ISCTE IUL

Instituto Universitário de Lisboa

Department of Information Science and Technology

Distributed sensing solution for home efficiency tracking

Carolina Aparício Dionísio

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master in Telecommunications and Computer Engineering

Supervisor

Prof. Dr. Pedro Joaquim Amaro Sebastião, Assistant Professor

ISCTE-IUL

Co-Supervisor

Prof. Dr. Nuno Manuel Branco Souto, Assistant Professor

ISCTE-IUL

October 2019

Resumo

Com a rápida evolução da tecnologia e dos dispositivos inteligentes, monitorizar os consumos domésticos é cada vez mais uma necessidade constante e que começa a ser feita de forma automática. Em pleno século XXI cada vez mais, são debatidas medidas para tornar o planeta um local mais sustentável, e isso depende de cada um de nós, a partir das nossas próprias.

É através do conceito da Internet das Coisas e do seu vasto conceito de conectar dispositivos, permitindo que estes comuniquem uns com os outros, que se torna possível converter uma casa normal numa casa inteligente de uma forma simples e revolucionária.

Esta dissertação apresenta a proposta de um sistema baseado numa rede de sensores sem fios, desenhada com o propósito de monitorizar e controlar os parâmetros ambientais de uma casa. De forma a obter um sistema eficiente e de baixo custo, foi feito um estudo para selecionar as melhores soluções de hardware e software para este sistema, permitindo ao utilizador, através de uma aplicação Android visualizar toda a informação, obtida pelos sensores, e consequentemente tomar decisões que tornem a sua casa mais sustentável.

A principal vantagem deste sistema, que o distingue dos outros, é que todos os componentes são pequenos, eficientes, com um baixo custo. Para além disso, é um sistema prático que permite uma fácil instalação e que se envolve com o meio à sua volta.

Palavras-chave: Internet das Coisas, Casas Inteligentes, Rede de Sensores Wireless, ESP32, LoRa, Android, Machine Learning

Abstract

With the rapid increase of smart devices, keeping track of household consumptions is a service that starts to be automated and a need that is becoming more and more constant. In the middle of the 21st century, more and more measures are being debated that can make the world a more sustainable place, and this is part of each one of us, from our own homes.

It is through the Internet of Things and its vast concept of connecting all devices, allowing them to communicate with each other, that it is possible to convert a normal home into a smart home in a simple and revolutionary way. This dissertation presents a proposal for a system based on wireless sensor networks designed for the purpose of monitoring and controlling environment parameters of a smart home. In order to obtain an efficient and inexpensive system, a study was made to select the best hardware and software solutions for this system, that also allows the user, through an Android application, to view all the information collected by the sensors, and consequently act in a way to make his home more sustainable.

The main advantages of this system, which distinguishes it from the others, are its small dimensions, its efficiency and the low cost associated. In addition, it is a practical system that allows easy installation and it can interact with the surrounding environment.

Keywords: Internet of Things, Smart Homes, Wireless Sensor Network, ESP32, LoRa, Android, Machine Learning

Acknowledgements

First of all, I would like to thank my supervisor, PhD Pedro Sebastião, and my PhD co-supervisor Nuno Souto for the orientation, support, motivation and encouragement given during the development of the dissertation.

To the Telecommunications Institute at ISCTE-IUL, thanks for providing all the material and resources needed for this dissertation.

I would like to thank my family, especially my mother Teresa and my sister Margarida, for all the support they provided me during these years. And for all the effort they did giving me the opportunity to get here, without them it would have been impossible.

To André Gloria, for all his advice and encouragement, a big thank you, there are no words to write all the support and patience, when I have always needed it. Your knowledge has no limits and I know that you will have nothing less than a successful career.

To my long-time friend, Catarina Duque, for all her support at this phase and for all her good advices, not only in this one, but in all the others.

Finally, a special thanks to all my friends and colleagues who accompanied me on this journey, specially to André Marques and Inês Sequeira for believing in my abilities, better than anyone else. To Gonçalo Simões, Jorge Rafael, Maria Inês Pires, Rui Passinhas and Raquel Duque for having been part of my academic life and for all their help and companionship in these 5 years of adventure. And to Miguel, for his support and all the patience in the world that he had with me.

Contents

List of Figures

List of Tables

Abbreviations

ADC	**A**nalogic **D**igital **C**onverters
AP	**A**ccsess **P**oint
BLE	**B**luetooth **L**ow **E**nergy
BSS	**B**asic **S**ervice **S**et
CSS	**C**hirp **S**pread **S**pectrum
DT	**D**ecision **T**ree
FFD	**F**ull- **F**unction **D**evice
FSK	**F**requency- **S**hift **K**eying
GPIO	**G**eneral **P**urpose **I**nputs **O**utputs
HTTP	**H**yper **T**ext **T**ransfer **P**rotocol
IDE	**I**ntegrated- **D**evelopment **E**nvironment
IoT	**I**nternet **o**f **T**hings
LDR	**L**ight **D**ependent **R**esistor
LoRa	**Lo**ng **Ra**nge
MAC	**M**edia **A**ccess **C**ontrol
ML	**M**achine **L**earning
MQTT	**M**essage **Q**ueuing **T**elemetry **T**ransport
MSE	**M**ean **S**quared **E**rror
P2P	**P**eer-to-**P**eer
RF	**R**andom **F**orest
RFD	**R**educed **F**unction **D**evice
SoC	**S**ystem-**o**n-a-**C**hip
SVM	**S**upport **V**ector **M**achine
Wi-Fi	**W**ireless **F**idelity

WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WPAN	**W**ireless **P**ersonal **A**rea **N**etwork
WSN	**W**ireless **S**ensor **N**etwork

Chapter 1

Introduction

The topic of this dissertation comes from the necessity to relate sustainability with the technology that surrounds us.

1.1 Motivation

In the course of the 21st century, the human being tends to change his routine with the evolution of technology, either to make the day-to-day life easier or for entertainment. The Internet of Things (IoT) has the capacity to make a simple physical device in a smart one, connected to the Internet with communication and computational capacity. In this way, the network has access to the device, knowing its position and its state, allowing communication with other devices over the Internet.

Due to this simplicity, IoT is a current reality, in a constant evolving state, being present in all sectors from entertainment to medicine. Consequently, it is easier to claim this reality is present in our homes in order to make them more efficient and intelligent, the designated smart homes. These types of homes are considered one of the most prominent applications of IoT, because the main characteristic of a smart home is automation and monitorization, resulting in the reduction of the human effort [1]. It is useful to monitor house expenses in real time, since this saves power and other resources, making the house safer and sustainable, improving the life quality of the user.

The monitorization of a smart home can be done according to the parameters the user intends to have, such as: temperature, gas, noise, humidity, brightness and water consumption. In order to perform this data acquisition a Wireless Sensor Network (WSN) is used, where some wireless sensors are strategically distributed around the house, to collect information and, after being transmitted and analyzed, to decide actions and alert the user. The WSN technology supports the IoT concept, since its main advantage is gathering measurements of parameters using devices connected to the Internet, where the results become immediately available to the user wherever he is. Nonetheless, the use of this technology depends on its own efficiency, the reliability and the safety of the data, since they can compromise the user.

The search to accomplish a sustainable way of consuming energetic resources of the planet earth, is a global concern, due to the tremendous consumption of natural resources by human beings, planet Earth's state is turning critical. With the technology being more present in the human life, thinking of a way to piece together sustainability with everyday use of equipment is a step forward in the right direction. Through IoT systems, due to their monitorization and automatization capabilities, it will be possible to reduce the impact human beings have in their consumption, sparing Earth's resources and saving monetary costs to the user.

1.2 Objectives

The main objective is to create an IoT system to monitor household consumptions, analyzing in real time, gathering data in the installation site, using a low-cost sensors network, in order to understand if there is any abnormal consumption or any dangerous situations. It should consequently lead to natural and material resources savings, as well as reducing monetary costs for the user.

In order to achieve this main goal, it is intended to develop a distributed solution based in WSN that gathers information according to pre-established parameters. Subsequently, this information will be sent to the platform, through an appropriate communication protocol given the range, number of nodes, reliability and price. The analyzed data will be sent periodically, except in dangerous situations, to a platform where the user can monitor it.

For this to work, groups of modules need to be developed in an individual way. They will then be put together to achieve best results and fulfill the proposed requirements. The modules to be implemented are:

1. **Platform** – Develop a smartphone application, that connects the sensors network to its user, that can see the information collected;
2. **Sensor Network** - Design and specify the low-cost sensor network, as well as chose the best sensors according to the chosen parameters;
3. **Communications** - Will be carried out a study to understand the best solutions, according to the home type, in order to enhance communications between the sensor's nodes and the server and vice-versa;
4. **Machine Learning Algorithm** - Data analysis to detect abnormal situations in real time, as well as analyzing user consumptions;
5. **Energy** - Will be carried out a study in order to interconnect the previous modules in a way to make possible to expand the sensors network lifetime.

It is necessary to ensure that these low-cost sensors network is easy to install and blends in with the space without interfering with his normal performance.

1.3 Scientific Contribution

This dissertation presents the following contributions:

- Design and implementation of the sensor network for the monitoring household consumptions;
- The development of a machine learning algorithm capable to predict events;
- Development and implementation of a mobile application for data visualization;
- Demonstrates the successful application of the system on real case studies.

The work developed in this dissertation resulted, besides this document, in one scientific article, published in an international Internet of Things conference and IEEEXplorer, contributing to the dissemination of our research and results.

- Dionisio C., Simões G., Glória A., Sebastiao P. and Souto N., "Distributed Sensing Solution for Home Efficiency Tracking", 2019 IEEE 5th World Forum on Internet of Things (WF-IoT)

Another article is currently under development to show the results obtained from this system.

Besides the published article, the work developed in this dissertation also helped in the research of another Master Thesis as well as for a PhD Thesis. From this contribution, three articles were published in international conferences and one journal paper is going to be published in October.

- A. Glória, C. Dionísio, G. Simões, P. Sebastião, and N. Souto, "WSN Application for Sus-tainable Water Management in Irrigation Systems," in 5th IEEE World Forum on Inter-net of Things WF-IoT 2019

- G. Simões, C. Dionísio, A. Glória, P. Sebastião, and N. Souto, "Smart System for Moni-toring and Control of Swimming Pools," in 5th IEEE World Forum on Internet of Things WF-IoT 2019
- A. Glória, C. Dionísio, G. Simões and P. Sebastião, "LoRa Parameters Self Configuration for Low Power End Devices" in 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC 2019)
- A. Glória, C. Dionísio, G. Simões, J. Cardoso, P. Sebastião and N. Souto, "Water Man-agement for Sustainable Irrigation System using Internet of Things, " in Transactions of the Institute of Measurement and Control

1.4 Structure of the Dissertation

This dissertation is organized by chapters, in chapter 2 a study of the best components according to the desired requirements of the system, is made. Chapter 3 describes the developed system as a whole and how everything is connected. Chapter 4 shows the necessary tests and the machine learning tests needed to implement the system. In Chapter 5, the final prototype is demonstrated and the results obtained are discussed. Finally, in chapter 6 the main conclusions are made, and the future work is presented.

The published article is included in Appendix A. Appendices B include a User & Technical Manual to help with the comprehension of the design and implementation of the developed project. In Appendix C the results obtained in the machine learning training case are displayed.

Chapter 2

State of the Art

IoT is a technological concept with a great development, with the potential to alter the way we see the technological world. The evolution of this concept is something that becomes more accentuated every day and that we begin to see in our own homes. In this chapter are presented the research area to develop the proposed work as well as all the necessary components. The first sections will introduce the concept of IoT in smart homes and how it is organized, as well as the importance of Wireless Sensor Networks. Then sections it will be discussed which are the best communication protocols, the best hardware and the best machine learning algorithms to create the desired system. Finally, the difficulties of similar work and how this system intends to overcome them will be presented.

2.1 Internet of Things

Over the years, the Internet has evolved in a brutally dramatic way, becoming a part of the daily life of the human specie [2]. The concept of IoT has come to us and has revolutionized and evolved the way we know and deal with the Internet today, being a neologism related to the extension of the internet network to the world of electronic objects [3].

IoT has the ability to turn a traditional physical object into an intelligent one, by exploiting is underlying technologies like embedded systems, communication technologies and sensor networks [4].Through IoT, each object gains the ability to perform tasks, share information, communicate and coordinate decisions with other devices in real time, allowing our day-to-day tasks to have a reduction in human effort.

When creating new IoT systems it is necessary to take into consideration that there are challenges that can jeopardize the entire system. The author of [5] addresses these issues such as security, connectivity, compatibility, data management and limited energy resource, being all these interrelated and not intended to compromise the user and all the information that the system has about the user, but also to ensure that the end product is of quality and compatible with several users so that it can be used for a long time. Thus, when developing an IoT system, it is necessary to consider all aspects so that these challenges are overcome, that is, to take into account the properties of communications protocols, hardware and software.

With the rapid development of IoT and technologies in all areas, homes have become intelligent, with strong computing and communication skills [6]. A Smart home is a house that connects devices that communicate with each other over the internet [7] with the purpose of improving living standards and security, as well as save resources.

A smart home is considered intelligent because it can monitor daily activities in real time, allowing the user to easily apply these features according to his needs, including detection of intruders, smoke, gas or water leaks, and also monitoring of household consumptions. Due to their enormous evolution and versatility, smart homes can be adapted according to the user's preference, and thus have several features [8], as can be seen in Figure 2.1 and detailed below.

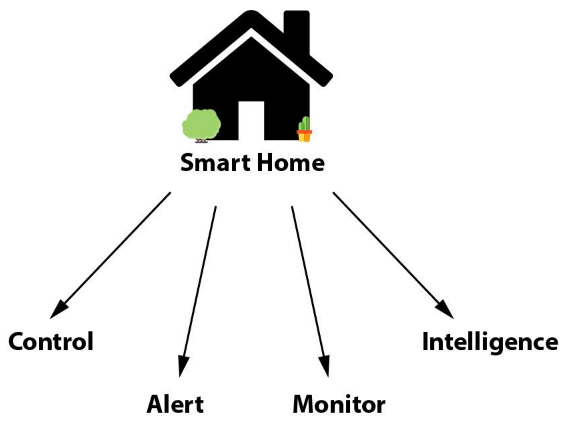


Figure 2.1: Features of a Smart Home

*M*onitor is one of the most important functions, through a network of sensors a smart home is capable of tracking every activity and take decisions and actions later.

*A*lert the system of a smart home is able to give alerts to the user in real time, whenever a sensor in the network finds an abnormal value, including gas leaks, spikes in temperature or in terms of security, the detection of intruders. These alerts can be sent by various means, namely by e-mail, message or by a smartphone application, leaving it to the user's discretion to choose the means he considers best.

*I*ntelligence is about making decisions automatically, without the need for user authorization. For that, it is necessary to integrate an Artificial Intelligence mechanism, allowing security functionalities, such as identifying abnormal or unexpected events, alerting the user while at the same time answering what to do to those abnormal events.

*C*ontrol this function is responsible for creating the autonomous part of a smart home, including automated activities such as turning ON/OFF a light bulb, lock/unlock doors, based on user preferences or habits, sensor values or in order to avoid potential hazards.

2.2 Wireless Sensor Network

A WSN is a network with several spatially distributed sensors, with the capacity of gathering, analyzing and distributing the data to monitor a certain event, such as temperature or noise. In the past, this type of network used to be wired, and were used to connect computers to the Internet [9]. Nowadays a network of sensors can group several devices, solely using wireless technologies.

This type of sensor networks consists of a certain number of nodes communicating with each other, wireless in a multi-hop way, this is, when two nodes can communicate with each other even if there is no direction between them by using other nodes as bridges.

As a rule, WSN's are designed to be energy efficient, due to their high scarcity, scalability, due to the high number of nodes, reliability, to inform when there is something wrong with the network, and robustness, because the nodes may be exposed to bad environmental conditions [3]. To have these characteristics, a network can have several types of sensors, as show in Figure 2.2.

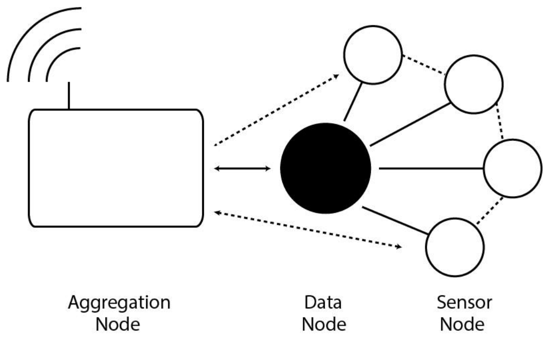


FIGURE 2.2: Type of nodes in a WSN

- Sensor node is the lowest level of a WSN network, being only responsible for collecting information and send them to other nodes, directly or through a data node data, so that the information can be processed.
- Data node is an intermediate node with a group of sensor nodes, achieved to act tasks in several nodes simultaneously.

- Aggregation node since it does not collect data, it just sends the information received from the other nodes to the server. This serves as a bridge between the sensor nodes and the server.

By using different types of nodes and a decentralized network, in which information is not contained in only one specific place, nodes can communicate with each other to achieve the common goal. The nodes are spatially distributed, depending on the typology they follow. As shown in Figure 2.3, the most common types of topologies are represented:

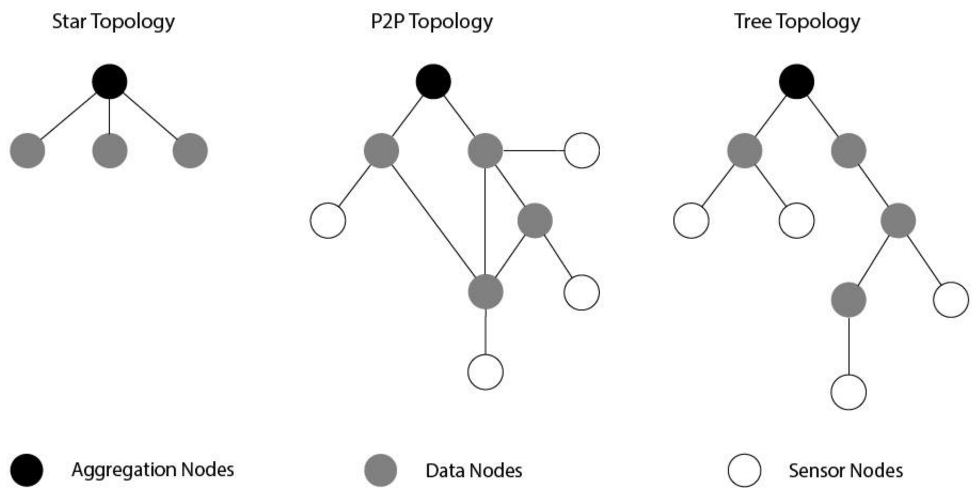


FIGURE 2.3: WSN's most common topologies

In Star topology, the sensor nodes are only one hop away from the coordinator, which in this case is the aggregation node, that will receive and send the messages to the server. This topology has a lower power consumption compared to the others, since the communication is practically direct, and it is not necessary to spend energy with many hops [10].

In both Tree and Peer-to-Peer (P2P), or Mesh, topologies, the communication are done with multi-hops. In Tree topology the sensor node reaches the aggregation node using the nodes that is connected higher up in the tree. P2P allows the transmission of data directly to the aggregation node as long as they are within the range, otherwise, the information is sent through intermediate nodes within reach, until the aggregation node. This type of topologies has the advantage of easy error detection but has a higher energy consumption [9].

For this system it is important to choose a topology that has a low level of complexity and low energy consumption, such as Star, but it is important that it is compatible with the communications protocol, since not all topologies are compatible with the communication protocols. In this way, the chosen typology will be made in the following subsection after the choice of the most appropriate communication protocol for this system.

2.3 Communications Protocols

One of the main components of IoT is communication [4] which, with the use of sensor networks, plays an important role in communication between the nodes and the server. Without the communication protocols, it would be unthinkable to create IoT systems. In this section a comparative study will be made of the best wireless communication protocol, as well as the best protocols on the server side.

2.3.1 Wireless Network Technologies

There are several wired and wireless communications protocols but bearing in mind that we are talking about monitoring a house with a network of wireless sensors, the communication protocols should follow the same path.

Currently there are several technologies, the most common are Wi-Fi, Bluetooth, ZigBee and LoRa, whose the main characteristics are presented in Table 2.1 [11] and detailed in the following subsections.

Table 2.1: Major Wireless communication protocols characteristics

Feature	**Wi-Fi**	**Bluetooth**	**ZigBee**	**LoRa**
Frequency [GHz]	2.4	2.4	2.4	0.433/0.868
Range [m]	1-100	10-100	10-100	2000
Nodes	32	7	65540	15000
Topology	Star	Piconet	Star/Mesh	Star
Power Consumption [mA]	100-350	1-35	1-10	1-10
Complexity	High	Medium	Medium	Low

2.3.1.1 IEEE 802.11 (Wi-Fi)

Wireless Fidelity (Wi-Fi) includes the IEEE 802.11 standard and provides the capability to connect various devices within a Wireless Local Area Network (WLAN). Wi-Fi has been approved in 1997 and it is been changing ever since. Initially it had a 1Mbps rate, with standard b and g upgrading it up to 11Mbps and 54Mbps at 2,4GHz, respectively and, with the introduction of 5GHz on Europe with standard h, to speeds up to 1Gbps in standard ac [12]. The range always depends on the version of the device, and the type of environment, whether it is outdoor or indoor, since if you have physical obstacles in between, the range is reduced.

The IEEE 802.11 architecture allows to use an infrastructure network, when connected to an Access Point (AP), or in an ad-hoc mode, allowing to the users to browse the internet at broadband speeds and fast data transfer, also being able to handle large amounts of data [13]. Using a topology based on a cellular architecture, each cell is called a Basic Service Set (BSS), which is a set of mobile or fixed stations. This way, all devices must be connected to the BSS, although it is not one, it resembles a star topology.

In the Table 2.2 are presented the advantages and disadvantages of the Wi-Fi protocol [14].

Table 2.2: Advantages and Disadvantages of the Wi-Fi protocol

Advantages	**Disadvantages**
-Decent coverage and outreach and can penetrate walls and other obstacles on the way	-Consumes significant power, the battery lifetime just can support 0.5 to 5 days without charging.
-Adding or removing devices from a Wi-Fi network is a simple process	-Radio waves in the network may interfere with other equipment
-Highly secure connection, with 128-bit AES encryption;	
-Supports networking over power lines, coaxial cables and phone lines;	
-Adding or removing devices from a Wi-Fi network is a simple process	

2.3.1.2 IEEE 802.15.1 (Bluetooth)

The standard IEEE 802.15.1 or as is commonly known, Bluetooth, is a wireless radio system designed for short-range and cheap devices to replace cables for computer peripherals, like keyboards and prints [13]. This standard is intended to be a secured and cheap way of connecting and transferring data among supported devices.

Bluetooth defines not only a radio interface, but a whole communication stack that allows devices to find each other and advertise the services they offer. This protocol can operate in two modes: slaver or master mode, with a maximum of eight devices — seven active slaves plus one master working together from a Piconet, which is the simplest configuration of a Bluetooth network.

In this protocol, slaves only communicate with their master in a point-to-point way, under the master's control, as show in Figure 2.4. In order to reduce power consumption, the slaves can switch from active mode to parked or standby mode. The most complex topology of Bluetooth is the scatternet, which are Piconets connected [12].

