Universidad de Extremadura

Escuela Politécnica

Máster en Ingeniería Informática

Trabajo Fin de Máster

Sensores para la medición de la capacidad de infiltración de suelos

Ismael Flores Pérez

Junio 2018

Universidad de Extremadura

Escuela Politécnica

Máster en Ingeniería Informática

Trabajo Fin de Máster

Sensores para la medición de la capacidad de infiltración de suelos

Autor: Ismael Flores Pérez

Fdo:

Director: D. Pablo García Rodríguez

Fdo:

Director: D. Pablo Durán

Fdo:

Tribunal Calificador

Presidente:–

Secretario:–

Vocal:–

Junio, 2018.

A Juan Flores Morán

"El propósito de la computación es la comprensión, no el obtener números."
-Richard Hamming-

Acknowledgments

- A mis padres, por aguantar mis situaciones de estrés, respaldar las decisiones que he llevado a cabo a lo largo de mi vida y aportar su punto de vista a la hora de afrontar los problemas que surgen en la vida.
- A mi hermana, por ser mi consejera y mostrar su constante apoyo en las decisiones que he llegado a tomar.
- A Bea, por ser la persona con la que comparto mis alegrías y los momentos más difíciles.
- A Pablo García, por ser mi mentor y brindarme oportunidades que han sido irrechazables.
- A Pablo Durán, por alimentar mi curiosidad día tras día, enseñándome aplicaciones de la propia *Ingeniería*.

Abstract

This document includes a study of the behavior of different materials using an Information System based on standard humidity sensors. The objetive of this study is determining if it is possible to use low cost solutions that could register information related to the behavior of different kind of soils, having variations in compaction processes and environmental factors.

The architecture of the designed Information System and its Software and Hardware components are defined. The objective of this study is to apply and to analyze different and newer technologies existing in Information Engineering on different studies made by other Engineerings.

This document evaluates diverse software methodologies to apply to develop the purpose solution. All the methodologies includes information about their main features, considering if they are appropriate to satisfy the requirements of this project.

The results obtained by the Information System are analyzed. An analysis is done to evaluate if the chosen solution is applicable to this study.

An annexed is included in this document. This annexed explains how to work with the developed Information System and it concludes with the required operations to maintain it. In the other hand, there is a section dedicated to explain most common issues.

Keywords:
Information System, NoSQL, Big Data, Arduino, Raspberry Pi, Humidity Sensor, MongoCheff, 3TStudio, Python, Ubuntu Server, NodeJS, C, Software, Hardware, MongoDB

Resumen

Este documento recoge el estudio del comportamiento de diversos materiales mediante el uso de un Sistema de Información basado en la utilización de sensores de humedad. El objetivo a alcanzar en este estudio es determinar si es factible optar por soluciones de bajo coste que registren información del comportamiento de distintos sustratos ante variaciones en procesos de compactación y factores ambientales.

Se define la arquitectura del sistema de información que ha sido diseñado junto con sus respectivos componentes *Hardware* y *Software*. La filosofía de este trabajo extrapola las nuevas tecnologías existentes en el campo de la Ingeniería Informática en estudios realizados por otras ingenierías.

En este documento también se evaluan las distintas metodologías de desarrollo software aplicables para desarrollar la solución propuesta. En cada metodología de desarrollo software se recogen principales características teniendo en cuenta si se adaptan a las necesidades requeridas por este proyecto.

Se realiza un análisis de los resultados obtenidos mediante el Sistema de Información desarrollado. Con este análisis se evalua si la solución escogida es apropiada para el estudio propuesto.

Al final del documento se adjunta un anexo con la información necesaria para trabajar con el Sistema de Información junto con las distintas operaciones a desempeñar si fuese necesario realizar tareas de mantenimiento o resolución de problemas comunes en el propio sistema.

Table of Contents

List of figures

Chapter 1

Introduction

This work studies the measuring reliability of an *Information System* based on low cost sensors, using *free Hardware with Arduinos* [1, 2] boards and ARM-based devices [3, 4].

1.1. Motivations

The main reason of the study is to apply Informatics Engineering solutions to innovate on other engineering fields. Moreover, it offers an opportunity to experiment with different technologies in real test environments. Those innovations may offer advantages such as the optimization of processes via automation.

In our case, this advantage is accomplished via a fast interpretation of results, avoiding mathematical calculations and deprecated tools. This leads to the minimization of the time needed to obtain results. Thus, speeding up breakthroughs in the state-of-the-art.

1.2. Objectives

The main objectives of our work are enumerated in this Section:

1. ***Apply Informatics Engineering knowledge to solve real life problems***.
2. ***Estimate the domain of the problem to be solved***.
3. ***Consider and analyse different technological alternatives***. Thus, finding an improved solution to a problem.
4. ***Design Information Systems useful for the automation of processes***.

1.2.1. Secondary objectives

The secondary objectives treated in this project are described in the following points:

1. ***Use technologies applied in BigData [5–7]*** e.g. NoSQL [8–10] databases, which are interesting when dealing with high dimension and non-structured data.

