DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

JOÃO ANTÓNIO FERREIRA MOURÃO CARTAXO
Licenciado em Ciências de Engenharia Eletrotécnica

Development of an IoT platform to monitor storage conditions and packaging optimization in industry environment

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa
Setembro, 2021

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA

Development of an IoT platform to monitor storage conditions and packaging optimization in industry environment

JOÃO ANTÓNIO FERREIRA MOURÃO CARTAXO
Licenciado em engenharia eletrotécnica e de computadores

Orientador: Professora Filipa Alexandra Moreira Ferrada, Professora Auxiliar, Universidade NOVA de Lisboa

Coorientadores: Engenheiro Hugo Pinto Nunes, sócio-fundador da AnalíticaES, Lda

Júri:

Presidente: Doutor Rui Santos Tavares, Professor Auxiliar, Universidade NOVA de Lisboa

Arguentes: Doutora Ana Inês Oliveira, Professora Auxiliar, Universidade NOVA de Lisboa

Vogais: Doutora Filipa Alexandra Moreira Ferrada, Professora Auxiliar, Universidade NOVA de Lisboa

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa
setembro, 2021

Development of an IoT platform to monitor storage conditions and packaging optimization in industry environment

To Madalena, Pedro, Filipa, António,
my parents, and Mariana

Acknowledgements

A special thanks to professor Filipa Ferrada and the engineer Hugo Nunes for the opportunity and all the monitoring of the project, it would not have been possible without these supervisors. From suggestions and opinions to the correction and testing of the project, as well as the availability of material. It was a great course with a great evolution, and therefore a big thank you for believing in my ability to develop the project.

Thank you to Duarte from DATAzero and Cooperativa Agrícola de Beja e Brinches for the opportunity to work in their installations and helping to develop and evaluate a real case scenario. This opportunity led us to better results and analyse technologies in the real case scenario permitting us to develop a stronger technology.

I am grateful for all my professors and the staff from the electrical and computer engineering department of the Faculdade de Ciências e Tecnologia specially robotics group for showing and teaching modern technologies that made me choose this area for the development of my master thesis.

To my friends, thank for being there when it matters and giving me the strength, I needed to finish the course.

Finally, I would like to thank my family, especially my brothers and sisters that encouraged me to continue my school course and were always there during the pauses. It would not be possible without their presence.

"If I have seen further than others,
it is by standing upon the shoulders of giants."

(Sir Isaac Newton)

ABSTRACT

Food production and storage is of major importance in the modern world. Markets and industry must guarantee supply to millions of people. Storing these quantities of food require a lot of management skills and large spaces with good conditions to ensure food quality. One particular area is the olive oil industry which requires large spaces and large tanks prepared with dozens of thousands of liters of capacity.

This storage can sometimes create problems in the management caused by misinformation provoked by human error in checking storage the status. One such Institution where this problem has appeared is Cooperativa Agrícola de Beja e Brinches (CABB), Portugal, which inspired us to study and develop the product of this thesis. In CABB it was verified that sometimes, due to its storage conditions, the registered quantities of olive oil stored in the tanks would not correspond to the real state, sometimes varying as much as 20 000 liters.

In this work, to facilitate olive oil management in CABB, we suggest the use of Internet of Things (IoT) technologies. This can rectify the verification of quantities of olive oil stored in tanks and indirectly improve the quality of information needed for better management of olive oil.

Tests revealed that the technology developed during this project might improve the functioning of CABB, making it better to read values associated to olive oil levels inside tanks.

Key words: Olive oil storage monitoring, liquid level reading, Internet of Things, Cyber Physical Systems

Resumo

A produção e armazenamento de comida é de maior importância no mundo moderno, os mercados e a indústria têm de garantir produtos a milhões de pessoas. Armazenar estas quantidades de comida requer muita qualidade de gestão e largos espaços com boas condições para garantir a qualidade da comida. Uma indústria onde se verifica esta necessidade é na produção de azeite que necessita de grandes depósitos com dezenas de milhares de litros de capacidade.

Este armazenamento pode por vezes criar problemas de gestão causados pela má informação provocada por erro humano. Uma das instituições onde se verificou o aparecimento deste problema foi na Cooperativa Agrícola de Beja e Brinches (CABB), Portugal, que nos inspirou para desenvolver o produto estudado nesta tese. Na CABB foi verificado que por vezes, dadas as condições de armazenamento, o registo de quantidades de azeite dentro dos tanques não corresponde ao estado real, por vezes com variação de valores acima dos 20000 litros.

Neste documento, para facilitar a gestão de azeite na CABB sugerimos o uso de serviços de Internet of Things (IoT). O uso desta tecnologia pode retificar verificação de quantidades de azeite armazenado em tanques e indiretamente melhorar a qualidade de informação necessária para a gestão de azeite.

Os testes revelam que a tecnologia desenvolvida durante o projeto poderá melhorar o funcionamento da CABB corrigindo a leitura de valores associados ao nível de azeite armazenado nos tanques.

Palavras-chave: Monitorização do armazenamento de azeite, leitura de nível de líquidos, Internet of Things, Cyber Physical Systems

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Acronymous

Symbol	Meaning
AI	-Artificial intelligence
AMR	-Automatic meter reading
BI	-Business intelligence
CABB	-Cooperativa Agrícola de Beja e Brinches
CPS	-Cyber physical systems
DJEG	-Direção Geral de Energia e Geologia
DP	-Differential pressure
GPRS	-General packet radio service
GPS	-Global positioning system
IaaS	-Infrastructure as a Service
IoT	-Internet of things
IPAC	-Instituto Português de Acreditação
LPWAN	-Low power wide area network
ML	-Machine learning
PaaS	-Platform as a service
RTD	-Resistance temperature detectors
SaaS	-Software as a service
SCL	-Serial clock time
SDA	-Serial data line

1 INTRODUCTION

1.1 Motivation & problem domain

Food production and storage is of major importance in the modern world, markets and industry must guarantee supply to millions of people. Food products come mostly from farms, but before consumption, this food must be processed and monitored in quality and state for it to be then dis-tributed and sold. With improvement in mind, many industries have been applying electronic devices (sensors, actuators, microprocessors, among others) to their production lifecycle, which can give valuable information during the whole process in real time. These devices can be installed in inaccessible places for humans and work for long time windows while being continuously exchanging information to a base. Also, technology advances have improved the quality of many electronic devices as well as their power consumption, so the collected data is generally reliable.

These circuits can be connected to networks, being able to transfer information for long distances. As the data is digitalized it becomes easier to monitor and generate statistics and predict future states of the production line. In addition, it is possible to have the factory spread around the world providing quality of life to workers. Another advantage using this technology is the insight it provides to managers and decision makers outside factory boundaries. The subjects that study these appliances are known by Internet of Things (IoT) and Cyber Physical Systems (CPS).

IoT and CPS can benefit industry by making it more efficient and reliable. Electronic devices remove human error from the production cycle while working continuously for long periods. This technology can change the mechanisms to control a production, machinery no longer needs the constant presence of a supervisor to start and prepare materials, most of these can be prepared from another place besides the physical place of the machine.

Currently, in Portugal many industries are being pushed to this digital transformation, as is the case of the olive oil production industry. One of the key drivers for such transformation is the need to monitor, in real-time, the level of olive oil inside the storage tanks in order to enhance the management of the storage and packaging processes in the olive oil production. In this context, Faculdade de Ciências e Tecnologia teamed up with AnaliticaES Lda[1] to develop and install an IoT system on an olive oil warehouse, the Cooperativa Agrícola de

[1] AnaliticaES Lda is a Portuguese inspecting entity of electrical installations acknowledged by IPAC (Instituto Português de Acreditação) and DJEG (Direção Geral de Energia e Geologia) with several years of experience on the certification of electric installations

Beja e Brinches[2]. This system should be able to monitor the amount of olive oil stored inside the tanks supported by an IoT platform, deliver information quality, improving the management of the product and reducing costs. As such to create an efficient system, a study is performed to identify its needs and the technologies used on similar projects.

1.2 Research context

According to PORDATA in ("PORDATA - Lagares e Produção de Azeite - Continente," n.d.), between 2011 and 2018, Portugal produced around 7.655.152 hectoliters of olive oil making around 956.894 hectoliters per year. The product is processed in large quantities and stored in large opaque tanks, however one of the main identified difficulties is to measure accurately the level of olive oil in such tanks.

With some tests performed in Cooperativa Agrícola de Beja e Brinches it was possible to check that the levels and quantities of the product stored in tanks are uncertain compared to the expected values. These values, i.e., the amount of olive oil that is stored in the tanks, influence the production and packaging rates, and the whole production lifecycle management, so it is extremely important that they are real. To surpass this problem, this work proposes an intelligent system designed to monitor and control the amount of olive oil that is, in fact, stored in the warehouse, while keeping this information refreshed and available on a network. The process can also benefit of an alarm system to detect unexpected situations/possible failures of the procedures, warning managers about these situations.

In this research work, it is intended to improve the information access and quality of the olive oil storage. Olive oil is a very important product in the Mediterranean market, and it is produced in very large quantities. As olive oil is stored in large opaque tanks, it is difficult to have access to viable values of the quantities of the good while stored in the tanks. In this context, this work is suited by the following question:

"How to organize and control information about olive oil stored in a warehouse tank to support storage monitoring and provide an efficient production management?"

A possible solution to this problem is to design a viable information acquisition system to provide the needed values, connect it to a small network and store it in a data base. Then it would be possible to develop a program to analyze the information and share it in a wider network accessible outside the warehouse. With the access to the data base, it would be possible to perform many types of analyses that could then predict future states helping management. Then it could be installed some actuators to actuate in the flow of stored olive oil from one tank to another without the presence of a worker and design a software to control it over the internet or another wide network.

[2] Cooperativa Agrícola de Beja e brinches is a Portuguese group of olive oil producers where it is produced stored and packaged olive oil to be sold

1.3 Research methodology

This project tries to find a solution for the current olive oil management problem caused by miss information on olive oil quantities in CABB. To accomplish this objective, we will be analyzing each challenge and study technologies to overcome the difficulties. Starting in the hardware choices to read volume inside tanks, to the communication technologies, and then interfaces to display information to the operators. To simplify development, we separate each work phase, so it is possible to focus on simpler tasks, each of these phases will be studied and developed following these steps:

1. Problem
2. Technologies
3. Development/Tests

Following, it is shown the structure of the document and described each section subject.

1.4 Work contributions and document structure

This work intends to improve the technology currently used in olive oil storage, by installing an efficient and intelligent system to monitor the amount of olive oil stored inside the tanks on a warehouse. Such system aims to prevent olive oil waste and contribute to a better management of the storage and packaging process, minimizing costs. In order to get this goal, an analysis and research of the technologies already used, and their advantages/throwbacks is conducted and will contribute with the:

- Design of a system to monitor olive oil tanks.
- Development of a small microcontroller/sensor network.
- Development of an IoT platform.
- Design of an application to visualize/monitor and control the enclosed tanks.

The remainder of this document is divided in 5 parts:

1. Section 2 – A review on the State of the Art: In this section it is analyzed and studied the technologies available on the literature and market and their applications.
2. Section 3 – Solution proposal - In this section it will be analyzed the system requirements and use cases necessary for the development.
3. Section 4 – Solution implementation - Choices of the Architecture and technologies to develop the system.
4. Section 5 – Tests and results - this section is reserved for test results and small observations where it is described the developed product.
5. Section 6 – Conclusions and future work - In this last section it is reviewed the whole project and it is suggested some future implementations and improvements.

2 Background

In this section it is studied some background information as well as many technologies that can be used to solve our problem. First it is studied olive oil to understand its needs in production and storage. Then there will be studied some technologies used to measure:

- Liquid's volume
- Liquid's temperature
- Olive oil acidity

Following, it is analyzed some CPS/IoT technologies, and some application examples that have some similarities with this project. Finally, it is descried CABB's current state and explained some of the problems this project is trying to solve.

2.1 Olive oil production

Olive oil is extracted from the olive fruit, and it is very important for food industry, mainly on Mediterranean. The olive oil industry must follow some steps, guaranteeing the quality of the product. The following subsections will give an overview of the olive culture and the production of its oil.

2.1.1 Story of Olive and Olive oil

The olive (Olea europaea L.) had its origins in Asia minor (Turkey, Syria), it is known that the oil that comes from its fruit has been used for more than 5000 years, since the Mesopotamian battles. The olive culture grew on the 14th century BC when it entered in the Greek market. Creating a big impact in economy. It was used as: food, medicine, perfume, illumination fuel, among others. The olive tree and its fruit were the face of some coins and were presented on old tumulus, symbolizing immortality. Today, olive oil, is still used extremely in culinary and health products. In Mediterranean countries, where it is mostly produced, is still very important for the economy and its food culture.

The production of this good is fully mechanical in order to maintain its chemical properties. Any chemical reactions could change its composition and deteriorate its beneficial characteristics. When the olive fruit gets to the industry, it is subject to various steps to extract its oil and get a product of quality at the end of the production. Based on (Souilem et al. 2017) and being present in some production lines, the highlighted steps on olive oil production are:

- Leaf removal - it is common to find leaves, rocks, and dirt when the fruit gets to the production line. These impurities raise the olive oil acid, making it less valuable in the market, also they might affect the product flavor so to keep quality, the fruit passes on a vibrating or rotating grid.

- Olive washing - before taking the fruit to the grinding, it must be washed to remove dirt particles and any trace of pesticides.

- Crushing - after cleaned and washed, the olive fruit will be crushed to form a paste, it will have all the fruits components: skin, lump, and the core. All these parts have important nutrients.

- Squeezing - the paste produced on the last step will then be slowly pressed to release its oil, it is a slow procedure for around an hour. After this, we have the first glance at the product.

- Storage – The product is stored in big tanks to let it rest and let the sediments go to the bottom. This phase can be different from producer to producer, in small companies, sometimes the product is bottled right after the squeezing and given to the people that brought the fruit. Saving a part to the industry that will sell it later.

The technology on olive fruit processing might vary considerably depending on the factory age and investment. The processes can also vary with the quantity of product to be processed. In (Scheffer-Dutra, Núñez-Reyes, and Bordons 2002), it is shown one application of the process as illustrated in figure 1:

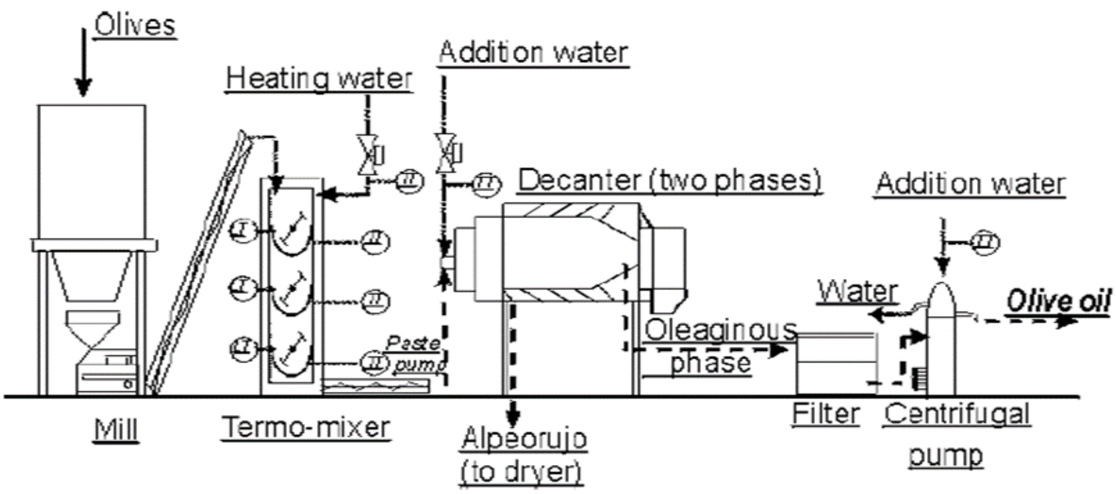


Figure 1– An example of Olive oil extraction method, (Scheffer-Dutra, Núñez-Reyes, and Bordons 2002)

This process can change from factory to factory, and the way they work can be different. In some cases, after getting the product, it can be added some conserving chemicals, to ensure that the product remains in eating condition for longer.

2.1.2 Olive oil storage

Olive oil is produced in large quantities being the Mediterranean Europe one of the most important markets, (Souilem et al. 2017). The producers need to store hundreds of thousands of liters of oil. Usually, the bigger mills are equipped with many tanks to keep the oil in good conditions over the year. Tanks need to be resistant to corrosion and be strong enough to bear the weight. These should also be opaque, (Caponio et al. 2005) demonstrates the problems caused by light exposure. The most common material used to build these deposits is inox due to its resistance to corrosion, the walls must be thick enough to bear high pressures. As security, they are reinforced with inox rings parallel to the base.

In Cooperativa Agrícola de Beja e Brinches, an olive oil manufacturer and warehouse, the storage tanks of olive oil are made of inox with around 10m in height with an area of almost $14m^2$. Due to the big dimensions of the deposits and their opacity, it is not simple to verify the volume of oil inside them, which create a problem managing the product. In this case it is favorable to design an electronic system capable of getting a trustworthy value on the volume inside the deposits. This would improve the managing by giving the real quantities present in the tanks.

2.2 Liquid state readings

In this next section, the characteristics of the liquids and the technologies to read them are explored. When working with the food industry, it is necessary to keep track of some properties of the goods, which can vary from temperature to acidity. Nowadays, there are already a good number of sensors that can communicate with integrated circuits, so it is possible to connect and adapt them to IoT/CPS.

2.2.1 Height and Volume

Volume is one important measure while preparing any food product, it is needed to control additives and control the production rate. Also, when in critical levels, might damage the equipment. In most cases we cannot take the volume right away but can find the height of liquid in the storage tanks. After getting the height value it is possible (knowing the tank shape) to find the volume of liquid inside. In this section there will be analyzed some methods for height reading. In (Nikolov, Nikolova, and Nikolov 2008) we can find a resume of the used methods.

- **Direct readings**

This kind of method works by checking the height on a marked object. It is very practical for small and fixed distance. The user can directly read the height looking at a scale on the object. This type of method is not very common when using informatized systems. For this method of measurement, the user can use Rulers which are used for very small measures, usually less than 1 meter. Another common instrument is the Metric Tape, a metric tape is a very useful object when needing to read fixed distances that do not need a very certain reading. There are tapes for a lot of ranges, like for example, 2m, 5m, 10m, 20m.

- **Measuring tank**

A measuring tank is very useful for reading small quantities of a liquid. This type of object is often used in its small equivalent for cooking with measuring cups. The disadvantage is it needs a full visible tank. It also can get bad readings if the liquid is oily or "dirty".

- **Mechanical systems**

This type of method works using mechanical properties of an object. Usually, these systems use a pulley connected to a float. The float follows the liquids height and the pulley rotate to a position directly proportional to the float's height.

- **Floats**

Floats are relatively simple to explain, we chose an object with the right density to float on the liquid and not on its vapor, then we measure its height, which is the harder part. Early float systems used mechanical components such as cables, tapes, pulleys, and gears to indicate the level. These may introduce some errors like float displacement, or it might get stuck in some place.

Latter float systems use electric or magnetic circuits to indicate the level. In (Kumar and Mandal 2016) they used a float to implement a capacitive circuit, using the liquid as a dielectric. The capacitance of the circuit would vary with the position of the float. The test results were satisfactory with the standard derivation around 0.04 (in comparison with the "best fitted value" for the liquids level).