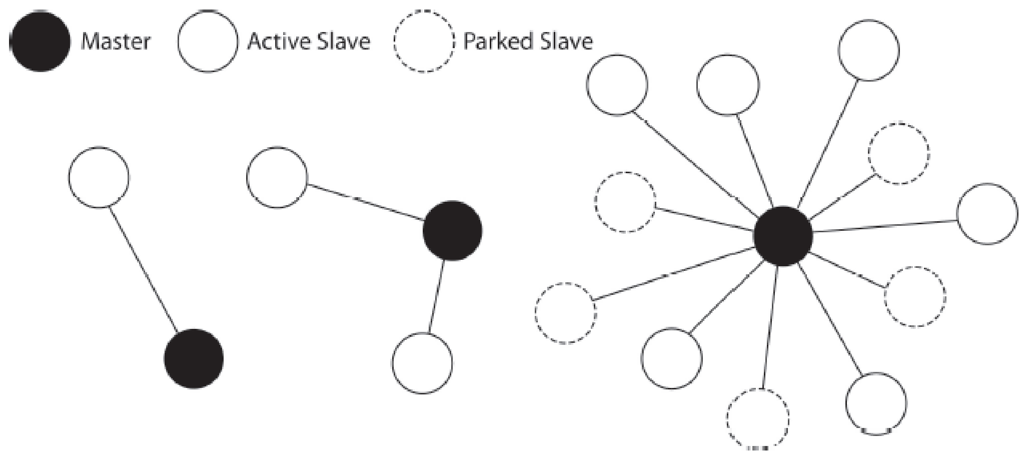


Figure 2.4: Bluetooth topology

One of the most recent versions, is the Bluetooth Low Energy (BLE) that as design specifically for IoT systems, because significantly reduces the power consumption of Bluetooth devices and enables long term operation using coin cell batteries [15]. In the Table 2.3 are presented the advantages and disadvantages of the Bluetooth protocol[14].

Table 2.3: Advantages and Disadvantages of the Bluetooth protocol

Advantages	**Disadvantages**
-One coordinator can control a numerous amount of slaves.	-Pairing all the network can be complex
-Widely Supported	-Limited number of nodes
-Self-organizing network	

2.3.1.3 IEEE 802.15.4 (ZigBee)

ZigBee is a bidirectional radio frequency wireless network standard which conforms to IEEE 802.15.4, a protocol that defines specifications for low power devices in a Wireless Personal Area Network (WPAN) with a focus on monitoring, control and sensor applications. This communication protocol provides self-organized, multi-hop and reliable networks with long battery life [7].

There are two types of devices that can participate in ZigBee Network: the Full-Function Device (FFD) and the Reduced Function Device (RFD). The FFD can operate in three modes: as a network coordinator, as a router and as end-node. In this network only exist on coordinator that create, control and maintain the ZigBee network and makes bridges to other networks. The RFD is intended for applications that are very simple, FFD can communicate with RFDs and other FFDs but the RFD can only communicate with an FFD. Because of this two modes, it is possible to use different topologies, like Star and P2P.

In the Table 2.4 are presented the advantages and disadvantages [14] of the ZigBee protocol.

TABLE 2.4: Advantages and Disadvantages of the ZigBee protocol

Advantages	**Disadvantages**
-Low power consumption	-Requires additional equipment
-Supports star, tree and P2P technologies	-Is incompatible with other network protocols
-Supports many slaves	
-One coordinator can control many slaves	

2.3.1.4 LoRa

LoRa which stands for Long Range is a physical layer using Chirp Spread Spectrum (CSS) modulation allowing a longer communication range than Frequency-Shift Keying (FSK) which is present in other communication protocols, without an increase in power consumption [16]. This modulation uses the star topology reducing the complexity and capacity of the network, with only hops between the sensor node and the aggregation node with a range of 2000 meters and a low power consumption. LoRaWAN is a bidirectional communication protocol that uses the LoRa physical layer in order to provide low power long-range communications [17]. This protocol provides a medium access control mechanism, enabling many end-devices to communicate with a gateway using the LoRa modulation.

While the LoRa modulation is proprietary the LoRaWAN is an open standard. Furthermore, this protocol is based on the protocols of the MAC layer of the WPAN and guarantees the security of the LoRa sensors by changing the use of encrypted symmetrical keys and two forms of encryption, at the MAC address level and at the network level [18].

In the Table 2.5 are presented the advantages and disadvantages [19] of LoRa.

Table 2.5: Advantages and Disadvantages of the LoRa protocol

Advantages	**Disadvantages**
-Low power consumption -Long communication range -Highly secure, with 128-bit AES encryption	-Low packet size, only 55 bytes per message

2.3.2 Server Side Communications

The information collected by the sensors is sent to the aggregation node by a wireless communications protocol. When arriving at the aggregation node it is necessary to transfer the information to the server, as these are usually in the cloud, it is not possible to communicate through wireless and wired protocol communications. Thus, it is necessary to have a protocol that supports the connection to the Internet, so that there is communication between the aggregation node and the server.

2.3.2.1 Hyper Text Transfer Protocol

Hyper Text Transfer Protocol (HTTP) is the most common protocol on the Internet, it can be used on IoT systems to send information to servers using a request/response architecture. HTTP can transfer a large number of data in tiny packets, which can possibly cause large overhead. So, due to this overhead communication for IoT via this protocol can cause serious bandwidth issues [20].

HTTP is operated over TCP/IP, being a unidirectional and synchronous protocol, that is, the client needs to start the connection with the server and only after the connection is made, the client can send the information, as shown in Figure 2.5 [21].

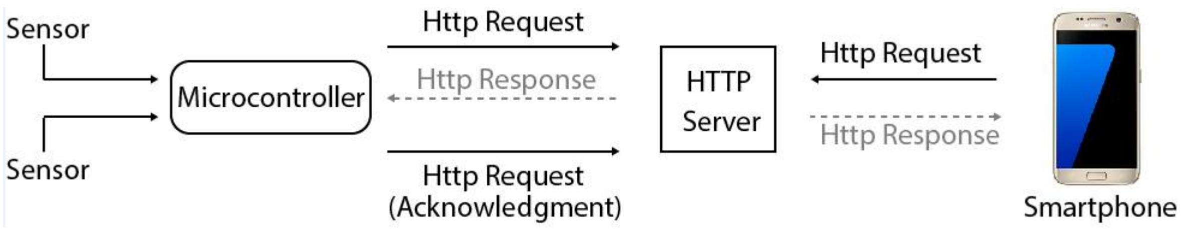


FIGURE 2.5: HTTP Request/Response messages

In IoT systems it is a long and unfeasible process, since a node can have several messages to send and must always establish a connection per message, ceasing to be a monitoring in real time once it causes delays in the network.

2.3.2.2 Message Queuing Telemetry Transport

Message Queuing Telemetry Transport (MQTT) is a protocol to data communication built on TCP protocol with low complexity. Is a many-to-many communication protocol, with three components: broker, publisher and subscriber [22]. The publisher sends a message with two elements, a topic and a message. This message is send to the broker, and the messages are transferred between multiple devices. A device registers as subscriber for topics of interests send by a publisher [20]. This exchange of messages can be seen in the Figure 2.6.

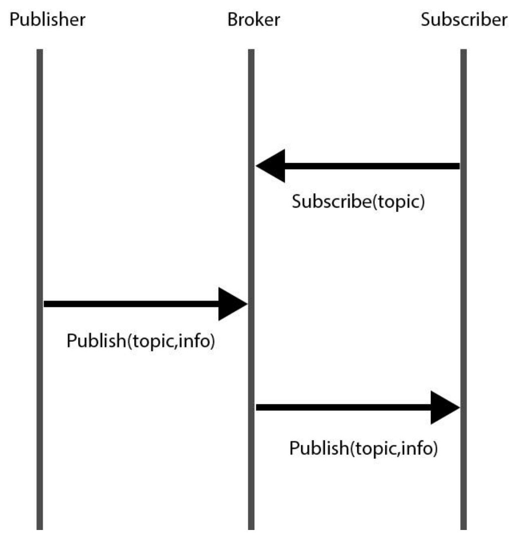


FIGURE 2.6: MQTT Publish/Subscribe process

Subscribers can only receive messages from the topics they have subscribed to, being the broker responsible to ensures that this happens, as well as to make available which topics are available. Unlike HTTP, with the publish/subscriber model it is possible for several publishers to communicate at the same time, so that one node does not need the other node to end its connection to the server [23].

MQTT is designed to send a message to one or more devices with less latency sending a large amount of data, being also considered the most favorable connection protocol for IoT, since it provides a simple implementation and flexible transition [21], it suitable due to its ability to provide routing for low power and low memory devices via a Wi-Fi connection [24].

2.3.3 Remarks

According to the author of [7], the communication protocol to be chosen should be the one that guarantees a better performance according to the criteria: Interoperability, Self-Management, Maintenance, Bandwidth and Power Consumption. Since it is not possible to perform tests on the final system, the communication protocol must follow these criteria and adapt to the intended system.

For the communication between the node sensors and the aggregation node, LoRa will be chosen, since it has a low power consumption offering a longer charging time to the nodes - something that Wi-Fi and Bluetooth cannot guarantee. It ensures a good range because in a house there are several physical obstacles, such as walls, that reduce the communication range, as in the case of Wi-Fi. ZigBee's exclusion criteria was that although both allows the Star topology which, as seen in section 2.2, was the one that seemed to be the most indicated, LoRa has a lower level of complexity and there is no risk of being incompatible with other network protocols.

On the server side, HTTP has proven to be unfeasible for this system due to its complexity and being a synchronous protocol, not allowing several nodes to send messages simultaneously. MQTT overcomes these difficulties, being also a secure protocol, with a good response time, lower battery and bandwidth.

2.4 Hardware Components

Hardware is a set of devices or physical objects which have the capability to retrieve data and follow the instruction.

An IoT system consists of several hardware components, which together and connected to each other, create an embedded and electronic projects system.

The choice of hardware in an IoT system is fundamental for its efficiency and for the main objective to be fulfilled, it is necessary to consider the characteristics of each module, if the modules are compatible and if these comply to the communication protocols. In this section we will study which are the best controlling platforms, transceivers and sensors to create the desired IoT system.

2.4.1 Controlling Platforms

Controlling Platforms are electronic systems that integrate a microcontroller as one of its components, being thebrain responsible for controlling the entire system, since they are able to retrieve, store and analyze data as well as integrate a fully functional communication scheme. The most common systems solutions include standalone microcontrollers or more advance system including System on Chip (SoC) devices.

SoC is a replacement for a computer, based on an electronic chip or integrated circuit containing the whole computing system. It typically runs on a single microcontroller unit and also exposes General Purpose Inputs/Outputs (GPIO) and is programmable using a compatible Integrated Development Environment (IDE) [25].

In todays market there are several solutions for this type of projects, each ideal for a particular specification. For an IoT system of our nature it is necessary to choose a board that is simple, but with a high performance, that can retrieve information from sensors, as well as transmit and receive messages from other nodes, also with the ability to perform under different specifications or environments. The most common boards that meet these requirements are the Arduino Uno, the ESP8266 and the ESP32.

2.4.1.1 Arduino UNO

Arduino is an open source platform which provides strong bases for software and hardware [26], based on a simple microcontroller input/output board and a development environment that implements the Processing language [27]. At the software level have their own Integrated Development Environment (IDE), where users can develop their sketches using C or C++ or use multiple already developed libraries, allowing to create project to all the Arduino compatible hardware.

Arduino has a great variety of boards, each with different functionalities, being the most common the Arduino UNO, that is represented in Figure 2.7. Composed of an ATMega328 microcontroller with 32 kB of flash memory, has the advantage of having an USB port, through which can be charged and programed, giving it more flexibility to the use.

FIGURE 2.7: Arduino UNO board

The entire Arduino system is compatible with various operating systems (Windows, macOS, Linux), and has become a great tool for hardware and electronics researchers since it provides a simple and low cost platform. However, it has the disadvantage of not being able to run several tasks simultaneously and have a low memory, not being able to singly control systems with numerous tasks [26].

2.4.1.2 ESP8266

ESP8266 is an Arduino compatible SoC, that integrates Wi-Fi and the ability to connect to the network using the TCP/IP protocol. The ESP8266 presents all the features of the Arduino Uno board with the advantage of providing an Internet connection, a lower price, higher internal memory, with 64 kB, and also a Flash memory of 4 MB that allows the storage of data without external devices [28].

The module is suited for IoT due to its small size as show in Figure 2.8, low cost and low power consumption.

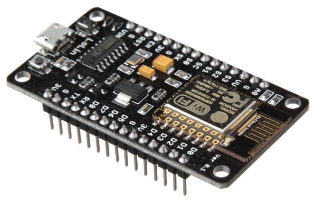


FIGURE 2.8: ESP8266 board

The ESP8266 microcontroller includes 10 GPIOS, with the disadvantage of having only 1 analog pin for analog sensors [22], and being an Arduino compatible hardware it is possible to program through the Arduino IDE.

2.4.1.3 ESP32

The ESP32 is very similar to the ESP8266 chip, being considered as its successor [29]. Combining all the advantages of ESP8266 and taking into consideration its disadvantages, the ESP32 comes along as the most advanced microcontroller in the market. With a better performance, it not only includes WiFi but also BLE in its dual core chip, improving also the GPIO ports, with 32 available, including 12 analog ports, and upgrading the memory to 520kB.

The excellent performance of the microcontroller is achieved due to dual core structure and a significant extension of the operational features. Besides all the stated above, the main feature that allows this board to be ideal for IoT systems is the ability to enter into a deep sleep mode when it is not sending information, reducing the normal electrical current consumption from 20 mA to $100\mu A$ [30].

This allows the node to be connected just for short periods of time, enabling it to be powered from batteries.

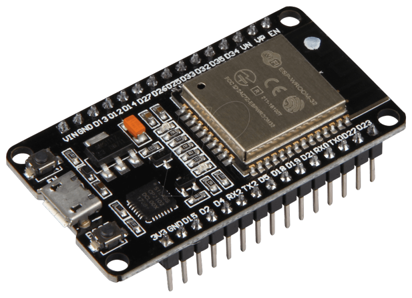


FIGURE 2.9: ESP32 board

Like the ESP8266, the ESP32 represented in Figure 2.9 is a good option for projects that need a SoC with a reduced size, Internet access, with a low cost that does not compromising system quality, low power consumption capability and with the addition of having the deep sleep mode.

2.4.2 Transceivers

Since the controlling platforms presented in the previous subsection do not have the capacity to transmit the communication protocol chosen in section 2.3, it is necessary to choose a transceiver that does this function, so that there can be communication between the nodes.

The RFM95W and RFM69HCW are transceivers feature the LoRaTM long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption [8,9], the Figure 2.10 shows an example of an RFM69HCW. Both radios transmit at a frequency of 868 or 433 MHz, a very low frequency compared to Wi-Fi, Bluetooth and ZigBee which use 2.4 GHz. Although the higher the frequency, the faster you transmit, the data we want to send is so small that the speed offered by these transceivers is not affected. In addition, it is much easier to transfer in LoRa than Wi-Fi and Bluetooth [31].

FIGURE 2.10: Transceiver RFM69HCW

The difference between both radios is slight, having the same main characteristics, which can be seen in Table 2.6 [32].

TABLE 2.6: Difference between RFM69HCW and RFM95W

Transceiver	**RFM69HCW**	**RFM95**
Range [m]	500	2000
Power Consumption [mA]	150	100

Given the similarity between the two, the radio to be chosen is the one with the greatest range and a reduced power consumption, criteria that the RFM95 meets.

2.4.3 Remarks

There is a great variety of controlling platforms that can be adapted for IoT projects, but it must be taken into account that the intended system requires a low level of complexity, with small dimensions, with several analog and digital ports for the various sensors and that can be powered by batteries. For this, it needs to have a low power consumption, so that the batteries have the greatest possible lifespan. As it turned out, ESP32 is the most viable option given the desired characteristics, and performs much better than its predecessor ESP8266, being widely used in a large variety of IoT applications. As far as the rest of the hardware is concerned, it has already been shown that the RFM95W transceiver was the best choice, as well as which sensors and how they would be distributed.

2.5 Software

After the aggregation node transmits the messages that contain the information collected by the sensors to the server, it is necessary to analyze and process all that information and check if the values go as intended and if they are not alert the user.

In the following subsections, the best machine learning algorithms are being studied, necessary to understand if there is any abnormality and to give sustainability tips to the user. In addition, the way in which the smartphone application will be developed, which is responsible for showing information from sensors and receiving system alerts from the server.

2.5.1 Machine Learning Algorithms

Machine Learning (ML) is a field of artificial intelligence that allows you to develop very precise software applications to predict results, even without being expressly programmed for this function [33]. By other hand, ML systems can learn from data, identify patterns and make decisions with minimal human intervention, the author of states that machine learning is then a method of data analysis that automates the construction of analytical models.

The way in which machine learning algorithms can predict results is due to the fact that the algorithm is given access to historical data, creating a dataset, letting the algorithm learn, that is, iteratively adjust the model of knowledge representation so that it improves its performance. After this learning process, the algorithm is able to make quality predictions in future situations that are related to historical patterns [34].

Machine learning is important because it allows you to build precise models, since it is possible to produce, quickly and automatically, models with a large and complex number of data.

There are several machine learning algorithms, but their effectiveness depends on the type of dataset that will be analyzed. According to the author the most efficient algorithms for the purpose of this system are presented below [35]:

- **Support Vector Machine (SVM)** - Used for classification, classifies data by building an n dimensions between two classes using the distance between the nearest points, differentiating between two classes with minimal errors [36]. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples [37]. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side, as it is possible to visualize in Figure 2.11 .

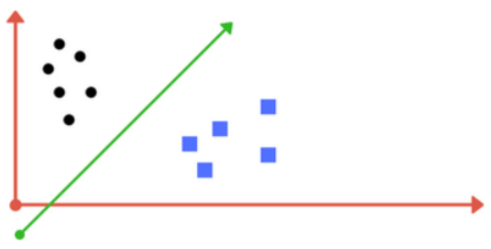


FIGURE 2.11: SVM Algorithm

- **Decision Trees (DT)** - Provide a classification via an hierarchical partition of training data, where a certain feature is used to split data, being this split done iteratively until the leaf node contains a significant number of particular records that can be used to classify data [35],[34]. According to

[38] this algorithm has some limitations because if there is a small change in the training dataset, it can result in a big change in the tree, not being able to predict with good prediction the next values.

- **Neural Network (NN)** - Works like the nervous system, using neurons in a bio-inspired system with multiple layers. Each neuron analyzes a part of the inputs, passing the information to the next layer and neurons, until valid output is reached [35], as it is possible to visualize in the Figure 2.12. This type of networks is ideal for nonlinear and complex problems, but as it implies big computational power is not ideal for IoT systems [34] data.

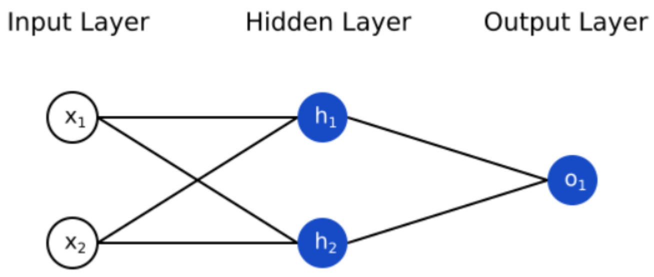


FIGURE 2.12: Neural Network Algorithm

- **Random Forest (RF)** - Highly applicable for classification problems, incorporates the process of aggregation, bagging and DT, selecting a subset of features from individual nodes of the tree, avoiding the correlation in the bootstrapped set [35]. The classification is computed by out of bag classification error.

2.5.2 Smartphone Application

The Smartphone application will serve as an information center, a place where all the consumption parameters and data are displayed. This information will be sent periodically, with the sensor values and a graph that compares the sensor data specifically, to ensure the user has all the details on demand.

With this information, the user can realize if any resource is being spent in an unnecessary way, and consequently saving money and natural resources. In addition, the software will forward an alert if any dangerous situation occurs, like for example, a gas leak.

To create a Smartphone Application, it is necessary to realize which is the best operating system where this application will occur. Currently there are two systems that are the most used: Android and iOS.

Android is an open source operating system developed by Google, based on Linux designed for mobile devices such as smartphones and tablets. Java is the language used as a base, the most common IDE is AndroidStudio and can be run on any computer [39].

iOS is developed by Apple, designed exclusively for the brand's smartphones and tablets, iPhone and iPad. Apple does not allow this operating system to be run on hardware that does not belong to it. The language used by iOS is Swift and to run a specific IDE for this language implies buying an Apple computer, which requires a large investment [40].

Globally, Android is the most used operating system, with a market share of about 74% , as show in Figure 2.13.

With all these information, it makes sense to develop the application for the Android operating system.

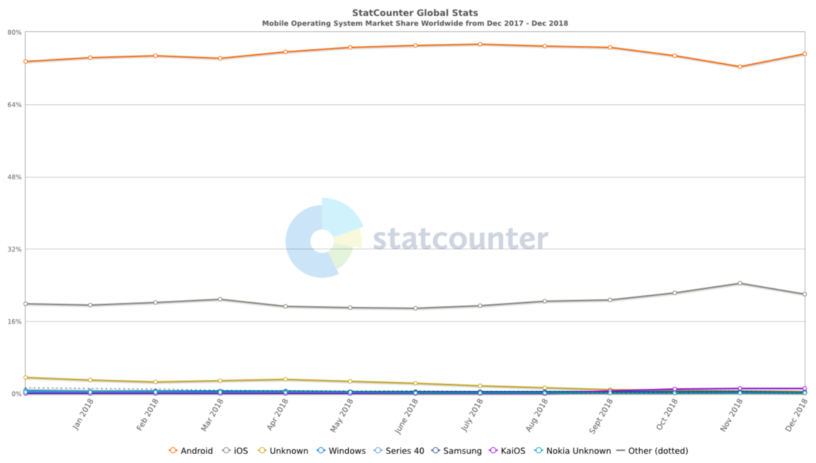


FIGURE 2.13: Global Operating System Market Share

2.6 Related Work

With the advances and improvements of today's technology, the necessity to create projects that can monitor homes to avoid risk situations and also keep track of consumptions, increased. The challenge in creating a project that can monitor a smart home, with reduced costs, in addition to being safe, adaptable and energy efficient, it is something we can see in various projects already developed.

In the literature it is possible to find several proposals [1] to monitor smart homes, using low cost hardware devices, such as Raspberry Pi and Arduino. But these proposals have limitations in the mode of transmission or in the central controller, not being sufficiently adaptable or efficiently energetic for long-term use.

The author in [41] developed a low-cost home automation system, using WSN technology to monitor and control the environment, safety and electrical parameters of an interconnected home using an Android application so the user can control over the devices in a smart home. The disadvantage of this system is the Wi-Fi communication protocol that has an high power consumption and low range, making it impossible to use in a large scale environment.

Imron Rosadi [42] implemented a WSN for a small area in a building that allows sending data in short distances, with the disadvantage that when packets have more than 10 bytes they get delayed, making packet size something to bare in mind in order to get the optimal transmission speed at the selected network configuration. Another disadvantage is that the network cannot operate more than five days without additional energy.

Minh-Triet Nguyen [43], offers a solution where the devices can transmit information up to eight floors without compromising is data flow, in other words, without losing any information packets. This can be useful to extend the monitorization from a simple house to a building.

There are already some IoT systems using WSN to monitor smart home, although with some limitations that compromises its use in an efficient way. Our proposal aims to answer the efficiency problem, using low power hardware, combined with a more optimized software and a long range, low power and reliable communication protocol, in order to improve the WSN lifespan.

Chapter 3

System Architecture

The main objective of this dissertation is to develop a decentralized solution based on a WSN that gathers information according to pre established parameters and subsequently send this information to a server, through an appropriate communication protocol given the range, number of nodes, reliability and cost. This data will then be analyzed and make available to the user through an Android interactive dashboard, that also warns the user through notifications when dangerous situations occurs.

In the previous chapter, several systems were described that were quite similar, but did not meet the desired requirements. Taking these systems into account, a new IoT system was developed from scratch, which considered several hardware and software modules, as can be seen in Figure 3.1.