2. ***Study new database management tools***. This allows to improve the management of high dimension databases.

3. ***Apply programming knowledge to sensor networks***.

4. ***Manage Unix systems***.

1.3. Organization of the memory

This document comprises 8 chapters and a bibliography section.

Chapter 1 offers and introduction to the selected study and the criteria for its design and components.

Chapter 2 presents the state-of-the-art.

Chapter 3 describes what methods are available to gather requirements according to the stakeholders' needs and presents which method was chosen for this project.

The design of the *Information System* [11, 12] is described in Chapter 4.

The materials and methods used for this work are addressed in Chapter 5.

The results of the study are shown and analysed in Chapter 6.

Finally, Chapter 7 enumerates the conclusions of the study and presents its future work.

Chapter 2

State of the art

This chapter explains techniques and studies related to this work. The techniques explained in this chapter are related to *Big Data*, information systems made with simple microcontrollers and Internet of Things (IoT) projects.

2.1. Big Data studies

Big Data is a recent technology based on systems which require huge amount of information. The information produced in Big Data systems refers to sensors and machines. These sensors and machines automatically produce information which is stored in *Information Systems*. The information Systems have resources to store huge amount of information, and also, can analyse the information stored to present results and predictions in **Real Time**.

There are studies which use this technology. Also, there are companies which use this technology as a business strategy. Some of them, are explained in this section:

- Google constantly develops new products and services that have Big Data algorithms. Google uses Big Data to refine its core search and ad-serving algorithms.
- Autonomous driving. Companies involve in Autonomous Driving use Big Data to design algorithms. These algorithms recognize objects, people and also, take information from the different external factors that have influence on driving experience [**?**].
- Energy Future Holdings Corporation is an electric utility company. The majority of the company's power generation is through coal- and nuclear-power plants. The company used Big data to install smart meters. The smart meters allows the provider to read the meter once every 15 minutes rather than one month.
- McLaren Racing Limited is a British Formula One team. The racing car team uses real-time car sensor data during car races, identifies issues with its racing

cars using predictive analytics and takes corrective actions pro-actively before it is too late.

- Verizon uses Big Data to enhance mobile advertising. A unique identifier is created when the user registers in the website. The identifier allows advertiser to use information from the desktop computer. Marketing messages can be delivered to you mobile phone using this information.

2.2. Information Systems based on microcontrollers

During last years, microcontrollers became more important on *Information Systems*. They have important features and can offer control to different devices to get information. Obtaining information from systems is one of the most important epics in Information Systems.

There are several studies where microcontrollers are used in Information Systems. Most important studies are explained in this section:

- Smart Room Temperature Controller Atmega [13]. This project is used as an alternative to control the temperature of a room. It measures the temperature of a room and notify the user when the threshold is reached.

- Automatic water plants [14]. The developer uses a microcontroller to control when it is needed to water plants. It has a designed circuit for its specific purpose.

- An Automated Metrics System to Measure and Improve the Success of Laboratory Automation Implementation [15]. It is an application which collects information from a distributed system. The data has collected over 20 months from at least 28 different workstations. This system prints charts and generates analysis with the stored information.

2.3. First studies of compaction methods

2.4. Proctor compaction test

2.5. Variables that affect the compaction of floors

2.6. Applications of the compaction processes

In this section it will be explained the main applications of the compaction of floors. It is important to say that in this study it is not pretended to use this technique for all its applications. In this study it is needed to use this technique to get more accurate measure of different floors.

The main applications of this technique are described in the following points:

- Increasing the density and the weight of a floor.
- Increasing load carrying capacity.
- Preventing soil from sinking
- Increasing the resistance of soils.
- Reducing water runoff.

2.7. Objectives of the compaction

This section, explains the necessary to compact floors and the main objectives found with this technique.

This process looks for a better performance in floors extracting air particles contained in the materials that form part of them. Compaction can also be applied to other processes. This project uses this technique to study how works the drying process in floors.

If we have this information about floors, we can also get information about how is the behaviour of a floor when the ambient variables are presented. It allows to get information about how to use some materials with adverse conditions. It also allows us to choose the best materials for determined conditions.

In this study this technique was used for the following purposes:

- Distributing the humidity in a floor.
- Extracting the air contained in a floor.
- Obtaining more accurate measures.

2.7.1. ...

Chapter 3

Requirements

This chapter exposes the requirements acquired after analysing the needs of the users. This analysis is required before the design of the system.

This analysis implies to organize several meetings to obtain the needs of the the users that were going to use the system. Listen to the needs of the client is required to look for a good solution to the main problem.

The solution was obtained after doing a good analysis of the tools available to develop the Information System. This solution is not unique, there are more available solutions, but in this project, we decided to developed the purposed solution because it was affordable in terms of time and budget.

