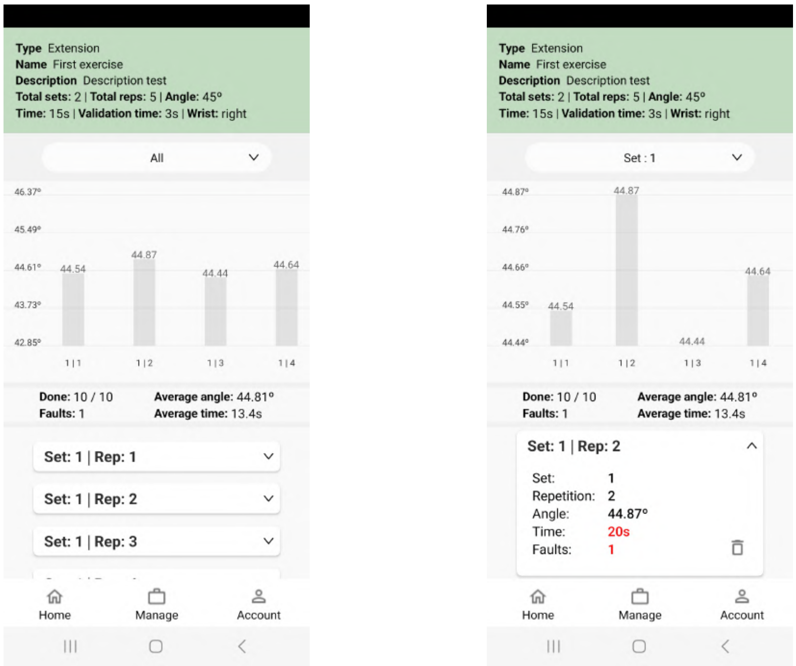


Figure 4.19: Interfaces of some pages where it is possible to manage plans and exercises. (a) Page of a patient's plans. (b) Interface to create plans for a patient through a physiotherapist's page. (c) Exercise list of a certain plan. (d) Page to add a new exercise to the plan



(a) (b)

Figure 4.20: Interface that displays the patient's progress during a given exercise. (a) Overall progress of all repetitions. (b) Summary of a repetition and a filtered graph to show only one of the sets of the exercise

In addition to the information available in the bar graphs, there are general statistics about the exercise that include the number of successfully completed repetitions, the total number of faults during the movements, the average of the angles performed in the repetitions, and also the average time interval for the execution of the movement. In a situation where the exercise is cancelled or paused and never resumed, it is possible to check how many repetitions were performed and how many were missed. The number of faults consists of the number of times a movement was considered incorrect due to a movement of the forearm during the exercise. With the information of the average angle and the average time interval to perform the repetitions, it is possible to understand if the patient was able to stabilize the wrist relatively close to the objective angle, and also if the movements were performed in the time intervals intended by the physiotherapist.

Finally, there is a list with detailed information regarding each repetition performed by the patient throughout the exercise. This way, the physiotherapist can observe the exact time, angle, and number of faults in each repetition. With the detailed information, it can be seen whether the patient took too long to perform the movement due to faults that occurred during the movement, or which repetitions were performed too fast.

Through the records of the progress of all physiotherapy exercises in a plan it is possible to conclude if the plan was well achieved or if there was any problem or inability of the patient to complete all the exercises. Thus, the physiotherapist can make a diagnosis of the patient's evolution regarding the wrist rehabilitation process, create new plans and, if necessary, change the intensity of the exercise according to the patient's condition.

It is important to note that a physiotherapist cannot be a patient, so the system does not allow the creation of training plans for physiotherapists, and consequently a physiotherapist cannot perform exercises. Thus, the aim of the physiotherapist application is to manage patients, plans, physiotherapy exercises and to observe the patients' progress.

4.3 Server and Database

The server component consists of a Spring Boot server implementation in Java that handles the communication between the application component and the database, which is a MongoDB instance. Besides all the advantages that both the Sring Boot server implementation and the MongoDB database can bring to the system, one of the choice criteria was to already have some experience in its use and implementation.

4.3.1 Server

The HTTP requests made on the mobile application reach the Spring server controllers, therefore four controllers were created so that the requests related to users, training plans, physiotherapy exercises, and exercise evaluations were received in a structurally independent way, being mapped according to the url's /users, /plans, /exercises, and /evaluations, respectively, with the prefix of /rest.

In the server implementation, the data object is handled in two ways, which is by data classes called DTO and DAO. When an HTTP request containing data arrives at a controller, it is processed as a DTO class, and therefore it has no business logic methods, only being used as simple, lightweight data transfer mechanisms. When the controller communicates with the service layer, these objects are converted into a DAO. There are four DAO classes, one for each object that will be stored in the database, these being, users, plans, exercises and evaluations. The system has five DTO classes, of which four are used to receive information from these four types of data objects, and another to receive and process the data sent in an authentication request. This DTO object used for authentication has a variable for the username or email address and another variable for the password, as well as the corresponding setters and getters functions. Any data object stored persistently in the database has a field called id which consists of a UUID that is a randomly generated 128-bit value used to uniquely identify each object, and which is then converted to a string type.

All fields of a **user** are required except the name, which is optional. The role field defines whether a user is a patient or a physiotherapist, however, any registration made in the application automatically assigns the user as a patient. For security reasons it is not possible to change the role field via HTTP request, and so if it is necessary to register physiotherapists in the system it must be done directly in the database. The id and username fields as well as the email are unique, and the email is validated by a regex that forces it to have a proper format with the local part placed before the at sign and the domain name placed after the same sign with a dot in its composition. The BCrypt encryption mechanism is used to encrypt the password during a user's registration to the system. This algorithm is designed for password hashing and is used to hash and salt passwords securely, meaning that a piece of random data is added to the password input before it is hashed and stored.

A **training plan** has three fields that are required and inserted by the physiotherapist, these being the name, description and date. The date defines the availability of the plan, that is, from which day the plan will be available to be performed. The fields doctorId and patientId, corresponding to the physiotherapist and patient identifiers, respectively, are automatically filled in, as well as the done field that defines whether the plan has already been completed or not. When creating a plan, the patient's identifier is received in the body of the request together with the rest of the information, while the physiotherapist's identifier is obtained through the session token that is received in the "Authorization" header with the suffix "Bearer". It is not possible to create a training plan if the physiotherapist identifier does not correspond to a user with the physiotherapist role, as well as if the patient identifier does not correspond to a user with the patient role.

A **physiotherapy exercise** stores the information regarding the identifier of the training plan to which it belongs and is automatically filled in during the creation request. The name and description fields are optional, while the rest of the information must be provided by the physiotherapist. It is necessary to define the type of exercise, which can be one of the four physiotherapy exercises proposed in this work, which are the flexion, extension, ulnar and radial deviation movements, and indicate whether the movement should be performed with the right or left wrist. It is also required to specify the number of sets, the number of repetitions in each set that are the same for all sets, the angle to be reached during the exercise, and the respective time required for its validation, as well as the recommended time to complete a repetition.

An **evaluation** consists of data regarding the progress made by the patient during the repetition of a set within a physiotherapy exercise. In other words, each evaluation contains information about the time it took the patient to perform the repetition, the angle that was reached and maintained by the patient to validate the repetition, the number of faults, as well as the identifier of the physiotherapy exercise, and also the repetition and set numbers so that the physiotherapist can identify what the repetition was and be able to observe how the patient performed it. All fields are required, and are automatically filled in when a training plan is completed.

The controllers map and handle the HTTP requests, and then data is converted from DTO to DAO classes and sent to the service layer. As well as controllers, there are four services and these are intended to handle the business logic, i.e. the business functionality regarding user data, plans, exercises and evaluations. This way, the database can be accessed and managed through CRUD operations in order to return exactly what is intended by the HTTP request.

The Spring Security framework[9] was configured and used on the server, which features a very powerful and customizable authentication, authorization and access control structure to add a layer of security at the application level. The authentication in the application is done with the user's username or email address and password credentials, and if the authentication is successful, a Json Web Token (JWT) is generated and returned, which defines a compact and self-contained way to securely transmit information between two sides as a JSON object. One of the claims used in the JWT has information regarding the user's username. The algorithm used to sign the JWT is called HS512, which is a json web algorithm name for Hash-based message authentication code (HMAC) using SHA-256. The HMAC is a cryptographic authentication technique that uses a hash function and a secret key, in which both the sender and the receiver have the same private key that is used to encrypt and decrypt the message on both sides. When a login request is made, a comparison is made between the password input and the user password stored in the system, both encrypted with the BCrypt algorithm, in order to validate the authentication.

There are two types of roles used to manage and access server resources, called ROLE_PATIENT and ROLE_DOCTOR, which correspond directly to the users' role field, respectively. These roles define whether a given user is allowed to access a server endpoint. The only requests that do not require authentication are the login and register requests, which are distinguished from other endpoints in the users' controller by having /auth in its url. All the requests to manage training plans, exercises and evaluations are restricted to physiotherapists, and the endpoints in the controllers have /doc in the url composition. All other HTTP requests only require a valid authentication, that is, a valid token from either a physiotherapist or a patient.