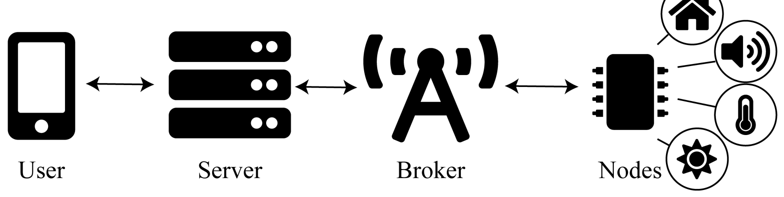


FIGURE 3.1: System Architecture

In this chapter the proposed architecture for a new IoT system will be described in detail, including the hardware and software modules that when used together create final system.

3.1 Hardware

The proposed system in terms of hardware, as represented in Figure 3.1, is based on a typical WSN, as described in Section 2.2, being composed by a set of sensor nodes and a aggregation node.

By using different types of nodes in a distributed way, we are ensuring that the information is not contained in a single place and allowing the collection of various parameters in different locations. These nodes follow the normal structure of a WSN, with the ability to collect data and communicate among each other. Besides that, by using a Star topology, the proposed network gains the advantage of being simple and flexible, allowing the addition and/or removal of nodes at any time, without compromising the network or being necessary to have human interaction.

Despite different functions inside the network, the aggregation node being responsible for broadcasting information and the sensors just gathering data, they share the same hardware in their main core. Each node was created in a way to promote efficiency, low cost and low-power, being used the best microcontroller, communication module and other hardware, as study on Section 2.4 and which will be described in detail in the next sections.

3.1.1 Aggregation Node

The aggregation node is the main node of the network, since it does not collect data, it just sends the information received from the other nodes to the server. In this case, this node serves as a bridge between the sensor nodes, responsible for collecting data, and the server, responsible for processing that information.

The protocol for the communication between the aggregation node and the sensor nodes was chosen based on the best low power and long-range wireless technology to serve this system in the most efficient way. As it was studied in the previous chapter, the best solution was to use LoRa, since it is a bidirectional communication protocol that can provide a low power, long range communications, as well as support a large number of devices and increase the battery life, with a reduced device cost.

To communicate with the servers, the MQTT protocol is used by the aggregation node, via a Wi-Fi connection. This dual communication feature for the aggregation node is due to the fact that Wi-Fi does not allow having several nodes simultaneously and due to the high power consumption, that counteracts the low power requirements for the system.

The constitution of the aggregation node can be analyzed in detail in Figure 3.2. It is possible to check that the aggregation node is composed by two main elements: a microcontroller and a radio module.

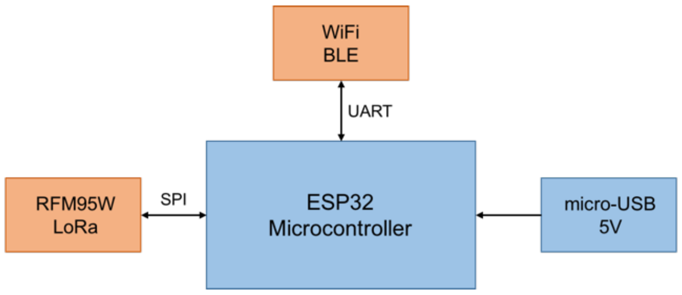


FIGURE 3.2: Aggregation Node Block Diagram

The first is an ESP32 chip, an ultra-low-power microcontroller that comes with a built in Wi-Fi and BLE, being a versatile, low price, high performance and a reliable solution for this system. This microcontroller is powered by a micro-USB 5V, due to the Wi-Fi high consumption, about 160mA, not being able to run on batteries for a long period of time, and the fact that although the ESP32 has a deep-sleep mode, which reduce the normal electrical current consumption, it is not possible for the aggregation node to use this feature, since it needs to be awake at every moment, listening to potential new messages, and to achieve a high level of reliability, no messages can be lost. Since the ESP32 cannot provide a LoRa connection, it needs to be coupled with a radio module capable of transmitting information between a multi-point network. The chosen module was the RFM95W, as it allows transmitting information, with individual node addresses, in a range of 2km between nodes, with a power consumption of 80mA when receiving or transmitting a message. Despite the high range, it is important to place the aggregation node in a strategic place between the nodes, in order to ensure that there is no loss of messages.

3.1.2 Sensor Node

The Sensor node is the lowest level of a WSN network, as well as the simplest. Their sole responsibility is to collect the information from the attached sensors and send it to the aggregation node. In order to guarantee that their responsibilities are being achieved, not only they use the same LoRa network as the aggregation node, but also some specific features must need to be taken into consideration.

A shown in Figure 3.3, and as said before, the sensor node shares the main core of the aggregation node, with an ESP32 microcontroller and the RFM95W LoRa module. The main difference between nodes is the way some of these modules are used, as well as the addition of an array of sensors.

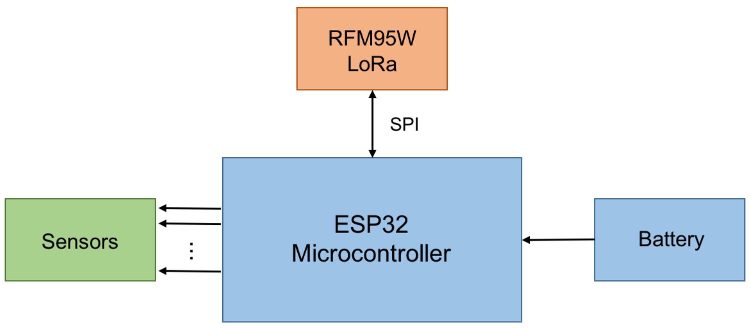


FIGURE 3.3: Sensor Node Block Diagram

The use of the ESP32 differs as it will not be using the Wi-Fi connection, making the LoRa the only communication protocol on the sensor node. As it is not using Wi-Fi, power consumption will be much lower compared to the aggregation node. When the node is not collecting data from the sensors, it enters in the deep sleep mode, where it only consumes 100μA. The total consumption of these nodes, as well as the batteries used and their duration, will be described in detail in Section 4.3.

Each node can have up to 8 sensors connected, since the ESP32 includes a high number of Analogic Digital Converters (ADCs), allowing the node to have different sets of sensors depending on room/division necessities, since the parameters among each division of a house may differ.

The sensor nodes are not all the same, since each room has its needs, i.e, a kitchen does not have the same needs as a room, for example. The distribution criteria of the sensors in each node are as follows:

- **Kitchen** - will be implemented with a gas sensor (MQ2), that has lower conductivity in clean air, because the sensor conductivity is higher when the gas concentration rises [44]. Due to the high hypotheses of failures in this division, this system intends to give the user warnings in case of gasleaks.
- **Garage** - A gas sensor (MQ7)[45], to obtain the levels of carbon monoxide with a detection range between 10 up to 500ppm.

For all of the above and the remaining divisions, each node has sensors for:

- **Humidity and Temperature** (DHT22) that measure the temperature between 40° and 80° Celsius [46], which uses a thermistor capable of measuring the surrounding air and allows new readings to be taken very accurately in a very short period, about 2 seconds.
- **Luminosity** (Light Dependent Resistor (LDR)) that works as a resistance, that whenever the light intensity increases its resistance decreases and from there it is possible to calculate voltage through the electrical intensity [47].

- **Noise** (sound sensor) [48], the main component of this sensor is a microphone, allowing to detect the intensity of the sound in the environment.
- **Motion Sensor** uses a pyroeletric sensor, which can detect levels of infrared radiation, and the hotter something is, the more radiation is emitted, being able to figure out if there's anyone in the room.

These parameters are considered common in an ordinary house and are sufficient to predict abnormal situations and promote sustainability tips. In addition, they have a low-cost, efficiency and a small size, as can be seen in Table 3.1:

Table 3.1: Sensor Features

Component	**Dimension [mm]**	**Estimated cost [€]**
DHT 22	25.1	8.96
PIR Motion Sensor	40	5.90
MQ2	23.3	4.98
MQ7	23.3	5.89
DFR0034	30	6.03
LDR	11.9	0.49

The values considered normalfor these sensors can be found in Appendix B - User & Technical Manual.

3.2 Software

As shown in Figure 3.1, besides the hardware, the proposed system is also composed by software modules. These modules are a fundamental part in the architecture, since they are responsible for the entire logic of this system. The software is divided in two modules, one responsible for processing the information through scripts that run directly in the hardware nodes and in the server, and another responsible for allowing the user to observe and control their network, through an Android mobile application. Each will be described in detail in the following sections.

3.2.1 Support Scripts

In order to fulfill the purpose of this system, collect, analyze and monitoring, in real time, the consumption/activities of a common house, there was a need to implement several scripts, in various programming languages (PHP, SQL, Python and C++), that work alongside the hardware and communication modules to meet the requirements of the system.

These support scripts, that are spread throughout the network, are responsible to keep the LoRa peer-to-peer network working, receive and analyze MQTT messages, store and analyze the sensor information, among other features. For that, a set of support scripts was developed, for each type of node, since we are working with a distributed network, and also for the backoffice in the server. The next section will describe why, how and where each of these scripts were implemented.

3.2.1.1 Sensor Node

Besides being the simplest node in the network, with the purpose to only collect sensor data and transmit them to the aggregation node, a way to this in a low-power and highly efficient fashion was needed. For that, and using the Arduino IDE and a C++ script, a script was developed to run directly on the ESP32 microcontroller and which is responsible to collect the information from the sensors attached, using the corresponding libraries.

The next step is transmitting those information to the aggregation node via the RFM95W LoRa module, and in order to be part of the LoRa peer-to-peer network, the RadioHead library [31] was used, responsible for configuration of the transceiver module, individual addressing and transmission and reception of

LoRa messages. This library is used to guarantee a good communication between the nodes, through a reliable LoRa connection, where no messages are lost. The RadioHead library allows communication to be made in two ways: with broadcast, without acknowledgment from the receiver, which is used when the aggregation node needs to send a message to the sensor node; and the second is to send a message to a specific address with acknowledgment from the receiver, which is used when the sensor nodes send data to the aggregation node [49].

The addressing done by the library allows the node to enter the network knowing in advance the address of the aggregation node, allowing it to be added without any complications.

Each message sent from the sensor node use the same structure in their message payload, making it easy for the aggregation node to retrieve the necessary information. The message consists of the node id, the sensor id, to know which parameter was measured, and the value obtained. As said in the hardware description, the sensor node when is not collecting information goes into deep sleep mode. To do this, the ESP32 own library allows to define a sleep period, after which the board is awaken via a hardware reset.

3.2.1.2 Aggregation Node

The aggregation node is the only point of communication between the server and the network of sensors, being its function to maintain the LoRa peer-to-peer network and distribute the messages in a bidirectional way. Figure 3.4 display this bidirectional communication scheme, that is divided into two parts, the MQTT implementation for communication with the server and the LoRa implementation for intra network communication.

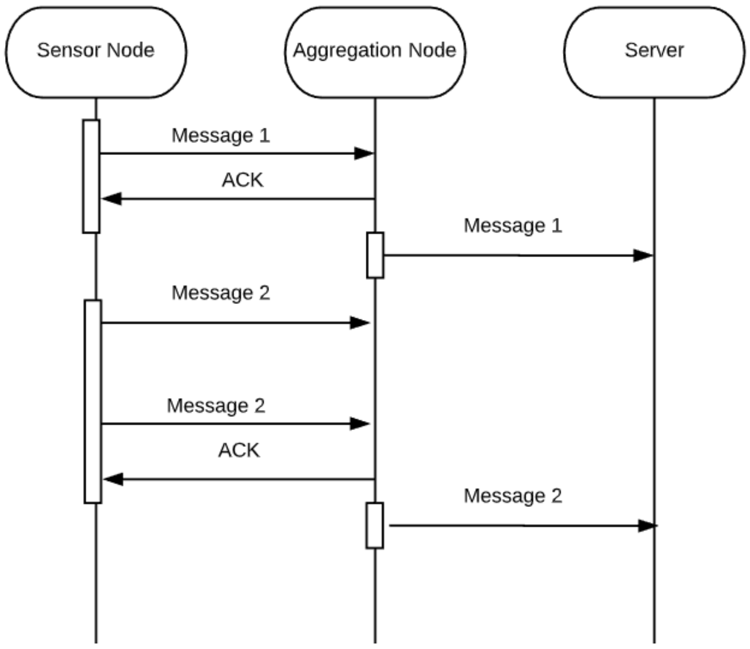


FIGURE 3.4: Exchange Messages schema between nodes

The LoRa implementation has the main function of receiving messages from the sensor nodes, using the RadioHead library.

So that messages are not lost, every time a message is received it sends back an ACK warning that the message has been delivered. Each time a Sensor node tries to transmit to the Aggregation node and this one not replies, the message is resent until an ACK is received, as show in Figure 3.4.

To forward messages to the server, as explained in Section 2.4, the MQTT protocol following a subscriber/publisher model is used. The aggregation node works as a publisher and takes the body of the message previously received from the sensor node and forwards the message, to a topic known to the server. For that, the PubSubClient library was used.

3.2.1.3 Server

The server script is where all the messages from the network are handled and analyzed. For that, a Python script was developed and deployed as a background service in the server.

The logic behind this script is divided into several parts, given its complexity and the fact that the server itself has several features: communication with the aggregation node, sending data to the database, analysis of the values received and send notifications when values considered abnormal are analyzed. Each of these features will be described in detail ahead.

An essential part of this script is to receive the sensor data that is sent via the aggregation node through the MQTT protocol. For that, the Paho library is implemented in order to allow the Python script to subscribe to the topic to where messages are being sent by the aggregation node. So, any message created in the network arrives in the server in real time, in order to be analyzed.

The first thing to be done after a new message is received, is to store the corresponding sensor values. Using the MySQLdb library a connection to a MySQL database was created, that through SQL commands is able to save the information. The need to have a database is due to the fact that we are dealing with a real time monitoring and consequently a large volume of data, so it is necessary to structure them to be later easily accessed.

It is possible to observe the database structure through its relational design in Figure 3.5, which goes according to the message established in the sensor node, that is, each node has a set/array of associated sensors. Each sensor and each node have a specific id, so that in the message it is possible to specify which is the division of the house and the parameter that was analyzed. This database is not accessed by users, so the information is not shared.

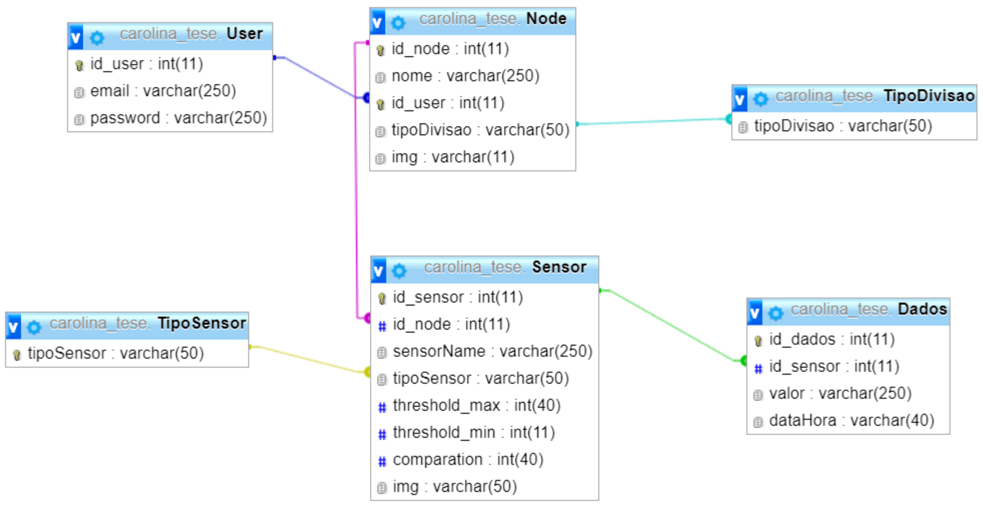


FIGURE 3.5: Database Relational Design

One of the main objectives of this script architecture is to notify the user, in real time, when abnormal values or new events are detected. In order to achieve this feature, the system needs to know what are considered abnormal values bearing in mind the sensor and the node. When a new message arrives, and after being stored, the system processes two types of analyses that go according to the functionalities that the system intends:

- **Threshold Analysis** - is the first analysis of the system and corresponds to the maximum and minimum values that the user has set in the application as normal values, if the sensor value is out of this value range, it is considered abnormal and a notification is sent to the user.
- **Machine Learning** - the system analyzes the values received and tries to predict some event. The algorithm is responsible for analyzing the new value considering the values obtained previously.

In all these analyses, the System sends to the mobile application, through MQTT, a notification warning the user of the event in question. The logic of the script and the analyses described is shown in Figure 3.6:

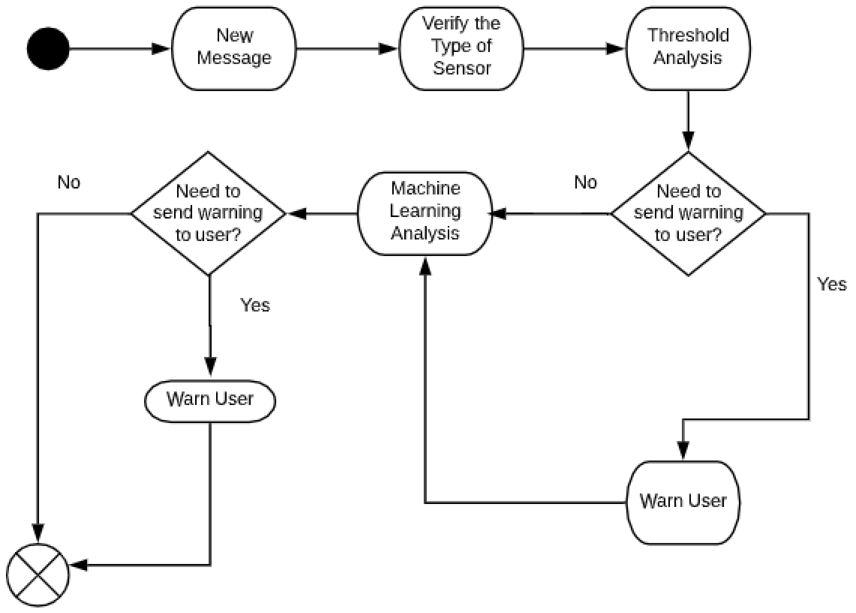

New Message
Verify the Type of Sensor
Threshold Analysis
Need to send warning to user?
No
Yes
Machine Learning Analysis
Need to send warning to user?
No
Yes
Warn User
Warn User

Figure 3.6: Server Logic

Each of these analyses will be described in detail in Chapter 4.

3.2.1.4 Communication with the Mobile Application

It is in the mobile application that the user interacts with the system and understands the values that were collected by the sensors. However, the information reaches the interface through scripts that are running in the application itself. These scripts have an objective: receive the notifications sent by the server and update the sensor data.

When on the server, the value is analyzed and if the server needs to send a warning to the use, this information is sent by MQTT. Therefore, the application script uses the Paho library, which allows the application to subscribe to the topics and act as a subscriber.

For the user to be able to visualize the sensor values and for the application itself to be able to show those values, it is necessary to query the database and make connection with it, for that, PHP scripts were developed, that allowed to search that information, through a request to the server. However, the application cannot make that request by itself, since it does not run on the server and consequently can not access the database for security reasons. So, to make a request to get the desired information from the server, it is necessary to use a WebService.

A WebService is a set of methods accessed and invoked by other programs using Web [50] technologies. When incorporated with the mobile application through its own library, the web service takes the PHP script with connection to the database, which runs in the background on the server, makes the request to the server and returns the data to the application and makes its interpretation, converting them to its own language. In addition to retrieving information from the database, the web service is also able to publish to the database, as it is used in the functionality of thresholds.

In addition to retrieving information from the database, the WebService is also able to publish to the database, as it is used in the functionality of thresholds.

3.2.2 Mobile Application

In order to provide the best user experience, an Android application was developed from scratch so that the user can view all the information collected and previously processed. The application is called "myHomeBoard" and its logo is shown in Figure 3.7.

FIGURE 3.7: Application Logo

The application was developed in Java using the official IDE for Android applications, Android Studio.

While developing the application it was necessary to consider which version of the operating system would be chosen, in order to be compatible with the largest number of users. The official Android website [39] was considered, where is possible to understand that the version with the highest percentage of users is 8.0 (Oreo), and as such it was the target version chosen. In contrast, the minimum version to be used is 5.0 (Lollipop).

As the name of the application indicates, the goal is to provide the user with a dashboard of their own home system, where they can check all the values whenever they want and in real time. The platform also allows the user to be warned when sensor values exceed the ones stipulated by the them, in the form of notifications.

Figure 3.8 represents the flow chart of the entire application, from the moment the user performs the authentication process to the moment he sees all the information of the sensor he wants, as well as all the permitted actions. The operation and logic of the application are described in the following subsections.

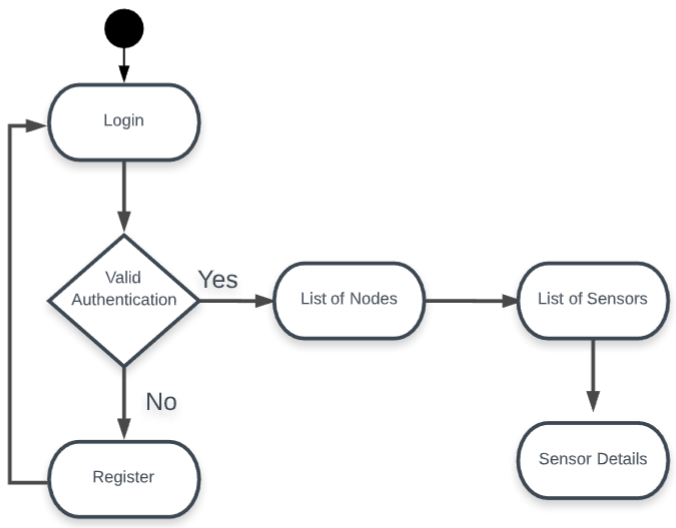


Figure 3.8: Application Flowchart

3.2.2.1 Authentication

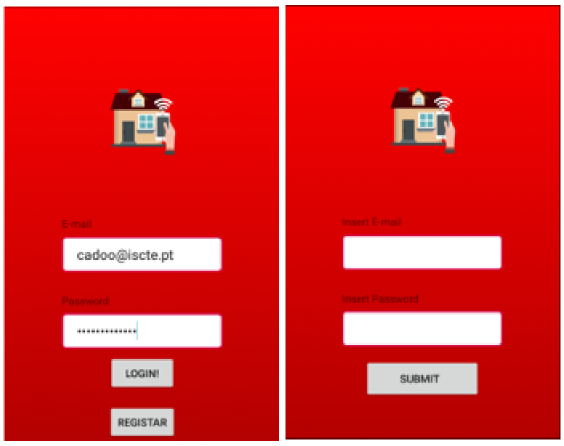


FIGURE 3.9: Authentication Activity (Login and Register)

In order to use the system, it is necessary to create a registration in the application so that the user can successfully authenticate himself. To register a new account, the user only needs to provide his e-mail and a password, that will be protected by an encryption algorithm, coming from the Encryption [39] library used in AndroidStudio, ensuring that both the password and also the information coming from the application is encrypted and decrypted correctly. Access to the rest of the application will only be done if the user has the correct credentials.