3.1. User requirements

This section explains the process followed to obtain information about the needs of the user. Here it is explained how the process was made and the steps that we followed to get the information to develop the purpose system.

To do this process, it was needed to organize meetings with one of the potential users of the system. Several meetings were needed to obtain all the requirements of the system. There are an amount of questions to obtain the requirements of the users. These questions are described in the following points:

- *What does the user need?* This is one of the most important things we need to ask to the user. We need to know exactly the desires of the user to check the requirements and, after that, decide if the project is affordable or not.

- *Who is going to use the system?* This question is needed to obtain the profiles of the users that are going to use the system. It is important to know if the user has experience with systems similar to the chosen solution. With this information we have an idea about the knowledge of the users and how must be the system for the different users that are going to use it.

- *When does the user want to get the results?* This question is used when there is a limited requirement related to time. In this case, there is a limit of time established by the user. The user needs to obtain the information as soon as possible.

- *How does the user want the results?* The user explains how must be the results. The user needs to manage the obtained information and it has to follow the requirements defined in the first step of the project . Also, the format to display the information is considered.

- *Where does the user want to have access to the system?* Here, it is defined if the system has to be accessible from everywhere or not.

- *Why does the user need the system?* This is the main reason to analyze if the purposed solution fits to the needs of the client.

After obtaining the answers to all these questions, we have an idea about how must be the system the users want. In this study, the user wants a customized and simple system to query information about materials and to get fast results to do analysis.

It is considered an affordable project because the available technology provides a potential solution to solve the main problem discussed in this document.

3.1.1. Measure requirements

The specifications of the measures have to be defined. The user gave some information about the available sensors that could be used to develop the required system.

The user does not need complex sensors, the user wants to estimate information and compare the results with other technologies available in the market.

The information measured by the sensors must be accessible for all the users. This information has to be interpreted with known values to use it for other studies.

The interpretation of the values must be considered, the measures are linked to real percentages of humidity.

3.1.2. Access requirements

The system has to be accessible for the users that are going to use the system. The access to the system and its limitations have to be explained.

Mobility and comfort are other requirements to consider. The system has to be accessible from different devices (e.g. computers, mobile phones or tables).

To satisfy this requirement it was established a wireless network to allow the access to the information obtained by the system. This requirement makes the system flexible.

3.1.3. Functionality requirements

The system has to monitor the information registered from sensors in Real Time. The system has to be available every time.

The information of the studied materials is registered. The objective of monitoring during long period of times is to obtain precise results in the studies. When the measures increase the predictions adjust better to real values.

The control of the system allows to detect possible issues on it (e.g. the system crashes). Also, the issues are tracked to fix them easily.

3.2. Technical requirements

This section is dedicated to describe the requirements related to the physical system and the technology used to develop the purposed system.

Here there is an explanation about why it was decided to use the technology described and the features of the designed system. This information makes easier to decide which technology is appropriated to design the system.

The designer has to think if the chosen technology satisfies the requirements of the users.

3.2.1. Hardware requirements

This section is dedicated to explain the requirements of the hardware used by the system.

The hardware has to be simple to use and transparent for the users. The users do not mind the election of the designer, they just want to use the system and do their work as better as possible.

The hardware has to be configurable because the user has a customized configuration to work. It should respond to the needs of the users.

The selected hardware has to have an easy deployment. It should be deployed everywhere, although the conditions are not the best for a normal system. The system has to work normal although there are in the ambient dust, noise or different item that can make interferences in the system.

3.2.2. System requirements

The system has to be accessible for the users. The user can query information without having a knowledge about how the systems works internally. The user does not need to configure anything in the system.

The system has to be auto-configured when the user wants to use it. It implies that it should be automatized and prepare to work every-time and every-where.

An important feature of the system is easy way to upgrade and its management. The system should be prepared for improvements if the user needs to upgrade it.

3.2.3. Software requirements

The software has to work transparently to the user. The user must not know how it works internally.

The software obtains information about some sensors. It stores the information into a system that is able to manage information. This information is accessible to the user. Also, the information is interpreted according to the requirements of the user.

It should be improvable and upgradeable if the needs of the users change.

3.3. Planning used in this project

One of the most important things needed to consider in a project is the existence of a Project Management. In this project, it was necessary to plan different scheduled meetings (to know the needs of the client), calculate timings related to investigation and analysis of the different alternatives which are interesting to apply to this project. Also, considering different death lines it is an important requirement (implementation plan).

It was important to consider the time needed to maintain the system after finishing its deployment and how to prepare the client to use the deployed system. Although the client has his/her knowledge to use the system, it was necessary to invest efforts to prepare a good documentation to allow to future developers to improve and to manage the full system.

If we have to enumerate the different things we needed to consider, the most important points are the following:

- Time needed to know the needs of the client.