- **Magnetic Level Gauges**

These kind of system uses a strong magnet attached to a float which will affect another magnet inside a gauge as shown in figure 2 found in ("Magnetic Liquid Level Gauges Working Principle - Level Measurement - Instrumentation Forum" n.d.). These magnets interact via magnetic forces on a magnetized bar. Then this bar will light up some led at the level of the magnet. On this type of system, the tank walls need to be non-magnetic, or it might influence the system readings.

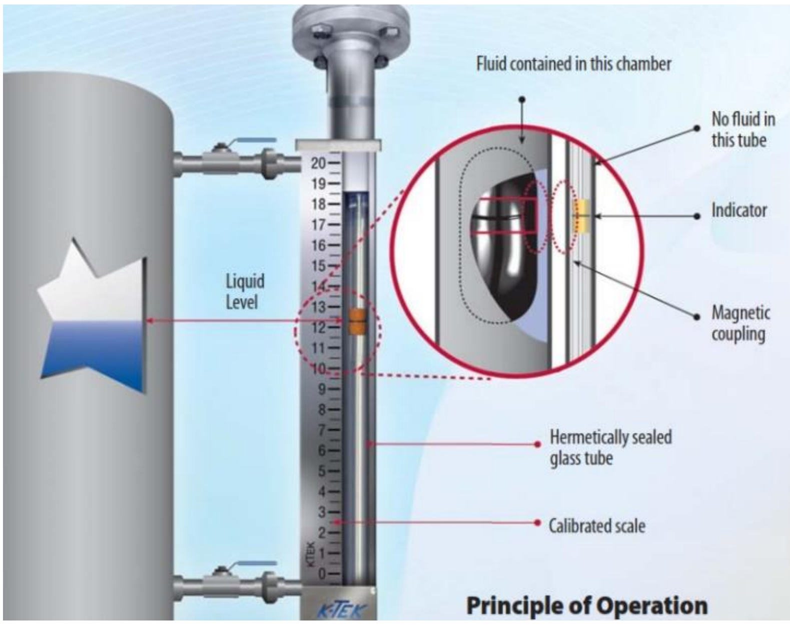


Figure 2 - Magnetic level gauges, ("Magnetic Liquid Level Gauges Working Principle - Level Measurement - Instrumentation Forum" n.d.)

➢ **Electronic Pressure readings**

With this kind of electronic device, it is possible to obtain the pressure in a point of measurement. With pressure values it is usually very direct to calculate volume if we have a regular form tank, following the next formula:

$$Pressure = \frac{Weight}{Area} = \frac{g \times m}{A} = \frac{\rho \times g \times V}{A} = \rho \times g \times h$$

Where g represents gravity acceleration, h the height of the liquid, ρ is density, A symbolizes area and V for volume. We can use these readers on some different ways, for example we can put the sensor on the bottom of the tank with a pressure reference in one side of the sensor, or we can instead use a bubbler type system with the point of the tube on the place we want to measure pressure.

- **Bubbler-type/differential pressure (DP)**

These machines measure the pressure needed to send bubbles out of a tube. A tube is attached to the tank at bottom with a hole to flow liquid and air. Then a machine on the top of the tank will put pressure on the tube till it sends bubbles out, then we know the pressure in the tube and can calculate the height of liquid outside the tube. If we know the density of the liquid, we can check the height and then calculate the volume inside the tank.

- **Submersible pressure transmitter**

Another application of the DP sensors is the submersible ones. On this method we can put the sensor on the bottom of the tank and give the sensor a pressure reference with an air tube for example. In (Nikolov, Nikolova, and Nikolov 2008) it is proposed a method to calculate liquid height and its density, using 2 DP sensors, in that experience, it is limited to 10m height but it can be expanded changing the hardware.

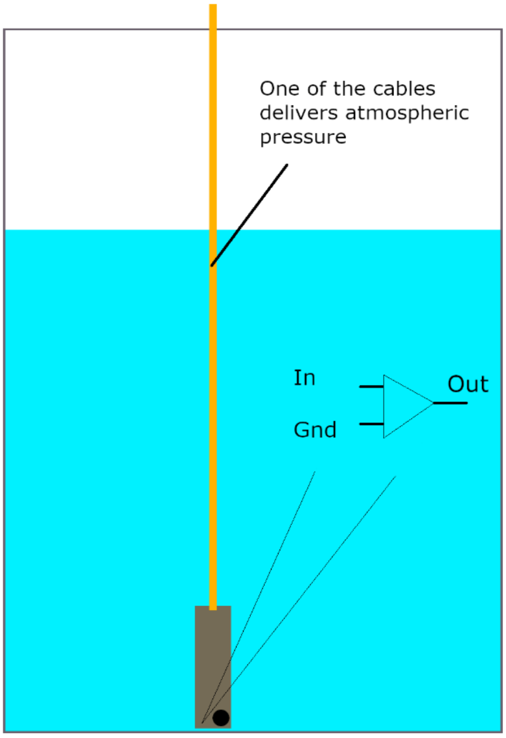


Figure 3 - Submersible pressure transmitters (differential pressure sensors)

- **Weight readers**

In many cases it is very simple to put a weight reader under a tank, the weight (when the density is constant) is directly proportional to the volume, making it simple to get the value of volume and height.

$$Mass = Volume \times Density$$

- **Load Cells**

There are many types of load cell, Pneumatic, Hydraulic, Capacitance, or Strain Gauge. These devices work by measuring the force applied on them. If we have the liquids density, we can easily calculate the volume inside the tank. It is then possible to get the level of product inside the tank if we know its shape.

The strain Gauge cell can check little distortions on a metal bar (the bar will deform with the pressure applied on it). These distortions vary with the pressure inside the box, so after getting these values it is possible to calculate the weight, and once again, we can then calculate the volume.

In (Wang et al. 2018) it was installed a system to measure water level using a gauge. In this test, the gauge was installed on the top of the tank with a PVC tube working as a float, as the level of water rises, the air inside the tube will be pushed upwards and it will apply a force on the gauge. This force is directly proportional to the level of water.

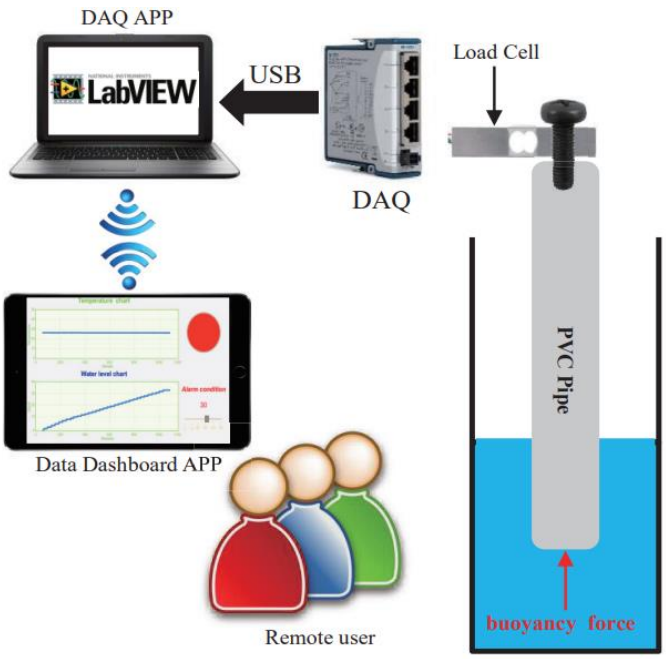


Figure 4 - Using a load cell to calculate water level, (Wang et al. 2018)

- **Using electric constants**

Usually, each material has different electric constant values, we can use this property to get the height values, as the height of the material will change the electric properties of the system. One example is the capacitance value of a fluid. Making a capacitor using the air (or another gas/liquid) and the liquid it is meant to measure; the capacitance value will vary with the height of the liquid.

- **Capacitance Transmitters**

On this type of technology, we simulate a capacitor with two dielectric materials, the air, and the liquid inside the tank. Fluids generally have dielectric constants significantly different from that of air, this technology requires a change in capacitance that varies with the liquid level. Then we can make a direct comparison with the capacity and the liquid level. This method was used in (Chetpattananondh et al. 2014) for water level monitoring, the results were very satisfactory with less than 1cm error during the tests. A rectangular pulse is sent to the capacitor, which will attenuate its fast changes in voltage, the changes are proportional to the circuit's capacitance. Then, a computerized method will analyze the capacitance equation and deduct the height of liquid and air in between the capacitor plates. Figure 5 represents the system used in (Chetpattananondh et al. 2014).

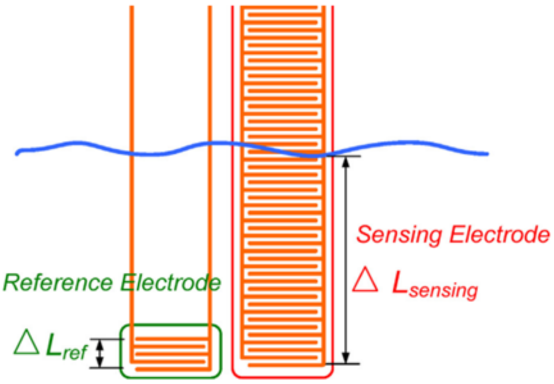


Figure 5 -Using a capacitive circuit to measure liquid height, (Chetpattananondh et al. 2014)

- **Send/Receive interval signal**

On this category the concept is similar among these technologies. An output signal is sent to the liquid triggering a timer, the output signal will then reflect on the product and come back, when de receiver detects the signal, the timer is stopped. The time of travel will be determined by the liquid height. Once again, with the height it is simple to get the volume inside the tank. A very similar system is used on boats to detect the height of water under the ship, represented in figure 6, found on ("Application of SONAR in Ships | Download Scientific Diagram" n.d.).

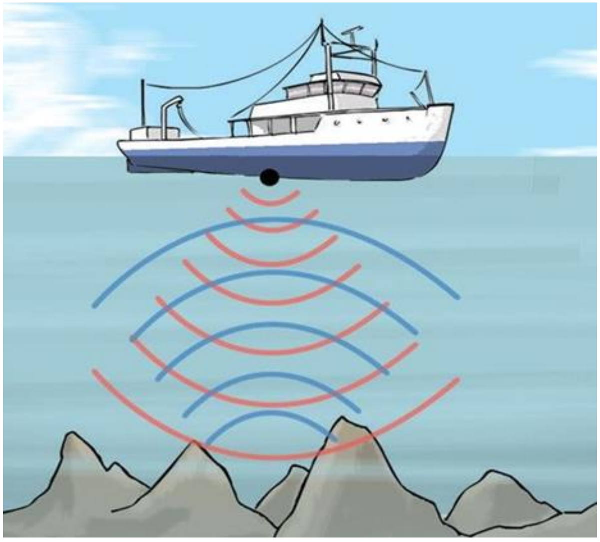


Figure 6 – Sonar working method, ("Application of SONAR in Ships | Download Scientific Diagram" n.d.)

- **Magneto strictive Level Transmitters**

An electrical signal is sent with a coil to a float on the liquid, this float is equipped with strong magnets which will make a distortion on the signal. The time needed to detect this distortion determines the height.

- **Ultrasonic Level Transmitters**

Distance is measured by the time it takes for an ultrasonic to be sent and come back. It is known that the speed of sound is around 340m/s, so we can directly calculate the distance it travelled before coming back. The disadvantage is that it only works in air or nitrogen and around 15ºC.

- **Laser Level Transmitters**

Works similarly to Ultrasonic Level Transmitters but instead of sound, uses a light signal. The disadvantage is it only works with opaque products.

- **Radar Level Transmitters**

This is another system that works following the same principal, but now the signal is sent via a horn or an antenna. The disadvantage on this method is it only works for liquids with a high dielectric constant (r). The reason is that the amount of reflected energy at microwave (radar) frequencies depends on it. If r is low, most of the radar's energy enters or passes through.

➢ **Optical**

Optical technology for level reading is mostly used when it is risky to implement electric circuits near the liquid. In some cases, like for metering the level of petroleum or another inflammable liquid, it might be dangerous to install any kind of circuit that can create a spark. In (Antunes et al. 2015) it was used optical fiber to transmit a light signal throw a cable with "grooves". As the liquid rises, these grooves will be fulfilled with the content on the tank and the signal power would change. With the study of this power change, it is possible to estimate the liquids height.

➢ **Summary**

The trend today is to replace mechanical and pressure-based measurement tools with systems that measure the distance to the fluid surface by a timing measurement. Magneto strictive, ultrasonic, guided-wave radar, and laser transmitters are among the most versatile technologies available. These can be very useful and relatively low priced, putting them among the best choices for most applications. In this next table we will find a resume on the advantages/limitations of each technology.

Table 1 - Liquid level reading technologies

Technology	Advantages	Limitations
Floats	Adaptable to wide variations in fluid densities	High installation cost; Floats might displace or get stuck inside the tank
Bubbler-type	Simplicity of design; Low initial purchase cost	Needs a pressurizer to get values; Is constantly putting gas that might affect the contents of the tank
Submersible pressure transmitter	Inexpensive, Wide range measurements; Can be isolated safely from the process; Measurements can be digitally networked for remote computer access	Need power and active electronic; Depends on the density and the temperature of the liquid
Load Cells	Does not need to be in contact with the product	Needs to be installed under the tank, which is unviable

Magnetic Level Gauges	Reluctant to errors	Complex installation Needs a structure outside the tank
Capacitance Transmitters	Very precise	Hard to install with large distances
Magneto strictive Level Transmitters	Very precise	Contact with the product
Ultrasonic Level Transmitters	Easy to install; Non-contact measurement; No moving parts; Can measure corrosive and volatile liquids	Need high power; Low accuracy; Not operate on vacuum or high-pressure applications; Expensive; Temperature correction is needed
Laser Level Transmitters	No Contact; Used in vessels with numerous obstructions; High level of accuracy (better than 1 mm)	Expensive; Fails if dust, smoke, etc. Sensitive to dirt
Radar Level Transmitters	Non-contact measurement; The transmission time is unaffected by ambient temperature and pressure fluctuations (can be used in closed tanks, where the liquid is turbulent and in the presence of obstructions and steam condensate)	Internal piping and multiple reflections can cause erroneous readings; Transmitter setup can be tedious and changes in the process environment can be problematic; The appropriate licenses must be obtained

2.2.2 Temperature

Temperature metering is very important on the food industry, some aliments will become poisonous when not conserved under some temperature. Also, with temperature control it might be possible to detect malfunctions on the temperature control systems.

Temperature readings have been of most importance throughout the years, so the methods to gather these values have been vastly developed. Usually, it is simple to find a cheap and easy to use chip/sensor to measure temperature values, if we have a relatively small temperature spectrum. In (Holloway, Nwaoha, and Onyewuenyi 2012) we find the table represented in figure 9, in which are shown some of the technologies for temperature reading and its advantages/limitations.

- **Contact**

  - **Thermocouple**

Based on (“Sensors Modules Thermocouple | Sensors Modules” n.d.), in some metals it is possible to create voltage between two pieces of the same metal at different temperatures, a thermocouple converts that voltage to a temperature gap between 2 points. This technology is usually used in temperature gaps of -270 to 3000 ºC.

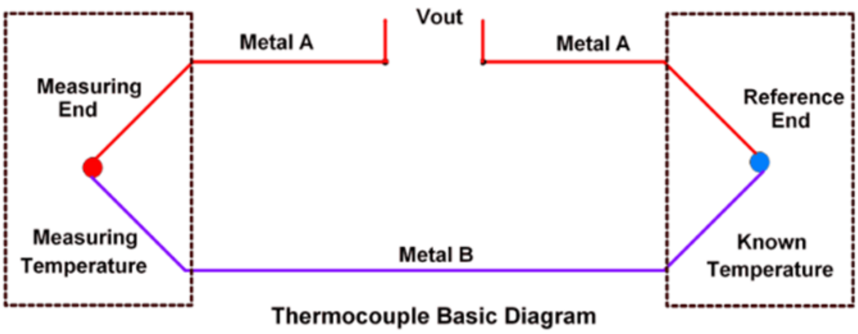


Figure 7 – Thermocouple device diagram, ("Sensors Modules Thermocouple | Sensors Modules" n.d.)

In (Yashiro, Ogawa, and Sasahara 2013) it was applied to measure the temperature in a cutting point of two carbon fibers. It is a good technology for rough works that do not need to much precision, it is relatively simple and cheap.

- **Resistance Temperature Detectors (RTD)**

RTD's are an electronic method of measuring temperature. The resistance of an electric component varies with its temperature, based on this theory, it is possible to track its temperature in function of its resistance. It is mostly used on projects with a wide range of temperatures (0-400 ºC). Based on (Maher et al., n.d.) we can verify that the resistance varies with a quadratic formula with the temperature as the variable. In this test, the function of resistance is almost linear using platinum with a 0.05% of rhodium.

➢ **Contactless**

- **Infrared readers**

This type of device read the infrared radiation emitted from an object, this radiation is proportional to the body's temperature. In ("What Is Infrared Radiation? IBT Informs! - Infrabiotech" n.d.) we can check the relation between infrared and thermal radiation as illustrated in figure 8. This technology can be used in industrial applications when it is important to keep track of machines and products temperature, it is practical to get values as it does not need direct contact between the worker and the machine (typically it can read a wide range of values like -20-500ºC). A portable infrared thermometer is often used as a common garage device, it can measure an engine temperature or even the temperature of meat while grilling it. The range of error in this kind of devices can vary, usually it is better as the cost rises but, in some cases, the extra certainty in its value might not be indispensable. In (Mahan et al. 2010) they compared two different type of infrared sensors, the idea was to verify if it was advantageous to choose the cheaper device. At the end of the study, as the values were close and the application did not need to much precision, it could be advantageous to choose the cheaper device.

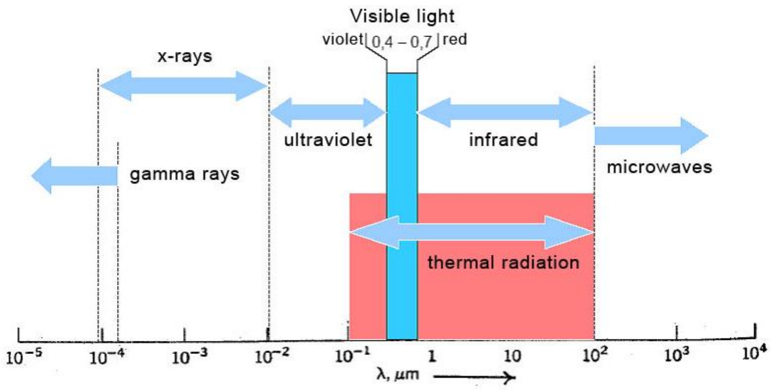


Figure 8 – Infrared and thermal radiation wavelengths, (Mahan et al. 2010)

Table 2 - Technology to read temperatures, (Holloway, Nwaoha, and Onyewuenyi 2012)

Sensor	Advantages	Limitations
Thermocouple	Self-powered Simple Rugged Inexpensive Wide variety Wide range	Nonlinear Low voltage Reference required Least stable Least sensitive
RTD	Most stable and accurate Area sensing More linear than thermocouple Most repeatable Contamination resistant	Expensive Current source required Slow response time Low sensitivity to small temperature change Self-heating
Infrared	No contact required Very fast response time Good stability over time High repeatability No oxidation/corrosion to affect sensor	High initial cost More complex support electronics Spot size restricts application Emissivity variations affect readings Accuracy affected by dust, smoke, and background radiation
Bimetallic	Simple, robust, and inexpensive Has good accuracy Can measure temperature in the range −40 to 550°C Can withstand 50% overage temperature measurement	Not recommended for measurement of temperature above 550°C The metals undergo permanent warp distortion Use limited to local mounting
Liquid-filled glass	More economical, versatile, widely used Rugged in construction, low maintenance Can be used for remote indication Stable in operation System provides enough power to drive the control mechanism	Compensation necessary for ambient (surrounding) temperature changes and long capillary tube For accuracy the pressure bulb should be large In case of error the entire system has to be replaced

2.2.3 Acidity

Acidity is one important factor when choosing an olive oil, it varies with the quality of the fruit and the production methods. Also, acidity on olive oil is sensible to light and temperature, so it is important to keep the product stored in a controlled room to keep its original properties and the products quality. Although not the most important factor when choosing the product for eating purposes (its acidity does not change the flavor of the oil), it is still a value that many people have in mind when buying it. However, olive oil is not an aqueous solution, its acidity comes from the free fatty acids, so it cannot be determined with a simple pH sensor.