3.2.2.2 List of Nodes

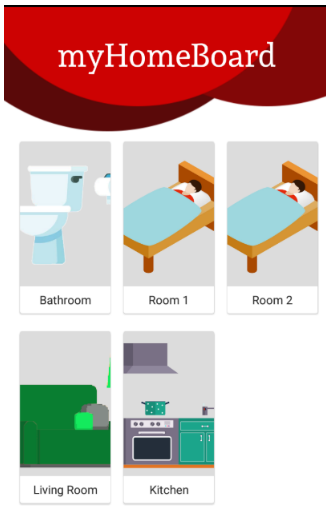


Figure 3.10: List of Nodes Activity

After the authentication process, the application displays all the nodes associated with that user. For that, a request is done to the database, and since a node can only be associated with one user, the user can only see their nodes. In Figure 3.10, the available nodes for that specific user are shown. Also, in this activity, the user is able to choose a specific node in order to get the corresponding sensors attached, which will prompt a new activity containing this information.

3.2.2.3 List of Sensors

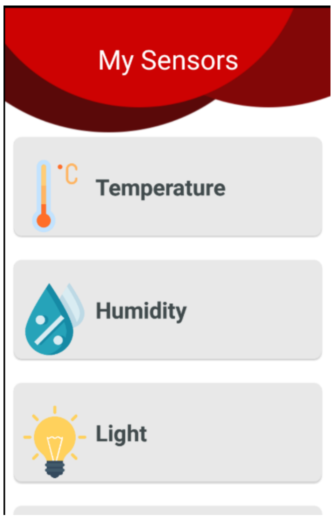
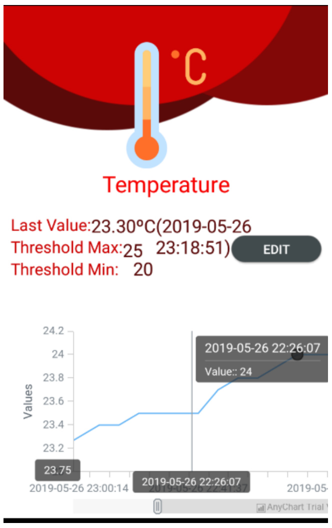


Figure 3.11: List of Sensors Activity

As can be seen in Figure 3.11, the list of sensors that are associated to the chosen node is shown, as well as the last values retrieved by them. As previously mentioned, each node can have different sensors, so the list of each node differs. When the user chose a specific sensor, a screen with all the sensor information is shown.

3.2.2.4 Sensor Details



Last Value:23.30ºC(2019-05-26 23:18:51)
Threshold Max:25
Threshold Min: 20

Figure 3.12: Sensor Details Activity

The last screen of the application, in Figure 3.12 allows the user to see all the details about a specific sensor. For that, besides the last values retrieved and the corresponding timestamp, a graph with all the values stored is presented. The objective of the graph is to give the user an understanding of its consumption, sensor variation and to understand if it can have more sustainable measures. Finally, it also allows the user to define the threshold values for that sensor, the minimum and the maximum values, which will be used by the server script to create warning for the user. This is done per sensor, and not by type of sensor, since two sensors of the same type can have different limits, for example if they are in different divisions.

3.2.2.5 Notifications

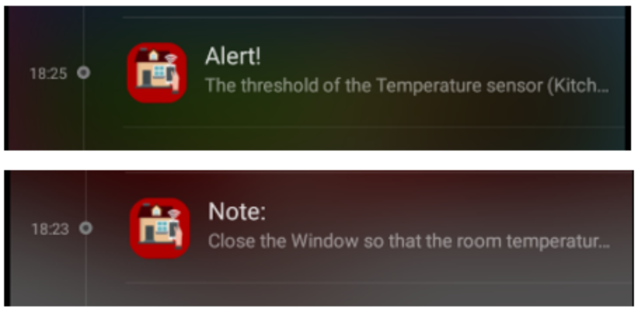

18:25
Alert!
The threshold of the Temperature sensor (Kitch...
18:23
Note:
Close the Window so that the room temperatur...

FIGURE 3.13: Types of Notifications

By observing Figure 3.13, we can observe the two types of notifications that the user can receive:

- Notifications in which the limits chosen by the user himself have been reached, and which may mean danger;
- Notifications with consumption suggestions, as explained above, the user can understand if there is any tip considering the environment, when approaching the threshold values.

Chapter 4

System Implementation

In the previous chapter, the architecture for the proposed system, as well why that solution is the best fit for this project, was studied and discussed. To check if the proposed architecture is viable and to check if all the assumptions made were right, there is a need to implement the complete system, including all the modules of hardware and software, in order to see if the system objectives are achieved or if any additional question, for example the consumptions being higher than expected, are raised.

In this chapter, an experimental test, where a small-scale system was implemented, were conducted to verify if all the modules can be connected, if the communications worked, as well as to verify if the chosen sensors needed to be calibrated. Also, an application scenario was done, very similar to the final one, in order to evaluate the best machine learning algorithm to use and to get results to train it. Finally, a power consumption test was performed, to check which batteries to use in the sensor nodes and how long will they last.

4.1 Laboratorial Tests

In order to evaluate if the proposed system is capable of working as proposed, each of the individual modules needed to be tested and then put together. For that, a set of laboratorial tests, in a controlled environment, were carried out, to understand if each node can perform their tasks, if the LoRa peer-to-peer network is able to sustain the communication between nodes and if MQTT is able to send messages from the network to the server.

The first thing tested, was if a message can be sent from one node to another using the LoRa network with the RadioHead library. The nodes were implemented to represent a aggregation node and a sensor node, and a message was sent every minute, to check the connection reliability.

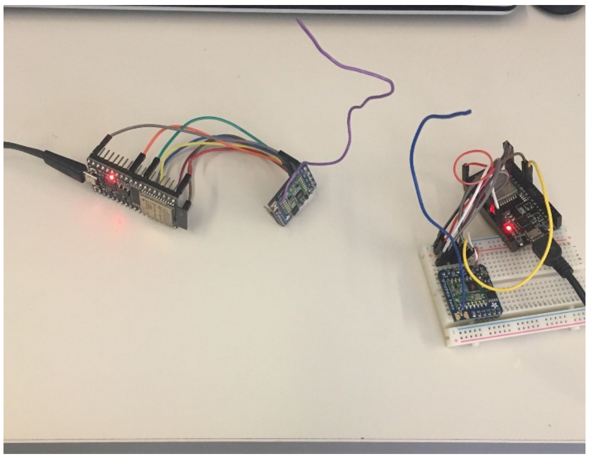


Figure 4.1: LoRa Communication between two ESP32 microcontrollers

Figure 4.1 displays the test implementation, initially one next to the other and then between divisions, to verify if there were messages lost when obstacles were placed in the middle. After a 30 minutes study, for each scenario, it was possible to check that no messages were lost. So, it can be verified that the RFM95W transceiver, studied in Chapter 2, fulfilled the requirements.

After assessing if the LoRa network was able to connect the nodes, a small implementation of the proposed WSN was done, as shown in Figure 4.2, with two nodes, one serving as aggregation node and the other as Sensor node with a temperature and humidity sensor (DHT22). In this way is also possible to comprehend if the sensor node is able to collect sensor information and transmit them to the aggregation node.

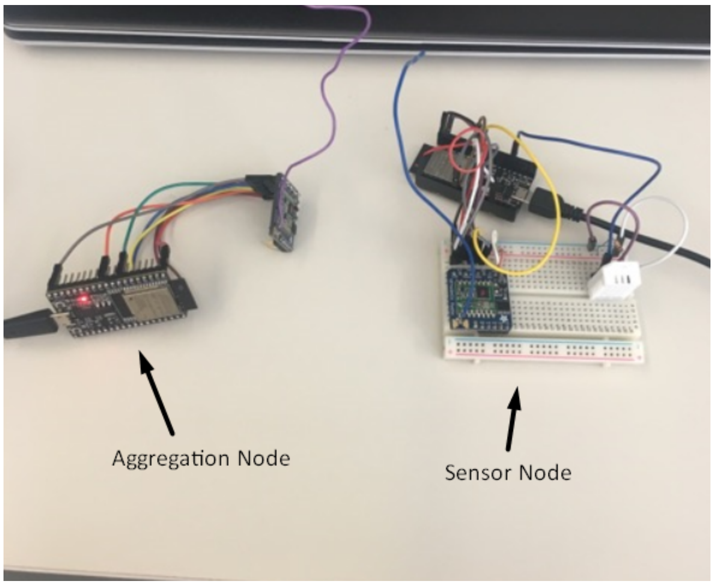
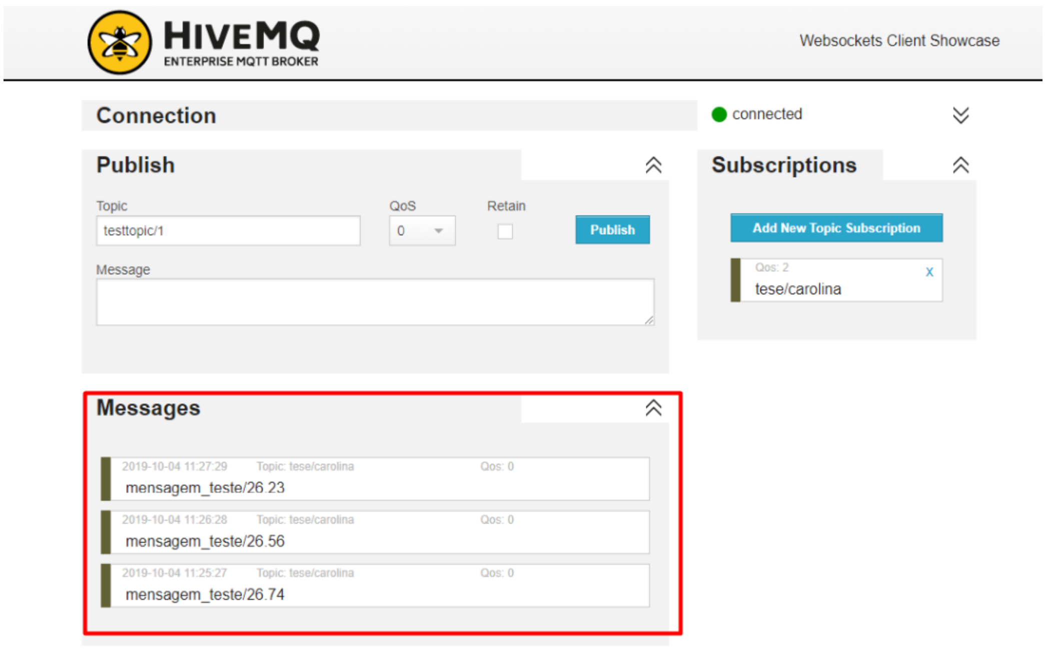


Figure 4.2: Implementation of WSN in laboratory tests

To complete the architecture, the MQTT protocol was implemented in the aggregation node, so that it forwards the sensor value to the server. For that, when a new sensor values was received via LoRa, the message was published into the corresponding MQTT topic. Figure 4.3 shows the MQTT logs where is possible to see that this implementation was done successfully, the server received [51], minute by minute, the values collected by the sensor.

Finally, and to check if the network can handle multiple nodes communicating simultaneously, a second node sensor was added. To better understand the results and how messages are exchanged, the nodes were left collecting and transmitting data for 3 days.

After that it was possible to conclude that the network can handle multiple nodes, but if both sensor nodes sent a message simultaneously, one of the messages was lost. This creates a reliability problem so, in order to create a more efficient solution, an ACK methodology was implemented, in which whenever a node sensor sends a message and does not receive a confirmation ack from the aggregation node, it resends the message after a specific interval until it receives the ack. The test was then repeated, and it was possible to verify that no messages were lost.



Topic: testtopic/1
QoS: 0
Retain
Message

Qos: 2
tese/carolina

2019-10-04 11:27:29 Topic: tese/carolina Qos: 0
mensagem_teste/26.23

2019-10-04 11:26:28 Topic: tese/carolina Qos: 0
mensagem_teste/26.56

2019-10-04 11:25:27 Topic: tese/carolina Qos: 0
mensagem_teste/26.74

FIGURE 4.3: Messages received using the MQTT protocol

These laboratorial tests allowed to understand that the proposed solution can fulfil its objectives, to adjust potential mistakes and also improve the communication scheme.

4.2 Machine Learning Training Scenario

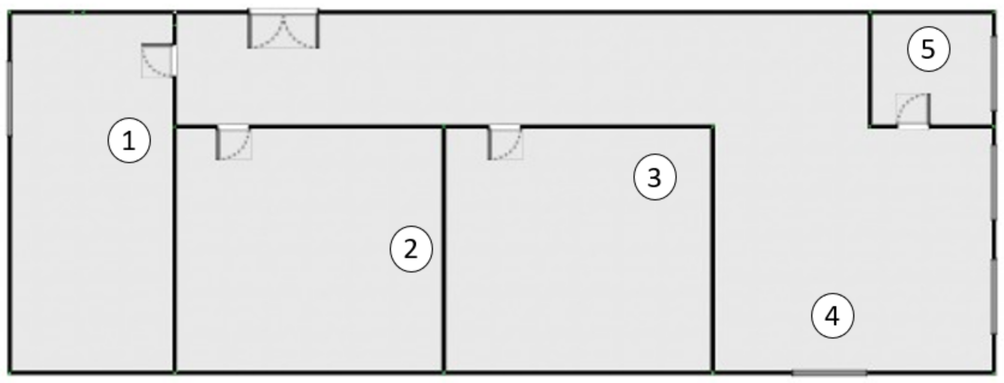
A fundamental part of this system is the machine learning algorithm, which is intended to predict events and give tips to the user. But to be able to predict, the algorithm needs to be trained to understand the differences in the environment, so that in a future event it knows what is happening, given that there are several possibilities of events, that is, different outputs.

In order to train a Machine Learning algorithm, there is a need to create a dataset containing data similar to the one that will be analysed in the future, also containing all possible outputs, since if an output never occurs in the dataset it is impossible for the algorithm to predict it.

In the proposed system, the Machine Learning algorithms will be used for analysing two scenarios:

- **Temperature and Humidity** - Analyse changes in the environment, so that the algorithm can predict what lead to that change and which actions can the user do to counteract, so that the temperature does not reach the undesired thresholds.
- **Motion, Light and Sound sensors** - Predict real-time events, comparing the values of these three values, the algorithm should have the ability to understand whether there is any light/electronic device unnecessarily connected.

For both scenarios, and in order to create the necessary dataset, a small case system was implemented using preliminary prototype with a WSN, composed by an aggregation node and four sensor nodes, as described in Chapter 3. Each of the sensor nodes was coupled with a temperature and humidity sensor, a light sensor, a motion sensor and a sound sensor and were distributed in a house as represented in Figure 4.4.



1-Bedroom 2-Aggregation node 3-Bedroom (without windows) 4-Kitchen and Living Room 5-Bathroom

Figure 4.4: Distribution of the nodes in the Machine Learning Training Scenario

The system, as said, was implemented in a real case scenario, in this case a house inhabited by 3 people, collecting data for a week, every 5 minutes (about

12 samples per hour), being all actions, for example opening a doors or window, stove, lights or heating turned ON/OFF, recorded, with time and location, in order to classify the collected data with all possible outputs. The first thing to conclude by this real case scenario, and another goal of this test besides the dataset creation, was that the network was able to work for a long period of time, without compromising the normal behaviour of the installation site, being able to create a dataset of 1500 samples. After the data being processed and stored, the IBM SPSS Modeler and Statistics software was used to test the Machine Learning algorithms, spoken in Chapter 2, and understand which is the best solution for our proposed system and to gather knowledge from new values. IBM SPSS Modeler and Statistics provides a tool to implement statistical learning, using a set of Machine Learning models. The stream used in SPSS Modeler, from collecting the data from the dataset, divide into train and test groups, train the models and finally apply the test groups to the train models, is represented in Figure 4.5.

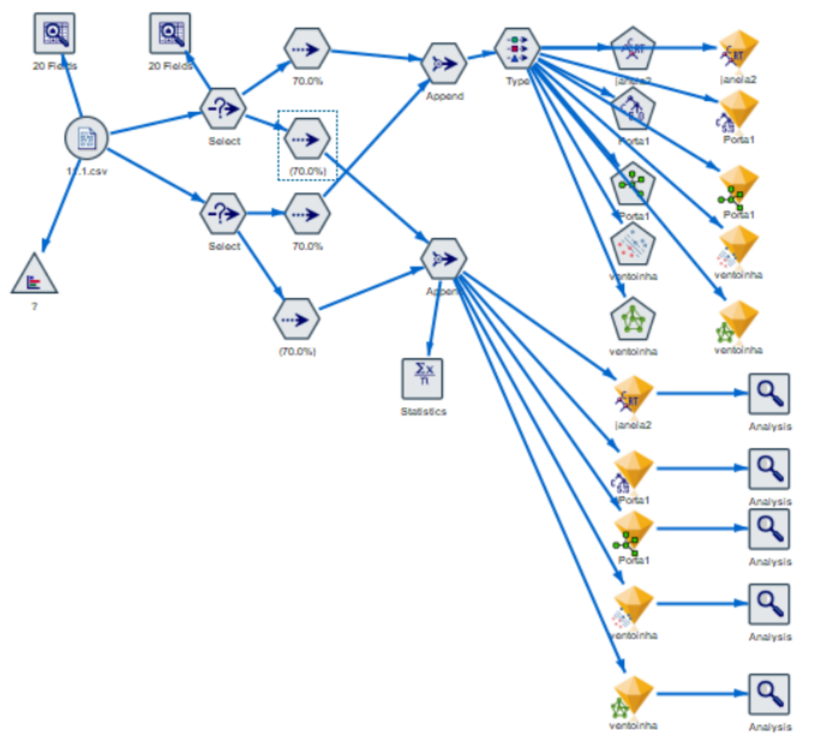


FIGURE 4.5: IBM SPSS Modeler Stream

In order to create, train and test the Machine Learning models, the dataset was divided into a train and test dataset, with 70% and 30% of the original dataset, respectively, for each type of classification. By having a test dataset, it is possible to know how effective the algorithm is.

To further evaluate the selected Machine Learning algorithms, their accuracy cannot be evaluated solely based on model accuracy, since sometimes the results do not show the true accuracy [35]. For that, the Huberty Index, a method used to evaluate model's precision based on the results facing the majority class rule, and the R^2 were used to evaluate the models, being defined by Equation 4.1 and 4.2, respectively.

$$HI = \frac{P_c - P_c^{def}}{1 - P_c^{def}} \tag{4.1}$$

where Pc is the model accuracy and P_c^{def} majority class value, in this case 60.7%,

$$R^2 = 1 - \frac{MSE}{S_y^2} \tag{4.2}$$

where MSE is the Mean Squared Error of the model and S_y^2 the dataset variance, in this case 0.205.

The following subsections, will explain in detail how the data collected were processed in order to create a dataset and the results obtained by each application scenario, given that they were created differently.

4.2.1 Temperature and Humidity Scenario

The goal of this scenario is for the algorithm to be able to predict what actions could be done, i.e. opening/closing doors or windows and turning OFF the AC when it is improperly connected, with the goal of creating a more sustainable environment.

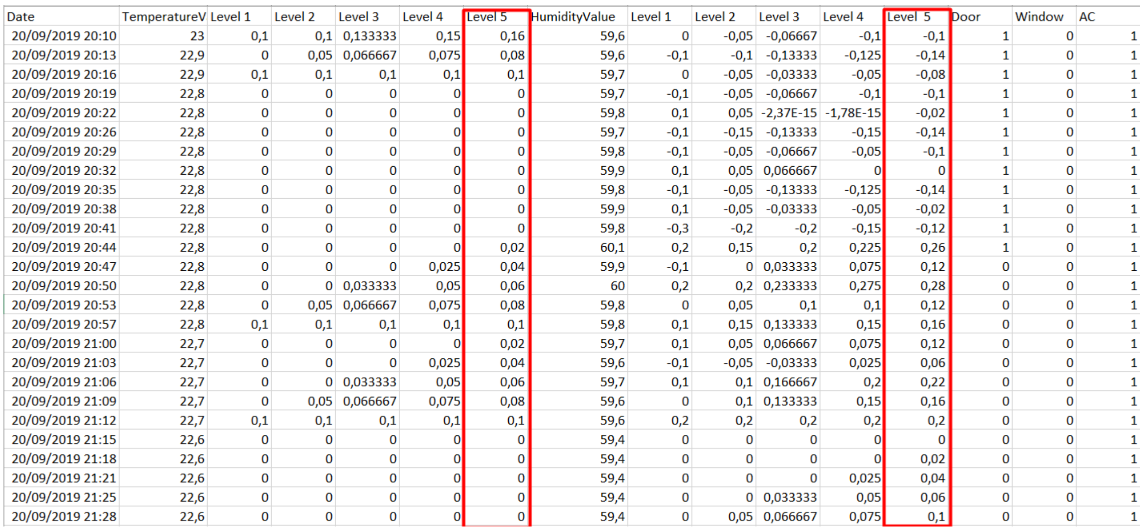
To create the necessary dataset, the collected data needs to be analyzed for each node, comparing it with the notes that were made. These notes indicate when a device was turned ON/OFF or any other action that can affect the temperature, allowing to correlate the collected data with these actions, indicating the corresponding output for each sample. The outputs are classified as 1 if they are ON/open and 0 if they are OFF/closed.

To fully apply this dataset to every possible scenario, the raw temperature and humidity values cannot be taken into consideration, since its very unlikely that two houses have the same conditions. So instead of using the collected values directly, the difference between samples was used, allowing the algorithm to truly evaluate spikes or other changes, that cannot be detected without comparing the new value to the previous ones. For that, a new dataset was created using the collected data, where for each value a level was created using Equation 4.3, which compares the new values to the 5 previous ones, creating 5 levels.

$$V = \frac{n_0 + \sum_{i=0}^{n} -n_i}{n} \tag{4.3}$$

In which n_0 is the new value, n is the level and n_i the n previous level.

The importance of these levels is to try to understand, how many values are needed to use to better predict that there was some change. A dataset sample is presented in Figure 4.6.