If an error occurs in any HTTP request, the server has an exception handler to correctly process each error with customizable messages. All requests that return lists of objects use pagination in order to return data dynamically, as well as to reduce loading times. The endpoints that are used to obtain these data structures have the possibility to add some parameters that allow to define the size of each page, a specific page number, and in some cases, an additional parameter to filter the list. An example of this additional parameter is in the case of the training plan listing where it is possible to filter only the plans of a specific patient. The size and number of pages parameters are optional, and by default a maximum of 10 first page items are returned. The structure returned by these requests consists of a Map<String,Object>, with information regarding the data list, the current page number, the total number of all items of the same type stored in the database, and the total number of pages according to the size requested in the HTTP request. Thus, it is possible to have a better insight into the data existing in the system.

[9] Spring Security is used to implement application-level security in Spring applications. It is available on the website: https://docs.spring.io/spring-security/reference/index.html

4.3.2 Database

The database was developed in MongoBb[10], which is an open source, documentation-oriented database. As a non-relational database (NoSQL), it is ideal for storing large amounts of data and it is horizontally scalable. A disadvantage of this type of database is that it is not recommended for executing complex queries, however, these types of queries are not needed in this system.

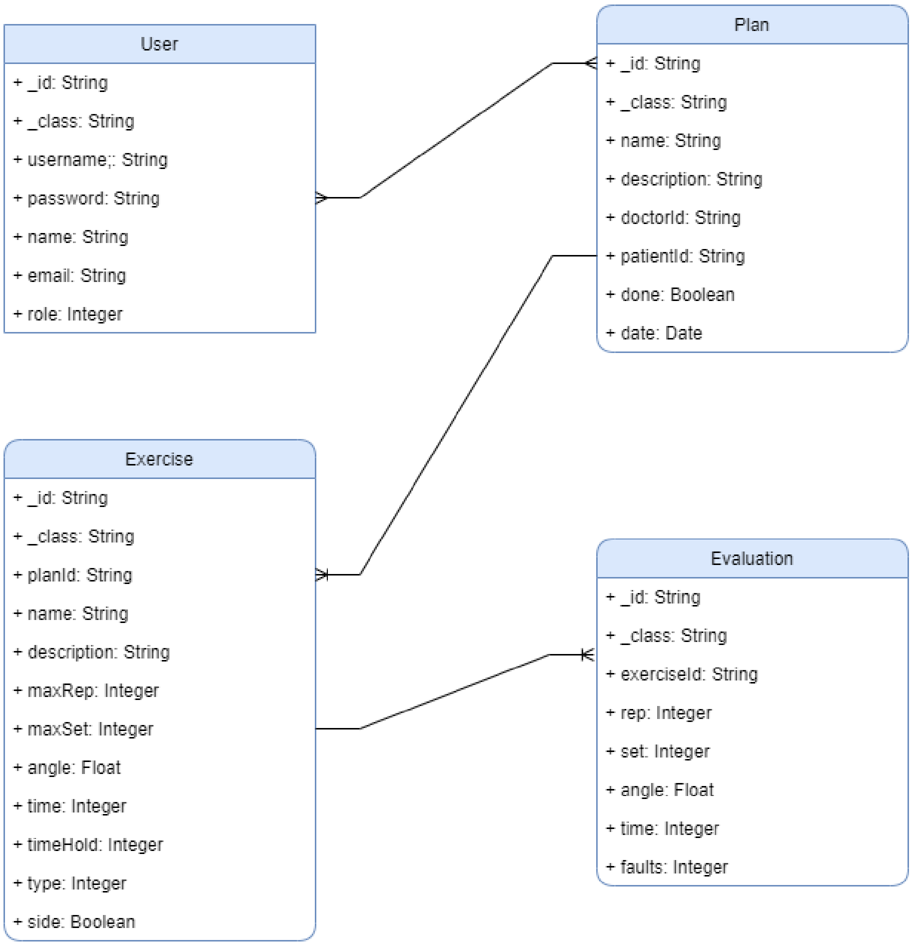


Figure 4.21: Database diagram

The database has four data collections, as shown in the diagram presented in figure 4.21, for storing users, training plans, physiotherapy exercises, and the exercise evaluations, referred to as users, plans, exercises, and evaluations, respectively. The data stored in each data collection is similar to the DAO data classes that the server uses to perform CRUD operations on the database.

Overall, a user can have zero or multiple training plans, and a training plan has many users, these being the patient who will perform the plan and the physiotherapist who created it. In turn, a training plan can have zero or more physiotherapy exercises, however,

[10] MongoDB is an open source NoSQL database management program. It is available on the website: https://www.mongodb.com/

an exercise is associated to only one training plan. Finally, a physiotherapy exercise can have zero or more evaluations, which consist of information about each repetition done by the patient. On the other hand, an evaluation is only associated with one physiotherapy exercise.

The server and database component will be running locally during the testing phase and all requests will be made within the same local network. Since no configurations have been made to make the server accessible via the Internet, it is necessary to ensure that HTTP requests from the mobile application are made to the IPv4 address of the local machine that hosts the server, and it is also necessary to take into account that Wi-Fi and Ethernet connections can result from connections outside the same network, and in some situations problems can arise due to this local communication. There is the possibility to host the server and database component on a cloud platform that will store and process the system data, as well as avoid restrictions on the use of the mobile application.

4.4 Summary

The system development began with the wearable motion-sensing prototype component in order to ensure that the reading of the sensors was done properly and that there was no issues with the two sensors reading simultaneously. Then, the application component was developed, more specifically the patient interface, to establish communication with the wearable motion-sensing prototype component. After that, the server and database component were implemented to allow the storage of persistent data, as well as the physiotherapist's application interface for the management of the data. It is interesting to have a final view of how all the pieces interact and what technologies were used in each layer. The figure 4.22 provides an overview of the final architecture of the system.

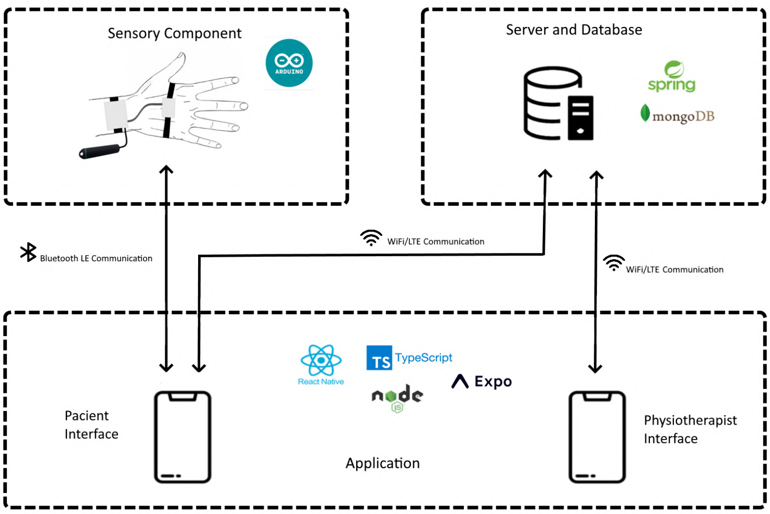


Figure 4.22: Final System Architecture

The system was developed and executed within a local network, so the server was hosted on a personal machine and the application requests sent over Wi-Fi on the same network. Once everything was working properly, the Android APK was generated so that the application could be easily installed on any mobile device. With the wearable motion-sensing prototype component reading, processing and sending the data correctly to the application, and all the information was flowing dynamically through the server and database, the system was ready to be tested and evaluated, as will be presented in the next chapter.

5

Evaluation

This chapter will describe the system evaluation and the results obtained. It is important to ensure that the system is working properly and has no limitations and problems in runtime. Several tests were performed to ensure that each layer of the system is performing well, such as integration tests and usability tests.

Integration testing consists of evaluating the overall functionality of the system and verifying that the entire system and communications are working properly, while usability testing verifies and ensures that the usability of the system is satisfactory in an environment with users using the application.