- Time inverted to invest the correct technology to develop the system.
- Time inverted to get all the components needed to develop the system.
- Time inverted to develop a correct implementation plan.
- Plan different meetings to expose the status of the project.
- Time needed to develop the solution chose by the stakeholder and the engineer.
- Resources used to deploy the system.
- Study the results obtained by the developed system.
- Time inverted to form the client to use the system.
- Time inverted to provide a good documentation to the client.
- Time inverted in tasks related to the maintain of the working system.

3.3.1. Software development methodology used in this project

Before starting to do all the tasks needed to develop the *Measure System*, it was necessary to decide which methodology was the appropriate for this project. There are several methodologies we can apply to start the development of this project. The different methodologies we could apply to this project are the following:

1. Waterfall development. It was discarded since the beginning because there is a low feedback from part of the client. This methodology was not the correct one for this project because the most important thing we consider is that the client has to know what is done in the project and the client has to participate in the development plan.

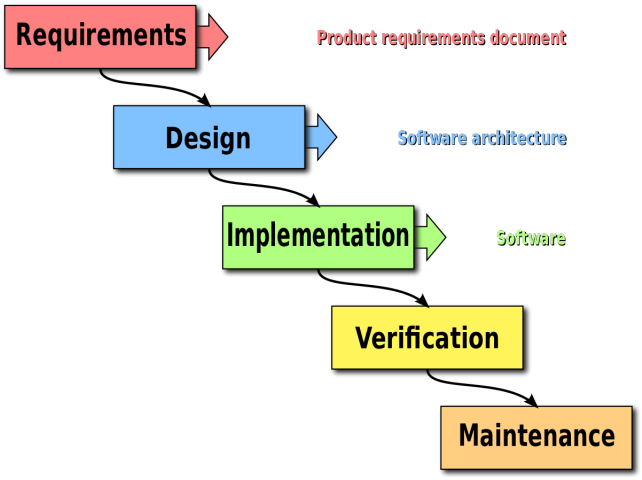

Fig. 3.1: Waterfall methodology

2. Spiral development. It is an interesting methodology because all the phases of the project are reviewed in all the steps, improving the quality of the final product. The worst thing of this methodology is that the client sometimes cannot participate in the development plan.

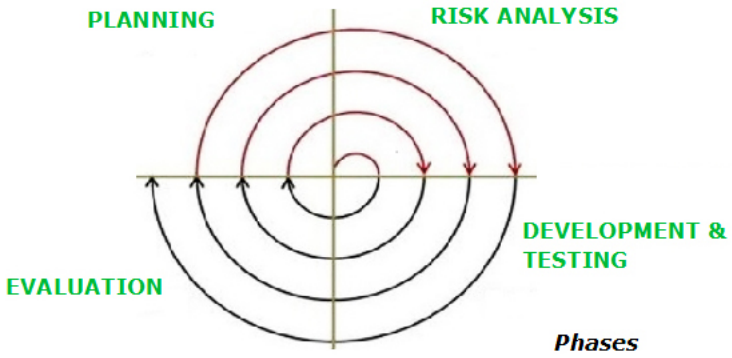

Fig. 3.2: Spiral model

3. V-Model. It is similar to the Spiral development but it has the inconvenient in the cost associated when it is needed to change features at the beginning of the project. Things are changed when the project or the step is in the final state and it has a high cost when it is needed to apply modifications.

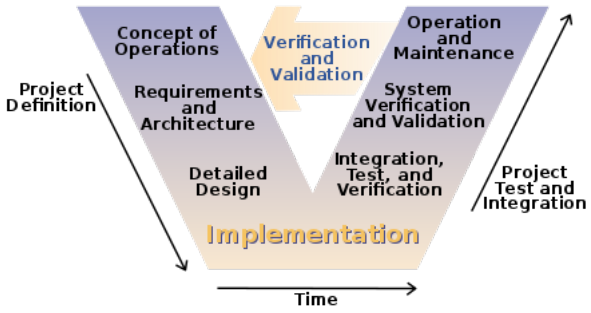

Fig. 3.3: V model

4. Scrum. This is one of the fittest methodologies we can apply to this project. The best characteristic of this methodology is that the client can participate in the development process of the product. It is flexible and the costs associated to changes are not high.