Date	TemperatureV	Level 1	Level 2	Level 3	Level 4	Level 5	HumidityValue	Level 1	Level 2	Level 3	Level 4	Level 5	Door	Window	AC
20/09/2019 20:10	23	0,1	0,1	0,133333	0,15	0,16	59,6	0	-0,05	-0,06667	-0,1	-0,1	1	0	1
20/09/2019 20:13	22,9	0	0,05	0,066667	0,075	0,08	59,6	-0,1	-0,1	-0,13333	-0,125	-0,14	1	0	1
20/09/2019 20:16	22,9	0,1	0,1	0,1	0,1	0,1	59,7	0	-0,05	-0,03333	-0,05	-0,08	1	0	1
20/09/2019 20:19	22,8	0	0	0	0	0	59,7	-0,1	-0,05	-0,06667	-0,1	-0,1	1	0	1
20/09/2019 20:22	22,8	0	0	0	0	0	59,8	0,1	0,05	-2,37E-15	-1,78E-15	-0,02	1	0	1
20/09/2019 20:26	22,8	0	0	0	0	0	59,7	-0,1	-0,15	-0,13333	-0,15	-0,14	1	0	1
20/09/2019 20:29	22,8	0	0	0	0	0	59,8	-0,1	-0,05	-0,06667	-0,05	-0,1	1	0	1
20/09/2019 20:32	22,8	0	0	0	0	0	59,9	0,1	0,05	0,066667	0	0	1	0	1
20/09/2019 20:35	22,8	0	0	0	0	0	59,8	-0,1	-0,05	-0,13333	-0,125	-0,14	1	0	1
20/09/2019 20:38	22,8	0	0	0	0	0	59,9	0,1	-0,05	-0,03333	-0,05	-0,02	1	0	1
20/09/2019 20:41	22,8	0	0	0	0	0	59,8	-0,3	-0,2	-0,2	-0,15	-0,12	1	0	1
20/09/2019 20:44	22,8	0	0	0	0	0,02	60,1	0,2	0,15	0,2	0,225	0,26	1	0	1
20/09/2019 20:47	22,8	0	0	0	0,025	0,04	59,9	-0,1	0	0,033333	0,075	0,12	0	0	1
20/09/2019 20:50	22,8	0	0	0,033333	0,05	0,06	60	0,2	0,2	0,233333	0,275	0,28	0	0	1
20/09/2019 20:53	22,8	0	0,05	0,066667	0,075	0,08	59,8	0	0,05	0,1	0,1	0,12	0	0	1
20/09/2019 20:57	22,8	0,1	0,1	0,1	0,1	0,1	59,8	0,1	0,15	0,133333	0,15	0,16	0	0	1
20/09/2019 21:00	22,7	0	0	0	0	0,02	59,7	0,1	0,05	0,066667	0,075	0,12	0	0	1
20/09/2019 21:03	22,7	0	0	0	0,025	0,04	59,6	-0,1	-0,05	-0,03333	0,025	0,06	0	0	1
20/09/2019 21:06	22,7	0	0	0,033333	0,05	0,06	59,7	0,1	0,1	0,166667	0,2	0,22	0	0	1
20/09/2019 21:09	22,7	0	0,05	0,066667	0,075	0,08	59,6	0	0,1	0,133333	0,15	0,16	0	0	1
20/09/2019 21:12	22,7	0,1	0,1	0,1	0,1	0,1	59,6	0,2	0,2	0,2	0,2	0,2	0	0	1
20/09/2019 21:15	22,6	0	0	0	0	0	59,4	0	0	0	0	0	0	0	1
20/09/2019 21:18	22,6	0	0	0	0	0	59,4	0	0	0	0	0,02	0	0	1
20/09/2019 21:21	22,6	0	0	0	0	0	59,4	0	0	0	0,025	0,04	0	0	1
20/09/2019 21:25	22,6	0	0	0	0	0	59,4	0	0	0,033333	0,05	0,06	0	0	1
20/09/2019 21:28	22,6	0	0	0	0	0	59,4	0	0,05	0,066667	0,075	0,1	0	0	1

FIGURE 4.6: Dataset for the Temperature and Humidity Scenario

The Figure 4.6 shows that the differences in values are significant from level 3 onwards, and that if this calculation were not made, it would be more difficult to predict the situations, since the differences may be minimal, for that reason level 5 was chosen.

After the creation of the dataset, it was put through IBM SPSS, as described above. And the results obtained were as expected, considering the study made in Chapter 2.

TABLE 4.1: Machine Learning Algorithm Results

Algorithm	**Accuracy** (P_c)**[%]**	**MSE**	**HI [%]**	R^2
Decision Trees (C5.0)	96.09	0.039	90.1	80.9
Random Forest	98.97	0.009	97.3	95.61
SVM	83.68	0.162	58.47	41.46
Neural Net	92.99	0.069	82.16	66.30

Considering that a dataset was created for each node and that these obtained very similar values, Table 4.1 shows the one that obtained better results, the node implemented in the kitchen/living room. The remaining tables will be in Appendix

C. Through the analysis of the table, it is possible to see that the algorithm with best accuracy is Random Forest.

Although the remaining algorithms have a very good accuracy, there is a very sharp decrease in the efficiency of the algorithm when using other methods to prove its efficiency, as is the case of the SVM and the Neural net. However, Random Forest is the only one that does not present a sharp decrease in efficiency indicators and is able to have values very similar to its accuracy.

Based on this analysis, Random Forest will be the machine learning algorithm to be implemented in the final prototype. We can also observe that although the temperature can vary in a very short interval, the algorithm can reach a good accuracy, so the monitoring interval will be kept at 5 minutes.

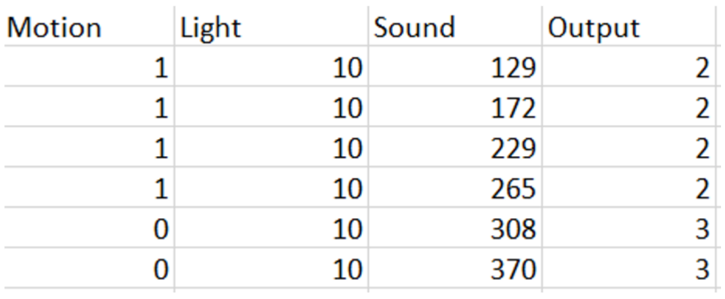
4.2.2 Noise, Motion and Light Scenario

The purpose of this test is to understand whether, when relating the values of three sensors, it is possible to encounter abnormal situations. These situations are based on whether the user has left an electronic device or lights turned ON unnecessarily. The sensors that are responsible for predicting these situations are: movement, light and sound.

The importance of machine learning in this scenario is that, with a suitable dataset, you can see if the new values received correspond to previously detected events and thus predict what is actually happening.

In order to do so, when collecting data from the sensors, the actions that occurred at the time were noted, so that when creating of the dataset it was possible to assign an output to the values. The defined outputs consider all the situations that occurred during the period in which the sensors were collecting information:

1. All OFF
2. Light ON + Television OFF
3. Light ON + Television ON
4. Light OFF + Television ON
5. Natural Light (Day) + Television ON
6. Natural Light (Day) + Television OFF



Motion	Light	Sound	Output
1	10	129	2
1	10	172	2
1	10	229	2
1	10	265	2
0	10	308	3
0	10	370	3

FIGURE 4.7: Dataset for the Noise, Motion and Light Scenario

Through Figure 4.7 it is possible to verify a small example of the 1500 samples that were analyzed for a specific node. Whenever the values arrive these will be analyzed through the machine learning algorithms using this dataset as training, in order to understand which output is occurring at that particular moment. If there is no movement, and the outputs indicate that something is on, a message will be sent by MQTT to the user to turn off the light or the electronic device, depending on the output.

Taking advantage of the results of the previous test and in order to check whether Random Forest maintains a good performance in a different data set, only this one has been tested in this scenario.

Through Table 4.2 we can observe that the Random Forest goes according to the intended and that the same machine learning algorithm will be used in both scenarios.

Table 4.2: Machine Learning Algorithm Results

Algorithm	**Accuracy** (P_c)**[%]**	**MSE**	**HI [%]**	R^2
Random Forest	97.12	0.016	92.7	92.19

4.3 System Power Consumption

One of the objectives established from the beginning, was to develop a system capable of being powered by batteries, consuming as little as possible in a low-power efficiency. For this purpose, the last test done was to analyse all the consumptions of the developed nodes. This was done using a digital multimeter, capable of detecting low values of current, such as the ones occurring during deep sleep cycles, as shown in Figure 4.8.

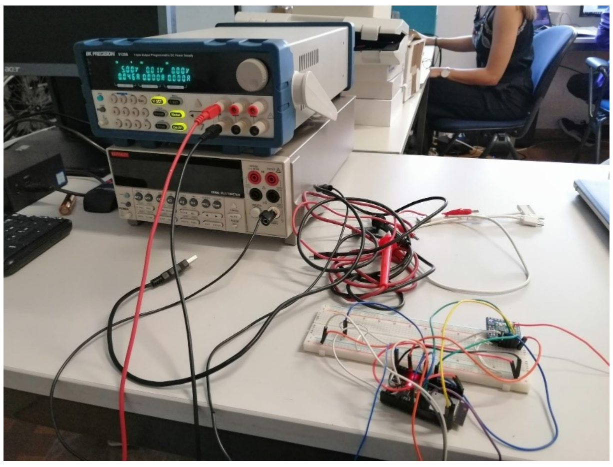


Figure 4.8: System Consumption Analysis

In Table 4.3 is shown the power consumption values obtained from the sensors and modules that will be used in this system:

Table 4.3: System Consumption Analysis

Component	**Type**	**Eletric Current Consumption [mA]**
DHT 22	Sensor (Temperature and Humidity)	2
PIR Motion Sensor	Sensor (Motion)	0.150
MQ2	Sensor (Gas)	92
MQ7	Sensor (Gas)	92
DFR0034	Sensor (Sound)	2
LDR	Sensor (Light)	-
RFM95 (transmitting a message)	Transceiver	80
ESP32 (in deep sleep)	Microcontroller	0.100

In order to calculate the system power consumption, we need to bear in mind the batteries that will be used. For this system two AA batteries were used, given their practicality and that they are easily replaced by the user. The capacity of the 2AA batteries is about 1500mAh (5400000 mAs).

In order to calculate the battery life, it must be considered that the sensor node takes about 2 seconds to collect and transmit through the RFM95W and that monitoring is done every 5 minutes (300 seconds), about 12 cycles per hour. When the sensor is not collecting and transmitting, it is deep sleeping and has a electric current consumption of only 0.100 mA. To determine the consumption of the system, calculations were then made based on the Equation 4.4, where the *Duration* is in seconds (s), the *Capacity* in mAs and the *ElectricCurrent* in mA:

$$Duration = \frac{Capacity}{EletricalCurrent} \tag{4.4}$$

In order to calculate the *Duration* of the system, first it is necessary to calculate its *Consumption* and have in mind that it differs from when it is transmitting ($C1$) and when it is deep sleep ($C2$), as can be seen in the Equation 4.5, where the *Consumption* of the system is measured in mAs:

$$ConsumptionSystem = C1 + C2 \tag{4.5}$$

Both capacities are measured by their power consumption and the time in which they are spending that consumption, where Electric current consumption is in mA and *Time* is seconds.

$$Capacity = \frac{PowerConsumption}{Time} \tag{4.6}$$

And finally, to understand how long it lasted in hours, the obtained value was divided by the number of cycles per hour,

$$Duration(h) = \frac{Duration(s)}{Cycles} \tag{4.7}$$

Based on these equations, we can understand the system power consumption, but we must account that the nodes differ, and consequently their power consumption. Table 4.4 is a synthesis table with the values obtained through the formulas presented for the best case, the node with better power consumption, and the worst case.

Table 4.4: System Consumption Synthesis

	Worst Case	**Best Case**
Consumption of the system to be be collected and transmitted	352.3 [mAs]	164.3 [mAs]
Consumption in Deep Sleep	29.8 [mAs]	29.8 [mAs]
Consumption	382.1 [mAs]	194.1 [mAs]
Duration	14132.45 [s]	27820 [s]
	49 days	**96.6 days**

As mentioned in the previous chapter, due to the high consumption that the Wi-Fi presents, the aggregation node cannot be powered by batteries, but can be connected directly to the outlet. We can observe that in the worst case, a node only needs to change batteries every 49 days.

We can conclude that in the worst case, the batteries can last about 49 days, and that in the best case can last about 97 days. Comparing these scenarios we can see that gas sensor has a very high consumption, accounting for more than half the consumption of the battery. Even so, it is possible to obtain a considerable value for the worst case.

Chapter 5

Results

In this chapter is presented the final prototype of the system, with all the modules implemented, including the logic in the server with all the logic of the system. It will be demonstrated how this prototype was implemented in a real case scenario and its behavior. Finally, an analysis of the results will be made, considering the system performance during the implementation in a real case.

5.1 Prototype

After performing the experimental tests and understanding the entire architecture of the system, a final prototype was achieved with all the components for its operation. In Figure 5.1 it is possible to visualize a Sensor Node base with all the implemented components and an aggregation node.

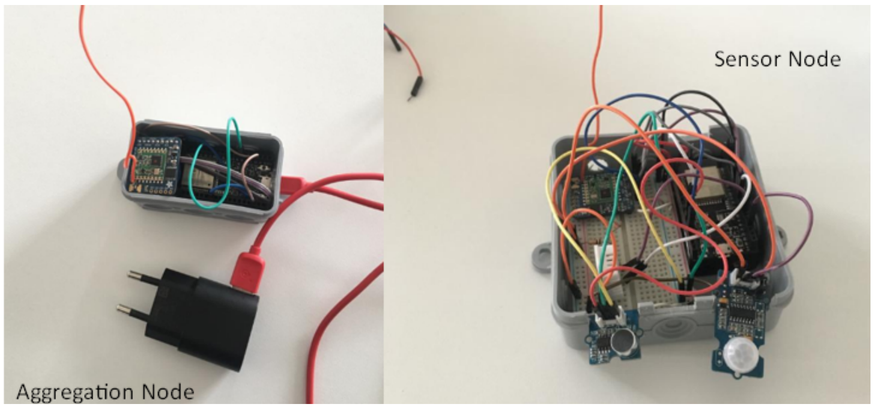
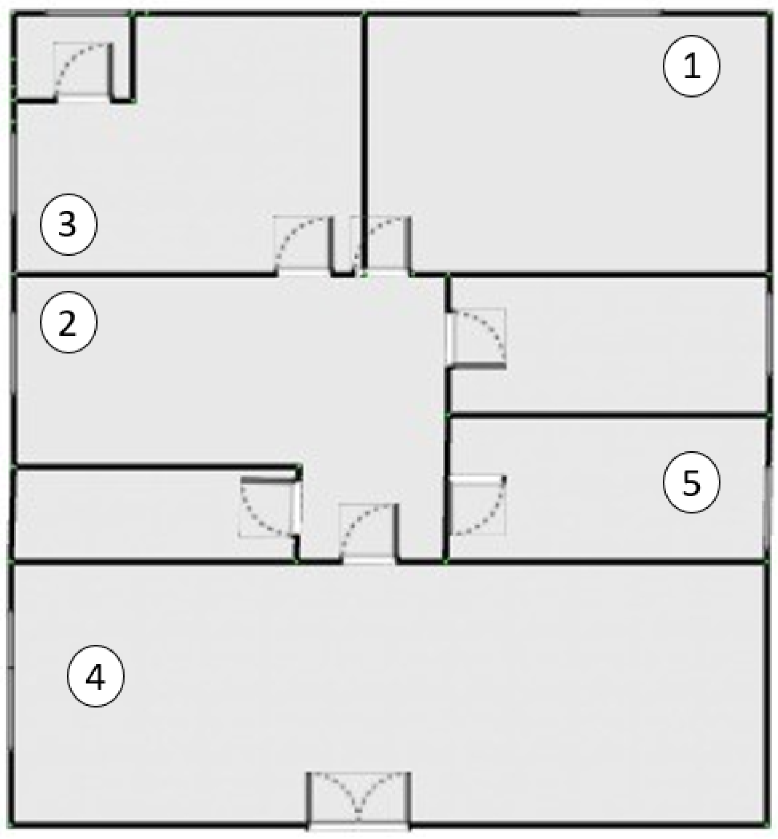


FIGURE 5.1: Aggregation and Sensor Node in detail

In order to implement the final prototype, it was necessary to take into account that the only way to verify the effectiveness of the system would be to implement it in a real case and in a different house than the one used to conduct the Random Forest training. This way, it is possible to test the efficiency of the system at a deeper level, since the implementation in a different scenario allows to evaluate the system ability to act under different circumstances and specifications and indirectly test other components, like communications, which, despite the various tests, is something that can always change their behavior from location to location, since houses can change architecture and placement, as well as and building materials.

Another criteria that lead us to choose another house to test the system, was the fact that the house where the final prototype was implemented, allowed a greater number of nodes simultaneously, since it had a higher number of divisions. Finally, a new scenario was also a way to test the trained machine learning algorithm, to see if it is able to adapt in a house where it was not trained.

Through Figure 5.2 it is possible to observe the layout of the house where the system was implemented.



1-Bedroom 2-Aggregation node 3-Bedroom 4- Kitchen 5-Bathroom

Figure 5.2: Distribution of the nodes in Real Scenario

For the test in this scenario, six nodes were implemented including, one Aggregation node and five sensor node, divided into a bedroom with windows, a bathroom, a kitchen, a living room and a garage, that is not shown in Figure 5.2, since the test house is in a building and the garage is on a different floor. The aggregation node was implemented in the most central room of the house, on a table near a socket, since it is not dependent on batteries. Figure 5.3 shows its implementation.

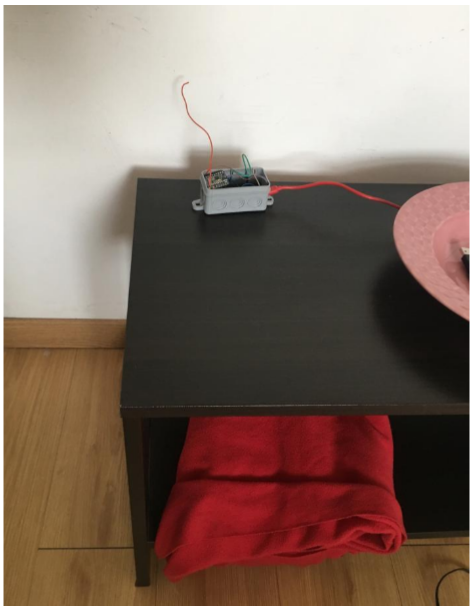


FIGURE 5.3: Aggregation Node Implementation

The sensor node implemented in the bathroom, Figure 5.4 , was placed on a bench, since the bathroom is a small room with only one window, not being necessary to place it in any strategic place.

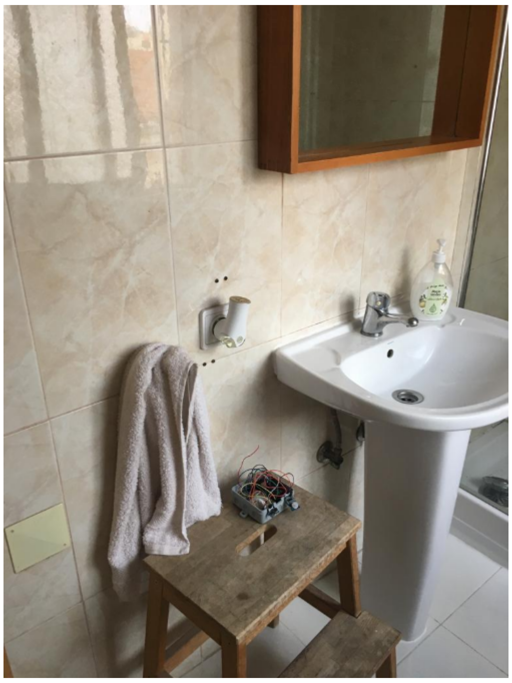


FIGURE 5.4: Bathroom Sensor Node Implementation

In both bedrooms the placement of the nodes was done in the most central place, in both cases a desk, as displayed in Figure 5.5.

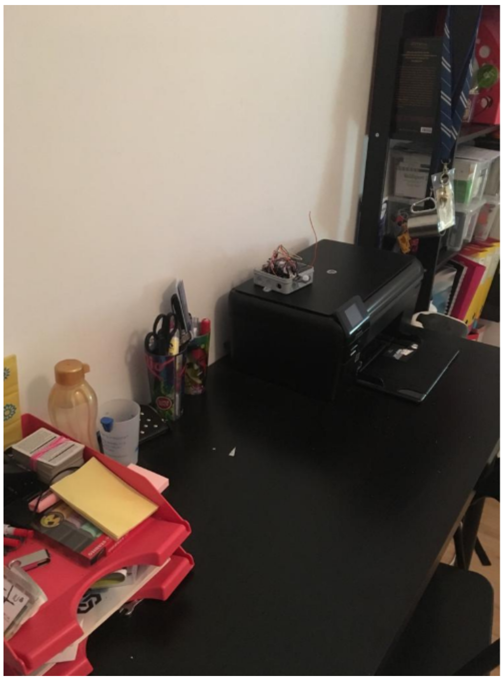


FIGURE 5.5: Bedroom Sensor Node Implementation

In Figure 5.6 , it is possible to observe that the kitchen node was implemented right next to the stove, which will easily identify when the user leaved the stove on.

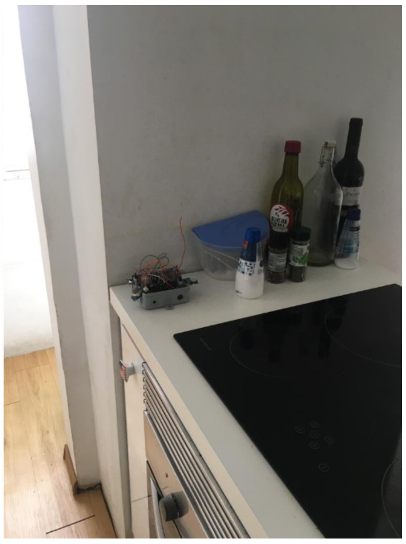


FIGURE 5.6: Kitchen Sensor Node Implementation

The system was implemented in an apartment, so the garage is on a different floor, with a difference of 4 floors, being the garage shared with other apartments. In Figure 5.7 the implementation of the garage node is presented.

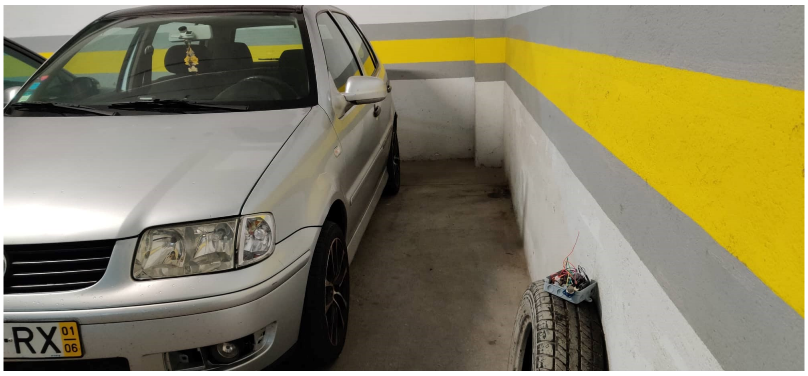


FIGURE 5.7: Garage Sensor Node Implementation

After the system implementation, it was left running for two straight weeks, collecting data every 5 minutes, as discussed in section 4.2.1, and specific test cases were created to verify if the system had the correct interpretation for each case. The test cases created were as follows:

- **Thresholds** - The thresholds were established through the smartphone application, very close to the values being collected, in order to know if the system responded effectively and sent the desired notification when a value exceeded the threshold.
- **Sensor, Light and Motion scenario** - All the situations described in Section 4.2.2 were tested, with specific action being done to check if the correct output is achieved.
- **Temperature and Humidity** - As the previous scenario, throughout the implementation, the situations in which the system should alert the user, suggesting an action, so that the thresholds were not reached, were performed, in order to analyze if the system can detect the correct output.

- **Communications** - To check if all communications scenarios worked, communication to the internet, in the aggregation node, was removed to see if the messages were lost.

Despite the test cases, the normal use of the system and its outputs were also tested.

5.2 Results

As previously mentioned, the system was implemented in a real case, during 2 weeks, monitoring every 5 minutes and test cases were created in order to verify the effectiveness of the system and all its components.

As far as the hardware itself, the sensor node was powered by batteries and at none of the nodes the replace the batteries, during the two weeks of implementation, was necessary. Regarding the sensors, the temperature and humidity sensor (DHT22) collected 4000 samples, in which 279 had no results, so it is necessary to considering a more accurate sensor in a future work.

For communications, communication to the Internet in the aggregation node was removed, to see what would happen to the collected messages sent by the sensor node. As expected, they were received after the aggregation node was connected, since the sensor nodes were continuously sending the messages until they received the confirmation ACK. Another very positive aspect in this system was that the sensor node implemented in the garage was able to successfully send all the messages and that it was not a complication for LoRa, since this room is on a different floor. On the MQTT side, the server was able to subscribe and receive the aggregation node messages, as well as send to the application without any problem.

In relation to the treatment of information and the scripts executed on the server, all these were correctly implemented in the final prototype and there were no problems in its implementation.

To understand the effectiveness of the thresholds analysis, test cases were created so that they were easily reached in order to understand if the server processed correctly and sent the notification to warn of the event, as can be seen in Figure 5.8 , where it is possible to view the reception of the message when the threshold is reached. There were no failures in this functionality, and all test cases were successfully performed.