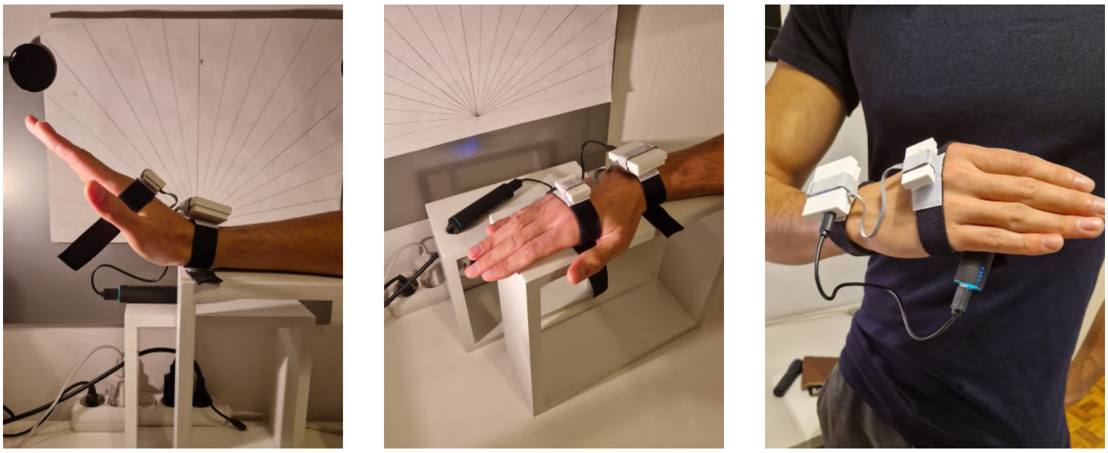
Initially, some tests were carried to find the best way to use and position the wearable motion-sensing prototype component on the wrist in order to take more correct readings, as well as to promote greater comfort and less interference during physiotherapy movements.

5.1 Sensor placement tests

As seen in the previous chapters, the wearable motion-sensing prototype component consists of two small boxes placed on the back of the hand and between the wrist and forearm, identified by the I2C addresses of the IMU sensors that are stored inside each box. One of the challenges was to find the best position of the box placed on the back of the hand, which has an IMU identified by I2C address 0x68, since it is the sensor responsible for measuring the orientation of the wrist movement.

The placement of this sensor was tested in three different ways, as explored in the work and study [21] conducted by Fong Jin Hiung and Idris Mat Sahat described in chapter 2, with these locations being the central area of the back of the hand, an area closer to the fingers, and another area closer to the wrist, referred to end, front and middle, respectively.

In order to validate the exact angle made by the wrist, manual measurements were performed and compared to the data collected by the sensors. The manual measurements were made by reading the angle reached by the wrist using a sheet of paper with different markings of the various angles. This way, the wrist joint was centered in the center of the referential, and as the wrist movement was made, it was verified that the angles collected by the system were equal to the angles reached on the sheet with the angle markings. The figure 5.1 shows the use of the wearable motion-sensing prototype component during this testing phase.



(a) (b) (c)

Figure 5.1: Wearable motion-sensing prototype component used on the right wrist during the testing phase. (a) Side view. (b) Top view. (c) Overview

The results obtained in this initial study were positive when the device was placed in all three positions, as the values collected by the sensor were very close to the expected angle. The central area of the hand proved to be the most adequate for the four physiotherapy movements proposed in this work. When the sensor was placed closer to the fingers, some complications occurred during the flexion movement, as the sensor could barely read angles above 70º. It must be taken into account that in the flexion movement the maximum wrist angle may reach up to 90º, and therefore this position of the sensor will not be efficient for exercises requiring these degrees of amplitude. When the sensor was placed at the closest part of the wrist, there were some problems during the ulnar deviation movement, with the problem again being related to the incapability of reaching higher wrist angles, with the maximum expected angle in this movement being 45º. In this case, values above 30º were almost never read. Therefore, the central area of the back of the hand was defined as the most suitable for the use of this prototype sensor, since, in general, the measurements obtained during the tests in this position were more correct for the various physiotherapy exercises than in the other positions. The second sensor placed between the wrist and forearm area also proved capable of performing the function of detecting any movement of the forearm.

5.2 Functionality tests: evaluation and correction

The system functionality testing can be achieved through integration testing, which will evaluate and measure all functionality and ensure that all system components are working and communicating correctly. These tests can be specific, to independently evaluate certain modules, but also to evaluate the interaction between several components at the same time.

This way, the functionality tests were conducted to validate and correct any problems that may arise during the execution of the exercises, as well as to determine whether the wearable motion-sensing prototype component is correctly collecting and processing data regarding the orientation of the movements. Therefore, three different angles were chosen for each physiotherapy exercise, and ten repetitions were performed for each angle. The aim of these measurements is to evaluate if the proposed angles were reached correctly, that is, if the values of the angles measured by the sensors correspond to the real angle made by the wrist. As explained in the previous section 5.1, the manual measurements were made using a sheet of paper with markings of the different angles to validate and compare the actual angle made by the wrist with the angle obtained by the sensors. An angle is considered as correct when it corresponds to the angle measured manually. A threshold of 3º was used to validate a performed angle, and it was necessary to maintain this angle for three seconds to complete the repetition. This threshold is used because achieving and holding an exact angle for a certain period of time is almost impossible due to the sensitivity of the sensors.

It is also important to note that the sensors were calibrated and that the data was processed on two occasions, as explained in chapter 4, one time in the wearable motion-sensing prototype component and the other in the mobile application. In the wearable motion-sensing prototype component an average is made of the different values collected by the sensors which is then sent to the application in time intervals of approximately 500 milliseconds. On the other hand, in the mobile application, an average is made of the angle values obtained during the validation time interval, which in this case is 3 seconds, and therefore an average of about six angles is made, since it is the result obtained from the quotient of 3 seconds by 500 milliseconds. These angles are exactly or very close to the objective angle up to a limit of 3º difference, which is the set threshold value. In all physiotherapy movements in which the hand is moved down towards the earth, these movements being flexion and ulnar deviation, the angles made by the wrist have negative values. On the other hand, when movements are made upwards towards the sky, i.e. extension and radial deviation movements, the angles are positive. As seen in chapter 4, the angle used to measure the inclination of the wrist during the four exercises is the pitch angle which corresponds to the rotation about the y-axis, since all the movements consist of moving the hand up and down. In any of the movements it is necessary that the forearm is fixed on a surface and, if it is moved, it is given as an incorrect movement, and it has to be corrected to resume the repetition so that the angle achieved with the wrist is not conditioned to a bad positioning of the forearm. It is also necessary to point out that for these tests the right wrist was used, although the system works equally well for the left wrist.

An Android smartphone with Android version 12 (API level 31), Bluetooth 5.2 and BLE compatibility was used to run the application and observe the angle values displayed on the exercise interface and compare them with the values shown on the arduino monitor series.

5.2.1 Flexion

In the flexion movement, three angles were chosen for this evaluation, which are -20º, -40º and -60º. As seen in chapter 2, the wrist angles performed during the execution of this movement range from 0º to 90º on a negative axis, with the forearm in pronation and the elbow bent to approximately 90º. The wrist flexion movement consists of moving the hand down until the wrist is at an angle of approximately -90º, and then lifting until it returns to the initial or neutral position of 0º in order to start a new repetition. This evaluation can be seen in the figure 5.2.

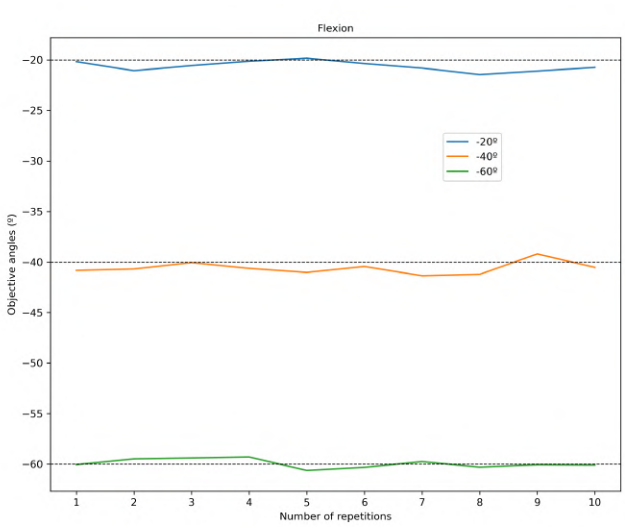


Figure 5.2: Flexion results

5.2.2 Extension

In the extension movement, three angles were chosen for this evaluation, which are 25º, 40º and 60º. The wrist angles performed during the execution of this movement range from 0º to 70º, with the forearm in pronation and the elbow bent to approximately 90º. The wrist extension movement consists of moving the hand upwards until the wrist is at an angle of approximately 70º, and then lowering until it returns to the initial or neutral position of 0º in order to start a new repetition. This evaluation can be seen in the figure 5.3.