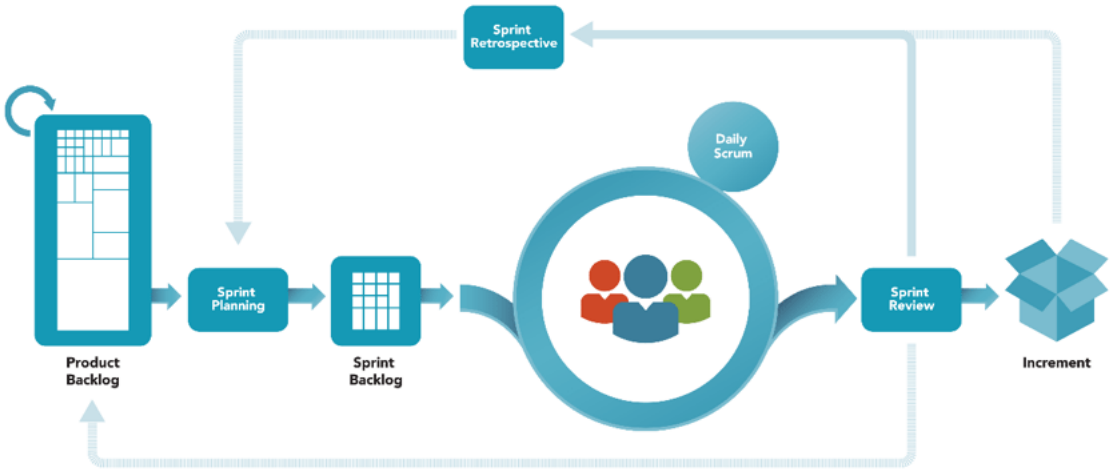

Fig. 3.4: Scrum methodology

After knowing the different methodologies we could apply to this project, it was decided to use a hybrid between Spiral development and Scrum model. The main reason to combine both technologies is the possibility to review the project in every step and the importance of the participation of the client in the project. The most

important thing is to consider the opinion of the client in every steps because the client is the person which can allow us to develop the correct solution to his/her needs.

The other methodologies are not the correct ones to apply to this project because it is more difficult to allow the client to participate in the different phases of the project.

Chapter 4

Design

This section is dedicated to explain the different important topics considered to develop a Measure System. It will be explained the architecture of the system explaining its different components, the physical components used to get information related to the different measured materials, how was though the communication flow between the sensors and the database manager and why it was chosen the different microcontrollers applied in the system.

There are other important requirements to consider for this project. One of these important requirements of the system is that it should be a flexible system where the different components used on it should have a low price. This point is important because it is necessary to choose the appropriate hardware to design the final solution.

Also, it will be explained why this solution is one of the best to apply to the of the client. This is an important topic to consider because to solve different problems it is possible to apply different solutions.

4.1. Architecture of the system

In this section, it will be explained the architecture of the designed system. There is an explanation about how was though the main system. Also, it is explained why this solution is one of the best to solve the exposed problem.

In the following picture it is possible to see how is the architecture of the designed system.

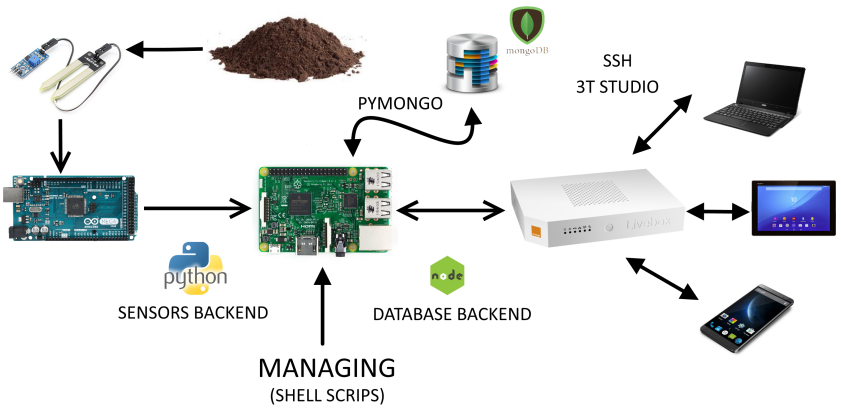

Fig. 4.1: System's architecture

In the picture, it is possible to see the different components of the system. All the components are explained in this section.

All the components are interconnected, using simple communication protocols. These protocols of communication will be the following:

1. Serial communication. It is reliable and it is possible to avoid wireless data transfers problems between the Arduino and the Raspberry Pi. Using a serial communication will allow us to get the information without losing measures. The communication is continuous and the delay between the different transfers is from five to ten seconds, avoiding problems in the communication, like flooding or missing information.

2. Ethernet communication. This communication allows us to connect the Raspberry Pi to an access point. This access point is reliable and will allow us to query the information stored in the system. We chose Ethernet to connect the Raspberry Pi to the router to avoid wireless communication problems in Raspbian (Operative System used by the Raspberry Pi).

3. Wireless communication. We chose wireless communication to connect different devices to the system to query the information of the system. Those devices have to have WiFi interfaces.

The main concept to chose all these protocols of communication is to allow a simple communication between the different devices to avoid heavy tasks related to the maintain of the system. Also, all the protocols used by the system are reliable and stable.

4.1.1. Data center processor

In this measure system it is important to know if the information that is going to be analyzed is going to be stored or not. After talking to the client, one of the most important requirements was that the system needed memory to storage the different results obtained after measuring the different materials.

With this information, it was though that it was needed to use a Database Manager to store all the information coming from the different sensors which are connected to the system. Also, it was important that the information measured by the system has to be obtained in a short period of time, doing this process continually. This requirement was fixed since the beginning of the project.

It is often applied Big Data technologies to problems where it is important to store continuous information in a short period of time. Big Data technologies allows us to obtain a big number of measures which allow us to analyze it after getting big amounts of information.