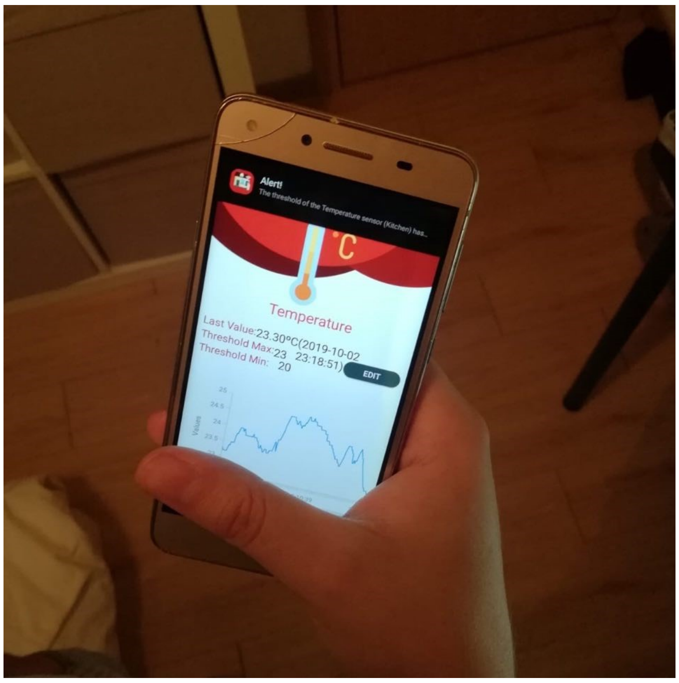

Alert!
The threshold of the Temperature sensor (Kitchen) has...
Temperature
Last Value:23.30ºC(2019-10-02 23:18:51)
Threshold Max:23
Threshold Min: 20

FIGURE 5.8: Alert Notification when threshold is reached

Although the thresholds functionality was successful and the server was able to identify the situations, considered abnormal by the user, the analysis of the machine learning algorithm obtained some disparities, as explained in detail in the next subsections.

5.2.1 Temperature and Humidity Scenario Results

For this scenario, when the dataset was created, the type of room, its characteristics and what is capable of influencing the temperature and humidity of the room were considered. The system knows which are the characteristics of the system since, in the message, the node id is sent, allowing the algorithm to always knows which dataset to use, based on division characteristics.

For this test case, the thresholds were considered, since it is with this values that the system relies to understand if it should send a notification to the user, warning them to do some actions to counteract the threshold being reached. So, two divisions were chosen, with characteristics similar to the training scenario. Figure 5.9 shows the graph of the test case, performed in a room with windows and represents a day and a half, of the two weeks of monitoring.

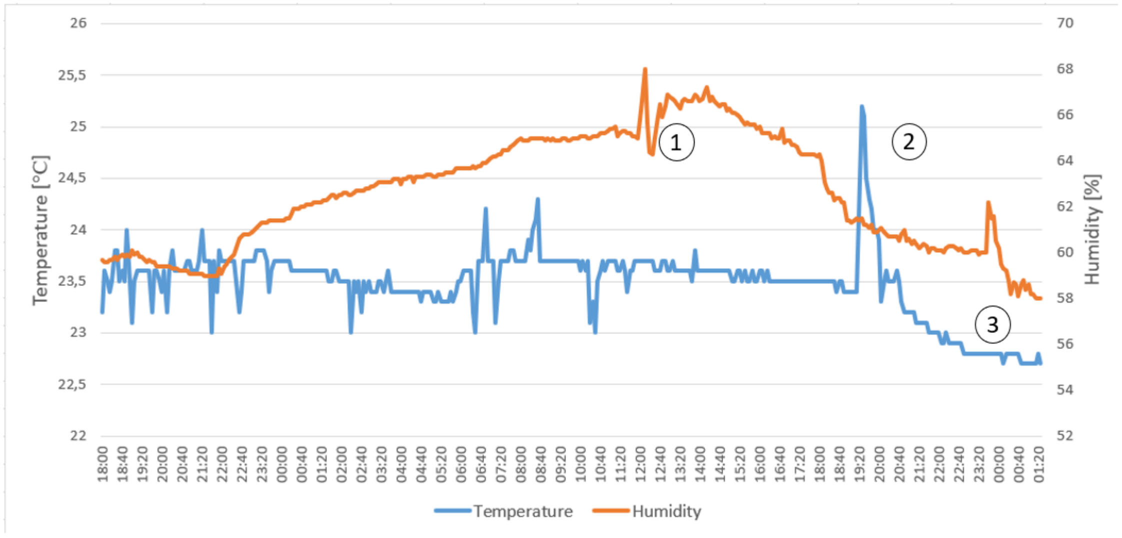


FIGURE 5.9: Results of Temperature and Humidity Temperature (Bedroom)

To understand the effectiveness of the dataset, a minimum threshold was defined, very close to the ambient temperature, and the windows were opened so that the temperature would decrease and the opposite case was also made, closing the windows and turning on the air conditioning, so that the temperature would increase to test the maximum threshold.

Three situations were made, represented in the graph, to understand if the system could perceive the difference in temperatures and send the notification to the user.

At scenario 2 and 3, the air conditioning was turned on, for heat and cold respectively, so that the system would adapt to, and the correct output was identified by the algorithm, with the notification being sent to the user. In Scenario 1 a window was closed, and the system did not detect that it should be open so that the humidity values would decrease, partly since the humidity variation was so low that the system was not able to detect the corresponding output that was occurring.

The other node that was used as a test, was the bathroom, since it is the room that suffers large variations in temperature and humidity, mainly when showers are taken. Figure 5.10 shows the values retrieved from this node for one day and a half of the two weeks of monitoring.

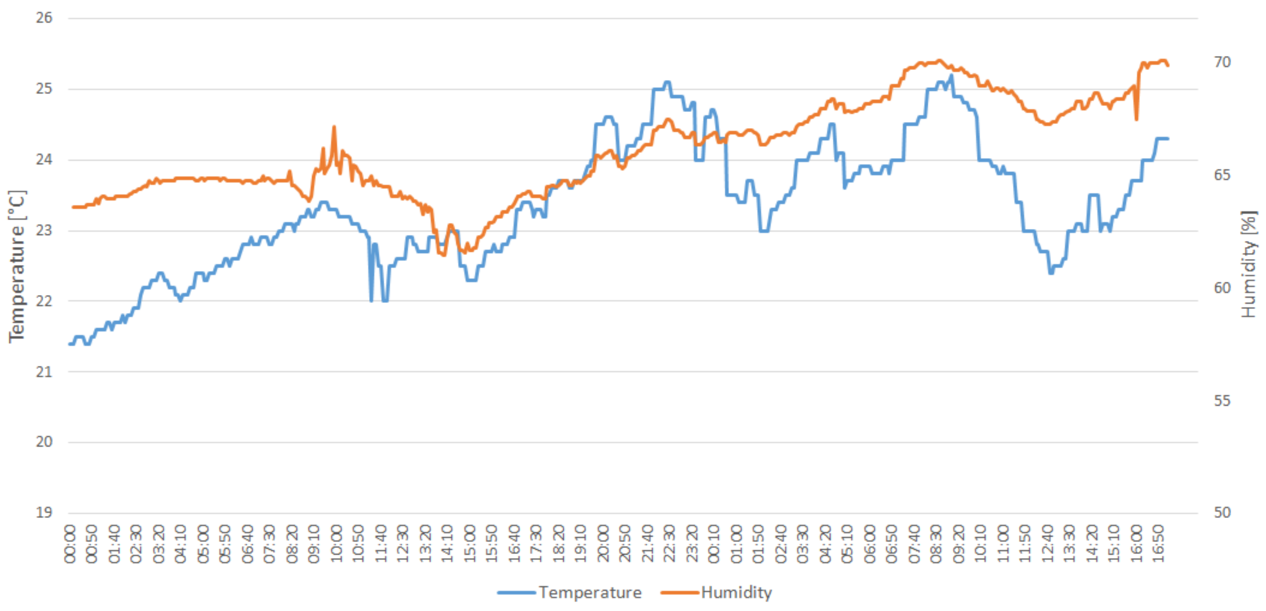


FIGURE 5.10: Results of Temperature and Humidity Temperature (Bathroom)

During this period, about 6 showers were taken and about 5 were recognized. This occurs when the algorithm detects differences in temperature and humidity levels, sending a notification to open the window when the values reached high peaks and another to close when it reached low peaks.

5.2.2 Noise, Motion and Light Scenario Results

Through Figure 5.11 it is possible to visualize the values obtained from the light, movement and sound sensors, values that correspond only to a sample of two days, where the desired test cases were performed. The node chosen was the one implemented in the kitchen, since it is the place where users spend most of their time, being also the place where there is a television, allowing for a better test performance.

The objective of this chart is to demonstrate which situations the system should recognize as abnormal situations, considering the values defined in the training dataset.

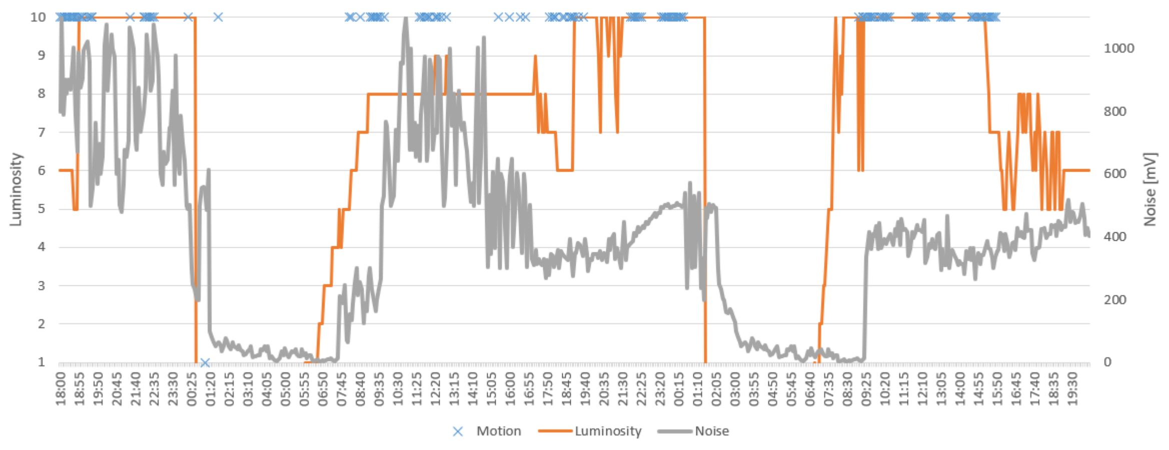


FIGURE 5.11: Results of Noise, Motion and Light Scenario

The test cases were repeated five times each, to see how many times the system recognized the action. The test cases go according to the previously defined outputs:

1. Light ON + Television OFF
2. Light ON + Television ON
3. Light OFF + Television ON

4. Natural Light (Day) + Television ON

5. Ligth ON + Natural Light (Day) + Television ON

In case 1 and 2, the system was able to successfully identify the five times that the light was on and that there was no movement, giving the user notification to turn OFF the light. But he only acknowledged three times, in case 2, that the television was ON.

For case 3 the five situations were recognized as the algorithm was able to realize that without movement, the noise comes from the television and that it is turned on unnecessarily.

The main failure of this system was found in the last two cases, in which the system, not only in these test cases but also during the course of the implementation time, sometimes recognized that the luminosity of natural light was equivalent to a light turned on, so false notifications were sent, since they were not in accordance with what was actually happening. This failure can be justified by the fact that the layout of the house is different in sunlight and even time itself. Despite this failure, the system was able to recognize 7 of the 10 attempts for these two cases.

Another failure detected, was to have situations, in case 5, in which it could not distinguish when the intensity of natural light was high and that a light was on. Despite this failure, he was able to understand that the television was improperly turned on.

In general, and outside the test cases, the system was able to identify the intended actions, except for the mentioned failures. This node was chosen to present the results, because it is a place of passage/central and leisure, and for that reason is much more susceptible to failures. In the other rooms there were no electronic devices, so notifications were only sent when the light was on without movement. In Table 5.1 it is represented the times that they have been set up, for a future work it is necessary to take into account that for a higher success rate, it will be necessary to create a dataset with more outputs, that is, a higher training.

Table 5.1: Cases of Success

Case	**Success/Attempts**	**%**
1	3/5	60
2	3/5	60
3	5/5	100
4 and 5	7/10	70

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, an IoT system was developed to monitor household consumptions, analysing and gathering data in real time on the installation site, using a low-cost sensors network, in order to understand if there is any abnormal consumption or any dangerous situations. The system fulfilled all the objectives set, and proved to be efficient, effective, reliable and user-friendly. All modules were properly tested before reaching the final prototype and its implementation, which was done in a different house from the test, proving that the system is versatile and adaptable.

Through the implementation made, both in laboratory and in real cases, it was possible to verify that the communications were in accordance with the intended objective. Regarding the communication between the aggregation and the node sensor, the communication was made using the LoRa protocol, transmitted through the RFM95W radio modules, capable of having several nodes at long distances and with a very reduced energy consumption, as observed by the power consumptions. Also there were no message losses due to the implementation of ACK mechanism for all messages. Regarding Server Side communications, the

MQTT proved to be a very efficient protocol, due to the practicality of its publisher/subscriber architecture, and advantageous over HTTP POST.

As far as the WSN hardware is concerned, the components chosen were in accordance with the requirements. Faced with the choice of the microcontroller, the ESP32, capable of supporting multiple sensors and being very energy efficient, as it allows for deep sleep mode, was the right choice. Due to all the energy efficiency of the chosen hardware, it was possible to meet the goal of creating a system supplied with only 2xAA batteries, and with a very high lifetime, in the worst case scenario of 49 days. The only problem encountered regarding the energy efficiency of the system, is the fact that the use of batteries in the aggregation may not be feasible since the energy consumption of Wi-Fi is quite high.

Through the Machine Learning test scenarios, it was possible to study which algorithm was the best to implement in the final prototype, which was the case of Random Forest, since it was the only algorithm that did not have a sharp decrease in efficiency indicators, not to mention that it was the best in terms of accuracy. When it was implemented, it was possible to verify that it was in accordance with expectations, although it was not able to predict about 26.47% of the test cases.

In addition to this analysis, an evaluation was also implemented through thresholds, perfectly capable of detecting dangerous situations, and warning the user according to the parameters defined by him.

A mobile application was developed, which allows the user to interact with the system and analyze the results obtained from the sensors, as well as a graph with the latest consumption. Through the work carried out it was possible to give the user the necessary tools so that he can monitor his domestic expenses and raise awareness to reduce his expenses.

It is then possible to verify that the developed system not only meets all the parameters defined as essential for a complete and intelligent system, but it is also possible to validate that it is at the level of the existing systems in the market and in the academic world, presenting a low-cost version with high quality, reliability and efficiency, which may in the near future be adjusted for the introduction of a new solution in the market.

6.2 Future Work

Although all objectives have been met, there are always changes that can be made in order to improve and optimize the system in a future work:

- **Actuator Nodes** - Add a new node type, an actuator node, to the existing WSN with a hardware very similar to the rest. This node is able to communicate with the other nodes in a bidirectional manner and take actions, such as turning a lights ON or OFF, as instructed by the user. Considering the Machine Learning algorithm developed would be an asset to the system.
- **Batteries in the aggregation node** - In order to make the system completely autonomous, find a solution capable of making the aggregation node wireless.
- **Mobile Application** - Make the application more dynamic and interactive with the user, adding new features, such as the possibility of adding nodes through the application and giving instructions to possible nodes actuators.
- **Machine Learning** - Through the mobile application indicate the objects that involve the environment and can influence the algorithm, since it was only created for testing in similar cases.
- **Extension to larger buildings or structures** - Take the sustainability objectives and developed network and apply it on a larger scale which includes structures with more than 20 divisions.

Appendices

Appendix A

Scientific Article Published

2019 IEEE 5th World Forum on Internet of Things (WF-IoT)

Distributed Sensing Solution for Home Efficiency Tracking

Carolina Dionísio(1,2), Gonçalo Simões(1,2), André Glória(1,2), Pedro Sebastião(1,2), Nuno Souto(1,2)
(1) ISCTE – IUL, Av. Forças
(2) Instituto de Telecomunicações, Av. Rovisco Pais, 1, Lisbon, Portugal
garss@iscte-iul.pt, cadoo@iscte-iul.pt, afxga@iscte-iul.pt, pedro.sebastiao@iscte-iul.pt, nuno.souto@iscte-iul.pt

***Abstract*— With the rapid increase of smart devices, keeping track of household consumptions is a service that starts to be automated. This paper presents a proposal for a system based on wireless sensor networks designed for the purpose of monitoring and controlling environment parameters of a smart home. In order to obtain an efficient and inexpensive system, a study was made to select the best hardware and software solutions for this system. This system allows the user, through an Android application, to view all the information collected by the sensors, and consequently act in a way to make his home more sustainable. The main advantage of this system is to take into account that all its components are small, practical and with high efficiency, in addition it allows an easy installation and in order to get involved with the inner environment, another advantage is to allow system interaction with the user.**

Keywords— Internet of Things, Smart Homes, Wireless Sensor Network, ESP32, LoRa, Android

I. Introduction

In the course of the 21st century, the human being tends to change his routine with the evolution of technology, either to make the day-to-day life easier or for entertainment. The Internet of Things (IoT) has the capacity to transform a simple physical device into a smart one, connected to the Internet with communication and computational capacities. Consequently, it is easier to claim this reality is present in our homes in order to make them more efficient and intelligent, the designated *smart homes*. These types of homes are considered one of the most prominent applications of IoT, because the main characteristic of a *smart home* is automation and monitorization, resulting in the reduction of the human effort [1].

The monitorization of a *smart home* can be done according to the parameters the user intends to have, such as: temperature, gas, noise, humidity, brightness and water consumption. In order to perform this data acquisition a wireless sensor network (WSN) is often used, where some sensors are strategically distributed around the house to collect information and, after being transmitted and analyzed, decide actions and alert the user. WSN technology supports the IoT concept, since its main advantage is gathering measurements using devices connected to the Internet, where the results become immediately available to the user wherever he is. Nonetheless, the use of this technology depends on its own efficiency, the reliability and the safety of the data, since they can compromise the user.

The search to accomplish a sustainable way of consuming energetic resources of the planet Earth, is a global concern, due to the tremendous consumption of natural resources by human beings, planet Earth's state is turning critical. Through IoT systems, due to their monitorization and automatization capabilities, it will be possible to reduce the impact human beings have in their consumption, sparing Earth's resources and saving monetary costs to the user.

Nowadays, the necessity to create projects that can monitor homes, to avoid risk situations and also keep track of consumptions, has increased, something we can see in various projects already developed.

The authors in [2] developed a low-cost home automation system, using WSN technology to monitor and control the environment, safety and electrical parameters of an interconnected home using an Android application so the user can control the devices in a smart home. The disadvantage of this system is the Wi-Fi communication protocol that has a high power consumption and low range, making it impractical to use in a large scale environment.

The authors in [3] implemented a WSN for a small area in a building that allows sending data in short distances, with the disadvantage that when packets have more than 10 bytes they get delayed, making packet size something to keeps in mind in order to get the optimal transmission speed at the selected network configuration. Another disadvantage is that the network cannot operate more than five days without additional energy.

In [4], the authors propose a solution where the devices can transmit information up to eight floors without compromising its data flow, in other words, without losing any information packets. This can be useful to extend the monitorization from a simple house to a building.

The main objective of this paper is to design and implement an IoT system to monitor household consumptions, analyzing and gathering data in real time on the installation site, using a low-cost sensor network, in order to understand if there is any abnormal consumption or any dangerous situations. It should consequently lead to natural and material resources savings, as well as reducing monetary costs for the user. There are already some IoT systems using WSN to monitor smart home, although with some limitations that compromises its use in an efficient way. This proposal aims to answer the efficiency problem, using low power hardware, combined with a more optimized software and a long range, low power and reliable communication protocol, in order to improve the WSN lifespan.

II. Proposed System Architecture

In order to achieve the main goal of this work is to develop a distributed solution based in WSN that gathers information according to pre-established parameters. Subsequently, this information will be sent to a server, through an appropriate communication protocol given the range, number of nodes, reliability and cost. This data will be analyzed and make available to the user through an interactive dashboard, also sending warnings via email or notification when dangerous situations occur. To develop this system and achieve the main goal it is necessary to consider the hardware and software solutions, which will be described, in detail, in the following subsections.

A. Hardware

The proposed system, represented in Figure 1, follows the structure of a normal WSN, composed by a few nodes with the ability to collect data and communicate with other nodes. This specific network is only constituted by a Broker and sensor nodes.

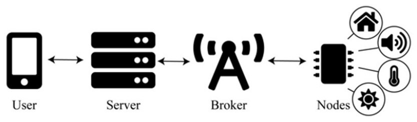


Fig. 1. System Architecture

The main node is the Broker, which is an aggregation node since it does not collect data, it just sends the information received from the other nodes to the server. In this case, the Broker serves as a bridge between the sensor nodes and the server.

The communication between the Broker and the sensor nodes, was chosen based on the best low power and long range wireless technology to serve this system in the most efficient way. The authors in [5] evaluated the main communication protocols associated with IoT and concluded that currently, the best choice was to use LoRa, since it is a bidirectional communication protocol that can provide a low power long range communications, and besides that, can support a large number of devices and increase the battery life, with a reduced device cost.

After receiving the data from the sensors nodes, the Broker transmits the information to the server using Message Queuing Telemetry Transport (MQTT), a protocol used in IoT systems due to its ability to provide routing for low power and low memory devices [6], via a Wi-Fi connection.

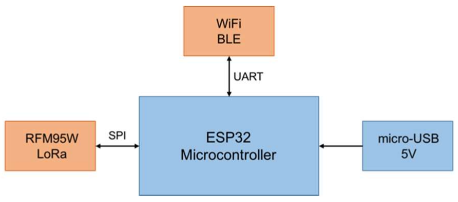


Fig. 2. Broke node block diagram

Analyzing Figure 2 in detail, we can verify that the Broker only consists of two main elements. The first is an ESP32 chip ultra-low-power microcontroller that comes with a built-in Wi-Fi and BLE, being a versatile, low price, high performance and reliable solution for a wide variety of applications and power scenarios [7],[8]. The other element is a RFM95W, a radio module capable of transmitting information using the LoRa protocol, capable of transmitting data between a multi-point network, with individual node addresses, ensuring a range of 2km between nodes [9].

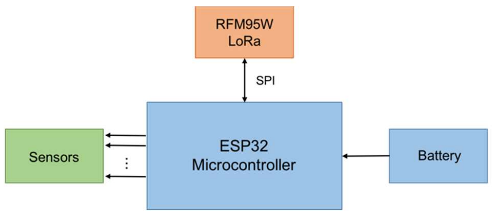


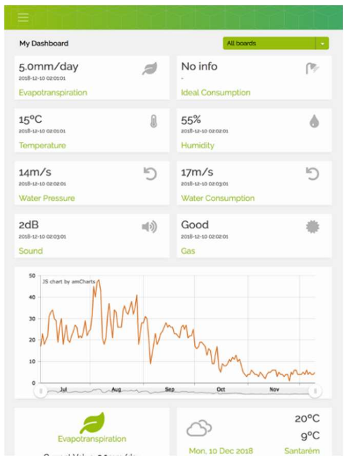
Fig. 3. Sensor node block diagram

As shown in Figure 3, the sensor nodes are composed by an ESP32 microcontroller, a RFM95W radio and an array of sensors. One of the ESP32 features is allowing the node to enter a deep sleep mode when it is not sending information, reducing the normal electrical current consumption form 20 mA to 150μA. This allows the node to be connected just for short periods of time, enabling it to be powered from batteries.

Each node can have up to 8 sensors connected, since the ESP32 includes a high number of Analogic Digital Converters (ADCs), allowing the node to have different sets of sensors depending on room/division necessities, since the parameters from different divisions of a house may differ.