Nowadays, the used methods to measure acidity in olive oil are mostly done on laboratory. This product is not an aqueous solution, so it cannot be analyzed with a simple pH sensor, it is usually determined using titration. There are some studies testing electrochemical methods to take this value, making the test cheaper and simpler. In (Baldo et al. 2019), a method using "platinum disk microelectrode, with the PLS regression method, EVOO acidity can be accurately quantified even at low FFA levels down to 0.2%". In (Grossi et al. 2014) it is suggested to estimate olive oil acidity by analyzing electrical impedance spectroscopy. The project pretended to develop a relatively cheap device and tested it in 39 different olive oil samples.

2.3 Cyber Physical Systems/Internet of Things

Based on (Liu et al. 2017), Cyber physical systems (CPS) are multidisciplinary systems to conduct feedback control on widely distributed embedded computing systems by the combination of computation, communication and control technologies. In (Broy, Cengarle, and Geisberger 2012), is stated that: "The vision of Cyber-Physical System (CPS) is that of open, ubiquitous systems of coordinated computing and physical elements which interactively adapt to their context, are capable of learning, dynamically and automatically reconfigure themselves and cooperate with other CPS (resulting in a compound CPS), possess an adequate man-machine interface, and fulfil stringent safety, security and private data protection regulations."

This technology links cyber systems with a physical state of a process in a loop. The constant feedback permits to keep track of the process state, helping the decision making of the machine and registering statistics of the procedure, if necessary. This interaction is implemented with the use of sensors and interfaces/connections between machines/systems. CPS supports the communication between distinct devices which can be applied in diverse areas, in (Broy, Cengarle, and Geisberger 2012) there are shown 2 scenarios related to Smart Health and Smart Mobility. A very simple application of this technology is a smart car lock, where a user with a smart phone or another device with Bluetooth connectivity can lock/unlock the car. The car is connected to a microcontroller device (equipped with Bluetooth technology) and if it receives a specific message, locks/ unlocks the doors.

Internet of Things (IoT) is known as the interaction between various machines, computers, sensors, and actuators that communicate via a network. "For about 30 years, it has been working with the idea of making all everyday objects a little more interactive", says ("IoT in Action - Towards Data Science," n.d.). It changes the way we get access to information and can be helpful in a lot of cases.

The concept had one of its first cases in 1982 with a Coca-Cola vending machine. In (Ornes n.d.) it is said that a group of computer science graduate students at Carnegie Mellon University in Pittsburgh, used to drink "Coca-Cola", but the vending machine was on the third floor and it was common to find that the stock was gone, or the drinks were not cool. To avoid these problems, they connected the machine to the university's computer network, providing them the information on stock and temperature. Since this event, the concept has changed a bit and spread to other research areas although the basic and simple idea is represented in this situation, a system that communicates via a network. In this given example it is possible to understand the interaction between CPS and IoT and its advantages, there is a CPS system implemented on the machine that permits to know its temperature and stock, then there is an IoT system connecting the information between floors. These technologies are commonly applied together, so many times most people will only say IoT instead of IoT/CPS.

IoT/CPS is not a specific word for a specific technology, there are studies to apply it on several areas, like smart houses, autonomous cars, or smart cities. We can see it already being used in smaller scales like smartphones connected to smartwatches or the car Bluetooth system. The potential of this technology keeps growing with the development of scientific areas like smart cities or smart cars, but it must be tested in small steps to guarantee security and efficiency before real applications.

The consensual definition of IoT is given by Veljko Milutinovic in (Kocovic et al. 2017):

> *"The Internet of Things is an emerging topic of technical, social, and economic significance. Consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way we work".*

Below, it is presented an example of an IoT/CPS application and some research on the state of hardware/technology used to install these systems.

- **Automatic meter readings as an example of IoT/CPS in smart cities**

Automatic meter reading (AMR) is a technology that permits collecting data automatically from sensor and send it via some network. This concept is majorly mentioned when talking about smart cities. The idea is to avoid periodical trips to the readers (electricity, water, gas) at each building, reducing time and people resources needed to do this kind of work. This technology is then used to make automatic billing, troubleshooting, and compute analytics. AMR systems take the advantages given by IOT and are being studied to be applied on real world situations. One example is given in (Holloway, Nwaoha, and Onyewuenyi 2012), where it is presented a project to apply AMR in water circuits in Vietnam.

In (Mlakić, Nikolovski, and Alibašić 2017), it was created an AMR system to read power on 3 electrical wires using open-source components, after analyzing its results, it was possible to see that a simple system of AMR can be economic and practical. This kind of technology can be expanded to share all kind of data if there is a sensor reader to give the right values.

2.3.1 Connection

Connection between devices is now a standard for nowadays technology electronics, for that, a communication protocol must be chosen. There is a variety of ways to connect more than one device, error detection, among others. In this section there will be introduced some of these connections and explain its applications with specifications for device reachability, power consumption and reliability.

- **Wireless**

  - **Cellular**

On a cellular network, there can be more than one access point. This technology works by sending electromagnetic waves from a device to an antenna, then the antenna sends that signal by cable to the receiver antenna, which then sends another electromagnetic wave to the receiver. It is mostly used for mobile communication as it can be very fast (specially post 3G communication). It is possible to connect microcontrollers to a phone antenna if we have a SIM card, which may need payment for it to be available.

Nowadays, this technology is very present because many people are phone users. The great thing of this protocol is that it is available almost all over the globe, so it significantly extends the range of the network in an IoT application. In (Souza et al., n.d.), it was designed a fire detecting system, using cellular communication to send alert systems when detecting any strange values. To use this service, the user pays a tariff, but as there are many options, it is usually a low-cost service, depending on what the user demands.

  - **Wi-Fi**

Wi-Fi permits a very fast transmission rate, it is ideal for the transmission of heavy files, like the streaming of a video. It has a big impact on IoT as it can connect many devices, its range vary with the quality of the devices, making it a great protocol to manage a factory with many sensors. The setback on this technology is its power consumption, it is very power hungry reducing the working time on battery powered devices.

  - **Bluetooth**

Bluetooth is very similar to Wi-Fi in the working process, but at a much lower transmission rate. It is mostly used when transmitting small blocks of data for example, word files, music, or small videos. It lacks in range, theoretically it can communicate at 100m but on real time situations it is limited to around 50m or less. It is low power hungry, making it a good technology for battery powered devices, like portable speakers.

  - **ZigBee**

According to (“Zigbee Wireless Networking - Drew Gislason - Google Books” n.d.), “ZigBee”, “Wireless control that simply works”. While most communication technologies tend to maximize the communication speed, ZigBee keeps it on low speeds to improve power consumption. This tech is used to control a led or keep track of states that do not need too much of a check rate. In this same book it is said that a ZigBee module can achieve 4+ years of working with only the power of 2 AA batteries, if the circuit is well dimensioned. Zigbee’s practical speed is around 25Kbps which is enough to transfer a few blocks of data per minute if the circuit monitoring just needs an update per minute (or a few per hour) this technology might be more than enough for the needs of the project. In (Miladinovic and Schefer-Wenzl 2018) it was used this protocol to communicate with a temperature and a light sensor in a medical surgery room. Light data would communicate via ZigBee and heavy data was directly connected to the base station.

- **Low power wide area network (LPWAN)**

LPWAN is a type of communication developed to improve IoT systems that do not need high-rate data transfer. Some projects can benefit from sacrificing fast communication in favor of better power lifetime. In a case where a farmer needs to check ground humidity in a plantation each km in line, it is not practical to install an electrical grid to power the sensors and the microcontrollers or replace batteries each week on the devices. In this case where we would probably rather have 2 readings per hour and a longer battery life instead of readings each second, it is of much use to install a LPWAN communication protocol. Bellow, based on (Mekki et al. 2018), there will be resumed 3 of these protocols.

- **Sigfox**

Sigfox is a limited half-duplex communication protocol developed to consume very low energy and to have very wide range. It can send a message to a device 40 km away in a rural setting, this reduces to around 10 km in an urban environment. Sigfox technology is limited to 140 daily messages each limited to 12 bytes which is around 6 messages per hour. The communication speed is around 100bps. REF

- **LoRaWAN**

This communication protocol allows half-duplex communication and higher speeds compared to Sigfox. Also, its messages can have 243 bytes so it can send more information in each message. In this technology there is not a limit number of daily messages (besides the time needed for each message).

- **NB-IoT**

"NB-IoT is a narrowband LPWAN technology which can coexist in LTE or GSM under licensed frequency bands." This technology can connect to 100K devices per cell and has a range from 1km up to 10km in a rural environment. NB-IoT offers a half-duplex communication and is the only that works on licensed LTE frequency bands (among these 3 examples).

- **Wired**

- **Ethernet**

Ethernet is a wired technology, is very fast, reliable, and works at lower power consumption. It is great for standing still devices that need a high data transfer rate. It is full duplex (can send and receive simultaneously), it is very common on gaming applications has it is stable. Sometimes Wi-Fi is unstable while transmitting too much data, giving a variable ping which might have some negative effects on some applications. The disadvantage of this protocol is it needs wires, so it is not great for mobile devices like a laptop or a mobile phone.

- **Ethernet/powerline**

Powerline is a technology that uses the power cables on a building to transfer data, it is very stable, and most times is still faster than Wi-Fi. It is a good option when the user has an old router, or the building walls attenuate most of wireless signals. It avoids having more cables spread around the building, making it an advantage compared to Ethernet, but it is also not as fast.

- **Wired vs Wireless**

In recent years, much attention has been paid to Wireless technologies due to their practicality, it is one less cable for each device. It works perfectly with laptops and phones as the user does not need to be in a specific place in the building. On the other hand, wired connections can reach much higher speeds due to having less devices in the line, reducing the risk of overlapping the transmission. Also, it is easier to recover and retransmit data as the points can check if the line is available or in use. Another point to check is the power consumption, Wi-Fi for example is very practical but it is power hungry, and when working with battery powered devices might be problem to choose the battery. To combat the power need on some of these protocols, there have been in development some LPWAN, these have some applications in smart cities as most needed information does not need to be refreshed every second permitting to use battery powered devices and not supercharge the electrical grid.

2.3.2 Hardware and communication

Over the years, there were developed many microprocessors with connection capabilities, these have a big impact on the development of IoT and CPS. Characteristics like reachability and power consumption are very important for the work, as well as a system compatible with a certain type of sensors. In this section it is compared some of these components, analyzing its advantages and disadvantages.

- **Arduino**

Arduino is an opensource hardware and software company. Arduinos are built in boards and are capable of reading inputs and then process outputs. There are a lot of different kinds of Arduino, also, the software for these devices has a wide range of libraries making it compatible with many electronic devices. These boards are equipped with a microcontroller instead of a microprocessor, and to program these devices it is needed an outside source (usually a computer), but after programmed, it is only needed a power supply and the software will run normally.

Arduino has been chosen for many projects, its uses vary from the simpler "turn on led" tutorial ("Getting Started with the Arduino - Controlling the LED (Part 1)" n.d.), to more complex applications, like controlling a motor speed over the internet, ("Motor Control Using NodeMCU - Arduino Project Hub" n.d.). Because it is open source, cheap, and has many users, it is very simple to find tutorials for a vast type of applications, being one of the preferred choices among teachers.

- **Raspberry Pi**

The Raspberry Pi is a low cost, credit-card sized computer, it can be plugged to a screen, mouse, a keyboard, and it will work just like a normal computer. The single board computers from Raspberry Pi were developed thinking about teaching applications. The brand wants this device to be used by kids all over the world as an introduction to programming and robotics subjects as it is a very easy way to learn how computers and programming works. This board can interact with the outside world and is used in a large variety of projects, from music machines to weather stations, (“5 Musical Instruments You Can Build With a Raspberry Pi” n.d.; “Build Your Own Weather Station - Introduction | Raspberry Pi Projects” n.d.).

- **BeagleBone**

BeagleBone is very similar to Raspberry Pi, both work like a computer and have similar sizes (90mm × 60mm).\ BeagleBone boards have good components and are fast for a microcomputer, also, these devices do not need an external SSD to work as they come with internal memory. In (Desai and Alex 2017), it was developed a device to monitor air pollution using BeagleBone. This gadget was connected in an IoT system that would store information on a cloud. It was also installed a machine learning service that would take data stored in the cloud and preform analysis to predict future pollution related results.

- **Minnowboard MAX**

Just like Raspberry Pi and BeagleBone, Minnowboard MAX is a small board microcomputer. It is powered by Intel and runs on a 1.91GHz Atom E3845 processor. The board’s schematics are available for download and the Intel graphics chipset has open-source drivers, (“MINNOWBOARDMAX-DUAL Reference Design | Application Processor | Arrow.Com” n.d.), also, it runs on Linux distributions being a good challenge for people who want to learn the way computers process data. It is fully open source making it a good instrument for people who want to develop software and test it directly on a processor. Like Beagle Bone, as the community is much smaller, it is much harder for a new user to start using the device (less tutorials, less drivers, and less people to help solving the problems).

- **Libelium Waspmote**

Waspmote is an open-source wireless sensor platform specially focused on the implementation of low consumption nodes. This device was designed to help the development of smart cities, this electronic use low power and slow data transfer (air pressure, temperature, humidity, are examples of information that occupy a few bytes, so the device does not need to use very fast communication protocols), making it very power efficient.

In (Azmi and Kamarudin 2017) it was developed a system to monitor the state of some patients, the device was equipped with a skin temperature sensor, a pulse sensor and a Waspmote. The choice of the Waspmote and the IEEE802.15.4 protocol was made due to its low power needs, providing a longer life cycle before it needed a recharge/replacement.

2.3.3 Cloud Computing

“Cloud computing is the on-demand delivery of IT resources via the internet”, says (“What Is Cloud Computing,” n.d.). It takes care of some of the needs for an IT application, like for example providing some memory to store data. Cloud Computing offers a lot to an enterprise, especially on the investment of IT equipment. In a case an enterprise needs to store a big amount of data and making it available around the world it would need: servers, programmers, server maintenance, technicians to handle support, among others, instead, the manager can hire a service and put those responsibilities on the service provider.

During the last decade, the amount of service providers has increased, this type of service can generate advantages for the providers and for the buyers. For the buyer it diminishes a lot the investment on electronic products, software developers and employees training, the provider set a cost to the service. The cost to each provided service depends on what will be offered to the buyer. Usually there are 3 kinds of services, reducing the buyer responsibilities to maintain. The 3 levels of services are known as: Software as a service (SaaS), Platform as a service (PaaS), Infrastructure as a service (IaaS). Based on (Camarinha-Matos 2010; "SaaS vs PaaS vs IaaS: What's The Difference and How To Choose – BMC Blogs" n.d.), the 3 cases are resumed in the next two tables.

Table 3 - Cloud Computing services advantage/disadvantages

Service	Advantages	Disadvantages
SaaS	• Managed from a central location • Accessible over the internet • Users not responsible for hardware or software updates	• As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change • As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change • As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change • As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change
PaaS	• As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change • As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change • As all the service is managed by the provider if any-	• Security on the cloud is always an unknown variable • If the buyer wants to change his PaaS provider, the current seller may not support the changing data, making it difficult to transact • If the buyer wants to change his PaaS provider, the current seller may not support the changing data, making it difficult to transact

	thing fails or is not as we expect we cannot control its change • As all the service is managed by the provider if anything fails or is not as we expect we cannot control its change	• The PaaS might not be optimized for the language chosen for the buyer • The control over the operative system is managed by the provider which may limit some performance
IaaS	• The most flexible cloud computing model • Easy to automate deployment of storage, networking, servers, and processing power • Hardware purchases can be based on consumption • Clients retain complete control of their infrastructure • Resources can be purchased as needed • Highly scalable • Cost varies depending on consumption	• Online security, all data is accessed by the internet and so the buyer must trust in the service provider security system • Has all the management is made by the customer, it is needed some investment on IT teams and some worker formation

Table 4 - Cloud computing service managing

On permises	IaaS	Paas	Saas
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Serevers	Serevers	Serevers	Serevers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

you manage
other manages

➢ **Cloud platforms**

Cloud platforms deliver cloud computing services. Cloud computing is growing in the commercial market sector as it offers low cost access to computing, storage and networks resources, (Bunch et al. 2010). The development and management of an efficient cloud platform needs scalable and fault tolerant structured data, besides all the supporting hardware with high data transferring rates and fast processing microprocessors. For an enterprise it might become much more affordable to rent the services instead of developing an IT department.

There are many cloud platforms in the market, and it generates valuable revenue, we can find big enterprises investing in these services like: Google App Engine, Microsoft Azure, Amazon Web Services.

➢ **IoT Platforms**

In (Hejazi et al. 2018) we find the following definition, "*IoT platforms consist of a huge number of objects connected around the world. It connects the edge of devices, gateways, and data networks to cloud services and applications*". In (Al-Fuqaha et al. 2015) it is said that for an IoT platform to work, there are needed 6 layers:

1. Identification – Distinction between devices connected along the network
2. Sensing – Raw data collectors (sensors/actuators)
3. Communication – Connectivity between devices
4. Computation – Data processing layer
5. Services – Information Aggregation Services
6. Semantics – Ability to extract knowledge

It is said in (Hejazi et al. 2018) that there are more than 30 IoT platforms, among them we can find Intel® IoT Platform, Microsoft Azure IoT Suite and IBM IoT Foundation Device Cloud.

➢ **Big Data Analytics**

Big data analytics is where advanced analytic techniques operate on big data sets, (Russom and Org 2011). It is about big data, analytics, and the way they work together. Big data was a problem in early 2000's, computation did not have sufficient processing power nor memory storage, but with the evolution of electronics following Moore's law, this problem was solved. When applied to Business Intelligence (BI), data analytics can be used to find patterns, or helping to find new opportunities on the market, taking advantage of Machine Learning (ML)/Artificial Intelligence (AI) technology. AI usually refers to an artificial creation of a system that can learn and get human like thinking. ML lets a computer space to learn from the data analyzed. Combining these two technologies, we get a program that can look for specific patterns, helping in many study areas.

In (Celesti et al. 2018), it is developed a system to prevent accidents on the roads. It proposes to connect devices (equipped with GPS, smartphone) to a network and for it to be constantly sending information about its position to a database. This data is analyzed to calculate speed, acceleration, detect sudden stops, among others. After the data processing the system can detect patterns and predict the state of the traffic, in a case of an accident or a sudden stop, it sends an alert signal to the cars near it possibly avoiding more accidents. In (Ingle and Phute 2016) it is said that Tesla uses a software called "map matcher" to detect traffic, any traffic congestion is automatically detected in each road segment.