There are several technologies that allow us to apply this solution. This topic was import to discuss. It was important to decide which technology is possible to apply to manage all the information measured by the different sensors we had connected.

Another important requirement to consider is that the system could grow in the future. The system could have more sensors connected to measure more information. Here it was clarified that the best solution to apply to this problem was a flexible technology as Big Data technologies.

In the following section, it will be explained the different topics related to the

Data center processor we chose for this system.

Data center storage

After having several meetings to get the different requirements of the system, the conclusion we reach is that the storage of the system was an important feature to consider. The *Measure System* needs to store lot of information which comes from different sensors which are connected to the Arduino.

Here, it was discussed if it was needed to use external devices to store all the information of the sensors. We decided to use the internal storage of the system because of its fast response when it is required to query information from the database. All the information which is stored in the database has not a big size, and although the storage is continuous, the size of the database is not going to be incremented so much.

With this information, it is possible to design a solution which will use a small storage device for all the information coming from the sensors. The storage of the measured information of the system is going to be done every five-ten seconds. It was defined this interval of transmission to obtain precise measures.

It is important to consider that the information has to be controlled to not fill the storage used by the system. When the database will grow it is needed to remove the information stored in the system.

Data center manager

This part of the system is going to be important in our design. The *Data Center Manager* is going to manage all the information stored in the system. Also, it will manage the serial communication needed to transfer the measures got from the different sensors we have connected to the system.

This device has to allow connection in case it is needed to do tasks related to maintain. Its maintain has to be simple and easy because it will have several processes in the background running. All these processes are going to be related to the communication to the sensors, backend of the database and queries to the MongoDB

database installed in the system.

As it was explained before, it was decided to use a Raspberry Pi 3 to control the main system. This device is going to give us the power needed to control the system, and also, it is going to be a cheap solution to deploy the explained system.

Data center controller

The data center controller of the system is going to be related to the *Data Center Manager* of the system. Both tasks are going to be managed by the Raspberry Pi 3. This device has enough power to manage both tasks. This is the main reason we chose to use a Raspberry Pi 3, as we explained before.

The data center is going to be important because it will be in charge of getting the information from the sensors and after getting this information, it will recognize the transmitted information storing it using the correct format for the database installed in the system.

4.1.2. Sensor controller

To control all the sensors of the system we decided to use an *Arduino Mega*. The main reason to use this device is its easy way to manage the information got from the different sensors we have connected in the system. It is easy to program and easy to maintain in case it is needed to do it.

We consider *Arduino Mega* as a great controller because in case we need to upgrade the system, it will allow us to do it. It has several connections (more than the *Arduino Uno*) and all these connection will allow us to connect more sensors to the system if it is needed to do it.

All the information transmitted by the *Arduino Mega* will have a specific format to allow the Data Center Controller to recognize the information that is going to be stored in the system. This format has to be simple and easy to interpreter by the system.

Data structures

One of the most important requirements of the system is that the information transmitted by the sensors has to have a specific format. This format has to be defined by the developer to allow the device to recognize the information transmitted in a easy way.

When the information is transmitted to the controller, the controller has to recognize the different parameters stored in the message. When the controller receives the information, it has to recognize from which sensor is coming the information with its current timestamp.

It is important to consider that the controller is not a normal computer and it is needed that the information transmitted has to be as simple as it is possible to simplify it.

Upgrades on sensors

Since the beginning of the project, it was though that the system will have to grow in the future. When we talk about growing, we are talking about connecting more sensors to the system.

All the upgrades done on the system have to be simple and have to avoid connection problems. We were thinking since the beginning how to do it, and the solution we designed has the possibility to connect N sensors to system without changing parameters in the *Data Controller*.

The *Arduino Mega* has to be prepared to allow the connection of more sensors on the system. It is needed to prepare the physical circuits to connect more sensors following the *datasheet* provided in this document.

The normal use case is that the user will only have to connect the sensor to the system as it is done with the others. When the sensor is connected, the system will recognize it instantly and it will start to record information from it.

4.2. Information flow

One of the most important things to consider to design the system is how is going to be the information flow. When we talk about *information flow*, we refer to from which parts of the system is going to flow the communication.

The way the communication is done in the system is the following:

1. The sensors will get the information from the substratum where they are connected.
2. The *Arduino Mega* has to process the information received from the sensors and it has to define the format used to start the communication with the Raspberry Pi 3.
3. Every five seconds the *Arduino Mega* will send information to the Raspberry Pi 3.
4. The Raspberry Pi has to listen continuously to the information which is coming from a serial port.
5. The Raspberry Pi has to receive the information received from the serial port, and after that, it has to parse the received information with its format.
6. When the Raspberry Pi 3 has processed the information, it has to store it with its correct format and its correct timestamp.
7. MongoDB's daemon has to manage the query launched by the backend of the Raspberry Pi 3 and it has to store it in the database.
8. MongoDB's daemon will wait for incoming connections where the user or the programmer will launch queries to the database. These queries could store information in the database or display the information stored in the database.