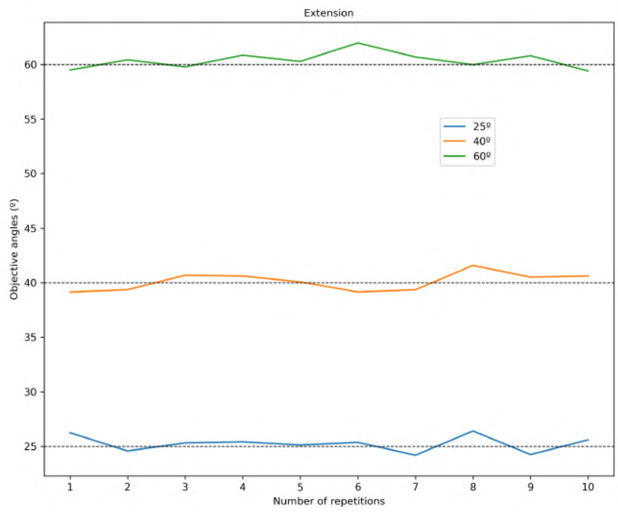


Figure 5.3: Extension results

5.2.3 Radial deviation

In the radial deviation movement, three angles were chosen for this evaluation, which are 5º, 10º and 15º. The wrist angles performed during the execution of this movement range from 0º to 20º, with the forearm in a neutral position, that is between supination and pronation, and the elbow bent to approximately 90º. The wrist radial deviation movement consists of moving the hand upwards until the wrist is at an angle of approximately 20º, and then lowering until it returns to the initial or neutral position of 0º in order to start a new repetition. This evaluation can be seen in the figure 5.4.

5.2.4 Ulnar deviation

In the ulnar deviation movement, three angles were chosen for this evaluation, which are -10º, -25º and -40º. The wrist angles performed during the execution of this movement range from 0º to 45º on a negative axis, with the forearm in a neutral position, that is between supination and pronation, and the elbow bent to approximately 90º. The wrist ulnar deviation movement consists of moving the hand down until the wrist is at an angle of approximately 45º, and then lifting until it returns to the initial or neutral position of 0º in order to start a new repetition. This evaluation can be seen in the figure 5.5.

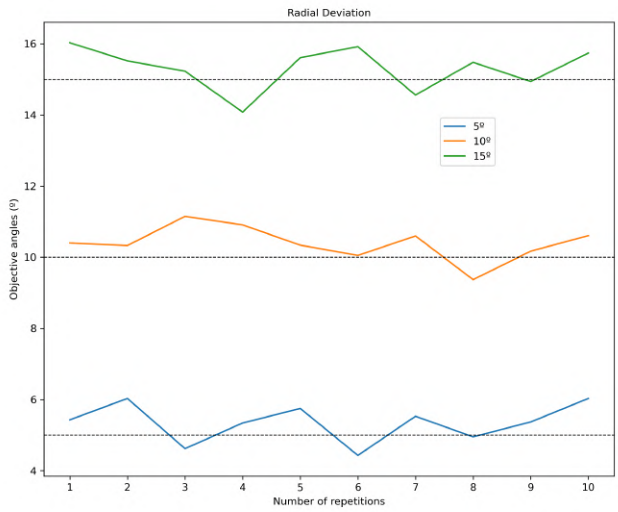


Figure 5.4: Radial deviation results

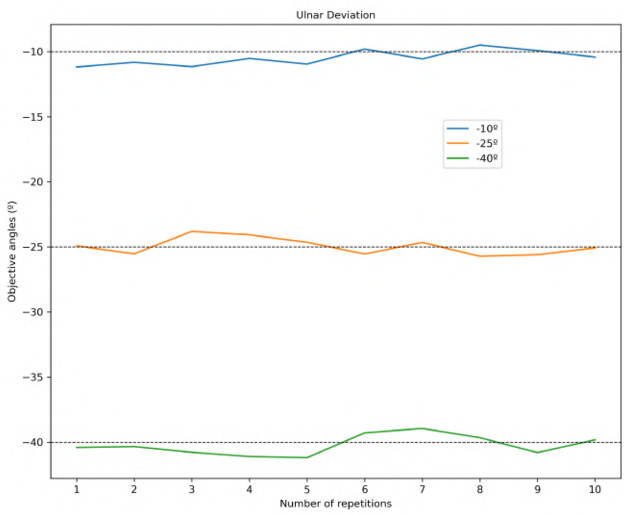


Figure 5.5: Ulnar deviation results

5.2.5 Results

Overall, the functionality tests were successful. All the angles performed by the wrist and displayed in the application were the same as their manual measurement, which means that the sensors are correctly collecting data from the orientation of the wrist movement. It can be seen from the graphs that the angles achieved oscillated very close to the target angle and within the defined limits, which is a very positive result.

The angles chosen to evaluate each type of exercise are within the range of angles that it is possible to make with the wrist when performing these movements, however, it was also tested and verified if it is possible to reach the maximum angles of these exercises with this system. Some problems occurred due to the wearable motion-sensing prototype component being used with velcro strips, and it was found that in situations where the angles were steeper, the velcro connections loosened slightly, which made it more difficult to get some angle readings, especially at the -90º angle of the flexion movement.

It is also necessary to consider the position of the sensor that will measure the movement of the forearm, since if it is placed too close to the wrist, movement limitations may occur, especially in the extension movement. On the other hand, any touch to the box that stores this sensor can cause the orientation of the sensor to oscillate and cause a movement that is considered to be incorrect. In summary, the system proved to be suitable and correct in reading the different angles selected for each physiotherapy exercise proposed in this work.

5.3 Usability tests

The interfaces of both the patient and physiotherapist application should run on an Android application on a smartphone, with the requirement that the android version must be higher than version 5.0 (API 21+) due to the Bluetooth communication support library function used in the application to negotiate the MTU size during communication between the application and the wearable motion-sensing prototype component.

In order to evaluate the system and obtain feedback from users, a set of user stories was developed, consisting of descriptions of tasks with the information needed to perform them. This way, a potential user can test specific parts of the system in order to identify whether the system presents a useful and intuitive interface. It is important that an examiner is present during the execution of these tasks to register any difficulty or problem the user may encounter.

The questionnaires were also an excellent tool for understanding whether the system was suitable for a user to use. This way, potential users can directly test the functionality of the system and answer a given questionnaire in order to evaluate the ease of use and intuitiveness of the application interfaces and wearable motion-sensing prototype component. Both patients and physiotherapists were asked about their experience when performing their user stories and also the System Usability Scale (SUS)[1], which is a reliable tool for measuring usability and provides an overview of subjective usability evaluations, was also used.

5.3.1 Pacient

The patient's user stories aim to evaluate a patient's actions when using the mobile application, as well as the whole process of connecting and using the wearable motion-sensing prototype component during physiotherapy exercises. Scenarios were created for the patient's functionalities to be used and tested, and some questions were also asked to get feedback and make an evaluation of the patient's experience when using the system.

5.3.1.1 Scenarios and questionnaire

In this assessment, the app is pre-installed on a smartphone given to the patient, and the patient will be asked to perform some scenarios. The reaction, concerns and problems shown by the patients will be registered. The following user stories were performed for the patients.

1. The patient wants to log in to the system with the credentials "patient1" and password "123", and go to the plan list page. Finally, the patient wants to see if there are any plans available for today. The story ends when the patient starts a training plan with the name "Plan one" that is scheduled for the day of the evaluation.

2. The patient wants to log in to the application and complete a training plan available for today. If the patient is not already logged into the system, the patient should log in with the credentials "patient1" and password "123". After authentication, the patient will check if a plan is already in progress, and if the patient finds one that is in progress and is not the "Plan one" plan, then the patient should cancel the plan, otherwise the patient should resume or start the "Plan one" plan. The patient must perform all physiotherapy exercises and finish the plan. The story ends when the patient returns to the plan list and checks that the plan that was executed is indicated as completed.

In order to evaluate and get feedback on the experience of using the system, some questions were asked following the performance of the patient in the user stories. The following questions were asked, where a rating between 1 and 5 is required, with 1 if the patient strongly disagrees, and 5 if strongly agrees.

1. Was it intuitive and easy to access your plan page and start a physiotherapy exercise?

2. Was the wearable motion-sensing prototype comfortable and easy to use?

[1] The system usability scale (SUS) provides a reliable tool for measuring the usability. More information is available at https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

3. Was the process of connecting the wearable motion-sensing prototype to the mobile device and sensors calibration easy to accomplish?
4. Regarding the interface presented while performing the physiotherapy exercise, with progress bars, interactive information and 3D model of the wrist to have a better follow-up during the movements in real time. Did you find the interface helpful and intuitive to finish the exercise successfully?
5. Any suggestions for improving or adding new features that you think would be useful to make the system more functional and useful given the scope for which it was designed, that is, to perform physiotherapy exercises in the right way and at the right time intervals?

5.3.1.2 Results

The tests were carried out on twelve patients, aged between 21 and 60 years, and overall the results were positive.

The navigation and interaction with the different patient interfaces in the application were positive and all patients were able to easily access their plans and exercises, as well as start a physiotherapy plan and exercise without any difficulty. Patients were able to identify the status of each plan and exercise through the icons and buttons available on the interfaces, so there was no difficulty in understanding the status of the progress of the plans and exercises after their completion, as can be seen in figure 5.6.