2.3.4 Application examples

In this section it is shown and studied three projects that implemented IoT platforms, "Water Factories in Vietnam using Automatic Meter Reading", "Monitoring the oil of a transformer" and "Online Monitoring and Controlling Water Plant System Based on IoT Cloud and Arduino Microcontroller". The first project particularity is that there are three distinct devices using different communication protocols for their distinct objectives. The second project has installed an alert system which is important for this work. Third project gives two distinct modes to manage water while giving all the information gathered through an interface connected to an IoT platform.

➢ **Water Factories in Vietnam using Automatic Meter Reading**

On 2018, a study about automatic meter readings (AMR) was published by Vu Chien Thang, (Chien Thang 2018). Automatic Meter Reading consists in the installation of automatic systems on the public services network, like water, gas, or electricity. These systems avoid people trips to get meters values and states. "The goal of AMR system is to help collect the meter measurement automatically and possibly send commands to the meters". This project developed three distinct devices: one to monitor water tanks (smart water sensor), another to check water flow on the pipes (smart water meter), and an internet gateway used to send information over the internet.

- **Smart water sensor**

This device reads pH, electroconductivity, and temperature of water in each tank, then, it sends these values to the base to be processed. With this gadget it is possible to get water monitoring in real time of the tanks around the country (if they are equipped with the device), also it is very practical to detect problems on the water without the delaying of human analysis.

For communication it was chosen General Packet Radio Service network (GPRS), this technology is standard for 2G and 3G protocols in mobile networks. The microcontroller chosen for this application was the Texas Instruments TivaC TM4C123GXL. This ultra-low-power device connected the 3 sensors (pH, electroconductivity, and temperature), and the Sim 908 module used for the GPRS communication.

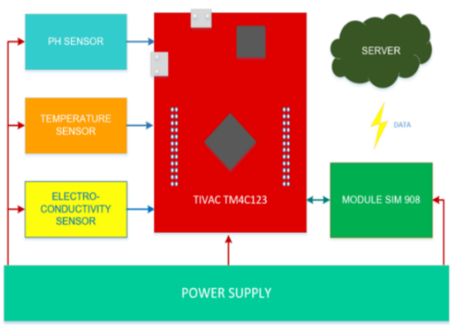


Figure 9 - The Functional Block Diagram of the Smart Water Sensor, (Chien Thang 2018)

- **Smart water meter**

The smart water meter has 2 functions, it measures the water flow automatically and send the water consumption periodically to the centre of operations. With this device it will not be needed to contract people to go to every house get the water readings, the readings will be sent via Zigbee networks to an internet gateway and then delivered through the internet.

This device has a flow sensor that detects and reads the flow passing through the sensor and with that value, it estimates the water consumed during that time interval. The protocol used for communication on this gadget is Zigbee, a very low power consuming technology. For this objective it was installed a DRF1605H, a Texas instrument with low transmission speed, low energy consumption, and low cost. The disadvantage using this protocol is that it will need another device prepared for the long-distance communication.

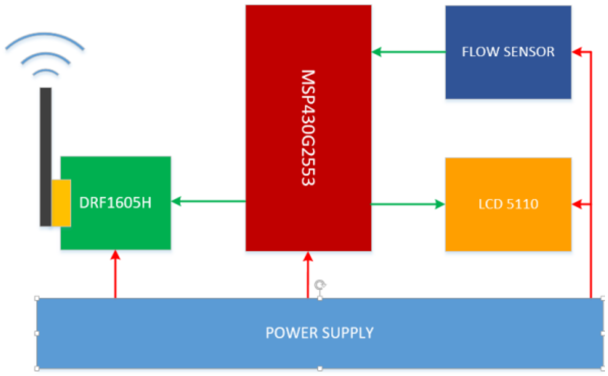


Figure 10 - Functional Block Diagram of the Smart Water meter, (Chien Thang 2018)

- **Internet gateway**

This device is used to receive and send the smart water meter values over the internet. It is used to allow the long-distance communication missing in the Zigbee protocol. To build this project it was once more chosen a Texas instruments device, TM4C1294 KIT. This board is equipped with a programable microcontroller that enables many communication interfaces. The choice for the Zigbee communication was again the DRF1605H and for the long communication it was used the ethernet available on the Texas board.

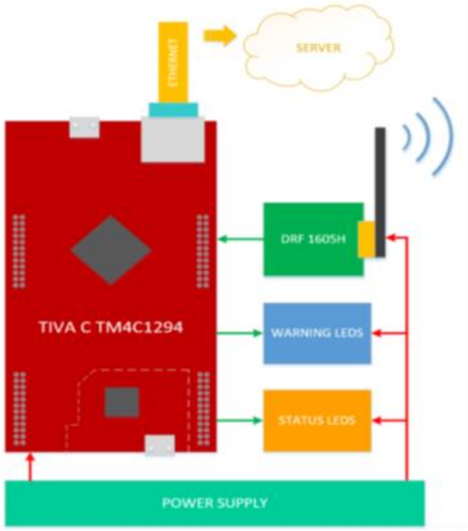


Figure 11 - Block Diagram of the Internet Gateway, (Chien Thang 2018)

- **Monitoring the oil of a transformer**

In (Pawar, Wagh, and Deosarkar 2018), it was studied a system to monitor the oil on a transformer. The block diagram can be found in figure 13. It comprises several sensors to get the values from the oil, an ADC to convert these signals and a PIC18F4550 to process them. Then, the values would be displayed in an LCD and stored on a buzzer. The system was developed in a way that any strange values would be communicated via GSM/GPRS protocols. This system is simple, it does not have many components and it is relatively small. The communication protocol used on the project gives a wide range if used with the cell phone networks.

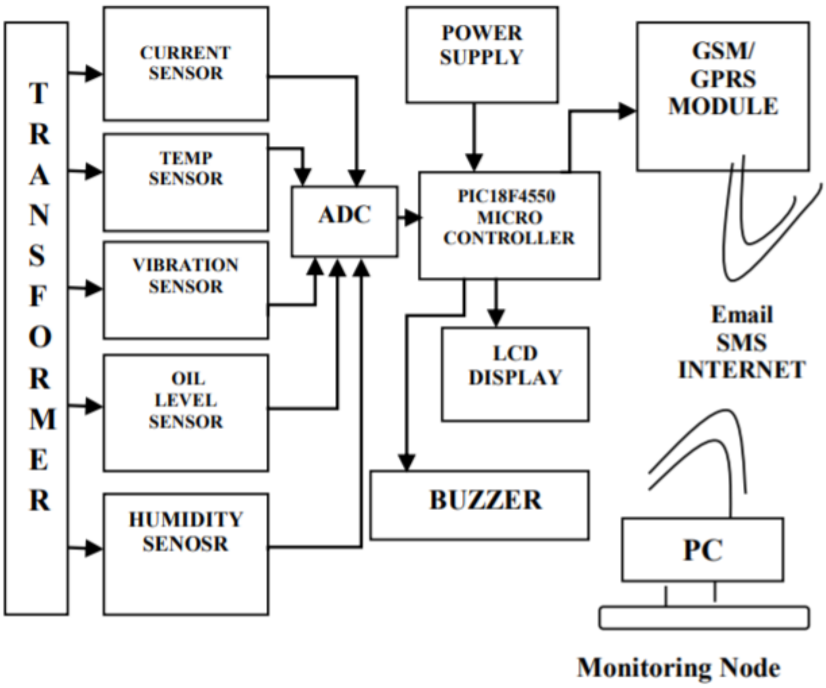


Figure 12 - Block diagram of distribution transformer monitoring system, (Pawar, Wagh, and Deosarkar 2018)

- **Controlling water circuit**

Water is essential to society's life and to preserve it, in (Abdullah 2017) it was developed an IoT system to monitor access to two water tanks. The system was accessible through a wireless network, the user had an interface with the information provided by the device. With the information given it was possible to manage the water tanks manually or automatically. The system was equipped with two ultrasonic sensors to read the water level, two pumps to move it, a relay to the water processing, an Arduino to communicate with these which was connected to a router.

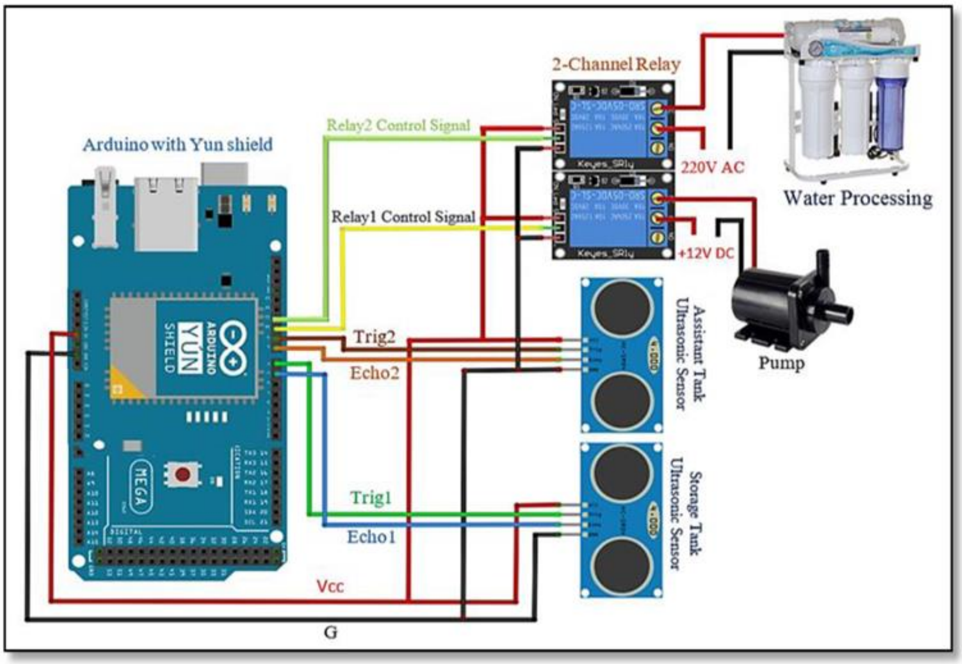


Figure 13 - Water Monitoring device, (Abdullah 2017)

To conclude, in these three systems we can find some similar challenges faced in this project. In the first project there are explained 3 distinct devices and the solutions used for the communication; the project connects 3 sensors to one microcontroller that communicate over the GPRS network. In our work, as the sensors are inside a warehouse with some metal objects, a Faraday cage effect weakens communication for the outside. The GPRS is not fully out of the game as it might be used as in the second project as an alert system installed outside the warehouse. In the third study, we analyze possible alerts to control with our system, as the example of trying to remove water (oil) from an already empty deposit. Also, it is important to study the behavior of the pumps changing with the mode the system is being controlled. There are some solutions in the market to monitor olive oil, but they are quite expensive and with proprietary management rights. Indea Technologies (a Spanish enterprise) claims to have one solution to the problem, using [Alma/System] ™, nevertheless it requires the olive oil industry to make an extreme economic investment either in equipment or service/application management. The approach proposed in this work aims to give a less expensive solution.

2.4 CABB Current state

2.4.1 Story and services

According to ("Associações Filiadas | Cooperativa Agrícola de Beja e Brinches, CRL" n.d.), CABB was the result of a coalition between 2 agricultural cooperative, Beja and Brinches. This occurred in 2008 in a historical movement in the Beja region. It became one of the principal agricultural producers of olive oil in Portugal. This coalition helped both cooperatives as the solutions completed each other, Beja was strong in the cereal production and Brinches was enormous in the production of olive oil.

In this work the focus is on the warehouse/meal in Brinches. In Brinches, the product arrives in the form of olive fruit. The fruit will go through the process described in the start of section 2 where it will be transformed to olive oil. Depending on the harvest as well as in the transformation process, the olive oil will get some characteristics that will define it. These characteristics are acidity, peroxide, organoleptic and spectrophotometry values. This new product will then be moved to large tanks on a warehouse. The olive oil will be kept stored until it is needed to be packed for selling. The facility is equipped with a packaging line that puts the olive oil in bottles, which will then be grouped in a larger package and then sold.

2.4.2 Existing processes & necessities

CABB produces and store large quantities of olive oil. This plant is equipped with a large mill that processes tons of olive fruit, which are then transformed in olive oil and stored in several tanks in a warehouse. Currently, these tanks are equipped with a small tube that goes along its height, which is used to verify the height of product inside. These tanks are around 10m high and $14m^2$ of base. It is required a person (operator) to look at this tube and verify the height of oil inside, which gets harder the fuller the tank is. It was possible to verify that sometimes the expected volume was not right with the real volume.

User interface and accessibility is also a big throwback in the installed method, information only passes in conversation, and the warehouse state is not represented anywhere. This provides errors and misinformation, which has been revealing bad for product management. Following, there is described the managing state of CABB in topics:

- **Data Storage**

Values are registered in paper and stored in an office outside the warehouse. Sometimes a paper is suspended in the tank with the information relative to the olive oil inside. There is no interface for data visualization, to view its level the worker needs to check the records or go to the warehouse.

- **Accessibility**

The information is only available on the warehouse and might not be actualized.

- **Data visualization**

There is no interface available, nor the information is automatically computerized.

- **Oil Transfers between tanks**

The oil transfers between tanks are made with the use of a pump, all pumping is managed manually, a worker will install the hoses on the tanks and connect them to the bombing system, then some other worker will turn on/off the bomb. There is no system to verify the oil mass/volume bombed besides having someone with a flashlight looking at a tube, so the control is always estimated.

- **Monitoring**

There is no informatized monitoring in the current state of the facility. Many problems might occur only to be verified days/weeks after as for example the transfers, we were told by the managers that sometimes oil would be pumped from the wrong tank creating problems related with flavor and other specifications. Another verified mistake was the wrong quantities registered in each tank; differences superior to 10 000 liters.

3 Solution proposal

In this section it is studied the requirements and the intended use cases of this system, which will be used as guidelines for the development of the product. Considering this first part, we will follow to the study of an architecture to implement the system, analyzing its levels and their purposes.

3.1 Requirement's identification

3.1.1 Context identification and understanding

This work intends to improve the information gathering of the amount of olive oil stored in a warehouse. This improvement will rely on the accessibility and fidelity of data by automatizing the system. This should bring workers comfort, reduce human error on data gathering, and increase confidence to the managers during decision making.

In the current state, not having an efficient and trustworthy system to get the volume inside the tanks has revealed to be a problem to the management of the cooperative. Eye information is not reliable (as the tubes get dirty) and the low number of readings might mislead the managers. It is very important to know the amounts of product to plan the sales and the tank cleaning schedule.

The permanent needing of a worker in the warehouse is also a problem that would benefit improvement. In the present, it is not simple to get information on the quantities outside the building, forcing the presence of someone in the warehouse.

Records are kept in papers, which are not very practical and occupy unnecessary space.

As the data is not informatized, there is no automatic interface to show data to the user. Also, it would be very positive to have a perceptible image representing the state of a tank and a list of the records during the times.

By the end of this project, it should be possible for a worker to get access to the amount of product as well as the description (acidity, peroxide, organoleptic values, ...) of the oil inside the tanks in the warehouse. It should also register all alerts related to the level during the storing of the product. In order to identify all the involved actors and understanding the functionality of the system, some scenarios were defined, as follows:

- **Scenario 1 – Stored olive oil**

The olive oil is stored in large opaque containers inside a warehouse, there is no method to gather information on quantities inside the tanks as there is not enough light to view the transparent part of a tube following the tanks height.

The new system should constantly verify the level of oil on the tank (using an electronic device) and show it on an interface through a computer screen, the value would also be registered in a database, containing the volume and the time stamp of the moment which the value was verified. It should also be possible to save the characteristics of the specific oil inside, so it could easily be checked. This will improve the oil management quality of life by having the needed information on the distance of a few clicks.

- **Scenario 2 – Historic**

The product is stored for many days and will sometimes be partially moved from tank to tank. The registered values, if in paper, most of the times will not be presented in a graph, which sometimes can lead to unexpected volumes of oil. The new system should automatically generate a graph with the values registered in a data base, presenting the user a historic of the level of oil in a specific tank. If sometime a user verifies that the quantity of oil in a tank is not expected, the user can look for the change in level of oil during a time window and look for when it changed. All values should be stored in a data base with easy access. This system should present the data in an easy and understandable graphic. It should also provide the historic of transfers in a list. All this information would be available on a data base which would reduce the number of papers on the office and improve the data presentation. Besides the levels and the transfers, there would also be the historic of alerts. These are needed to detect anomalies that might be caused by an accident (for example a leak in a tank), the system would be capable of detecting these variations and register them in the database.

- **Scenario 3 – Adding/updating information**

The oils are tested after being put on the tanks, this new system would let the user create new oils with its specific characteristics and save on the database.

Olive oil tanks will change their contents and the new oils will need to be added/updated to stay true.

Some workers will change their position in the factory, for example a machine worker that becomes a supervisor with new authorizations which will then be able to do other type of functions.

- **Scenario 4 – Transfer oil inside the warehouse**

Sometimes it is needed to mix the content of two tanks, it might be to change the flavor or the acidity. This requires the pumping of product and the verification of volume transferred. This system would do it automatically with the use of bombs and the height controlling system. The user would only have to define which tank to take from and to and then chose the quantity to move. This would reduce human error by showing all the characteristics of the oils in the transfers and requiring an administrator to allow the transfer.

Having into consideration the scenario examples of the proposed system, the main actors of the system are identified as:

1. Technician - Preparation of the software and connection between points in the network. Management of accesses to the information to the data base.
2. Administrator - Administrator will be able to work all the information but the levels read from each device.
3. User Employee - View the information registered in each tank and verify alerts related with volume change.

3.1.2 Functional & non-functional requirements

The requirements analysis focuses on establishing the system characteristics in terms of user needs and is essential in the process of a system design and development. Starting from the description of the scenarios, a set of functionalities that the system should offer were identified. The functionalities are organized into two main categories: functional requirements and non-functional requirements.

➢ **User Requirements**

The proposed system should deliver reliable information on the amount of product inside the tanks and show it in an explicit interface. The user should not only be shown the amount of olive oil but also the fullness of the tank. This can avoid some overfilling and accidents inside the warehouse. The system can also present another info useful to the user, like the type of olive oil inside each tank, with all its main characteristics defined.

➢ **Functional Requirements**

The principal problem that the CABB is facing, is the misinformation on the level of olive oil in each tank, so this system should propose the development of a technology that would show the user the levels of olive oil inside the tanks.

This system should concentrate information in a single device, making it easier to organize data for presentation. This will demand a connection between several devices.

Information should then be stored in a database. This will demand an API capable of receiving information and then communicate it to the database. Information should have the level and the time stamp associated. This value should not change much in time, so there is not much difference in sending the time stamp so much as saving it in the database time stamp (time at which the message was received).

It is important to know the product inside each tank, the installed product should be able to save and present this information to the user. Also, the installation can expand, and some functions should only be made by a supervisor, in order to guarantee this, a user access and permission approach has to be explored. It is also important that the data is available on the cooperative network, this could reduce the amount of people working and walking on the warehouse and favor working on other places connected to this network.

Finally, this system will need a user interface, this should make data reading easy and provide all the interfaces to use the data base. It is very important that it shows all the desired information in a simple way.

Table 5 - Functional Requirements

Requirement ID	Requirement	Necessity	Comment
1	Autonomous olive oil level reading	Must	Should be Contactless
2	Periodical Reading	Must	
3	Sensor Wireless Network	Want	There is no need for high-speed communication
4	Database with the values/timestamp/dispositive	Must	
5	Database with other relevant information	Want	
6	Interface showing each tank and corresponding level	Must	
7	Interface showing the level during a determined period	Must	
8	Interface for extra information[3]	Want	Used for each tank describing its content
9	Login system with different user types and permissions	Want	
10	Alert system detecting oscillations on the level of the tanks	Must	
11	Pumping Actuators	Want	Pump a controlled quantity of olive oil to another tank

[3] We want to save additional data in the project, helping management and organization (olive oil characteristics, list of transfers related to a tank, list of alerts, user databases...).