B. Software

In order that the user can check all the information that has been gathered from the sensors, an Android application serves as an information center, a place where all the consumption parameters and data is displayed. This information will be sent periodically, with daily reports as well as graphic charts every month, to ensure the user has all the details on demand.


My Dashboard
All boards

5.0mm/day
2018-12-10 02:01:01
Evapotranspiration

No info
-
Ideal Consumption

15°C
2018-12-10 02:01:01
Temperature

55%
2018-12-10 02:02:01
Humidity

14m/s
2018-12-10 02:02:01
Water Pressure

17m/s
2018-12-10 02:03:01
Water Consumption

2dB
2018-12-10 02:03:01
Sound

Good
2018-12-10 02:02:01
Gas

Evapotranspiration

20°C
9°C
Mon, 10 Dec 2018
Santarém

Fig. 4. Android Application Mockup

With this information, the user can realize if any resource is being spent in an unnecessary way, as show in Figure 4, either daily or monthly, consequently saving money and natural resources. In addition, the software will forward an alert if any dangerous situation occurs, like for example, a gas leak.

The process when a new sensor value arrives at the server is simply a comparison with a threshold value for the type of sensor. This process is done in real time, ensuring that the user is alerted when the problem starts. Figure 5 shows a flowchart of the proposed algorithm.

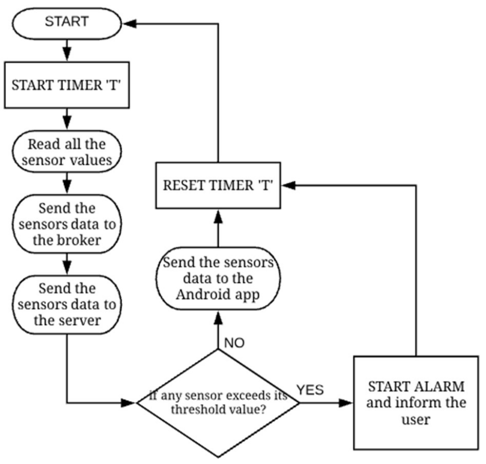

START
START TIMER 'T'
Read all the sensor values
Send the sensors data to the broker
Send the sensors data to the server
If any sensor exceeds its threshold value?
YES
NO
START ALARM and inform the user
Send the sensors data to the Android app
RESET TIMER 'T'

Fig. 5. Proposed data analizes algorithm flowchart

To determine the best threshold for each sensor, an exhaustive analysis of the sensor data in multiple environments needs to be done. In some cases, such as carbon monoxide, the threshold value can be obtained online, since it is a common issue for human beings. Others, such as temperature and humidity, the threshold needs to be adapted to every scenario, so a Machine Learning algorithm can be applied in order to get the individual value for each sensor.

To make sure that all the system performs as needed, it is necessary to guarantee a good communication between the nodes, through a reliable LoRa connection, where no messages are lost. For this, the RadioHead library [10] is used, which allows the individual addressing of each node. Communication can be done in two ways: with broadcast without acknowledgment from the receiver, which is used when the Broker needs to send a message to the sensor node; and the second is to send a message to a specific address with acknowledgment from the receiver, which is used when the sensor nodes send data to the Broker.

To make sure that all types of communication are reliable, when the Broker needs to send a message to a specific node, it sends a broadcast message to all nodes with the desired node ID, if the node is the node with the message ID it sends an ACK to the Broker, if not, the node discards the message. Through this messaging algorithm it is guaranteed that the messages are delivered and that the IoT system is reliable.

III. Proposed System Implementation

Designed and presented the system architecture capable of monitoring household consumptions, analyzing and gathering data in real time, a real case scenario can be proposed. Firstly, is important that laboratory tests be performed so that all parts of the system do not fail in their implementation. Also, it is necessary to define which sensors will be distributed by each node, since each division has different needs, and lastly decide the best position for the broker.

The proposed implementation will be done in a house with the following features for each type of division.

- Kitchen: will be implemented with a gas sensor (MQ2) that has lower conductivity in clean air, because the sensor conductivity is more higher along the gas concentration rising [11], a water pressure and consumption sensor (water flow sensor) [12]. Due to the high hypotheses of failures in this division, this proposal intends to give the user warnings in case of gas or water leaks.
- Bathroom: to control water consumption will be implemented with water pressure and consumption sensor (water flow sensor) that consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, rotor rolls and the speed changes with different rate of flow [12].
- Garage: A gas sensor (MQ7) [13], to obtain the levels of carbon monoxide with a detection range between 10 up to 500ppm.

For all of the above and the remaining divisions sensors for humidity and temperature (DHT22) that measure the temperature between -40º and 80º Celsius [14], luminosity (LDR) that his applications include smoke detection, automatic lighting control [15], an electrical consumption sensor (electricity meter sensor) that can transform AC signals of large current into small amplitude signals. [16] and noise (sound sensor) [17] will be implemented. According to the type of division and its needs, these sensors will be warning the user if conditions such as temperature and humidity values can affect the power consumptions, for example contributing to the correct use of air conditioner in a bedroom or living room. Also combining these measurements with the luminosity sensors can recommend or even directly control the opening or closing of the blinds to help control the temperature.

Since the proposed system is flexible, it is possible to add as many nodes as the divisions of a house, including multiple nodes per division, without jeopardizing the efficiency of the house or network.

With the implementation of the system, it will be possible to visualize the information collected by the parameters in the developed application and later the consumption graphs.

IV. Conclusions and Future Work

In this paper, we described a monitoring system designed using WSNs technology in order to make an ordinary home into a smart one with the objective of keeping track of consumptions to promote sustainability. The proposed system relies on low power WSNs that can communicate either with the online server as well as between nodes spread far apart. Modules that can communicate over LoRa and MQTT like the RFM95W and ESP32 microcontroller were chosen for the implementation due to their characteristics, which allow this system to be efficient, low cost and with a good battery life.

A proposal for the implementation of this system was described with details on how each division can be equipped with a set of sensors and how with that data the user can be warned of potential failures or misusage of consumptions.

The next step is to implement the proposed system and test it in real environments in order to evaluate its efficiency, to ensure that the main goal is achieved, in order to promote sustainability and reduce the household consumptions. Another step is to create actuator nodes, capable of acting in the environment based on the data collected by the network, such as turn on/off lights, blinds or other appliances.

Although this project was designed for smart homes, it can be extended to buildings or even commercial areas. This way it would be possible to promote sustainability in larger environments affecting several users at the same time.

ACKNOWLEDGMENT

This work was supported in part by ISCTE – Instituto Universitário de Lisboa from Portugal under the project ISCTE-IUL-ISTA-BM-2018 and by the FCT - Fundação para a Ciência e Tecnologia and Instituto de Telecomunicações under project UID/EEA/50008/2013.

REFERENCES

[1] C. Paul, A. Ganesh, and C. Sunitha, "An overview of IoT based smart homes," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 43–46, 2018.

[2] N. Vikram, K. S. Harish, M. S. Nihaal, R. Umesh, A. Shetty, and A. Kumar, "A low cost home automation system using wi-fi based wireless sensor network incorporating internet of things (IoT)," *Proc. - 7th IEEE Int. Adv. Comput. Conf. IACC 2017*, vol. 100, pp. 174–178, 2017.

[3] I. Rosadi and S. P. Sakti, "Low-cost wireless sensor network for small area in a building," *Proc. - 2017 Int. Semin. Sensor, Instrumentation, Meas. Metrol. Innov. Adv. Compet. Nation, ISSIMM 2017*, vol. 2017–Janua, pp. 115–118, 2017.

[4] N. Nguyen, L. Nguyen, and T. Nguyen, "On the Design of Gateway Node for Smart Grid Home Network," *2015 Int. Conf. Commun. Manag. Telecommun.*, pp. 57–61, 2015.

[5] A. Gloria, F. Cercas, and N. Souto, "Comparison of communication protocols for low cost Internet of Things devices," *South-East Eur. Des. Autom. Comput. Eng. Comput. Networks Soc. Media Conf. SEEDA-CECNSM 2017*, pp. 1–6, 2017.

[6] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," *Proc. - 2017 Int. Conf. Eng. MIS, ICEMIS 2017*, vol. 2018–Janua, pp. 1–6, 2018.

[7] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," *2017 Internet Technol. Appl. ITA 2017 - Proc. 7th Int. Conf.*, pp. 143–148, 2017.

[8] Espressif Systems, "Esp32 Series - Datasheet," 2018. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Accessed: 07-Dec-2018].

[9] Adafruit, "Adafruit RFM69HCW and RFM9X LoRa Packet Radio Breakouts," 2018. [Online]. Available: https://learn.adafruit.com/adafruit-rfm69hcw-and-rfm96-rfm95-rfm98-lora-packet-padio-breakouts/overview. [Accessed: 05-Dec-2018].

[10] Adafruit, "Adafruit RadioHead," *Adafruit RadioHead*, 2018. [Online]. Available: https://github.com/adafruit/RadioHead. [Accessed: 07-Dec-2018].

[11] W. E. Technology, "MQ2 -Semiconductor Sensor for Combustible Gas." [Online]. Available: https://www.pololu.com/file/0J309/MQ2.pdf. [Accessed: 10-Dec-2018].

[12] W. E. Technology, "Water Flow sensor - Datasheet." [Online]. Available: https://lib.chipdip.ru/583/DOC000583441.pdf. [Accessed: 08-Dec-2018].

[13] W. E. Technology, "Toxic Gas Sensor - MQ7 - Datasheet." [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7 Ver1.3 - Manual.pdf. [Accessed: 10-Dec-2018].

[14] L. Aosong Electronics Co., "Digital-output relative humidity & temperature sensor/module." [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf. [Accessed: 09-Dec-2018].

[15] Sunrom, "Light Dependent Resistor - LDR - Datasheet." [Online]. Available: https://www.sunrom.com/get/443700. [Accessed: 07-Dec-2018].

[16] I. Studio, "Electronic Brick of Electricity Meter Sensor." [Online]. Available: ftp://imall.iteadstudio.com/Electronic_Brick/IM120710018/DS_IM120710018.pdf. [Accessed: 07-Dec-2018].

[17] DFRobot, "Analog Sound Sensor - Datasheet." [Online]. Available: https://github.com/Arduinolibrary/DFRobot_Sound_Sensor/blob/master/HYLD 9767 Specification.pdf?raw=true.

Appendix B

User & Technical Manual

ISCTE IUL

Instituto Universitário de Lisboa

Department of Information Science and Technology

User & Technical Manual

Carolina Aparício Dionísio

Supervisor

Prof. Dr. Pedro Joaquim Amaro Sebastião, Assistant Professor

ISCTE-IUL

Co-Supervisor

Prof. Dr. Nuno Manuel Branco Souto, Assistant Professor

ISCTE-IUL

October 2019

Contents

List of Figures

Chapter 1

Developed Platform

In order to provide the best user experience, an Android application was developed from scratch so that the user can view all the information previously collected and processed. In the next sections it is discussed how the mobile application was developed and how the information was handled on the server.

1.1 Mobile Application

The smartphone application serves as an information center, a place where all the data are displayed. The information is sent to the application on a periodic basis, allowing the user to realize if any resource is being spent in an unnecessary way and send alerts if there is any abnormal situation. A study was made to understand the best operating system to develop the application and, from then on, it was concluded that the best option was Android. For this reason, the application was developed in Java using the official IDE for Android applications, Android Studio.

About the application developed for the system it is discussed below what are the main functions for the user and how it is implemented. In addition, through Figure 1.1 it is possible to visualize the logic of the application.

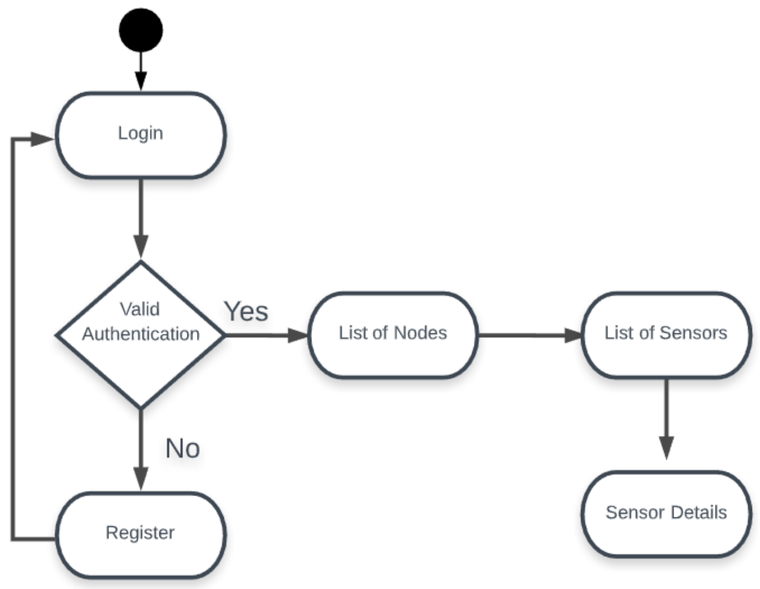
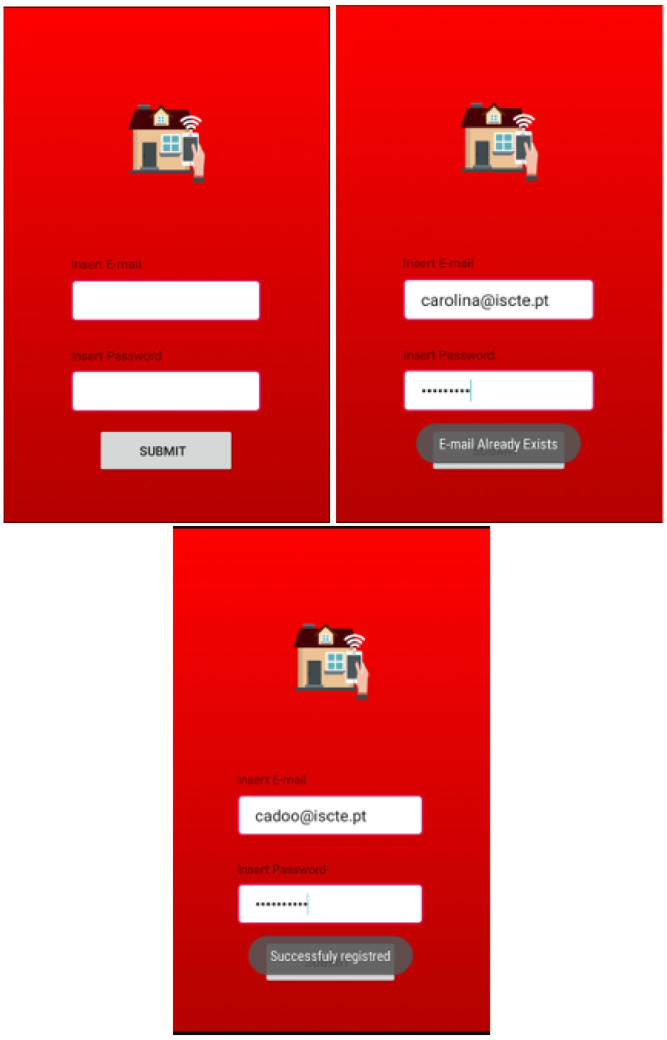


Figure 1.1: Application Flowchart

1.1.1 Authentication

To enter the application, the user needs a valid registration, otherwise the access is denied. The fields necessary to make a registration, i.e., an e-mail and a password. In Figure 1.2 is demonstrated the case where this e-mail is already registered, an alternative e-mail is required to the user and the case where the registration was successful and the interface without any kind of error.



carolina@iscte.pt

E-mail Already Exists

cadoo@iscte.pt

Successfuly registred

FIGURE 1.2: Register Activity

After the user has created a valid record, he can proceed to a very similar page, where he can login. If the user has invalid credentials, a message appears to warn that the login is not correct, shown in Figure 1.3.

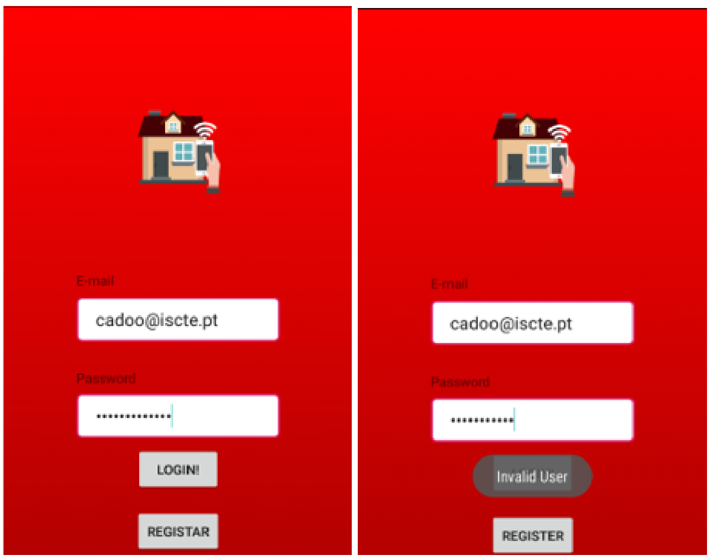


FIGURE 1.3: Login Activity

1.1.2 List of Nodes

When the user has a correct authentication, he is forwarded to the list of nodes available, represented in Figure 1.4, each one representing a division where the node is installed. The node list is information from the database, so it may differ from user to user.

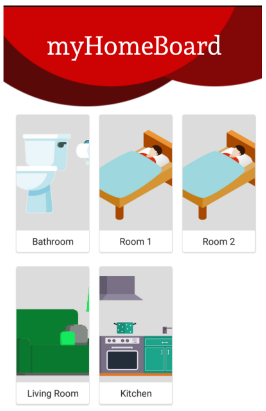


FIGURE 1.4: List of Nodes Activity

1.1.3 List of Sensors

After the user chooses the selected node, he is forwarded to the list of sensors that node has available, as can be seen in Figure 1.5. As in the node list, the information may differ depending on the user.

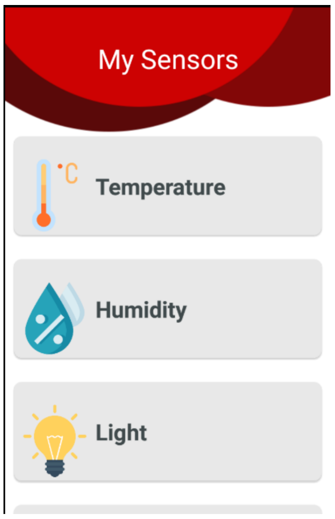
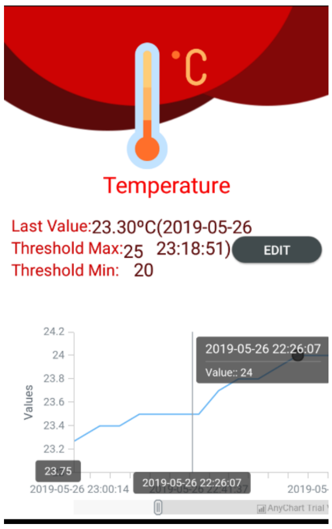


FIGURE 1.5: List of Sensors Activity

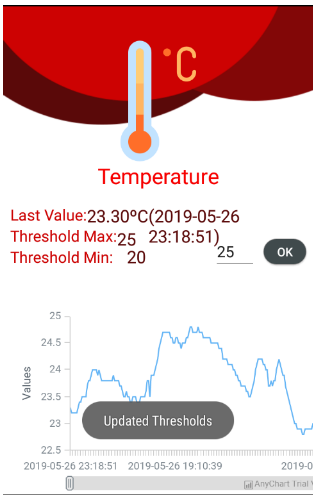
1.1.4 Sensor Details


Temperature

Last Value:23.30ºC(2019-05-26 23:18:51)
Threshold Max:25
Threshold Min: 20

FIGURE 1.6: Sensor Details Activity

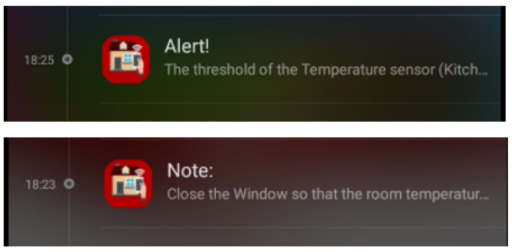
The last screen of the application allows the user to see all the details about a specific sensor. For that, besides the last values retrieved and the corresponding timestamp, a graph with all the values stored is presented. The user can scroll through the figure and see all the values he pretends to. In addition, the user can edit the minimum and maximum thresholds whenever he wants to, this is done per sensor, and not by type of sensor, since two sensors of the same type can have different limits, for example if they are in different divisions. Figure 1.7 shows the message that occurs whenever the user changes the thresholds.



Last Value:23.30ºC(2019-05-26 23:18:51)
Threshold Max:25
Threshold Min: 20
25

FIGURE 1.7: Updated Threshold Message

1.1.5 Notifications



18:25 Alert!
The threshold of the Temperature sensor (Kitch...

18:23 Note:
Close the Window so that the room temperatur...

FIGURE 1.8: Types of Notifications

By observing Figure 1.8, we can observe the two types of notifications that the user can receive:

- Notifications in which the limits chosen by the user himself have been reached, and which may mean danger;

- Notifications with consumption suggestions, as explained above, the user can understand if there is any tip considering the environment, when approaching the threshold values.

1.2 Server

The server is responsible for storing all the information collected, and for making all the necessary analyses so that the system can alert the user of possible events. It is through the server that the mobile application gets the information to show the user in its interface.

1.2.1 Database

Given the high number of information collected not only from sensors, it was necessary to structure this information and implement a database on the server itself, through the MySQL database. The relational diagram of the database is represented in Figure 1.9.

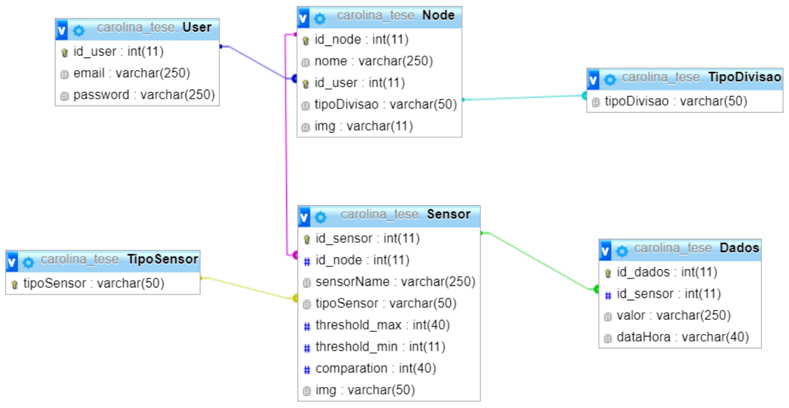


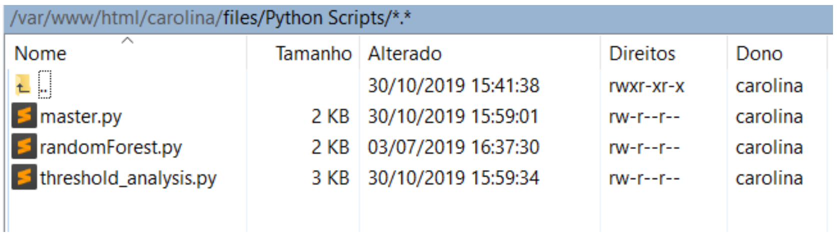
FIGURE 1.9: Database Relational Design

- **User** - This table is responsible for storing user information, such as e-mail and password. It is from these parameters that the verification is made if there is a valid user to authenticate in the mobile application. The primary key is its id;
- **Node** - Responsible for indicating the nodes that each user has. By having the type of division, it is possible to understand which division we are analyzing when the message arrives at the server. The image attribute, allows you to know which image to use in the application, depending on the type of node. Its primary key is idNode;
- **TipoDivisao** - It allows us to know which division is being addressed and which treatment will be done, since each division has its own characteristics;
- **Sensor** - This table assigns an id to each sensor, which is its maximum and minimum threshold. Each sensor knows to which node it belongs, since it is related to the node table via the foreign key idNode. Its primary key is idSensor;
- **TipoSensor** - This table only has the functionality to know which type of sensor;
- **Dados** - It is in this table that all the information coming from the sensors is stored, the message sent to the server is taken into account in this table, since the message indicates the value obtained by the respective sensor (idSensor) and the date and time that was obtained. The primary key of this table is idData.