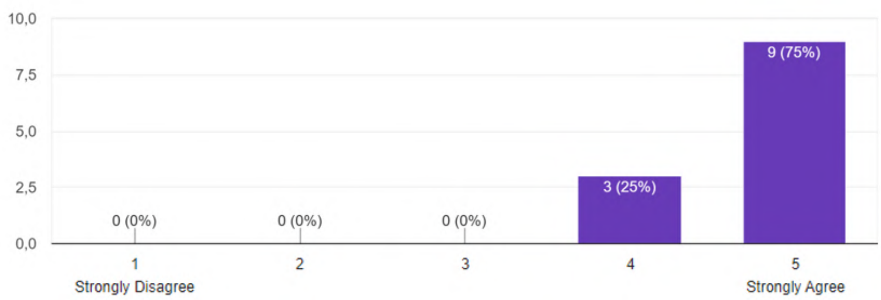
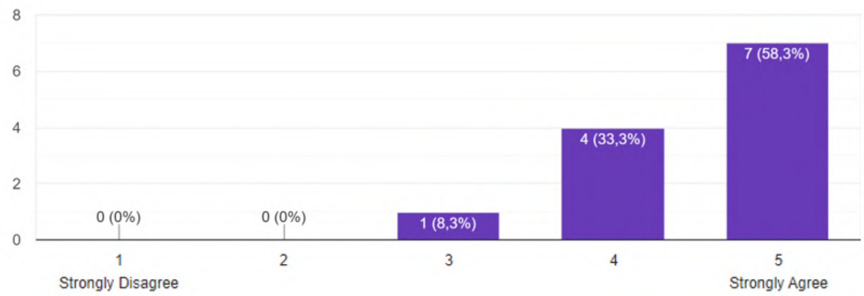


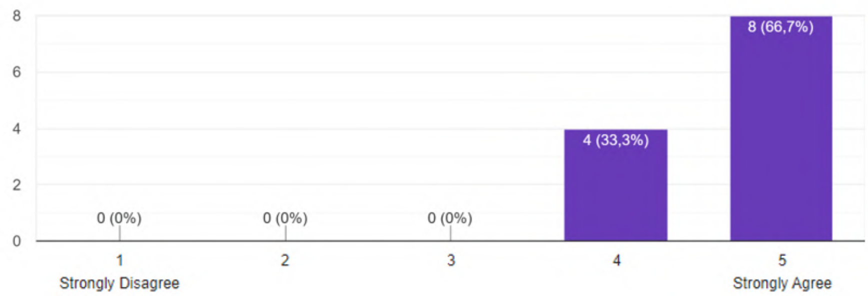
Figure 5.6: Graph of the patients' evaluation on the first question asked

The phase of placing and connecting the wearable prototype to the mobile device was explained in advance by the observer, as the physiotherapist is expected to give initial explanations and instructions before a patient uses the application independently. Through the various interfaces showing the progress and steps to take to establish the connection and calibrate the sensors, all patients were able to successfully perform these steps and proceed to the interface to perform the exercise. Overall, the wearable motion-sensing prototype was found to be comfortable and easy to use, as shown in figure 5.7.

The interface presented during the exercise is one of the most important interfaces of the application, where a patient is expected to successfully and without difficulty perform



(a)



(b)

Figure 5.7: Graphs of the patients' evaluation regarding the wearable motion-sensing prototype. (a) Second question. (b) Third question.

all repetitions and sets of a given exercise. Since this interface has several interactive components, it is important to understand if these components are intuitive and enough for the patient to have a great experience. The audio component was helpful as the patient did not have to be fully aware of the visual interface of the application to understand the state of the exercise while performing the movement. Due to the fact that the 3D model component decreases the performance of the application because of the many renderings that are performed as the data is collected by the sensors and imported into the graphics component, there is an increase in the delay in displaying the angle that is currently being performed, and therefore the interactive graph component was better received. All patients were able to successfully perform the exercise to the end and finish the respective plan. The evaluations made by the patients about the interface of the exercise can be seen in the figure 5.8, and most of the feedback was related to the design and how the information was presented in the interface and not about the performance of the system's functionalities, such as placing the objective angle more prominently next to the angle being performed, and also complementing the information in the temporary messages that appear throughout the exercise with the objective angle that should be performed. Some improvements were made based on the evaluations carried out, namely the display of more temporary informative messages that appear on the screen to indicate the progress of the exercise, such as indicating whether a repetition has been started or successfully validated.

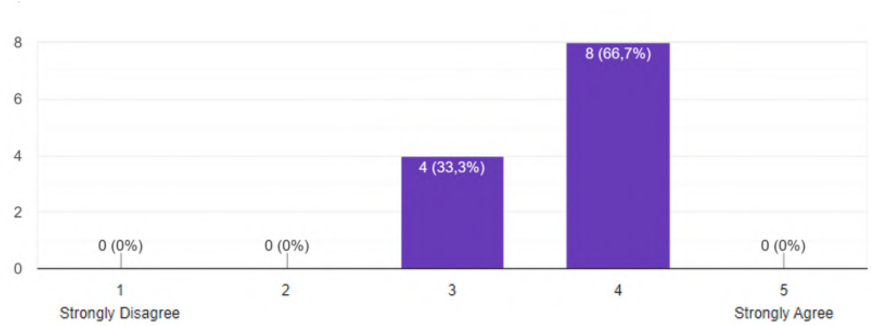


Figure 5.8: Graph of the patients' evaluation regarding the last question about the exercise interface

The last question presented in the section 5.3.1.1 does not appear in the graphs of the patients' evaluation since its purpose is to obtain suggestions for improving the system and not to evaluate certain features of the system. These are some suggestions of functionalities considered of interest by patients to be implemented in the system.

- Audiovisual support for patients with visual impairments or disabilities. Thus, it is of interest that the whole application has informative audios that explain the steps to be performed so that the visually impaired patient can perform the physiotherapy plans and exercises without any difficulty.
- System notifications to alert when a plan or exercise has been added by a physiotherapist, or if it is available to be performed by the patient.
- Use a kind of watch strap instead of the velcro straps to hold the case between wrist and forearm.
- When the movement is incorrect indicate through an image or video how to correct the problem.

After testing the patient interfaces in the application and the use of the wearable motion-sensing prototype component, the patients were asked to answer the questionnaire presented by SUS. The average SUS score is 68%, and scores below this average indicate problems with the design that need to be researched and resolved, while values above indicate the need for minor design improvements. The score of the system developed in this work was 83.3%, which on the scale is equivalent to good and very close to excellent, that would be above 85.5%.

5.3.2 Physiotherapist

The physiotherapist's user stories aim to evaluate the actions of a physiotherapist when using the mobile application, such as the management of the users, training plans, physiotherapy exercises and the monitoring of a patient's evolution. Scenarios were also created for the physiotherapist features to be used and tested, and some questions were asked to get feedback and make an assessment of the physiotherapist experience when using the system.

5.3.2.1 Scenarios and questionnaire

The application is pre-installed on a smartphone given to the physiotherapist who will also be asked to perform some scenarios designed to include the functionality present in the physiotherapist's application interfaces. The following user stories for physiotherapists were carried out.

1. The physiotherapist wants to access the system with the credentials "physio1" and password "123", and search for the patient whose username is "JohnDoe". When navigating to the page for managing the user, the physiotherapist decides to create a training plan for today with the name "Today Plan" and description "Test description". Once the plan is created, the physiotherapist adds a physiotherapy exercise of the extension movement with the name "First exercise", description "Do slow and controlled movements", the angle to be reached is 35.5°, two sets of ten repetitions, an interval time of 30 seconds to perform a repetition and lastly and a validation time of 3 seconds. The story ends when the physiotherapist creates the physiotherapy exercise.

2. The physiotherapist wants to access the system with the credentials "physio1" and password "123", look for the patient whose username is "JaneDoe" and access the information regarding the progress of the exercise with name "First exercise" from plan "Plan one". When accessing the page with the progress made by the patient during the exercise, the physiotherapist decides to look at the graph showing all the repetitions made in the first set of the exercise. The story ends when the physiotherapist sees the detailed information of the second repetition of the first set.

As was done previously for the patients, some questions were asked after the performance of the physiotherapist on the user stories in order to evaluate and get feedback on the physiotherapist's experience using the system. The following questions were asked, where a rating between 1 and 5 is required, with 1 if the physiotherapist strongly disagrees, and 5 if strongly agrees.

1. Were you able to easily access the management page of a user's plans and exercises?

2. Did you find the statistics presented for each exercise performed by a patient useful for making an assessment of the patient's progress?

3. Do you think the system is useful and improves the experience and process of evaluating and monitoring wrist physiotherapy exercises?

4. Any suggestions for improving or adding new features that you consider helpful in making an assessment of the patient's progress in the wrist rehabilitation process?

5.3.2.2 Results

The tests were performed with a specialist at a clinic that provides specialized, individual and personalized physical therapy treatments.

The purpose of these tests is to ensure that there are no problems using the physiotherapist interfaces in the application, and also to understand whether the interface that allows the monitoring of a patient's progress in a particular physiotherapy exercise is useful and intuitive.

The navigation and interaction with the different physiotherapist interfaces in the application went well, and the physiotherapist was able to easily access and manage the physiotherapy plans and exercises for a given patient. The creation of both the plans and exercises was successfully performed, and there was no difficulty in filling out the fields on the forms.