➢ **Non-functional Requirements**

This work will consider a set of restrictions to the good functionality of the system. Data must be very reliable as the new application might not leave space for the worker to read the values in case of a breakdown. It should be secure to grant that the data is not lost or changed overtime. The system needs to be simple to use, this should avoid major problems when handled by a new user. It should also respect the different types of users as the access and permissions should vary from a common user to a supervisor. The development of the new system should be open for scalability and improvement, built in layers making it easier for changes during the time.

3.1.3 Use cases

Starting from the list of functional requirements identified in the previous section, a set of use cases of the proposed system were identified. These use cases are represented via a diagram of use cases. In this requirement analysis process, two main diagrams were elaborated, one for the installation phase (Figure 14) and another for the operation phase (Figure 15).

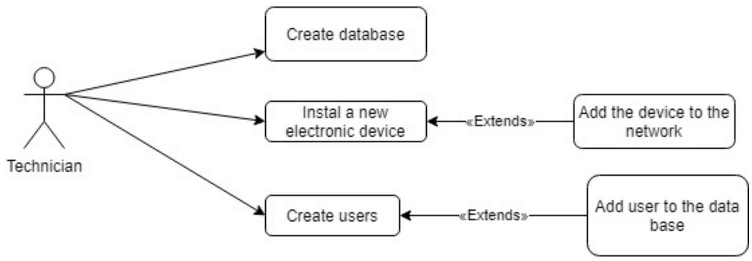


Figure 14 - Installation phase use case

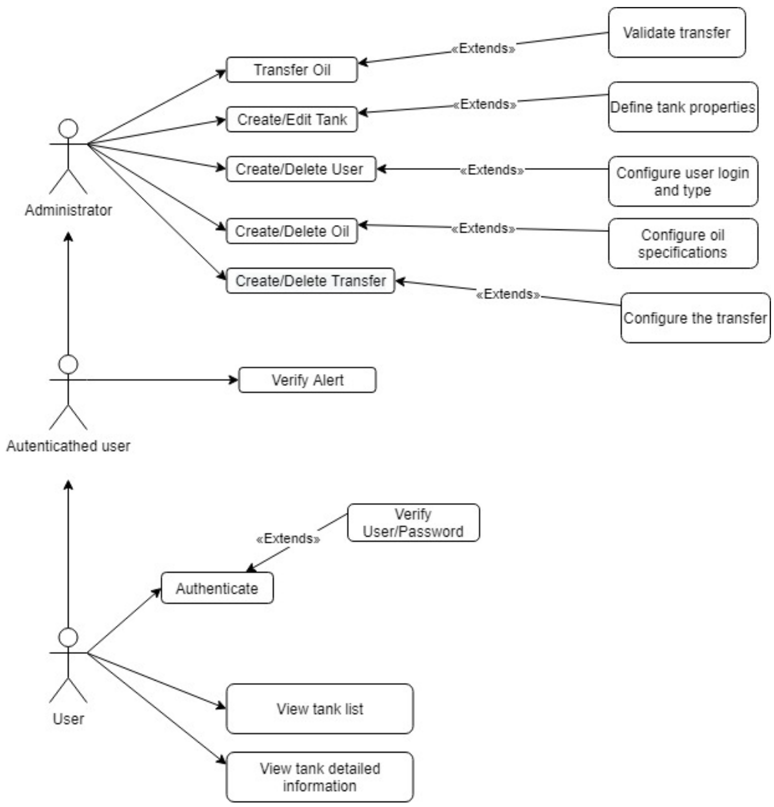


Figure 15 – Operation phase use Cases diagram

Developing this phase of the project should open doors to a future scenario where there would be a fully automated system capable of transferring the product tank to tank, with a simulation functionality and possible predictions for future states.

3.2 Conceptual Architecture

In this section it will be presented the conceptual architecture for the proposed system based on the requisites shown before.

The system will be divided in 3 parts:

1) Infrastructure - The infrastructure will contain the devices installed in the warehouse, will be responsible for their communication and for the data validation. The processed information should then be uploaded to the service layer.
    a) IoT connectivity - Sensor network gathering information
    b) Storage and resources management - Information analysis and error detection
2) Service - This layer will be responsible for the data saving and management, will also work as a bridge for the communication being connected with the other 2 layers. It is of major importance as it will work as the brain of the system. It will receive all the information from the other 2 layers, process it and answer requests.
    a) Transfers management - Registration of transfers of olive oils and possible sales
    b) Historic data management - Histogram of olive oil quantities registered from each device
    c) User registry - Information and permissions for each user
    d) Alerts - Data analysis and detection of possible problems

e) Data analytics - Processing of data to predict possible future scenarios
f) Tanks monitoring - Page showing all the sensors installed on the warehouse (each one representing a tank)

3) Web Application - The web application will be the interface for the human/machine, it will present the data and permit the user to work with the system layer. It will process the data given from the user and validate it to send requests to the service layer.
   a) Updated data - automatic update of data keeping the information fresh and valid
   b) Requests processing - Process the user requests to the service layer
   c) Validation and notification - validation of data before making the requests to the service layer and presentation of information in a user-friendly way

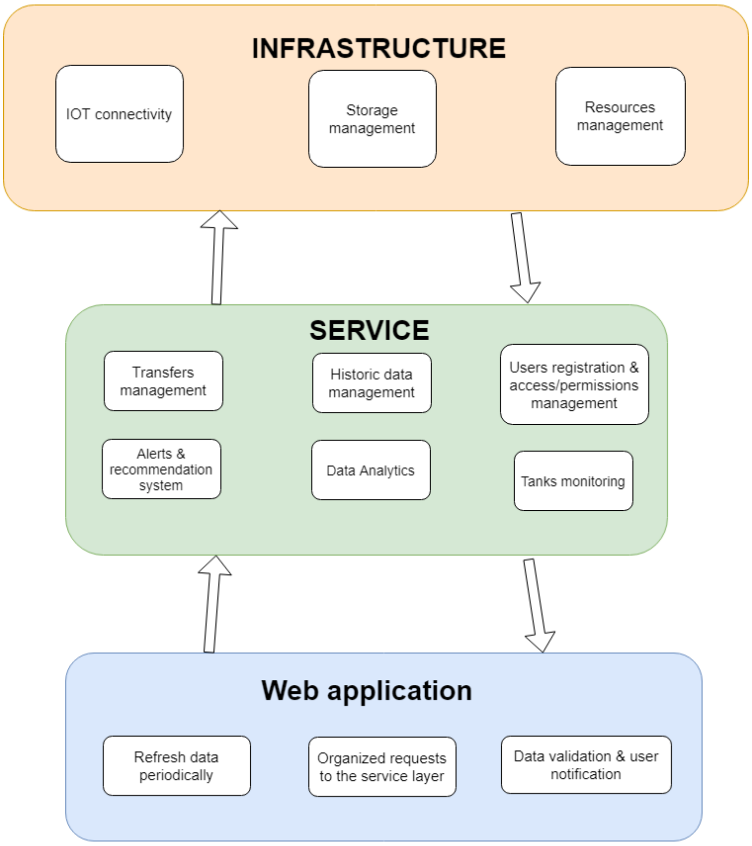


Figure 16 - Conceptual architecture

To implement the infrastructure layer there should be installed a network for communication. A network transmitting values/requests and communicating overtime with a central device connected to the service layer. The data sent to the service layer should be evaluated and verified before saved or communicated, this will reduce errors and unnecessary occupied space. If there is a pumping system, this information should be previously analyzed to avoid possible problems, for example, pumping from an empty deposit.

The service layer will receive and process data from both other layers, information from the warehouse to monitor and from the web application to process/save. This would be the memory and the brain of the whole system, containing all the information and history about the warehousing of the olive oil, the information about users, and others (alerts, transfers, ...).

The web application layer will work as an interface human/machine, will offer user friendly interface for requests, and will process the data received from the service layer and present it in an intuitive way for the user. All the processing done in this layer should be mainly to reduce unneeded requests to the service layer, for example a login without the user space filled.

4

Solution implementation

This section presents the implementation of the proposed conceptual architecture. Firstly, the proposed general system architecture is described and then the implementation aspects are defined considering all the involved limitations, advantages and disadvantages. It is also analyzed the development and constraints of the proposed system to install in the CABB. Same choices made during the development of the project are further justified.

4.1 System architecture

The system will be divided in 3 main parts as proposed on section 3, Infrastructure, service, and web application, each of these will then be divided for a total of 5 main parts:

- **Infrastructure**
  - **Sensor Network**

A sensor network to gather all the information needed for each tank, the information should be gathered and then sent to the network base

  - **Network base**

In this part, all the information should be processed, if a value does not make sense, it might be worth to wait some extra seconds and receive a new value to verify instead of instantly uploading it. Also, it is not needed to spam the same value for hours, it should only send new information when the value changes (maintaining a maximum delay guarantying it is possible to verify if the system is still working).

- **Service**
  - **ThingsBoard**

ThingsBoard will be used as a simple free way to test the system in real life, it offers us a free online database. In the future, ThingsBoard and the local server can emerge and be deployed in an online server.

- **Local server**

In the local server it will be saved extra information (olive oils characteristics, transfers, user database...) and processing it, it will be the backend and will be responsible to connect with ThingsBoard while processing the requests from the Web application.

- **Web application**
    - **Web application interface**

The web application will be responsible for the interaction with the user, presenting all the needed information while providing the tools necessary to manage all the information.

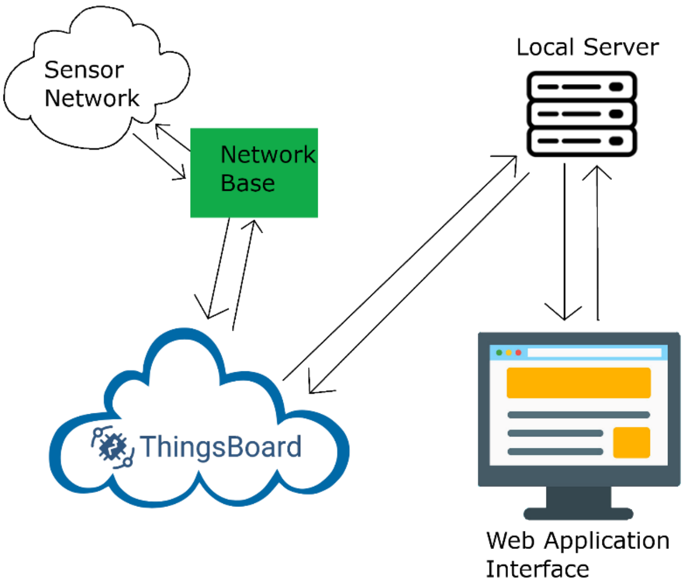


Figure 17 - High-level communication architecture

4.2 Components & technologies

➢ **Reading System**

There are several methods possible to obtain the distance values, so there were evaluated their advantages/disadvantages and talked to the administrators about the options. During a brainstorming about the devices, its characteristics, and prices it was decided that it would be preferable to use a contactless reading system due to the following reasons: Olive oil drops some sediments, which will require the CABB to clean the tanks time to time, so the options would be to either make a strong and reliable system that could take the pressure of cleaning the tanks, or a system simple to remove and then install again; It is important to avoid contact with food products and guarantee its quality; Submersible pressure devices are expensive and it would be complicated to take them out for the tank cleaning, besides the reading system would have to be prepared to deal with the lees.

After some research and discussion it was decided to use the Lidar Lite V3, a Garmin device, , which is Arduino compatible. Lidar Lite V3 is a “near infrared laser” dispositive used to calculate distance by measuring a time delay between a sent signal and its reflection (“Lidar Lite v3 Operation Manual and Technical Specifications,” n.d.). In the data sheet it claims that for distances >5m the distance error is smaller than 100mm ($100mm/10000mm \approx 1\%$) so the measured error (in our system) should always be inferior to 1%. This device is compatible with Arduino language, when installed with the connections shown in Annex 1. The device should use I2C protocol to communicate with the Arduino device. I2C protocol provides easy communication without data loss, uses only two wires for communication and it is a fast communication protocol (Mankar et al. 2014). To assembly the laser we will need to install a capacitor and wire 4 cables:

- Power supply (5V)
- GROUND (reference voltage)
- SDA (Serial Data line)
- SCL (Serial Clock line)

➢ **Processing laser values**

Lidar Lite is compatible with Arduino language which facilitates the work, Arduino boards are relatively cheap and there are many to choose for the different uses. Also, Arduino has a wide range of user tutorials and applications. To communicate between the laser and the board it will be used the I2C communication protocol, so the board needs the 4 pins referred in the previous item (5V, GROUND, SDA, SCL). There will also be needed 2 specific libraries, “Wire.h” used to communicate via cable and “LIDARLite.h” which is a library used for the laser setup and communication.

➢ **Communication protocol between devices**

The warehouse is a workplace with people walking around in low light environment, with a slippery floor, also, while transferring oil between tanks there are installed hoes on the passage. Considering this we thought it would be preferable to have the minimum number of objects laying around on the hall and choose a wireless communication protocol. After evaluating the available technologies, it was opted to choose Bluetooth Low Energy.

In this phase of the project in particular, it is not needed a very fast communication protocol, information is only a few bytes long and a few data blocks per hour. But as there is also the possibility of expanding the project, it is important to design a system that is fully prepared for further development.

➢ **Choosing the main board**

Choosing the board, there was a focus on the compatibility, performance, and communication possibilities. For single device systems it is common to use the Arduino Uno which is a low-priced board with a good controller performance. For wireless networks it is usually used the ESP86 which is equipped with Wi-Fi technology. Very similar to the ESP86 is the ESP32 which is equipped with not only the Wi-fi but also Bluetooth communication protocol. There are other boards with other possibilities but using one of these there is space for expansion, for example creating a control system for a pump that would require a fast communication protocol. Another characteristic of the ESP32 is that it is compatible with Arduino. Considering the information given before we opted to use the Espressif ESP32 Devkit as the microcontroller (Figure 18).

Figure 18 – Espressif ESP32 Devkit

In ("Getting Started with the ESP32 on Arduino IDE [Full Guide 2020]" n.d.) we can find the pin layout for the board as illustrated in Figure 19.

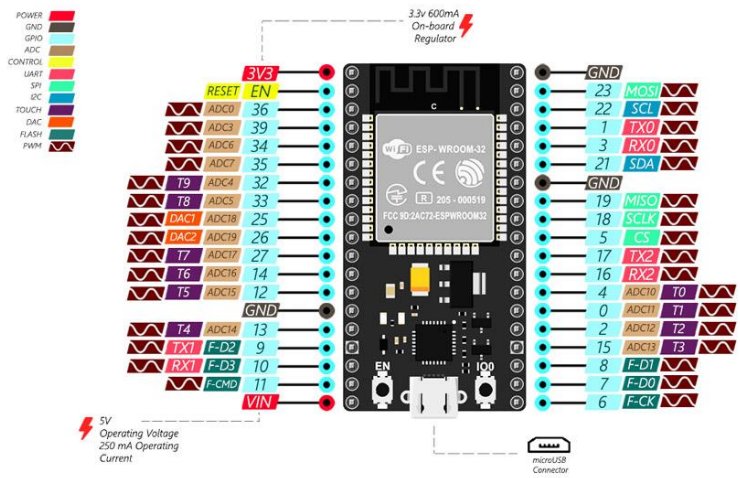


Figure 19 - ESP32-WROOM-32 Pin layout

- **Warehouse central device**

After having the values from the ESP's, it will be needed to send it to a data base. This will require a device connected via cable to outside of the warehouse. It is also needed a device with some independency and ability to reprogram in case of needing to solve bugs, there will not be full access to the CABB, and it would be really useful to have remote access. We also have to consider that due to its location it needs to be a small object. One such device that offers these abilities is the Raspberry Pi. With its credit card size and low power consumption it is fairly easy to place anywhere (as long as there is a power font and an ethernet cable). In this project it was opted to use the raspberry as normal desktop to access remotely, this way it is possible to connect the client ESP32 via USB, as illustrated in Figure 20. The information given from the Bluetooth network will be processed and then uploaded to ThingsBoard.

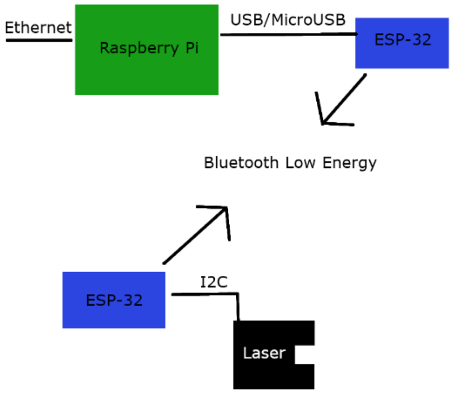


Figure 20 - Electronic installation sketch

4.3 ThingsBoard

"ThingsBoard is an open-source IoT platform that enables rapid development, management and scaling of IoT projects. Our goal is to provide the out-of-the-box IoT cloud or on-premises solution that will enable server-side infrastructure for your IoT applications.", ("What Is ThingsBoard? | ThingsBoard" n.d.).

ThingsBoard offers cloud computing services, more specifically IaaS services. It is mostly designed for sensor networks. ThingsBoard is equipped with many widgets and interfaces to visualize data, which can be helpful for people that do not need a special interface and are only making tests. A big advantage in using this platform is its python REST API (they also provide a Java REST API), having a prepared API will be very helpful and reduce the work needed in communication between platforms. ThingsBoard let the user choose assets, devices, telemetry names, being very simple to set a bigger system as it can be very organized.

ThingsBoard offers free IoT services and has online services that will be important for the development and test of this system. In this project, ThingsBoard is used to connect the system over the internet and utilize its data processing tools for simpler progress. The communication relies, as previously mentioned, on the REST API available as a python library. The use of an API will simplify the development speeding of the whole process.

ThingsBoard will be saving values sent from the CABB as well as its timestamps that will be important for the design of evolution graphs. Figure 21, represents the data model implemented in ThingsBoard.

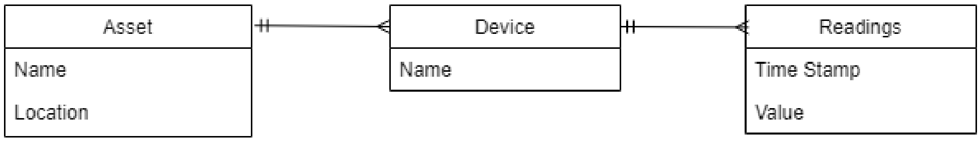


Figure 21 - ThingsBoard data model

Readings will be saved in on different devices that will let us know which tank they represent. The asset is meant to represent the building where the service is installed, in the current study case it will not be fully used as there is only one installation.

4.4 Web development tools & technology

To save additional information we chose to keep it on a local server for further development. In the future and after some tests, it should be possible to deploy the full system to an online server and access it from any browser.

On this Data Base there will be saved a few types of data with the information on tanks, users, alerts, types of olive oils, logins, and transfers. The data will be saved in SQL format, using SQLite, "SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine", ("SQLite Home Page" n.d.).

In order to present data to the user it was decided that it would be a good choice to develop a web application, this would permit to make it available in most phones tablets and PC's, thus not needing a different application for each device/operative system. This part is divided in two phases, the backend that would be responsible for the data management and store part of it and the implementation of the services, and the frontend that would be responsible for the interaction with the user.