With all these steps, the system will start to record information coming from the sensors. All the stored information will be available for the user in case the user needs to query information from the system to get results.

4.2.1. Sensor information flow

It is required to use a device which will have the possibility to get information from different sensors. This device has to be connected to all the sensors and it has to have enough efficiency to manage all the information coming from the different sensors of the system.

The information has to be received by the controller (in this design, the *Arduino Mega* will do all this stuff) and it has to choose the correct format to send the information.

When the device has all the information of all the sensors, it has to create a message with all the information of the sensors to send it to the target device (in this case the controller which will store the information in the database).

The device will build a message with a specific format, and when the message is built, it will send it with the chosen format.

4.2.2. Storage information flow

When the system receives a message, it has to process it, and after that, it has to decide if the information has be stored or not.

The system will be listening to all the incoming messages over a serial connector. This serial connector is managed by the system. The system will control the port linked to the serial communication.

When the system receives a message on its serial port, it will analyze the received message and it will check its format. If the format of the message in the correct way, it will parse the message and it will send the information to the database.

The backend will control the communication to the database. This communication is controlled by the system, and it will send the information to the database

using different queries supported by the database controller.

The MongoDB daemon will stay listening for all the queries launched by the system. This daemon supports concurrent connections to the database.

If there is a message with the incorrect format, the backed will just discard the message. This concept will allow the database to have consist and reliable information on it.

4.2.3. Query information flow

In the requirements explained in previous sections, it was decided to allow the system to store information. Also, it was important to define that the user has to have access to the database created by the system to retrieve its information.

This concept has to be explained with the following understanding:

- The system has to support queries which allows the developer or the backend to store information in the database. We define this queries as internal queries used by the main system to feedback the database.
- The system has to support external queries which will allow the users to query the information stored on it. All these queries are external to the system. The external queries are not going to modify the database, they are going to display the information stored in the system.

With this information, it is clarified how is going to be the query information flow in the system. It is needed to define it because all the queries which are going to be done in the system to store information has to be done inside the device.

The external queries are dedicated to show information of the database that the user will use to prepare different analysis of the data stored in the system.

Chapter 5

Materials and methods of study

This section of the memory explains the aspects related to the materials used to build the project. Additionally, it describes the methods used to obtain results and the technologies applied to develop the system.

5.1. Materials

To carry out this project, it was necessary to use numerous components supplied by the *University of Extremadura*. These components are characterized by their diversity and their reduced cost. The prototype designed for the study uses free hardware components. Free hardware components have good features like their flexibility when building a system is required.

It was decided to used the aforementioned components since the manufacturer provides a lot of documentation. The documentation of these components allows to develop new features and provide flexibility and customization to the final user.

There are several components which form part of this study. These components are hardware devices and software features. Also, tools and frameworks are relevant in this project. All of them allow to develop the desired solution. This information is listed in this section.

5.1.1. Arduino Mega

It is more powerful than Arduino Uno, Arduino Nano and Arduino Micro. It has more available connections used as inputs and outputs. Thus, making possible to create more complex systems.

In this study we decided to use and Arduino Mega because it is a device which scales better than other models. Having a higher number of available connections allows the developer to extend easily the system.

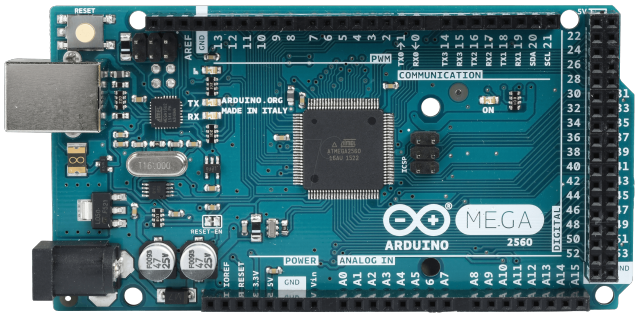

Fig. 5.1: Arduino Mega

Arduino is an embedded board which allows to program it using its framework, *Arduino IDE*. With this framework it is possible to develop small and powerful programs which allow the control all the functions available in the system. The programs are developed in the C programming language using the *Arduino IDE*. The *Arduino IDE* includes many libraries and allows the developer to create his own libraries to customize the control of this board..

The most important capabilities of the *Arduino Mega* are described in the following points:

- ***Analogical outputs***. These interfaces allow the programmer to read and control sensors. The read voltage of the sensors goes from the 0 value to 1023 value.
- ***Digital outputs***. The modulation of the read signal uses binary values instead of curves to represent the information. Also, there are digital signal which convert the output values to analogical signals using the *PWM* technology.
- ***Processor***. This processor is a microcontroller board base on the ATmega2560. Its throughput reaches 16MIPS at 16MHz which is enough for embedded systems.
- ***Flash Memory***. This memory is used to upload the developed software. Also, there is a section in this memory which is dedicated to the bootloader of the device. This bootloader allows the device to boot up.