The physiotherapist easily identified and accessed the page showing the patient's progress throughout a given exercise. The bar graph showing the angles validated in each repetition was well received, as was the list of detailed information regarding each repetition. Overall, the results were positive, as can be seen in table 5.1.

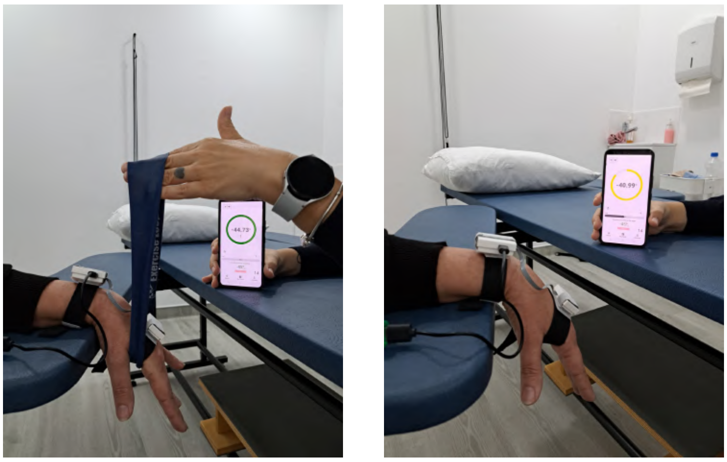
Table 5.1: Physiotherapist's evaluation on the questions asked

Questions	Answer
Was it intuitive and easy to access your plan page and start a physiotherapy exercise?	5
Did you find the statistics presented for each exercise performed by a patient useful for making an assessment of the patient's progress?	4
Do you think the system is useful and improves the experience and process of evaluating and monitoring wrist physiotherapy exercises?	4

Based on some aspects pointed out by the physiotherapist, small changes were made to the system in order to improve the diagnosis, such as adding information about the number of faults, that is, the number of times that incorrect movements occurred during the exercise, as well as highlighting with red colors these irregularities. Also, the time intervals for performing the movements were highlighted for the repetitions in which this time is exceeded or if it is less than half the time set by the physiotherapist. Some general statistics were also added regarding the performance of each physiotherapy exercise, concerning the number of repetitions successfully performed, total number of faults, average number of the objective angles performed over the several repetitions, as well as the average time interval to perform the repetitions.

Similar to the patient evaluation, the last question presented in the section 5.3.2.1 is only intended to get suggestions for improving the system and not to evaluate certain features of the system.

One last observation made by the physiotherapist was that it would be interesting to distinguish between exercises with and without load, and therefore the physiotherapist also used the application to define two different exercises for a patient, one exercise to be performed with a load and other without, as shown in the figure 5.9. In order to do this, the physiotherapist added a description to the exercises stating whether or not a load should be used. Both exercises were performed successfully.



(a) (b)

Figure 5.9: Tests performed with the physiotherapist in the clinic. (a) Weight-bearing exercise. (b) Off-load exercise

Finally, the physiotherapist was also asked to answer the questionnaire presented by SUS, and the result was 75%, which on the scale is equivalent to a good.

5.4 Evaluation of the cost of the system

A requirement of the system is that it should be accessible and used by many patients and physiotherapists, so it is of interest that the system should be built and used at a low cost. It has been made an estimate of the price of the material used in the wearable motion-sensing prototype component, as well as the cost of operating the system.

In the table 5.2 it is possible to see the price of the material used, with the total price of the production of the wearable motion-sensing prototype component being around 37.10 euros. The prices are an estimate based on the current market, and it should also be taken into account that there is currently a crisis in chip production, which has caused stock-outs and a consequent increase in prices, especially for sensors and microcontrollers. In summary, the wearable motion-sensing prototype component is composed by 1 microcontroller, 2 IMUs, 1 power bank, connection wires, 2 boxes to store the sensors and

Table 5.2: Table of an estimate price for the material used in the wearable motion-sensing prototype component

Material	Cost
1x Microcontroller TTGO T7 v1.3	7€
2x IMU MPU-9250/6500	2 x 10€
1x Powerbank Kunft 2000mAh 5V	5€
2x 3D Print boxes	3€
20cm Connecting wires	0.10€
2x Bands elastic fabric 2.5cm x 30cm	1.5€
6x Velcro strips 2.5cm x 4cm	0.5€

the microcontroller, and lastly elastics and velcro strips to use the boxes in the wrist region. There are other implementation options, such as using different microcontrollers, boards, sensors, power banks and material to use the wearable motion-sensing prototype component, which will produce similar results.

Regarding the application component, an Android smartphone is required to run the application and must have BLE compatibility, as well as an Android version above 5.0. Currently, there are several models of smartphones at affordable prices, and their use has been growing more and more. In this work the APK application was installed directly on the mobile device, however, the application installation can be provided through virtual shops, such as Google services, and this is the only cost associated with the application component, which is the cost related to software distribution.

The server can be hosted either locally or in cloud services. In this work the server ran locally, so the only cost is related to the physical machine which in this case has 6 cores, 16 Gb of RAM and 1 TB of memory, as well as power and internet costs. However less powerful machines can easily run the server and instance of the MongoDB database. Alternatively, there are several companies that provide Cloud services, such as Google, Azure, AWS, among others. This way, it is possible to use virtual machines with enough capacity and power to run the system. Cloud services allow to host the system without major complications and complex configurations, as well as there is no need to maintain the hardware, and in the short term is advantageous because the prices are affordable.

5.5 Summary

In order to make an evaluation and validation of the system in terms of correctness, functionality and usability, a series of tests were performed.

Experiments were made on the positioning of the sensor placed on the back of the hand that is performing the physiotherapy exercise with the aim of measuring the orientation of the wrist movement. It was concluded that among the three regions tested for its positioning, the central area of the hand proved to be the most suitable for the four physiotherapy movements proposed in this work, since it was generally possible to measure the angles made by the wrist, while there were complications in reading the angles in certain exercises in the other two positions.

The wearable motion-sensing prototype component was tested for validation and correctness, as it is intended to assess whether it is correctly collecting and processing data related to movement orientation during physiotherapy exercises. That said, different angles were chosen for each physiotherapy exercise and several repetitions were performed to assess if the proposed angles were achieved correctly when compared to the manually measured angle. The tests were carried out successfully, as the data of the angles performed by the wrist were read correctly by this component and did not oscillate outside the limits defined for their correct validation.

Tests were carried out with the users of the system, patients and physiotherapists, in order to obtain general feedback on the system and possible improvements, as well as to understand if the system interfaces are accessible, intuitive and useful to promote a better experience in performing physiotherapy exercises and assessing progress within the wrist rehabilitation process. This way, user stories were selected for both types of users and some questions were asked.

Overall, the system components are easy and intuitive to use, and many of the problems encountered during testing have already been corrected. A brief analysis of the cost of the system was carried out, especially with regard to the wearable motion-sensing prototype component, and it proved to be affordable for clinics and independent use.

6

Conclusion

This chapter describes the main conclusions of the work carried out in the development and evaluation of the system. It will also present and list some future recommendations that can improve, correct and expand the system implemented in this dissertation.

6.1 Conclusions

Nowadays it is becoming more possible to observe the impact of technology in the world, such as technologies for the fast development of mobile applications, as well as the rise of new sensors and more powerful microprocessors. The physical services are progressively adopting new technologies and infrastructures that allow their remote and more ubiquitous use. This way, these services are migrating to the Internet world, where it seeks to minimise the need for physical interaction, and thus promote the fast delivery of services without any impediment or barrier in a physical and temporal space.

The work carried out in this dissertation presents the prototype of a system designed to improve the rehabilitation process of a problem associated with the wrist joint, as well as to help the evaluation and monitoring of the process. This system is intended to be used in a medical context, and it is desired for the platform to be accessible and easy to use for both the patient and the physiotherapist. This way, the patient can improve the physical condition through the use of the prototype at any time, in an independent way and without any restriction. On the other hand, the physiotherapist can evaluate and follow the progress of physiotherapy exercises performed by patients outside the office.

It was of interest to develop a low-cost system so that it could be accessible and used by any patient and physiotherapist. Another important consideration was to build the wearable motion-sensing prototype component to be easy to use, and of reduced volume and weight, in order not to limit or restrict any of the physiotherapy movements proposed in this dissertation. The system performed well during the testing phase, as it was possible to correctly reach and measure the angles made by the wrist during physiotherapy exercises, and the results obtained in the tests performed with patients and physiotherapist were positive. However, the system presents some problems when the wearable motion-sensing prototype component is not properly used and positioned, or because the velcro straps used in this prototype loosen slightly, causing minor setbacks during the execution of the movements.

Overall, the general goal of the system and of this dissertation have been accomplished.

6.2 Future Work

In the course of defining the proposed objectives and developing the system in this dissertation, additional work was discussed and taken into consideration to expand the system with new functionalities, as well as to improve its features and performance.