1.2.2 Scripts

In the server several scripts run in background, being these scripts divided in two parts: the ones that go to get the information from the database to be shown in the application and the scripts that analyze the obtained information.

To analyze the information, Python scripts were created, which are running in the background on the server. Besides analyzing these scripts, they are also responsible for sending the information to the database.



/var/www/html/carolina/files/Python Scripts/*.*

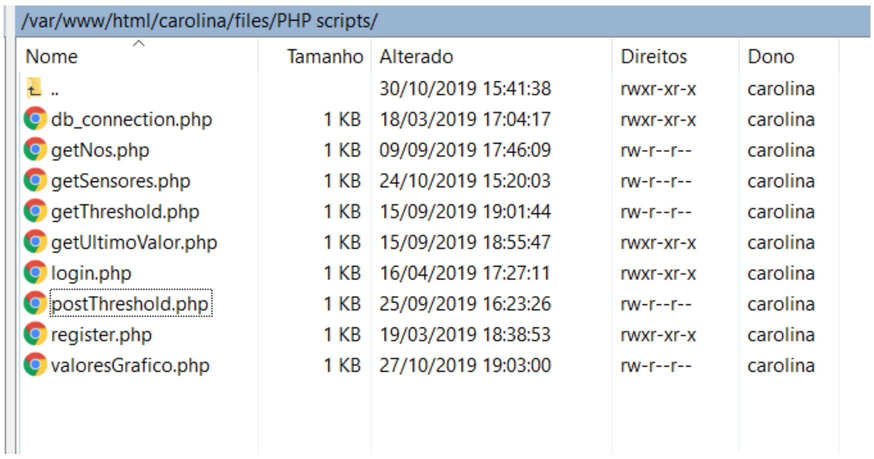
Nome	Tamanho	Alterado	Direitos	Dono
..		30/10/2019 15:41:38	rwxr-xr-x	carolina
master.py	2 KB	30/10/2019 15:59:01	rw-r--r--	carolina
randomForest.py	2 KB	03/07/2019 16:37:30	rw-r--r--	carolina
threshold_analysis.py	3 KB	30/10/2019 15:59:34	rw-r--r--	carolina

FIGURE 1.10: Python Files

Through Figure 1.10 it is possible to view the python files used:

- **master.py** - Responsible for making the connection to the database for the data to be sent to it. This is also where the MQTT library, Paho, is implemented, responsible for receiving messages from the aggregation node.
- **thresholdAnalysis.py** - It is in this file that the thresholds analysis is defined, and if there is any situation in which they are exceeded, they send through MQTT a notification to the mobile application.
- **randomForest.py** -This is where the script of the machine learning algorithm is defined, and can predict the results taking into account the division under analysis.

For the user to be able to visualize the sensor values and for the application itself to be able to show those values, it is necessary to query the database and make connection with it, for that, PHP scripts were developed, that allowed to search that information, through a request to the server, since the application can not access the database itself for security reasons.



/var/www/html/carolina/files/PHP scripts/

Nome	Tamanho	Alterado	Direitos	Dono
..		30/10/2019 15:41:38	rwxr-xr-x	carolina
db_connection.php	1 KB	18/03/2019 17:04:17	rwxr-xr-x	carolina
getNos.php	1 KB	09/09/2019 17:46:09	rw-r--r--	carolina
getSensores.php	1 KB	24/10/2019 15:20:03	rw-r--r--	carolina
getThreshold.php	1 KB	15/09/2019 19:01:44	rw-r--r--	carolina
getUltimoValor.php	1 KB	15/09/2019 18:55:47	rwxr-xr-x	carolina
login.php	1 KB	16/04/2019 17:27:11	rwxr-xr-x	carolina
postThreshold.php	1 KB	25/09/2019 16:23:26	rw-r--r--	carolina
register.php	1 KB	19/03/2019 18:38:53	rwxr-xr-x	carolina
valoresGrafico.php	1 KB	27/10/2019 19:03:00	rw-r--r--	carolina

FIGURE 1.11: Python Files

In Figure 1.11 are represented the files used in PHP whose functionality is:

- **dbConnection.php** - File responsible for making the connection to the database to be included in other files;
- **getNos.php** - Obtain the available nodes that the user has to be seen in the node list of the application;
- **getSensores.php** - Obtain the available sensors for each node, to be demonstrated in the list of sensors in the application;
- **getThreshold.php** - Responsible for going to the database to know the thresholds of each specific sensor;
- **getUltimoValor.php** - Fetch the last value of each sensor, to be available in the sensor details;
- **postThreshold** - When the user changes the threshold values, this file is responsible for sending that change to the database;
- **valoresGrafico.php** - Responsible for retrieving all the values of a certain sensor, so that in the application it is possible to make the graph to show to the user;

- **login.php** - responsible to verify that the user's credentials are correct;
- **register.php** -is from here that a new user is registered and the information is sent to the database.

Chapter 2

Prototype

This chapter describes the necessary procedures to use the developed prototype.

2.1 Hardware Components

Regarding the prototype developed, the user needs to have knowledge of the Hardware in order to implement it correctly. For this, he needs to know how to distinguish the two types of existing nodes: the aggregation node and the sensor nodes.

The aggregation node is only responsible for sending the messages to the server and cannot be charged by batteries, so it needs to be powered by a charger with a micro usb input. This node consists of an ESP32 microcontroller, an RFM95W transceiver, which are inside a small box, as shown in Figure 2.1. The user can implement the aggregation node by simply connecting it to the plug in a division of his choice.

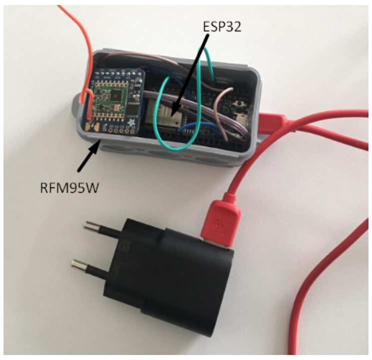


FIGURE 2.1: Aggregation Node

In Figure 2.2 is represented the electrical circuit of the aggregation node:

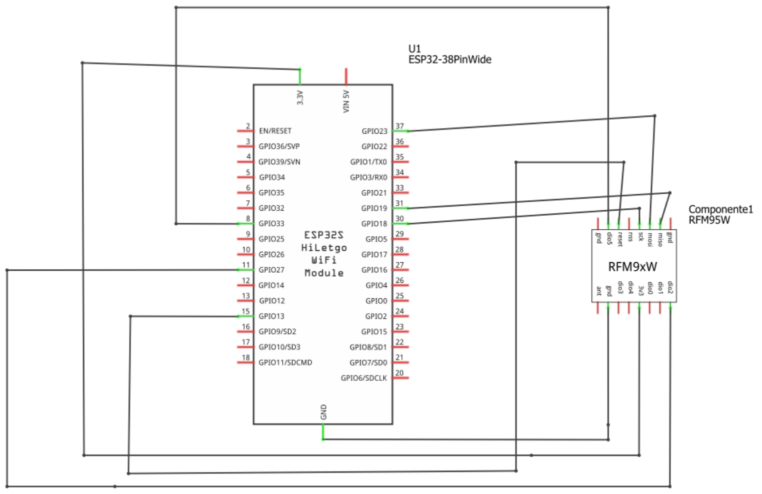


FIGURE 2.2: Aggregation Node Electric Circuit

The sensor nodes are responsible for collecting all the information and can have a maximum of 8 sensors per node, powered with 2 AA batteries. Through Figure 2.3, we can see how the ESP32 board is connected to the batteries.

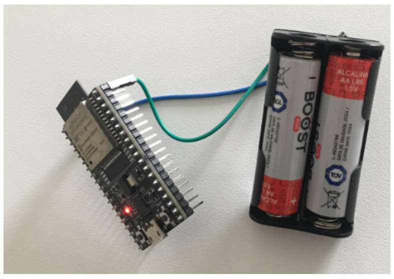


FIGURE 2.3: ESP32 with 2 AA batteries

Each node has an ESP32 microcontroller, an RFM95W transceiver, 2 AA battery, a set of sensors connected to the board. In Figure 2.4 is represented the electrical circuit for the sensor node.

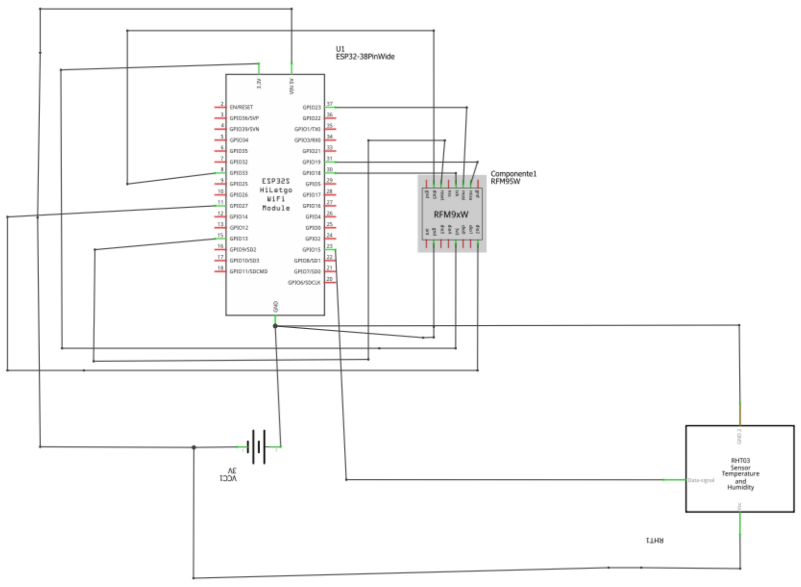

U1
ESP32-38PinWide
ESP32S HiLetgo WiFi Module
Componente1
RFM95W
RFM9xW
VCC1
3V
RHT03 Sensor Temperature and Humidity
RHT1

FIGURE 2.4: Sensor Node Electric Circuit

2.2 Arduino IDE

All the Hardware was configured using the Arduino IDE, which is available on the brand's website (www.arduino.cc). The code developed was in C/C++ and the libraries used were as follows:

- **Paho** (https://github.com/256dpi/arduino-mqtt)
- **RadioHead** (https://github.com/dragino/RadioHead)
- **DHT22**(https://github.com/adafruit/AdafruitSensor)

The following files have been created with this IDE:

- **AggregationNode.ino** - The purpose of which is to configure the broker. It is in this file that the MQTT is configured, responsible for sending the information to the server, and where the RadioHead library responsible for connecting the RFM95W transceiver to the ESP32 board is implemented;
- **SensorNode.ino**- It is this file that is responsible for having the sensors configured and able to collect the outputs. The RadioHead library is also implemented in this file so you can send the information to Aggregation node. From this script, it is possible to configure the message that will be sent with the node information.

2.3 Sensor Normal Values

Taking into account the values obtained by the sensors it is important that the user is able to understand which are the values considered normal taking into account the sensor in question, since it may not be aware of them. For the Temperature and Humidity sensors it will be considering that the user knows which values he feels more comfortable and that for the motion sensor and light there are no values considered normal.

- **Sound**:
  - Very Low : 10 a 20 dB
  - Low: 30-40 dB
  - Normal: 50-70 dB
  - High: 80-120 dB
  - Very High: from 120 dB
- **Gas**: The values given take into account the threshold values for each type of gas:
  - Carbon Dioxid: 0,5%
  - Metane: 5%
  - Butane: 1,5%
  - Propane: 1,8%
  - Hydrogen: 4%

Appendix C

Machine Learning Results

Kitchen/Living Room Node (P_c = 60.7%) (S_y^2 =0.205)

Table C.1: Machine Learning Algorithm Results- Kitchen/ Living Room Node

Algorithm	**Accuracy (P_c)[%]**	**MSE**	**HI [%]**	R^2
Decision Trees (C5.0)	96.09	0.039	90.1	80.9
Random Forest	98.97	0.009	97.3	95.61
SVM	83.68	0.162	58.47	41.46
Neural Net	92.99	0.069	82.16	66.30

Bedroom (without windows) Node (P_c = 58.8%) (S_y^2 =0.230)

Table C.2: Machine Learning Algorithm Results - Bedroom (without windows) Node

Algorithm	**Accuracy (P_c)[%]**	**MSE**	**HI [%]**	R^2
Decision Trees (C5.0)	90.96	0.09	78.05	57.14
Random Forest	91.64	0.082	79.70	60.95
SVM	67.12	0.327	20.19	55.71
Neural Net	71.23	0.283	30.17	34.76

Bedroom Node (P_c = 60.12%) (S_y^2 =0.210)

Table C.3: Machine Learning Algorithm Results - Bedroom Node

Algorithm	**Accuracy (P_c)[%]**	**MSE**	**HI [%]**	R^2
Decision Trees (C5.0)	94.51	0.055	86.23	76.08
Random Forest	94.91	0.05	87.23	78.26
SVM	76.44	0.168	40.80	19.12
Neural Net	82.60	0.173	56.37	24.7

Bathroom Node (P_c = 60.6%) (S_y^2 =0.205)

Table C.4: Machine Learning Algorithm Results - Bathroom Node

Algorithm	**Accuracy (P_c)[%]**	**MSE**	**HI [%]**	R^2
Decision Trees (C5.0)	95.52	0.045	88.62	78.04
Random Forest	97.59	0.023	93.88	88.78
SVM	83.68	0.163	58.57	41.46
Neural Net	91.38	0.079	78.12	61.46

Bibliography

[1] C. Paul, A. Ganesh, and C. Sunitha, "An overview of IoT based smart homes," *Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, no. Icisc, pp. 43–46, 2018.

[2] K. Routh and T. Pal, "A survey on technological, business and societal aspects of Internet of Things by Q3, 2017," *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–4, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8519898/

[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[5] V. Govindraj, M. Sathiyanarayanan, and B. Abubakar, "Customary homes to smart homes using Internet of Things (IoT) and mobile application," *Proceedings of the 2017 International Conference On Smart Technology for Smart Nation, SmartTechCon 2017*, pp. 1059–1063, 2018.

[6] T. Malche and P. Maheshwary, "Internet of Things (IoT) for building Smart Home System," *International conference on I-SMAC*, pp. 65–70, 2017.

[7] S. S. I. Samuel, “A review of connectivity challenges in IoT-smart home,” *2016 3rd MEC International Conference on Big Data and Smart City, ICBDSC 2016*, pp. 364–367, 2016.

[8] J. Bangali and A. Shaligram, “Energy efficient Smart home based on Wireless Sensor Network using LabVIEW,” *American Journal of Engineering Research (AJER)*, no. 12, pp. 409–413, 2013.

[9] E. A. D. G. Reina, S. L. Toral, F. Barrero, N. Bessis, “The role of ad hoc networks in the internet of things,” *in Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence*, pp. 89–113, 2013.

[10] M. Brzozowski, K. Piotrowski, and P. Langendoerfer, “A cross-layer approach for data replication and gathering in decentralized long-living wireless sensor networks,” *Proceedings - 2009 International Symposium on Autonomous Decentralized Systems, ISADS 2009*, pp. 49–54, 2009.

[11] A. Gloria, F. Cercas, and N. Souto, “Comparison of communication protocols for low cost Internet of Things devices,” *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2017*, pp. 1–6, 2017.

[12] E. Ferro and F. Potortì, “Bluetooth and Wi-Fi wireless protocols: A survey and a comparison,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[13] J. S. Lee, Y. W. Su, and C. C. Shen, “A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” *IECON Proceedings (Industrial Electronics Conference)*, no. January 2014, pp. 46–51, 2007.

[14] A. Hafeez, N. H. Kandil, B. Al-Omar, T. Landolsi, and A. R. Al-Ali, “Smart home area networks protocols within the smart grid context,” *Journal of Communications*, vol. 9, no. 9, pp. 665–671, 2014.

[15] K. Cho, W. Park, M. Hong, G. Park, W. Cho, J. Jihoon, and K. Han, "Analysis of latency performance of bluetooth low energy (BLE) networks," *Sensors (Switzerland)*, vol. 15, no. 1, pp. 59–78, 2015.

[16] J. Marais and R. Malekian, "LoRa and LoRaWAN Testbeds: a Review," *IEEE Africon 2017 Proceedings*, vol. 24, no. 4, pp. 1496–1501, 2017.

[17] L. Angrisani, P. Arpaia, F. Bonavolonta, M. Conti, and A. Liccardo, "LoRa protocol performance assessment in critical noise conditions," *RTSI 2017 - IEEE 3rd International Forum on Research and Technologies for Society and Industry, Conference Proceedings*, pp. 0–4, 2017.

[18] A. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan, "Evaluation of Lora and LoRaWAN for Wireless Sensor Networks," *IEEE*, vol. 0, pp. 45–46, 2016.

[19] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of Lora: Long range & low power networks for the internet of things," *Sensors (Switzerland)*, vol. 16, no. 9, pp. 1–18, 2016.

[20] Bharati Wukkadada ; Kirti Wankhede ; Ramith Nambiar ; Amala Nair, "Comparison with HTTP and MQTT In Internet of Things (IoT) - IEEE Conference Publication," *2018 International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*, no. Icirca, pp. 249–253, 2018. [Online]. Available: https://ieeexplore-ieee-org.e.bibl.liu.se/document/8597401

[21] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT," *ICCEREC 2016 - International Conference on Control, Electronics, Renewable Energy, and Communications 2016, Conference Proceedings*, pp. 1–6, 2017.

[22] D. Ghosh, A. Agrawal, N. Prakash, and P. Goyal, "Smart Saline Level Monitoring System Using ESP32 And MQTT-S," *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services*

(Healthcom), pp. 1–5, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8531172/

[23] N. Tantitharanukul, K. Osathanunkul, K. Hantrakul, P. Pramokchon, and P. Khoenkaw, "MQTT-Topics Management System for sharing of Open Data," *2nd Joint International Conference on Digital Arts, Media and Technology 2017: Digital Economy for Sustainable Growth, ICDAMT 2017*, pp. 62–65, 2017.

[24] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017*, vol. 2018-Janua, pp. 1–6, 2018.

[25] R. K. Kodali and K. S. Mahesh, "A low cost implementation of MQTT using ESP8266," *Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics, IC3I 2016*, pp. 404–408, 2016.

[26] A. Nayyar and E. V. Puri, "A review of Arduino Board's, Lilypad's & Arduino Shields," *2016 International Conference on Computing for Sustainable Global Development*, pp. 1485–1492, 2016.

[27] A. A. Galadima, "Arduino as a learning tool," *Proceedings of the 11th International Conference on Electronics, Computer and Computation, ICECCO 2014*, pp. 1–4, 2014.

[28] P. Srivastava, M. Bajaj, and A. S. Rana, "IOT based controlling of hybrid energy system using ESP8266," *2018 IEEMA Engineer Infinite Conference, eTechNxT 2018*, pp. 1–5, 2018.

[29] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," *2017 Internet Technologies and Applications, ITA 2017 - Proceedings of the 7th International Conference*, pp. 143–148, 2017.

[30] Espressif Systems, "Esp32 Series - Datasheet," 2018. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32{_}datasheet{_}en.pdf

[31] Adafruit, "Adafruit RadioHead," 2018. [Online]. Available: https://github.com/adafruit/RadioHead

[32] ——, "Adafruit RFM69HCW and RFM9X LoRa Packet Radio Breakouts," 2018. [Online]. Available: https://learn.adafruit.com/adafruit-rfm69hcw-and-rfm96-rfm95-rfm98-lora-packet-padio-breakouts/overview

[33] Y. Tan and G. J. Zhang, "The application of machine learning algorithm in underwriting process," *2005 International Conference on Machine Learning and Cybernetics, ICMLC 2005*, no. August, pp. 3523–3527, 2005.

[34] M. Mamdouh, M. A. Elrukhsi, and A. Khattab, "Securing the Internet of Things and Wireless Sensor Networks via Machine Learning: A Survey," *2018 International Conference on Computer and Applications, ICCA 2018*, no. Section II, pp. 215–218, 2018.

[35] R. Saravanan and P. Sujatha, "A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification," *Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018*, no. Iciccs, pp. 945–949, 2019.

[36] Q. Tang, X. Ge, and Y. C. Liu, "Performance analysis of two different SVM-based field-oriented control schemes for eight-switch three-phase inverter-fed induction motor drives," *2016 IEEE 8th International Power Electronics and Motion Control Conference, IPEMC-ECCE Asia 2016*, no. c, pp. 3374–3378, 2016.

[37] M. Ross, C. A. Graves, J. W. Campbell, and J. H. Kim, "Using support vector machines to classify student attentiveness for the development of personalized learning systems," *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, vol. 1, pp. 325–328, 2013.

[38] L. Li, W. Chou, W. Zhou, and M. Luo, “Design Patterns and Extensibility of REST API for Networking Applications,” *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 154–167, 2016.

[39] Android, “Android.” [Online]. Available: https://www.android.com/

[40] Apple, “Apple.” [Online]. Available: https://www.apple.com/pt/

[41] N. Vikram, K. S. Harish, M. S. Nihaal, R. Umesh, A. Shetty, and A. Kumar, “A low cost home automation system using wi-fi based wireless sensor network incorporating internet of things (IoT),” *Proceedings - 7th IEEE International Advanced Computing Conference, IACC 2017*, vol. 100, pp. 174–178, 2017.

[42] I. Rosadi and S. P. Sakti, “Low-cost wireless sensor network for small area in a building,” *Proceedings - 2017 International Seminar on Sensor, Instrumentation, Measurement and Metrology: Innovation for the Advancement and Competitiveness of the Nation, ISSIMM 2017*, vol. 2017-Janua, pp. 115–118, 2017.

[43] N. Nguyen, L. Nguyen, and T. Nguyen, “On the Design of Gateway Node for Smart Grid Home Network,” *2015 International Conference on Communications, Management and Telecommunications*, pp. 57–61, 2015.

[44] W. E. Technology, “MQ2 -Semiconductor Sensor for Combustible Gas.” [Online]. Available: https://www.pololu.com/file/0J309/MQ2.pdf

[45] ——, “Toxic Gas Sensor - MQ7 - Datasheet.” [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7Ver1.3-Manual.pdf

[46] L. Aosong Electronics Co., “Digital-output relative humidity & temperature sensor/module.” [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

[47] Sunrom, “Light Dependent Resistor - LDR - Datasheet.” [Online]. Available: https://www.sunrom.com/get/443700

[48] DFRobot, "Analog Sound Sensor - Datasheet." [Online]. Available: https://github.com/Arduinolibrary/DFRobot{_}Sound{_}Sensor/blob/master/HYLD9767Specification.pdf?raw=true

[49] C. Dionisio, G. Simoes, A. Gloria, P. Sebastiao, and N. Souto, "Distributed Sensing Solution for Home Efficiency Tracking," no. 1, pp. 825–828, 2019.

[50] C. Zhang and X. Yin, "Design and implementation of single-service multi-function Webservice," *2011 International Conference on Computer Science and Service System, CSSS 2011 - Proceedings*, pp. 3912–3915, 2011.

[51] H. M. Server, "MQTT Websocket Client." [Online]. Available: http://www.hivemq.com/demos/websocket-client/