4.4.1 Django

In this phase it is described the development of the backbone of our application, it should gather all the information and organize it, this requires the implementation of the ThingsBoard API to communicate with ThingsBoard and then answer the information requested from the frontend. It is also this part that verifies the users and logins (information available might vary with the user logged). In other words, it will be responsible for: security, communication, and organization. To develop the backend, it was chosen Django python tools.

"Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design", ("The Web Framework for Perfectionists with Deadlines | Django" n.d.). Web sites can be separated in two classes, static (display the same information for every user) or dynamic (change the information based on each user), ("Static vs Dynamic Website: What Is the Difference?" n.d.). If building a small and simple website with 2 or 3 information pages, a static website developed in html is the way to go. It does not need the complexity of a recursive connection with a database, and it will always show the same information. A dynamic website will show information based on a state which will usually require a database and user interaction. For this project it is required a dynamic website that will refresh periodically and will change interactions depending on the user.

Django offers database tools that are employed in this project. The implemented database follows the model of Figure 22:

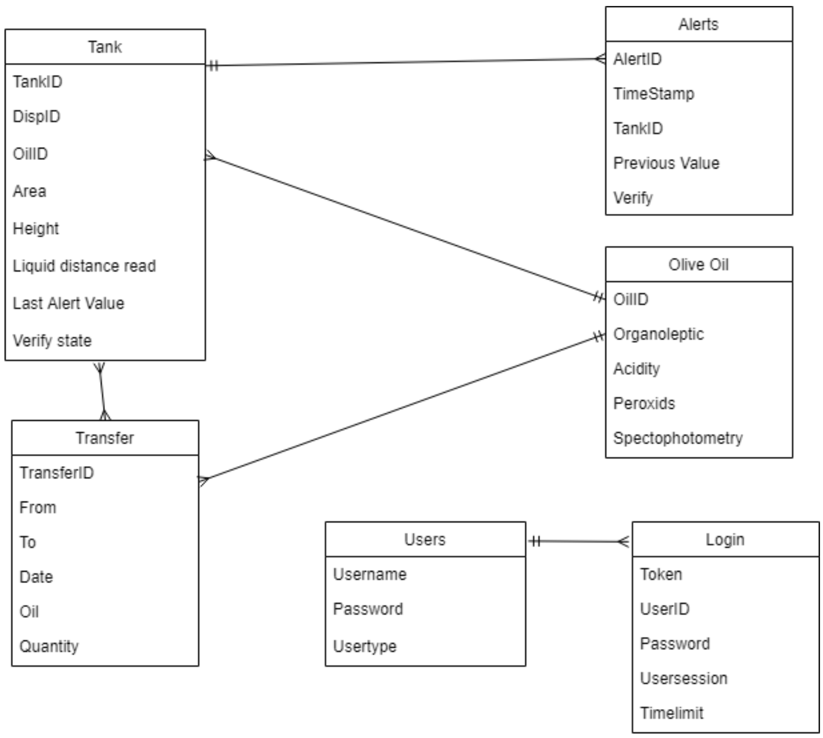


Figure 22 - Django data model

There are many ways to develop a dynamic website, it is possible to install many features and change appearance based on the client needs. With experience on developing these sites it can get a little frustrating because, although most sites are different, many features end up working similarly and it gets annoying to reinvent those same features, so developers end up creating libraires and optimizing them, these are called frameworks, ("Python Web Development with Django - Jeff Forcier, Paul Bissex, Wesley J Chun - Google Books" n.d.). Django is a consequence of many developers working together for the same final, it is an opensource framework that simplifies the development of websites.

In this project it will be majorly used the database calls developed in this framework. Django let us create data structures and will automatically prepare the database to save them with call methods. We will also use the update and library management that let us change a structure without having to recreate the whole saved data.

Our backend will be divided in 4 major files explained in the next table:

Table 6 – Data Processing files in the backend

Models (python)	This file is responsible to specify the data structures that we will be saving in the local database. This will simplify managing the data and will help in the creation of new blocks if needed.
Serializers (python)	Serializers will be used to facilitate communication, they will automatically verify if the JSON formats are valid before they are processed.
URL (python)	The URL's page is responsible to send the requests to the right function.
Views (python)	Views is the main file of the backend, it is responsible to process the requests, validate data, control user access and also communication with ThingsBoard. It will also be responsible to convert the Distance values measured by the laser in tons (by converting the distance in height of liquid, then measure volume, and finally with the density, calculate the mass).

The communication between ThingsBoard and the Django will be made via internet to improve remote development and testing. As said before, it will be used ThingsBoard REST API for python that simplifies the requests. A simple representation is shown in the following image (Figure 23):

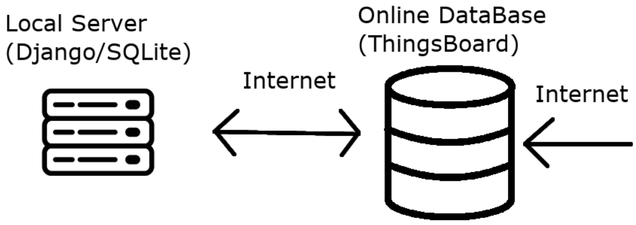


Figure 23 - ThingsBoard and Django

Although we developed this platform, we kept ThingsBoard on the cycle as it would let us test the project installed in the facility, and so we needed to install and test an API to facilitate our work, tb-rest-client. Tb-rest-client translates and process our requests without needing to develop our own requests to ThingsBoard from root.

Using the tb-rest-client to communicate with ThingsBoard revealed to be a goof choice and practical to use although it was needed some time to test and get used to the functioning. The tests were mostly meant to enlighten the format of the messages received from the ThingsBoard, this required several attempts to understand the work process and the meaning of variables in the requests. To help developing this phase of the project it was used Swagger online tools ("API Documentation & Design Tools for Teams | Swagger" n.d.), which simplified the course of software development taken for the project. Sometimes we would not get the intended information from the API calls which would later reveal (most cases) to be requesting formats as well as missing information on the JSON messages.

4.5 Interface

4.5.1 Frontend

In this phase, the part responsible for the user interface is developed. The communication with the backend is performed by sending requests in JSON format. Then, the information delivered by the backend is presented to the user. To develop the frontend, it was used JavaScript and its libraries, particularly React. These tools let us develop an interactive and user-friendly page, necessary for a practical product. It was considered the client preferences for the interface, focusing on the presentation of the information required for each page, and implemented them.

The frontend is the more complex in number of files, but this organization will facilitate the development by having each file dedicated to a simpler function. Some features will only be available for the administrators, these are indicated with the word admin. Following, there are described some important files for the development of this interface.

Table 7 - Frontend Files

Admin Alerts	Responsible to show the management of alerts for the administrators.
Admin Oils	This tab will let the user add a new oil entry to the database. This will then be available to select on the tank descriptions and the oil used in the transfers.
Admin Page	This page is a tab pane that permits the administrator to choose the data that will be edited.
Admin Tank	In this tab, the administrator can create, edit, or delete a tank.
Admin Transfers	Here we can manage the transfers of oils done in the cooperative.
Admin users	User management.
Homepage	For now, it is just the reception page, but we can prepare it to show curiosities or news about the cooperative.
Login Page	This page will permit login/logouts in the system, the result of the actions on this page will influence the appearance of the site.
Tank Design	It is responsible for the tank presentation.
Tank Details	Tank Details is a page with the details of the selected tank, these are the history in volume, the oil inside and the alerts history of the tank.
Tanks	In this page we will be presented an interface with all the tanks with a system installed, it is the most important page in the

	program. The page will refresh every 2 minutes to present oscillations in the tanks (which might trigger alerts).

4.6 Full system overview

Finally, it is presented the full system model in Figure 24. Starting from the top left we can find the lasers connected to the ESP32's via cable using the I2C communication protocol. Next, a Bluetooth low energy network connects the ESP's, using a request/response method. The client ESP (on the right), requests the values to the server ESP's (on the left). These values are sent to the Raspberry Pi via USB. With the help of a router with internet connection, the Raspberry Pi sends the values to the ThingsBoard and these are saved in a different "dispositive" that represent the Laser which reads the distance value.

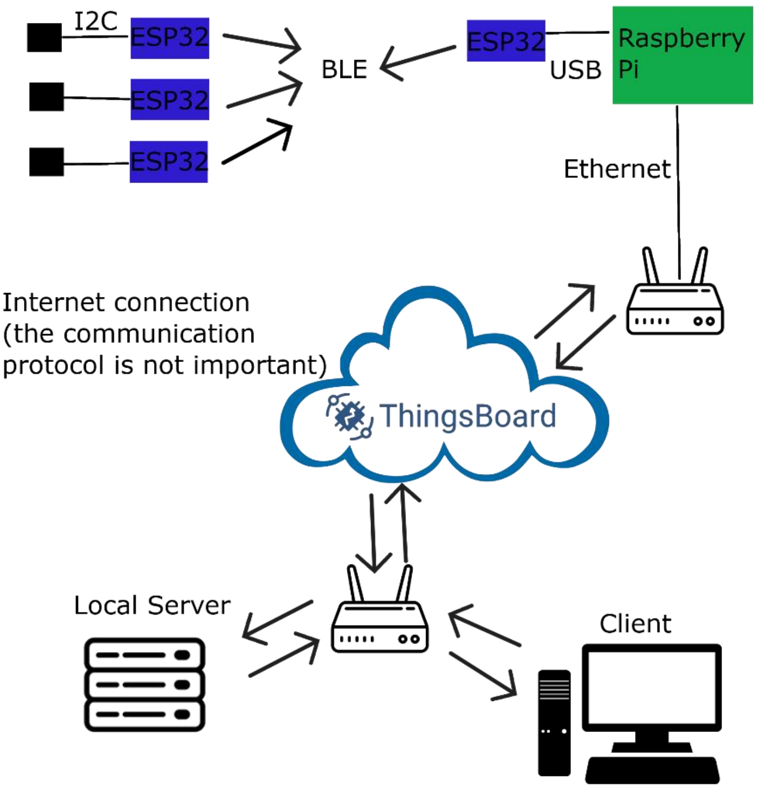


Figure 24 - Full system architecture

The second phase consists of the local server. An API is used to communicate with the ThingsBoard. The local server saves the other necessary data and processes the client requests/responses. The client is provided with a webpage with a friendly interface where he/she can access/manage data.

This system is very flexible, most parts are independent, which permits expansion, and it can easily change for other applications. Reusable software has a great impact on the development of new tools, it reduces time and resources investment by not requiring the "reinvent of the wheel". In this particular case, with a few changes on the ESP32's, the sensor and the interface, the application could read temperatures instead of distances, becoming a whole different product in spite of using mostly the same software. In another different scenario could also do both just by adding a new sensor and change the messages format (and later adapt the interface).

5
Tests and Results

5.1 Tests

In this section it is analyzed the result tests done within the development of this project. The tests were conducted for all levels progressively. At each new phase we advanced with the system, we will present the tests in the sequence which they were made.

- **Warehouse**

Inside the warehouse we face a closed building full of inox tanks and metal components. The olive oil is stored inside the tanks, which are around 10m height. The room is a big block of around (60 × 15)m area. The space between tanks is a thin rectangle from one limit to another following the 50m line. There is a network cable from the office to the warehouse that will be connected to the router with internet access. The room is equipped with 20 + 4 tanks, 20 of 10m height 120 000 liters and 4 of 8m height (Figure 25). The taller tanks are equipped with a tube (transparent in one side) needed to verify the filled height of the tank although smaller ones have a float locked in a cable that will move a metallic object outside the tank. Both methods are limited, in case of the tube, olive oil leaves debris that dirty the tube; in case of the float, administrators already verified that the metallic object sometimes would get stuck and not move at all.

Figure 25 – Olive Oil Warehouse

The access to the top of the tanks is very limited: they are very tall (almost to the limit of the ceiling); floor is slippery; there is no space for an elevator machine; it is not simple to mount an access structure. Liquid movements are made from tubes that are only installed when needed, these will connect to a pump and allow passage to where the oil is needed.

- **Testing the reading values**

The lidar LITE v3 was assembled to an Arduino Uno board and uploaded a fast cycle to the Arduino. It consisted of presenting the laser value each millisecond. This was meant to test the laser capacity for fast cycles as well as its variance. As expected from reading the laser sheet, it was verified that the values would vary +/-8 cm although the product sheet assert that the values may vary +/-10cm from the real distance. The laser was tested for different distances although all in the range of our application, 0.2m-12m. In bigger distances there was some limitations in space, so the tests started from pointing the laser to the roof in 3-4m distance, later using walls to test.

In this kind of tests, it was needed some carefulness with angles, in a 3m distance, a 10º angle will vary the value in ~5cm, $(1 - \cos(10) = 1.519\%)$ as illustrated in Figure 26. Although 5cm seems a small error, in a 12m distance it would be 20cm, which in our application could mean 2500 liters. Some of the variance we obtained during tests may have come from the lack of precision on the laser position.

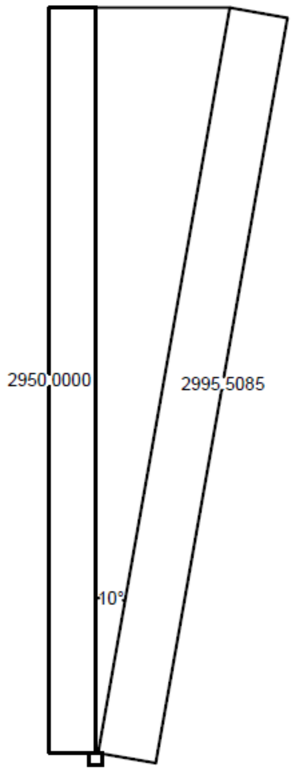


Figure 26 - 10º difference in distance (values in mm)

Later, it was discussed it would be better to have the laser installed inside a tube with a float inside, this would fix the laser position and the float would be made in a (near infra-red, 905nm wavelength) reflective material. To test how the laser would respond to the tube it was bought a 2” PVC pipe which worked very well as the laser would perfectly fit and stay in the right position. The results were great as the values would not float near as much as in open air, later it was assembled a very simple object to verify the floating idea. The test consisted of a carboy cut at the top with part of a PVC pipe stick with tape (Figure 27).

Figure 27 - Carboy with a taped PVC tube

Figure 28 - Float inside the PVC tube

With this small device we verified if the laser would follow the level of water inside and tested if the float would not get stuck. In Figure 28 it is shown the second float, the first one was slightly larger and would sometimes get stuck, the second float worked fine with the tests.

At some point we verified that sometimes the laser would stop responding and we would only be able to read "nack" (meaning error) on the console. Later, we realized it was due to a broken cable that would lose connection in some positions. Since we mounted a new cable, we never verified that same error again.

- **BLE network**

The first tests to experiment the BLE network were based on only 2 devices. There were some tests inside a building with walls. The results were good enough to proceed with the idea, we had some walls that would require signal reflection to communicate. There was noise in this path which would make it harder to detect the signal (Wi-Fi, GPRS, Bluetooth). The objective was to verify the communication reach, how it would respond with disconnecting/reconnecting, and if it were able to reach the destiny with obstacles.

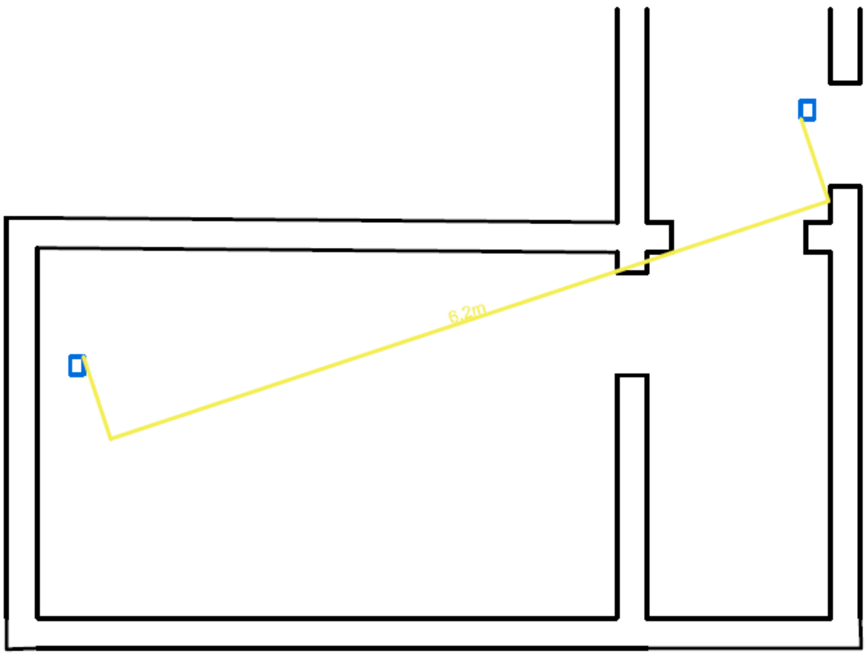


Figure 29 - Testing ESP32 reach with obstacles

Moving the right ESP32 upwards on the diagram would make it harder to make a connection, although sometimes it would be able to communicate. In the positions represented on the diagram there would always be a connection. In a straight line it can reach distances of 20+ meters which would be more than enough to proceed the experiment to the real application.

After the reaching tests there were added more devices to the network, the limit was 4 devices, 1 server and 3 clients the client would not be able to connect with a fourth server with the code developed. As it is also possible to install a laser on the client, this means we can have 4 deposits in each network.

Although we were able to communicate between devices, sometimes the ESP32 re-started (periods over 10 hours) and was not able to reconnect. Also, sometimes it randomly loses connection between devices. After some more testing we realized that the right sequence to assure it to connect would always pass by starting the server after all the clients were ready. To solve this problem, we decided to make a reboot sequence that would reboot after the occurrence of some errors, for example the disconnection of a client for more than 2 minutes. Although it is not the most efficient procedure, it works very fine, as communication losses rarely pass ten minutes. After some testing, it was opted to develop a software following the chart presented in Figure 30:

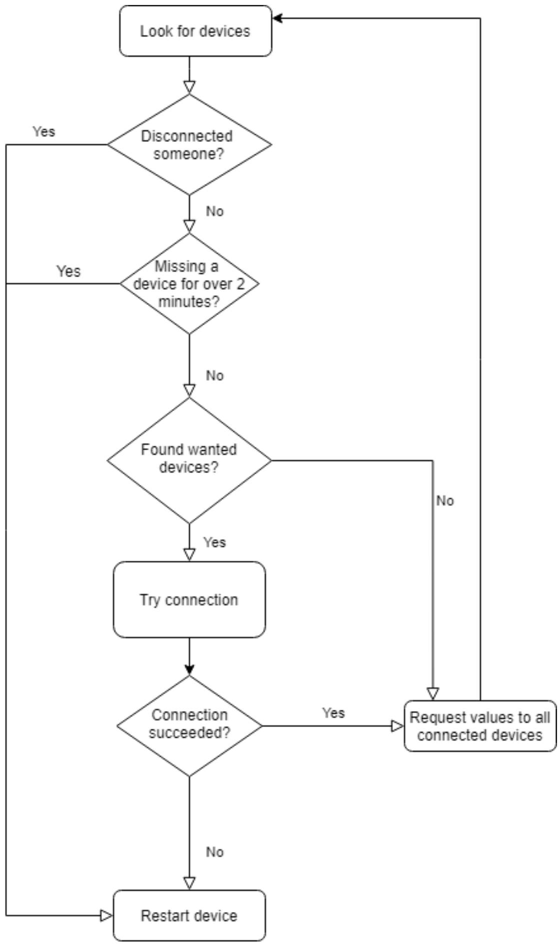


Figure 30 - ESP32 behavior sequence

- **Sending information to ThingsBoard**

As explained in section 4 of this work, the network (Bluetooth low energy network, ESP32's) data would be sent to a raspberry using an USB cable (which would also power the client ESP device). To read the information it was chosen to use the "serial" library. It was developed a cycle that would upload the data to ThingsBoard when receiving new information. The first connections were made using MQTT Paho library that is available for python language. It is quite simple to use as the user would only need to initiate the program with his ThingsBoard credentials and set a line to upload. This test was running for 3 weeks before it started giving problems. After a while, ThingsBoard stopped communicating and accepting new values. Verifying audit logs, it was concluded that this unsuccess was due to the high uploading rate that the tests were using, there were uploadings each 2 seconds which was extremely unnecessary for this type of data. It was then added a much bigger delay (10 minute) between updates that apparently solved the problem (the error did not appear during the 2-month testing period).