An expansion of the system would be to include **new physiotherapy exercises** for the wrist, and also **integrate new sensors** in order to correct bad postures while performing movements. There are open and closed kinetic chain exercises, and it may be interesting to adapt the system to allow these two types of exercises. Currently, the system only supports open chain exercises for the wrist, which consists of moving the parts of the body furthest from the limb root, called the distal part, which is the case of the hand. This way, the parts of the body closest to the limb root, the proximal part, are immobile, which in this case is the forearm. In order for the system to also support closed chain exercises, the sensors would have the purpose of validating the objective angles of the exercises and detecting if there is any undesirable movement. Therefore, depending on the exercise being performed, the sensors will take on a certain function.

6.2.1 Wearable motion-sensing prototype component

The wearable motion-sensing prototype component could be improved in terms of **using better material** to allow both the boxes that hold the sensors and the microcontroller to be tightly and securely attached to the wrist.

Some types of actuators could be included in order to allow a more interactive feedback during the exercises, such as sound actuators, vibration, or even LED lights to tell the patient that the movement is incorrect, or that the repetition has been successfully validated.

6.2.2 Server and database

An improvement in this component could be **migrate the system to a server available on the Internet**, with its own domain. This way, it would no longer be necessary to use the IP of the local machine that ran the system during the development and testing phase.

6.2.3 Application

In this work, the application was developed only for Android, so it would be interesting to **make the application also compatible for iOS**. In general, the design of the interfaces could be improved, as well as including more animations and increasing the overall interactivity of the application.

In order to improve the communication between the patient and the physiotherapist, a **message exchange system** could be included, where either one could contact the other when necessary.

It would also be interesting to **improve the profile page** by adding features to edit and remove the account, and also display personal information or personal statistics.

Bibliography

[1] P. A. Abdalla and A. Varol. "Advantages to Disadvantages of Cloud Computing for Small-Sized Business". In: *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. June 2019, pp. 1–6. doi: 10.1109/ISDFS.2019.8757549 (cit. on p. 37).

[2] N. Ahmad et al. "Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications". In: *International Journal of Signal Processing Systems* 1.2 (Jan. 2013), pp. 256–262. doi: 10.12720/ijsps.1.2.256-262 (cit. on p. 21).

[3] R. Ambar et al. "Development of a Home-based Wrist Rehabilitation System". In: *International Journal of Electrical and Computer Engineering* 7.6 (Dec. 2017), pp. 3153–3163. doi: 10.11591/ijece.v7i6.pp3153-3163 (cit. on pp. 19, 20).

[4] M. M. M. Aye. "Analysis of Euler Angles in a Simple Two-Axis Gimbals Set". In: *International Journal of Mechanical, Industrial and Aerospace Sciences* 4.9 (Sept. 2011). doi: 10.5281/zenodo.1330465. url: https://doi.org/10.5281/zenodo.1330465 (cit. on pp. 27, 28).

[5] I. Ben Abdallah, Y. Bouteraa, and C. Rekik. "Web-based robot control for wrist telerehabilitation". In: *2016 4th International Conference on Control Engineering & Information Technology (CEIT)*. Dec. 2016, pp. 1–6. doi: 10.1109/CEIT.2016.7929100 (cit. on pp. 12, 13).

[6] G. Botelho. "Sistema para Avaliação de Postura com Utilização de Sensores: SAPI-Sistema de Avaliação de Postura Ininterrupta". In: 2021 (cit. on pp. 2, 3).

[7] T. Carter et al. "Accuracy and Reliability of Three Different Techniques for Manual Goniometry for Wrist Motion: A Cadaveric Study". In: *The Journal of hand surgery* 34 (Sept. 2009), pp. 1422–8. doi: 10.1016/j.jhsa.2009.06.002 (cit. on p. 9).

[8] L. Correia. "Development of a web and mobile based system for remote physiotherapy using motion sensors". In: 2021 (cit. on pp. 2, 3).

[9] E. B. Dam, M. Koch, and M. Lillholm. *Quaternions, Interpolation and Animation*. Vol. 2. Department of Computer Science, University of Copenhagen, 1998 (cit. on p. 29).

[10] S. Danbatta and A. Varol. "Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth Wireless Technologies Used in Home Automation". In: *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. June 2019, pp. 1–5. DOI: 10.1109/ISDFS.2019.8757472 (cit. on p. 31).

[11] S. M. Darroudi and C. Gomez. "Bluetooth Low Energy Mesh Networks: A Survey". In: *Sensors* 17.7 (June 2017). ISSN: 1424-8220. DOI: 10.3390/s17071467. URL: https://www.mdpi.com/1424-8220/17/7/1467 (cit. on p. 33).

[12] D. Deponti, D. Maggiorini, and C. E. Palazzi. "DroidGlove: An android-based application for wrist rehabilitation". In: *2009 International Conference on Ultra Modern Telecommunications Workshops*. Oct. 2009, pp. 1–7. DOI: 10.1109/ICUMT.2009.5345442 (cit. on pp. 17, 18).

[13] F. J. Dian, A. Yousefi, and S. Lim. "A practical study on Bluetooth Low Energy (BLE) throughput". In: *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. Nov. 2018, pp. 768–771. DOI: 10.1109/IEMCON.2018.8614763 (cit. on pp. 32, 33).

[14] B. Eisenman. *Learning react native: Building native mobile apps with JavaScript*. "O'Reilly Media, Inc.", 2015 (cit. on p. 36).

[15] J. Eschweiler et al. "Anatomy, Biomechanics, and Loads of the Wrist Joint". In: *Life* 12.2 (Jan. 2022), p. 188. ISSN: 2075-1729. DOI: 10.3390/life12020188. URL: https://www.mdpi.com/2075-1729/12/2/188 (cit. on pp. 6, 9).

[16] E. Fresk and G. Nikolakopoulos. "Full quaternion based attitude control for a quadrotor". In: July 2013, pp. 3864–3869. DOI: 10.23919/ECC.2013.6669617 (cit. on p. 59).

[17] C. Gordon. "Quaternion Representation of 3D Orientation and Rotation for Sensor Fusion Applications". In: 2021 (cit. on p. 52).

[18] G. Gridling and B. Weiss. "Introduction to microcontrollers". In: *Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group* (Feb. 2007) (cit. on pp. 25, 26).

[19] M. H. Haider et al. "Comparison of Complementary and Kalman Filter Based Data Fusion for Attitude Heading Reference System". In: vol. 1919. 1. AIP Publishing LLC. Feb. 2017, p. 020002. DOI: 10.1063/1.5018520 (cit. on p. 26).

[20] M. Hazman et al. "IMU sensor-based data glove for finger joint measurement". In: *Indonesian Journal of Electrical Engineering and Computer Science* 20 (Oct. 2020), pp. 82–88. ISSN: 2502-4752. DOI: 10.11591/ijeecs.v20.i1.pp82-88 (cit. on p. 14).

[21] F. J. Hiung and I. M. Sahat. "Development of wrist monitoring device to measure wrist range of motion". In: *IOP Conference Series: Materials Science and Engineering* 788.1 (Apr. 2020), p. 012033. DOI: 10.1088/1757-899X/788/1/012033. URL: https://dx.doi.org/10.1088/1757-899X/788/1/012033 (cit. on pp. 10, 11, 82).

[22] javatpoint. *Spring Boot Architecture*. URL: https://www.javatpoint.com/spring-boot-architecture (visited on 12/05/2022) (cit. on p. 38).

[23] M. Jones, J. Bradley, and N. Sakimura. *Json web token (jwt)*. Tech. rep. 2015 (cit. on p. 39).

[24] M. Jones, B. Campbell, and C. Mortimore. *JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization Grants*. Tech. rep. 2015 (cit. on p. 39).

[25] A. Lacava et al. "Securing Bluetooth Low Energy networking: An overview of security procedures and threats". In: *Computer Networks* 211 (2022), p. 108953. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2022.108953. URL: https://www.sciencedirect.com/science/article/pii/S1389128622001335 (cit. on p. 32).

[26] P. LaStayo and D. L. Wheeler. "Reliability of passive wrist flexion and extension goniometric measurements: a multicenter study". In: *Physical therapy* 74.2 (1994), pp. 162–74. DOI: 10.1093/ptj/74.2.162 (cit. on p. 9).

[27] G.-h. Lee et al. "A study on the performance comparison of 3D file formats on the web". In: *International journal of advanced smart convergence* 8.1 (2019), pp. 65–74 (cit. on p. 71).

[28] S.-H. Lim. "Experimental Comparison of Hybrid and Native Applications for Mobile Systems". In: *International Journal of Multimedia and Ubiquitous Engineering* 10.3 (Mar. 2015), pp. 1–12. DOI: 10.14257/ijmue.2015.10.3.01 (cit. on p. 35).

[29] T. Lopes. "Gait Monitor and Analyzer: A Wearable Sensor-Based Gait Analysis System for Physiotherapy". In: 2021 (cit. on pp. 2, 3).

[30] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[31] D. Lu and T. Liu. "The application of IOT in medical system". In: *2011 IEEE International Symposium on IT in Medicine and Education*. Vol. 1. 2011, pp. 272–275. DOI: 10.1109/ITiME.2011.6130831 (cit. on p. 2).

[32] S. Ludwig et al. "Comparison of attitude and heading reference systems using foot mounted MIMU sensor data: basic, Madgwick, and Mahony". In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*. Ed. by H. Sohn. Vol. 10598. International Society for Optics and Photonics. SPIE, 2018, pp. 644–650. DOI: 10.1117/12.2296568. URL: https://doi.org/10.1117/12.2296568 (cit. on p. 27).

[33] M. M. Abdellatif and W. Mohamed. "Telemedicine: An IoT based Remote Healthcare System". In: *International Journal of Online and Biomedical Engineering (iJOE)* 16.6 (May 2020), pp. 72–81. DOI: 10.3991/ijoe.v16i06.13651. URL: https://online-journals.org/index.php/i-joe/article/view/13651 (cit. on pp. 30, 31).

[34] S. Madakam, R. Ramaswamy, and S. Tripathi. "Internet of Things (IoT): A Literature Review". In: *Journal of Computer and Communications* 3.5 (Apr. 2015), pp. 164–173. DOI: 10.4236/jcc.2015.35021 (cit. on p. 1).

[35] S. Madgwick et al. "An efficient orientation filter for inertial and inertial/magnetic sensor arrays". In: *Report x-io and University of Bristol (UK)* 25 (2010), pp. 113–118 (cit. on p. 27).

[36] A. Maier, A. Sharp, and Y. Vagapov. "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things". In: *2017 Internet Technologies and Applications (ITA)*. 2017, pp. 143–148. DOI: 10.1109/ITECHA.2017.8101926 (cit. on p. 26).

[37] L. Markley et al. "Averaging Quaternions". In: *Journal of Guidance, Control, and Dynamics* 30.4 (July 2007), pp. 1193–1196. DOI: 10.2514/1.28949 (cit. on p. 29).

[38] *MongoDB*. URL: https://www.mongodb.com/ (visited on 12/19/2022) (cit. on p. 39).

[39] P. MR et al. "Reliability and concurrent validity of a new iPhone® goniometric application for measuring active wrist range of motion: a cross-sectional study in asymptomatic subjects". In: *J Anat* 230.3 (2017), pp. 484–495. DOI: 10.1111/joa.12568 (cit. on p. 10).

[40] A. Nag, S. C. Mukhopadhyay, and J. Kosel. "Wearable Flexible Sensors: A Review". In: *IEEE Sensors Journal* 17.13 (2017), pp. 3949–3960. DOI: 10.1109/JSEN.2017.2705700 (cit. on pp. 21, 22).

[41] M. Narasimhappa et al. "MEMS-Based IMU Drift Minimization: Sage Husa Adaptive Robust Kalman Filtering". In: *IEEE Sensors Journal* 20.1 (Sept. 2019), pp. 250–260. DOI: 10.1109/JSEN.2019.2941273 (cit. on p. 22).

[42] R. Nussbaum et al. "Systematic Review of Mobile Health Applications in Rehabilitation". In: *Archives of Physical Medicine and Rehabilitation* 100.1 (2019), pp. 115–127. ISSN: 0003-9993. DOI: 10.1016/j.apmr.2018.07.439. URL: https://www.sciencedirect.com/science/article/pii/S0003999318311754 (cit. on p. 8).

[43] P. Que, X. Guo, and M. Zhu. "A Comprehensive Comparison between Hybrid and Native App Paradigms". In: *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*. Dec. 2016, pp. 611–614. DOI: 10.1109/CICN.2016.125 (cit. on p. 35).

[44] *React Native*. URL: https://reactnative.dev/ (visited on 01/12/2022) (cit. on p. 37).

[45] E. J. Rechy-Ramirez, A. Marin-Hernandez, and H. V. Rios-Figueroa. "A human–computer interface for wrist rehabilitation: a pilot study using commercial sensors to detect wrist movements". In: *The Visual Computer* 35.1 (Jan. 2019), pp. 41–55. DOI: 10.1007/s00371-017-1446-x (cit. on pp. 15, 16).

[46] D. Rose. "Rotation Quaternions, and How to Use Them". In: (2015). URL: http://danceswithcode.net/engineeringnotes/quaternions/quaternions.html (cit. on p. 59).

[47] D. Rose. *Rotations in Three-Dimensions: Euler Angles and Rotation Matrices*. 2015. (Visited on 10/17/2022) (cit. on pp. 28, 29).

[48] G. Saggio et al. "Resistive flex sensors: a survey". In: *Smart Materials and Structures* 25.1 (Dec. 2015), p. 013001. DOI: 10.1088/0964-1726/25/1/013001. URL: https://dx.doi.org/10.1088/0964-1726/25/1/013001 (cit. on p. 22).

[49] S. Al-Sarawi et al. "Internet of Things (IoT) communication protocols: Review". In: *2017 8th International Conference on Information Technology (ICIT)*. July 2017, pp. 685–690. DOI: 10.1109/ICITECH.2017.8079928 (cit. on p. 30).

[50] A. G. de Saúde LTDA. *Manual de Goniometria Medição dos Ângulos Articulares*. 2016. URL: https://acegs.com.br/wp-content/uploads/2016/06/MANUAL-DE-GONIOMETRIA-FINAL.pdf (visited on 09/14/2022) (cit. on pp. 9, 10).

[51] N. Semiconductors. "I2C-bus specification and user manual". In: *NXP Semiconductors* 4 (2014) (cit. on p. 53).

[52] M. Solís-Peña and A. Villa-Parra. "Device to Guide Hand Rehabilitation Routines Based on Pressure Signals". In: *Procedia Computer Science* 160 (2019). The 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2019) / The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2019) / Affiliated Workshops, pp. 659–664. ISSN: 1877-0509. DOI: 10.1016/j.procs.2019.11.031. URL: https://www.sciencedirect.com/science/article/pii/S1877050919317314 (cit. on pp. 18, 19).

[53] *Spring*. URL: https://spring.io/ (visited on 12/14/2022) (cit. on p. 38).

[54] M. Stokes. "Improving the performance of skeletal mesh animations in the Blender game engine". In: *Eastern Washington University* (2014) (cit. on p. 70).

[55] K. T'Jonck et al. "Optimizing the Bluetooth Low Energy Service Discovery Process". In: *Sensors* 21.11 (2021), p. 3812. ISSN: 1424-8220. DOI: 10.3390/s21113812. URL: https://www.mdpi.com/1424-8220/21/11/3812 (cit. on pp. 33, 34).

[56] S. Tachakra et al. "Mobile e-Health: The Unwired Evolution of Telemedicine". In: *Telemedicine Journal and e-Health* 9.3 (2004), pp. 247–257. DOI: 10.1089/153056203322502632 (cit. on p. 2).

[57] M. Takaiwa and T. Noritsugu. "Development of wrist rehabilitation equipment using pneumatic parallel manipulator -Acquisition of P.T.'s motion and its execution for patient-". In: *2009 IEEE International Conference on Rehabilitation Robotics*. 2009, pp. 34–39. DOI: 10.1109/ICORR.2009.5209563 (cit. on p. 13).

[58] K. Tarchanidis and J. Lygouras. "Data glove with a force sensor". In: *IEEE Transactions on Instrumentation and Measurement* 52.3 (July 2003), pp. 984–989. DOI: 10.1109/TIM.2003.809484 (cit. on pp. 23, 25).

[59] P. Visconti et al. "Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses". In: *International Journal on Smart Sensing and Intelligent Systems* 11.1 (June 2018), pp. 1–25. DOI: 10.21307/ijssis-2018-005. URL: https://doi.org/10.21307/ijssis-2018-005 (cit. on p. 15).

[60] H. W. Wallmann. "Overview of wrist and hand injuries, pathologies, and disorders: Part 1". In: *Home Health Care Management & Practice* 23.1 (2011), pp. 60–62. DOI: 10.1177/1084822310378982 (cit. on p. 8).

[61] H. W. Wallmann. "Overview of wrist and hand injuries, pathologies, and disorders: Part 2". In: *Home Health Care Management & Practice* 23.2 (2011), pp. 146–148. DOI: 10.1177/1084822310384683 (cit. on pp. 6, 8).

[62] Y. Wei and Q. Xu. "An overview of micro-force sensing techniques". In: *Sensors and Actuators A: Physical* 234 (Sept. 2015), pp. 359–374. ISSN: 0924-4247. DOI: 10.1016/j.sna.2015.09.028. URL: https://www.sciencedirect.com/science/article/pii/S092442471530145X (cit. on p. 23).

[63] H. Zhou, H. Hu, and N. Harris. "Application of wearable inertial sensors in stroke rehabilitation". In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2005, pp. 6825–6828. DOI: 10.1109/IEMBS.2005.1616072 (cit. on p. 10).