After some more time and experience, it was decided to change the MQTT Paho library for the tb_rest_client. This software seems more organized and complete. For the time being this is still the library used in the project.

Olive oil inside the deposits is mostly a stationary value and it is unnecessary to periodically save a constant number. When this technology matures, we will change the conditions to upload and adapt to the needs of the project. For the time being it was chosen to kee the periodic updates as we have two of the servers generating random values (Figure 31). These are used to verify the distance which we can communicate in the CABB's warehouse (as the values are random, the watchdog would not make sense).

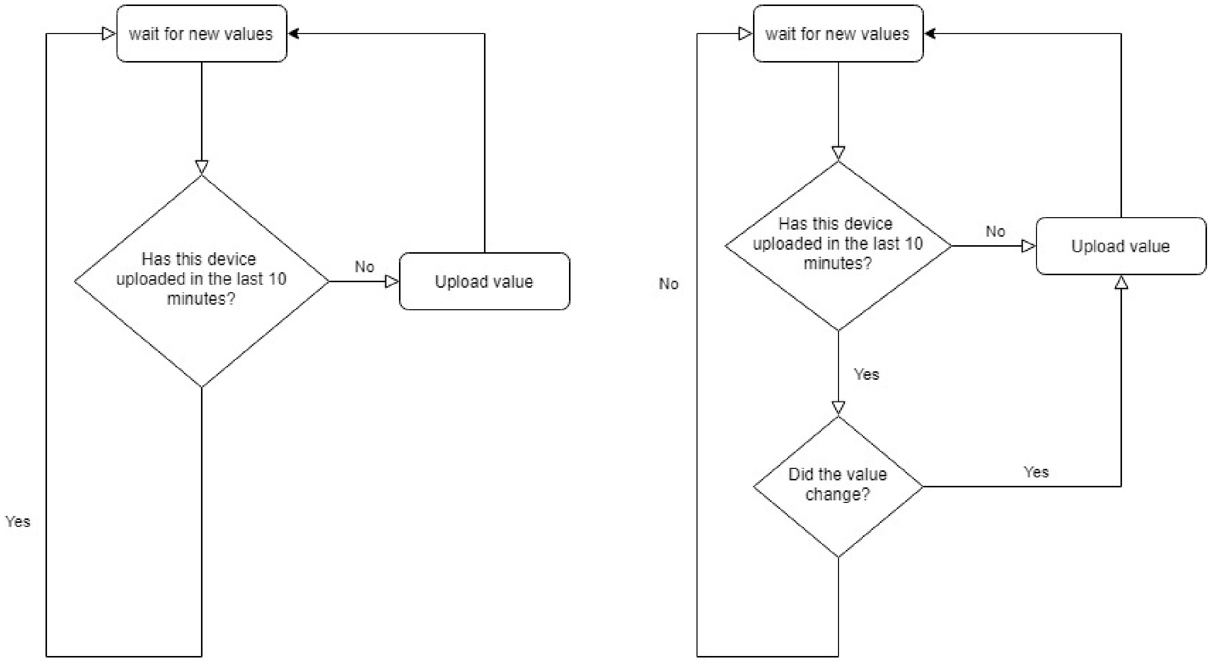


Figure 31 - Comparison of behaviors (installed vs intended)

Another throwback that was encountered during the test phase was the power supply of the CABB. During the testing period, CABB has been having some trouble with its switchboard that would casually shutdown the network and the Raspberry Pi. To bypass this problem, the raspberry was settled up to run a script that would initiate the program on its launch.

This also allowed us to verify when the problem occurred, we just had to navigate to ThingsBoard and verify the time which there had not been updates (using the ThingsBoard online dashboard tool).

- **Local server and ThingsBoard requests**

The local server started from being a simple site host that would request data from ThingsBoard to become a total platform with most of the data processing and analyzing. This processing and request handling capabilities became important to develop a product that made sense, and so there were a lot of tests between each phase. In the next table it is presented a little resume on the responsibilities of this part of the project.

Table 8 - Data base and methods

Data	**Data processing**
Users	Verification of users and their access to certain requests
Tanks	Verification of level, possible alerts, and the type of olive oil inside
Alerts	Notification in the principal page and registry in their tank
Olive oils	Characteristics of each oil
Transfers	Registry of transfers

- **Interface**

The interface did not have many limitations, only one important request: it should show all the list of tanks in the cooperative and change color dependently on the percentage filled. A suggestion mockup as how the user could visualize the tank's level is depicted in Figure 32.

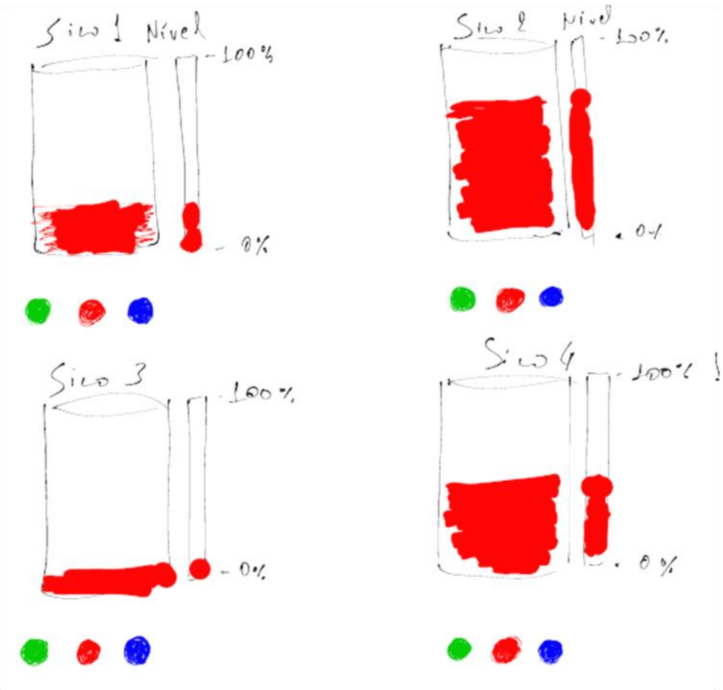


Figure 32 - Visualization of the tank's level mockup

Based on the mockup we came to one of the first iterations of the tanks page, shown in Figure 33. Each circle represents a tank, with this design we can present more tanks which was one of the client requests for the interface.

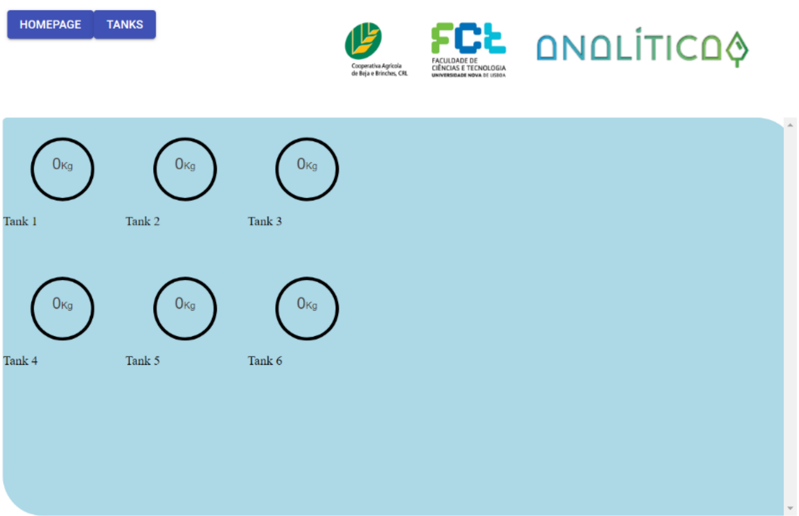


Figure 33 – First version of tanks page

At this point there were still many problems in design and functioning: the tanks would have to be added in the source code instead of being added by a user, the colors were not working, calls to ThingsBoard were not polished, there was still data missing, the webpage was not centered, and the dimensions would not change depending on the monitor dimensions, among others. Later iterations of the webpage solved most of these issues, the current application is presented in the following figures.

Figure 34 – Homepage

The home page (Figure 34) is very simple, it is only meant to welcome the user and present shortcuts for the remaining pages as well as to provide the links for the partnerships made in this work.

Figure 35 - Login page

In the login page (Figure 35), the user as the possibility to login and to logout. The tools provided to the user will differ on its user type, administrators are provided with more functionalities which are important for the cooperative, for example, the possibility to add a new tank, eliminate users, verify alerts, among others.

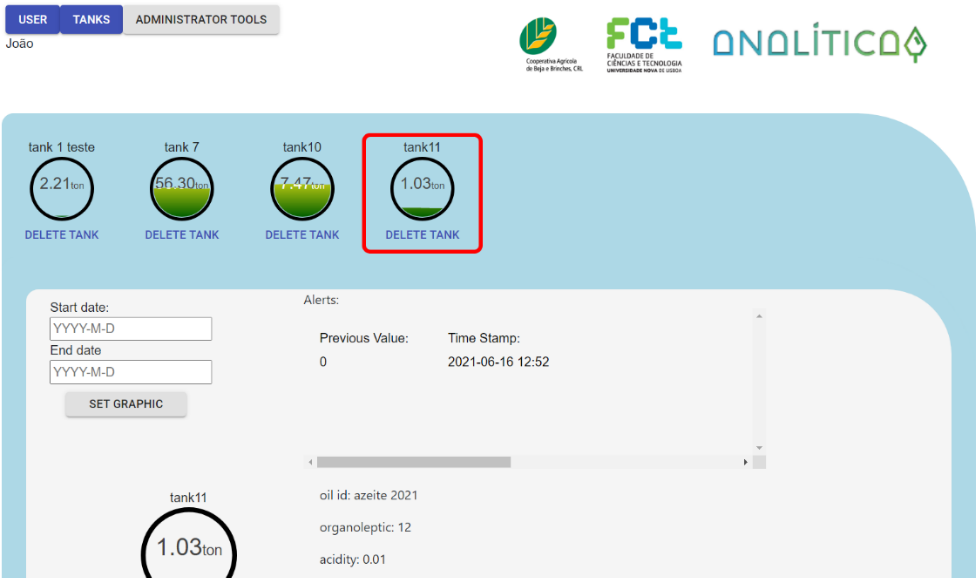


Figure 36 - Tank's page

In the tanks page (Figure 36), it is presented all the tanks registered in the platform. If well set up, they show the mass of product inside the tanks in tons. This measure unity was chosen by the manager of the cooperative (the quantity of olive oil is registed using mass instead of volume). The red square limiting the fourth tank means that there is an unverified alert related to that tank.

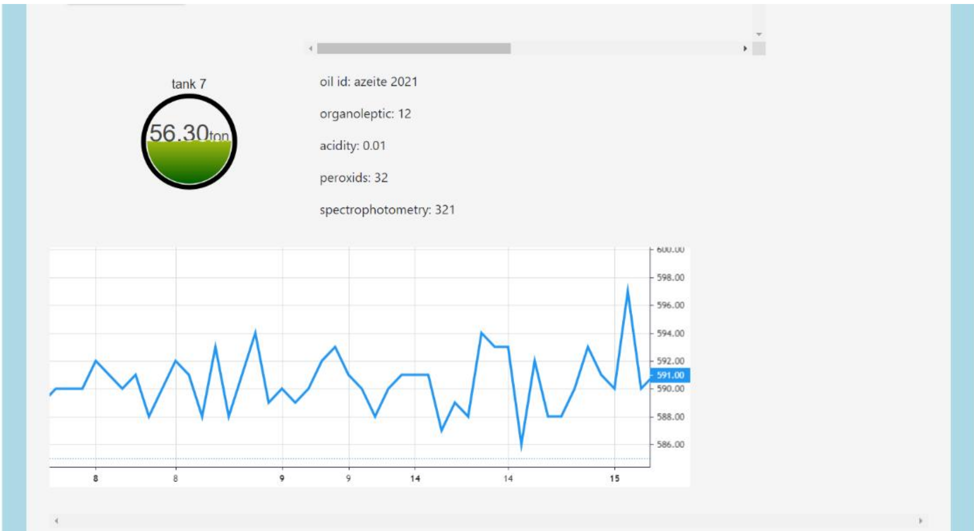


Figure 37 - Tanks page with a chart showing the level during a time interval

When clicking any of the tanks in that list, a smaller page will appear (grey page under the list), in this page there will be presented the characteristics of the olive oil inside (Figure 37). It is also possible to verify the history of level of product during a closed time interval.

- **Real case test**

It was installed a version of the system inside CABB, which has been running since 8th April 2021. We have been testing durability, reluctance to electrical problems, network capability, and reconnection issues.

Figure 38 - View from the top of the test tank

In Figure 38 it is shown the warehouse from the top of tanks, the four closer tanks are the reserve ones, and are the less volatile in terms of volume changing. As the risk was smaller, CABB used one of these to make the tests. In zone 1 we have the electrical devices used to test the system and in zone 2 (at ground level, 6-8meters lower than the image) there are 2 extra devices generating random values. These are being used to test the network resistance to the environment.

In Figure 39 we can check what is marked has zone 1 in Figure 38. Inside the box there is a raspberry pi connected via Lan, an ESP32 client connected via USB to the raspberry and another ESP32 server connected to a laser inside the tank via cable.

Figure 39 - Assembly on top of the principal tank

5.2 Discussion

When testing in the real field, we were faced with a new problem. Most days we verified that we would lose connection at around 7PM (only gaining access on the next day at 9AM on weekdays). On weekends we would not regain access which could be caused by 2 different problems, either the system would lose internet connection, or the electricity would go down. As we are using VNCviewer which let us use raspberry remotely, we verified that no matter which programs we opened, the next day all were closed, leading us to deduct that the electricity was cut during night periods and weekends.

This creates 2 problems; we need to reconnect the whole Bluetooth network; the software must be running at start. The microcontrollers software always starts with the booting so as long as they are able to reconnect, the system can work. For the raspberry it will be used its starting features to start the software with the booting. Crontab[4] let us schedule the start of a script. We tried using it with its "reboot" option but unsuccessful. Following some suggestions on the raspberry forums we found one that advises the user to instead try the "Auto-Start" document. This was successful and it was chosen as the way to go. This solution has been working since 13 April 2021, and to avoid sudden shutdowns, it was installed an UPS with some battery that can extend the periods in which the raspberry is working.

The overall testing results were very positive as the assemble worked together with no major problems. In each phase we had to retouch some loose ends clean up bugs and adapt the system to the real case use, which highly improved the capacity to adapt to other problems that were yet to come. We firmly believe that it is a good addition to the CABB.

The project is highly adaptable, and it has the chance to become a good tool for future projects. The technologies used were all opensource reducing the overall cost of the system. The combination of tools/technologies used to implement the project leave room for further development.

[4] Crontab is a program from Unix that lets the user schedule the execution of a command in the device.

6

Conclusions and future work

Every day IoT technologies and devices are increasing their influence in the industrial world, principally in monitoring applications. It is very important for small industries to keep up with technology as it can reduce costs and improve the efficiency of the systems.

Internet services are growing and becoming simpler every day. It has become an important tool as it works really efficiently and simplifies development of new systems. Although it works really efficiently and it is a great technology, it is required a whole other system to interface with the real world for then to be presented in the computer screen. However, this interface is not as easy to project and build, thus requiring some planning and studies, and this is the problem we are trying to fix in this investigation.

6.1 Synthesis of the developed work

The main objective of this work was to imagine, project, and install an IoT platform capable of monitoring the storage of olive oil. The development of this work is divided into 5 main phases:

- **Study of olive oil production, storage, and characteristics**
- **Review on monitoring technologies, sensors and IoT technologies**
- **Analysis of system requirements and possible architectures**
- **System development**
- **Tests and results**

Based in the study and after verifying technologies we rapidly realized that for this project we would focus mainly on volume control, the technology to evaluate acidity is mostly done in laboratory and temperature is not really a key factor on producing and storing this product.

In this specific application it was opted to use what we thought it was the best application, to test other technologies we would require much more investment. More accurate systems are expensive and may require other hardware/software to function, besides the installation does not need that much precision to perform.

With this work we can emerge the olive oil production in the technologic world and improve the performance of the production and storage. We built a central base with the possibility of extension to more complex features, meaning that there is still space for future improvement.

6.2 Suggestions for future work

- **Big Data analytics and Artificial intelligence**

Big data analytics has been gaining influence in the technology world as it is very important for Artificial intelligence. It can help us predicting future states making it a great tool to planning. In this particular project, it could help preventing food waste, which is one of the big contemporary themes.

- **Automatic transfer of fluids**

We intend to fully automatize the final system, this would minimize operator risks and would overcome human limitations on the process. It is one of the bigger steps as in this case, a bad test could cost a lot of olive oil loss. It is also one big investment, requiring electrical valves on every in/out tube.

- **More efficient network**

In the current state of installation, the Bluetooth low energy seems to be a good communication protocol, but in further development it might be required a faster communication with fewer reconnections throughout the time, particularly in case of developing the automatic transfer of fluids.

- **Extension of the system**

For further development and a better system, it would be good if at least we could have 4 lasers working in the planned state. This would permit us to test the setup one step further and transform the system into something that CABB could really rely on.

- **Better interface and extra features (if wanted/necessary)**

The interface was built on user preference, but as it took much time to install the real interface in the CABB network (due to the pandemic constraints) it was not possible to fully adjust the interface with the administrator's feedback. It is very important that the system is adapted to the user, or it may be much less valuable to the system.

- **Deploying the project on the internet**

In future iterations of the server/interface, it might be worth to deploy the system in an online server. This would get us another type of power over the system and extend its use.

For now, it is possible to get the olive oil height values online, but the users/local server is only hosted in a local platform which limits the online features (This will also be required to review security features and adapt to the intended scenario).

- **Adding other sensors**

In future it could also add new sensors to network without changing much of the already developed platforms, this system was developed modularly and left space for improvement and extensibility. It is quite easy to add a new sensor to the ESP32s such as a temperature meter or a humidity sensor. We would then just need to adapt the network by changing the messages format and adapt their processing in the central base.

BIBLIOGRAPHY

"5 Musical Instruments You Can Build With a Raspberry Pi." n.d. Accessed September 8, 2020. https://www.makeuseof.com/tag/pi-musical-instruments/.

Abdullah, Ali Najim. 2017. "Online Monitoring and Controlling Water Plant System Based on IoT Cloud Computing and Arduino Microcontroller," no. 4: 1230–40.

Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications." *IEEE Communications Surveys and Tutorials* 17 (4): 2347–76. https://doi.org/10.1109/COMST.2015.2444095.

Antunes, Paulo, João Dias, Tiago Paixão, Esequiel Mesquita, Humberto Varum, and Paulo André. 2015. "Liquid Level Gauge Based in Plastic Optical Fiber." *Measurement: Journal of the International Measurement Confederation* 66 (April): 238–43. https://doi.org/10.1016/j.measurement.2015.01.030.

"API Documentation & Design Tools for Teams | Swagger." n.d. Accessed June 16, 2021. https://swagger.io/.

"Application of SONAR in Ships | Download Scientific Diagram." n.d. Accessed May 28, 2020. https://www.researchgate.net/figure/Figure-Application-of-SONAR-in-ships_fig1_325895659.

"Associações Filiadas | Cooperativa Agrícola de Beja e Brinches, CRL." n.d. Accessed September 6, 2021. http://www.faaba.pt/associacoes-filiadas/30.

Azmi, Noraini, and Latifah Munirah Kamarudin. 2017. "Enabling IoT: Integration of Wireless Sensor Network for Healthcare Application Using Waspmote." In *AIP Conference Proceedings*, 1808:020010. American Institute of Physics Inc. https://doi.org/10.1063/1.4975243.

Baldo, M. Antonietta, Paolo Oliveri, Sabrina Fabris, Cristina Malegori, and Salvatore Daniele. 2019. "Fast Determination of Extra-Virgin Olive Oil Acidity by Voltammetry and Partial Least Squares Regression." *Analytica Chimica Acta* 1056 (May): 7–15. https://doi.org/10.1016/j.aca.2018.12.050.

Broy, Manfred, María Victoria Cengarle, and Eva Geisberger. 2012. "Cyber-Physical Systems: Imminent Challenges." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7539 LNCS:1–28. https://doi.org/10.1007/978-3-642-34059-8_1.

"Build Your Own Weather Station - Introduction | Raspberry Pi Projects." n.d. Accessed September 8, 2020. https://projects.raspberrypi.org/en/projects/build-your-own-weather-station.

Bunch, Chris, Navraj Chohan, Chandra Krintz, Jovan Chohan, Jonathan Kupferman, Puneet Lakhina, Yiming Li, and Yoshihide Nomura. 2010. "An Evaluation of Distributed Datastores Using the Appscale Cloud Platform." In *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, 305–12. IEEE. https://doi.org/10.1109/CLOUD.2010.51.

Camarinha-Matos, Luis. 2010. "ICT INFRASTRUCTURES FOR COLLABORATION."

Caponio, Francesco, Maria Teresa Bilancia, Antonella Pasqualone, Ewa Sikorska, and Tommaso Gomes. 2005. "Influence of the Exposure to Light on Extra Virgin Olive Oil

Quality during Storage." *European Food Research and Technology* 221 (1–2): 92–98. https://doi.org/10.1007/s00217-004-1126-8.
Celesti, Antonio, Antonino Galletta, Lorenzo Carnevale, Maria Fazio, Aime Lay-Ekuakille, and Massimo Villari. 2018. "An IoT Cloud System for Traffic Monitoring and Vehicular Accidents Prevention Based on Mobile Sensor Data Processing." *IEEE Sensors Journal* 18 (12): 4795–4802. https://doi.org/10.1109/JSEN.2017.2777786.
Chetpattananondh, K., T. Tapoanoi, P. Phukpattaranont, and N. Jindapetch. 2014. "A Self-Calibration Water Level Measurement Using an Interdigital Capacitive Sensor." *Sensors and Actuators, A: Physical* 209 (March): 175–82. https://doi.org/10.1016/j.sna.2014.01.040.
Chien Thang, Vu. 2018. "Computer Network and Information Security." *Computer Network and Information Security* 8: 44–50. https://doi.org/10.5815/ijcnis.2018.08.05.
Desai, Nitin Sadashiv, and John Sahaya Rani Alex. 2017. "IoT Based Air Pollution Monitoring and Predictor System on Beagle Bone Black." In *2017 International Conference On Nextgen Electronic Technologies: Silicon to Software, ICNETS2 2017*, 367–70. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICNETS2.2017.8067962.
"Getting Started with the Arduino - Controlling the LED (Part 1)." n.d. Accessed September 8, 2020. https://www.circuitbasics.com/arduino-basics-controlling-led/.
"Getting Started with the ESP32 on Arduino IDE [Full Guide 2020]." n.d. Accessed April 15, 2021. https://electropeak.com/learn/getting-started-with-the-esp32/.
Grossi, M., G. Di Lecce, T. Gallina Toschi, and B. Riccò. 2014. "A Novel Electrochemical Method for Olive Oil Acidity Determination." *Microelectronics Journal* 45 (12): 1701–7. https://doi.org/10.1016/j.mejo.2014.07.006.
Hejazi, Hamdan, Husam Rajab, Tibor Cinkler, and Laszlo Lengyel. 2018. "Survey of Platforms for Massive IoT." In *2018 IEEE International Conference on Future IoT Technologies, Future IoT 2018*, 2018-January:1–8. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/FIOT.2018.8325598.
Holloway, Michael D., Chikezie Nwaoha, and Oliver A. Onyewuenyi. 2012. *Process Plant Equipment: Operation, Control, and Reliability*. *Process Plant Equipment: Operation, Control, and Reliability*. John Wiley and Sons. https://doi.org/10.1002/9781118162569.
Ingle, Shantanu, and Madhuri Phute. 2016. "Tesla Autopilot : Semi Autonomous Driving, an Uptick for Future Autonomy." *International Research Journal of Engineering and Technology*. www.irjet.net.
"IoT in Action - Towards Data Science." n.d. https://towardsdatascience.com/iot-in-action-a8b7fac83619.
Kocovic, Petar, Muthu Ramachandran, Reinhold Behringer, and Radomir Mihajlovic. 2017. *Emerging Trends and Applications of the Internet of Things*. *Emerging Trends and Applications of the Internet of Things*. IGI Global. https://doi.org/10.4018/978-1-5225-2437-3.
Kumar, Brajesh, and Nirupama Mandal. 2016. "Study of an Electro-Optic Technique of Level Transmitter Using Mach-Zehnder Interferometer and Float as Primary Sensing Elements." *IEEE Sensors Journal* 16 (11): 4211–18. https://doi.org/10.1109/JSEN.2016.2544960.
"Lidar Lite v3 Operation Manual and Technical Specifications." n.d.
Liu, Yang, Yu Peng, Bailing Wang, Sirui Yao, and Zihe Liu. 2017. "Review on Cyber-Physical Systems." *IEEE/CAA Journal of Automatica Sinica* 4 (1): 27–40. https://doi.org/10.1109/JAS.2017.7510349.
"Magnetic Liquid Level Gauges Working Principle - Level Measurement - Instrumentation Forum." n.d. Accessed June 18, 2020. https://instrumentationforum.com/t/magnetic-liquid-level-gauges-working-principle/5242.
Mahan, James R., Warren Conaty, James Neilsen, Paxton Payton, and Stephen B. Cox. 2010. "Field Performance in Agricultural Settings of a Wireless Temperature Monitoring

System Based on a Low-Cost Infrared Sensor.” *Computers and Electronics in Agriculture* 71 (2): 176–81. https://doi.org/10.1016/j.compag.2010.01.005.
Maher, Anthony, Vijayalakshmi Velusamy, Daniel Riordan, and Joseph Walsh. n.d. “Modelling of Temperature Coefficient of Resistance of a Thin Film RTD Towards Exhaust Gas Measurement Applications.”
Mankar, Jayant, Chaitali Darode, Komal Trivedi, Madhura Kanoje, and Prachi Shahare. 2014. “REVIEW OF I2C PROTOCOL.” *International Journal of Research in Advent Technology* 2 (1). http://www.ijrat.org.
Mekki, Kais, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. 2018. “Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT.” In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018*, 197–202. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/PERCOMW.2018.8480255.
Miladinovic, Igor, and Sigrid Schefer-Wenzl. 2018. “NFV Enabled IoT Architecture for an Operating Room Environment.” In *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, 2018-January:98–102. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/WF-IoT.2018.8355128.
“MINNOWBOARDMAX-DUAL Reference Design | Application Processor | Arrow.Com.” n.d. Accessed September 8, 2020. https://www.arrow.com/en/reference-designs/minnowboardmax-dual-meet-minnowboard-max-based-on-atom-e3825-series-soc-dual-core/0743729e400e59218c0bb2c6678f9f64.
Mlakić, Dragan, Srete Nikolovski, and Emir Alibašić. 2017. “Designing Automatic Meter Reading System Using Open Source Hardware and Software.” *International Journal of Electrical and Computer Engineering (IJECE)* 7 (6): 3282. https://doi.org/10.11591/ijece.v7i6.pp3282-3291.
“Motor Control Using NodeMCU - Arduino Project Hub.” n.d. Accessed September 8, 2020. https://create.arduino.cc/projecthub/shubhamsantosh99/motor-control-using-nodemcu-9582d3.
Nikolov, Georgi Todorov, Boyanka Nikolova, and Georgi Nikolov. 2008. “Virtual Techniques for Liquid Level Monitoring Using Differential Pressure Sensors Design and Development View Project Geothermal and Solar Systems-VET-TU-Sofia View Project VIRTUAL TECHNIQUES FOR LIQUID LEVEL MONITORING USING DIFFERENTIAL PRESSURE SENSORS.” Vol. 9. https://www.researchgate.net/publication/264848936.
Ornes, Stephen. n.d. “The Internet of Things and the Explosion of Interconnectivity.” Accessed April 8, 2020. https://doi.org/10.1073/pnas.1613921113.
Pawar, Rohit R., Priyanka A. Wagh, and S. B. Deosarkar. 2018. “Distribution Transformer Monitoring System Using Internet of Things (IoT).” In *ICCIDS 2017 - International Conference on Computational Intelligence in Data Science, Proceedings*, 2018-January:1–4. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICCIDS.2017.8272671.
“PORDATA - Lagares e Produção de Azeite - Continente.” n.d. https://www.pordata.pt/Portugal/Lagares+e+produção+de+azeite+++Continente-3457.
“Python Web Development with Django - Jeff Forcier, Paul Bissex, Wesley J Chun - Google Books.” n.d. Accessed April 28, 2021. https://books.google.pt/books?hl=en&lr=&id=M2D5nnYlmZoC&oi=fnd&pg=PT31&dq=django&ots=vZTMJqaPMT&sig=NUIkVf7K5DqE6aj2iELZCec5lD4&redir_esc=y#v=onepage&q=django&f=false.
Russom, Philip, and Tdwi Org. 2011. “BIG DATA ANALYTICS FOURTH QUARTER 2011 TDWI RE SE A RCH Co-Sponsored by BIG DATA A N A LY TIC S FOURTH QUARTER 2011 TDWI BEST PRACTICES REPORT Introduction to Big Data Analytics.”

“SaaS vs PaaS vs IaaS: What’s The Difference and How To Choose – BMC Blogs.” n.d. Accessed June 11, 2020. https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/.

Scheffer-Dutra, C. B., A. Núñez-Reyes, and C. Bordons. 2002. “Predictive Control of an Olive Oil Mill with Multi-Objective Prioritization.” *IFAC Proceedings Volumes (IFAC-PapersOnline)* 35 (1): 85–90.

“Sensors Modules Thermocouple | Sensors Modules.” n.d. Accessed June 20, 2020. https://www.electronicwings.com/sensors-modules/thermocouple.

Souilem, Safa, Abdelilah El-Abbassi, Hajar Kiai, Abdellatif Hafidi, Sami Sayadi, and Charis M. Galanakis. 2017. “Olive Oil Production Sector: Environmental Effects and Sustainability Challenges.” In *Olive Mill Waste: Recent Advances for Sustainable Management*, 1–28. Elsevier Inc. https://doi.org/10.1016/B978-0-12-805314-0.00001-7.

Souza, Rodrigo, João Lopes, Cláudio Geyer, Anderson Cardozo, Adenauer Yamin, and Jorge Barbosa. n.d. “An Architecture for IoT Management Targeted to Context Awareness of Ubiquitous Applications.”

“SQLite Home Page.” n.d. Accessed April 17, 2021. https://www.sqlite.org/index.html.

“Static vs Dynamic Website: What Is the Difference?” n.d. Accessed April 28, 2021. https://wpamelia.com/static-vs-dynamic-website/.

“The Web Framework for Perfectionists with Deadlines | Django.” n.d. Accessed April 17, 2021. https://www.djangoproject.com/.

Wang, Sheng Wei, Chen Chia Chen, Chieh Ming Wu, and Chun Ming Huang. 2018. “A Continuous Water-Level Sensor Based on Load Cell and Floating Pipe.” In *Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018*, 151–54. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICASI.2018.8394554.

“What Is Cloud Computing.” n.d. https://aws.amazon.com/what-is-cloud-computing/.

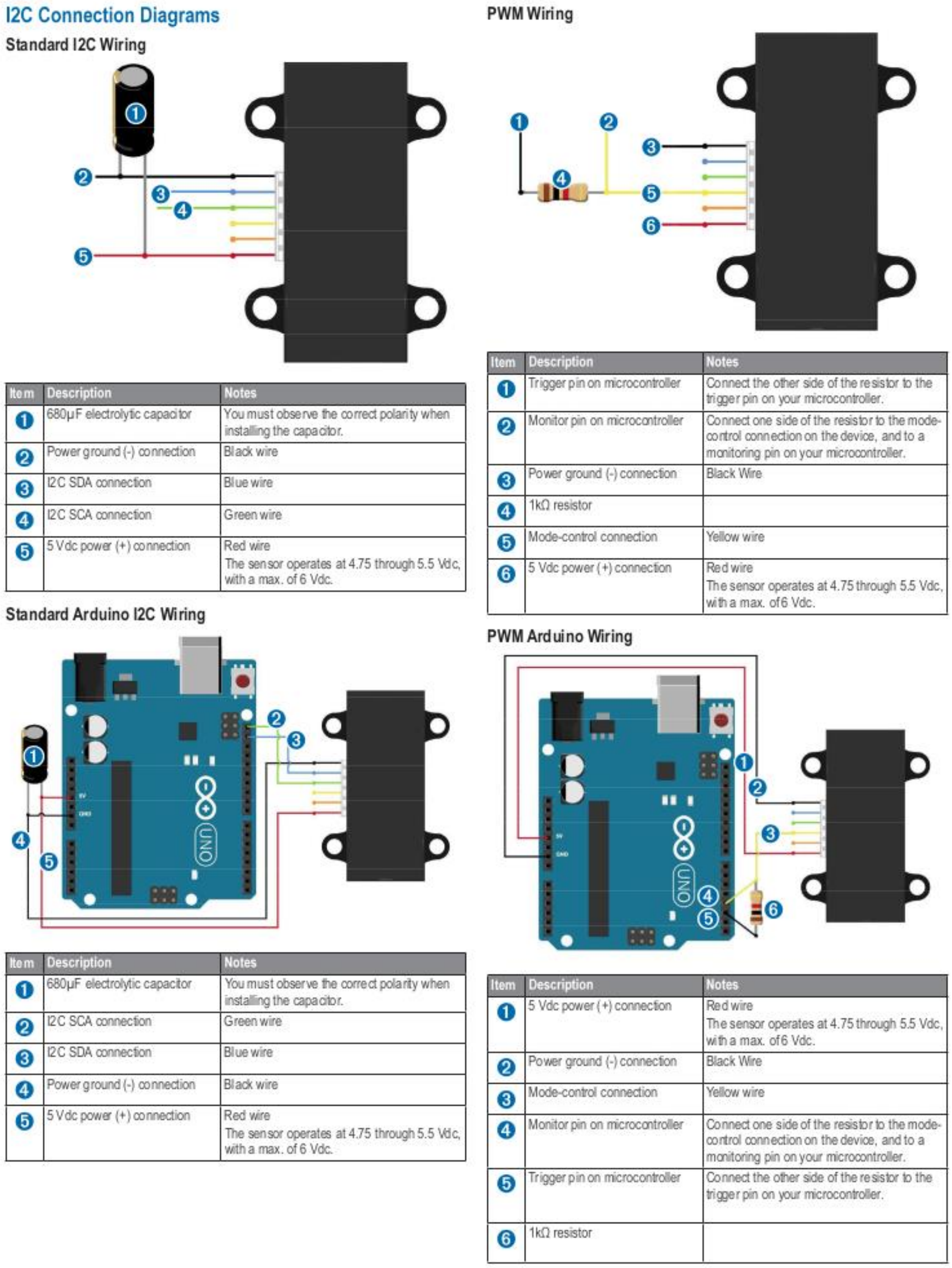
“What Is Infrared Radiation? IBT Informs! - Infrabiotech.” n.d. Accessed June 22, 2020. https://www.infrabiotech.de/en/technology-243/infrared-radiation.html.

“What Is ThingsBoard? | ThingsBoard.” n.d. Accessed May 1, 2021. http://thingsboard.atjiang.com/docs/getting-started-guides/what-is-thingsboard/.

Yashiro, Takeshi, Takayuki Ogawa, and Hiroyuki Sasahara. 2013. “Temperature Measurement of Cutting Tool and Machined Surface Layer in Milling of CFRP.” *International Journal of Machine Tools and Manufacture* 70 (July): 63–69. https://doi.org/10.1016/j.ijmachtools.2013.03.009.

“Zigbee Wireless Networking - Drew Gislason - Google Books.” n.d. Accessed June 25, 2020. https://books.google.pt/books?hl=en&lr=&id=up8Oa7456I8C&oi=fnd&pg=PP1&dq=Zigbee&ots=IFKr1-yQzo&sig=FLv6k4f1mUrgOSD92lo9Ay45cf0&redir_esc=y#v=onepage&q=Zigbee&f=false.

Annexes



I2C Connection Diagrams

Standard I2C Wiring

Item	Description	Notes
1	680µF electrolytic capacitor	You must observe the correct polarity when installing the capacitor.
2	Power ground (-) connection	Black wire
3	I2C SDA connection	Blue wire
4	I2C SCA connection	Green wire
5	5 Vdc power (+) connection	Red wire The sensor operates at 4.75 through 5.5 Vdc, with a max. of 6 Vdc.

Standard Arduino I2C Wiring

Item	Description	Notes
1	680µF electrolytic capacitor	You must observe the correct polarity when installing the capacitor.
2	I2C SCA connection	Green wire
3	I2C SDA connection	Blue wire
4	Power ground (-) connection	Black wire
5	5 Vdc power (+) connection	Red wire The sensor operates at 4.75 through 5.5 Vdc, with a max. of 6 Vdc.

PWM Wiring

Item	Description	Notes
1	Trigger pin on microcontroller	Connect the other side of the resistor to the trigger pin on your microcontroller.
2	Monitor pin on microcontroller	Connect one side of the resistor to the mode-control connection on the device, and to a monitoring pin on your microcontroller.
3	Power ground (-) connection	Black Wire
4	1kΩ resistor	
5	Mode-control connection	Yellow wire
6	5 Vdc power (+) connection	Red wire The sensor operates at 4.75 through 5.5 Vdc, with a max. of 6 Vdc.

PWM Arduino Wiring

Item	Description	Notes
1	5 Vdc power (+) connection	Red wire The sensor operates at 4.75 through 5.5 Vdc, with a max. of 6 Vdc.
2	Power ground (-) connection	Black Wire
3	Mode-control connection	Yellow wire
4	Monitor pin on microcontroller	Connect one side of the resistor to the mode-control connection on the device, and to a monitoring pin on your microcontroller.
5	Trigger pin on microcontroller	Connect the other side of the resistor to the trigger pin on your microcontroller.
6	1kΩ resistor	

Figure 40 - Lidar lite v3 datasheet

2021

João António Ferreira
Mourão Cartaxo

Development of an IoT platform to monitor storage conditions and packaging optimization in industry environment

NOVA SCHOOL OF SCIENCE & TECHNOLOGY