5.1.2. Raspberry Pi 3

It is an ARM-based microcontroller. This device has a processor used in mobile devices like mobile phones, tablets and Chromecasts. It has a good performance and a low energy consumption. It allows the installation of light *Unix* distributions.

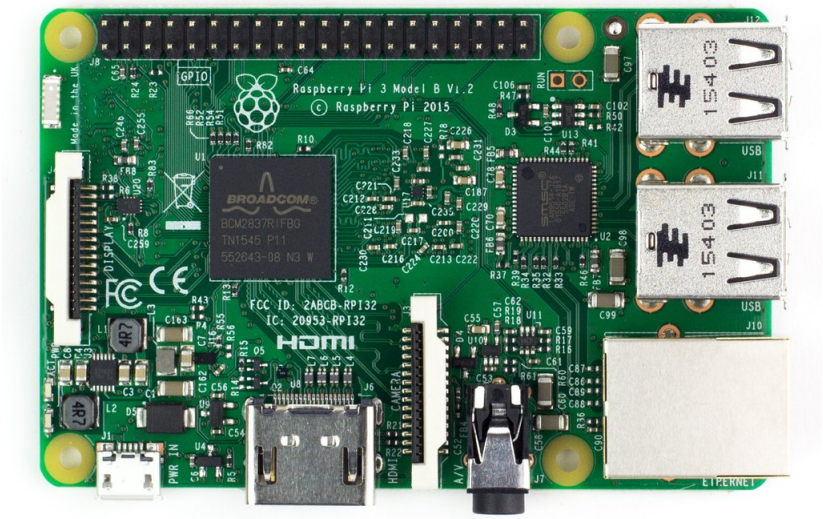

Fig. 5.2: Raspberry PI 3

This project works with a *Raspberry PI 3* using *Raspbian Freezy* as operative system. This platform has a good support and it is proven to be stable. The community of Raspbian is strong and brings customized solutions for embedded systems.

Raspberry PI 3 is a device that has useful features e.g. storing information coming from sensors. It is able to manage a NoSQL database managing petitions and queries coming from users and developers.

The most outstanding features of the device are enumerated as follows:

- *Mobility.* This device is small and easy to move to other places due to its wireless adapters.
- *Portability.* It is possible to install platforms in external storage and run the installed platform in every Raspberry PI 3.
- *Maintenance.* The maintenance of a Rasperry PI is low. It mainly involves the run of Software update. It is a device that does not have air cooling or other mechanical devices and it is possible to use it in industrial environments under harsh conditions.
- *Cost.* As aforementioned, the cost of the device and its components is low. For instance, its SD Card, power supply, network adapters or peripherals.

Knowing the capabilities of this device, in the following section it will be analysed the technical specifications of the *Raspberry PI 3*:

- *Processor.* It is based on ARM architecture. These processors have a low consumption power and good performance. Parallel and concurrent programs can be deployed such as a solution for complex systems.
- *Graphic processor.* This microcontroller allows to process images and textures. It is appropriated to develop solution where image processing is required.
- *Memory RAM.* It uses 1GB LPDDR2 memory which is just enough to run simple and advanced applications.
- *GPIO connections.* All these interfaces are used to connect peripheral and sensors. Sensors are programmed over these interfaces. It is common interface used in embedded systems.
- *SD slot.* This socket allows to connect a SD card where the operative system is deployed. It supports large and fast SD cards which are dedicated to the operative system and applications.
- *HDMI output.* Used to connect monitors and big screens.

- *Mini-Jack output*. Allows to connect the device to audio systems. Also, it is used to connect sensors which requires this connector.
- *USB ports*. These interfaces are used to connect external peripheral devices.
- *Ethernet interface*. It has a standard 1Gbit Ethernet via *RJ45*.
- *Wi-Fi adapter*. It is integrated in the board and provides mobility to customized systems.

5.1.3. Router

It is a useful component which creates a network to allow the user to connect to the system and retrieve all information needed. There are some possibilities to initialize a connection with the system, but using a separate network is better. The user can use a mobile device forgetting about using a physical connection to the system. The RJ45 connector could be used but limits the mobility of the system.

The brand of the router used is a Livebox 2. Its features are enough to control and access the main system.

Fig. 5.3: Router Live Box 2

The router is used to create a communication between the *Raspberry PI 3* and the computer used by the user. Therefore, it is possible to retrieve information from the NoSQL database and manage queries to system.

In case it is needed to modify some parameters of the configuration of the system, this network helps the administrator to manage them.

The router has a direct connection to the *Raspberry PI 3* via an Ethernet Cable. It was decided to use this connection because it is a reliable media. Using a wireless connection might create problems in the communication, generating latency and jitter to the access of the system and complicating the querying of information in real time.

5.1.4. Laptop

The user will use this device to query information from the system. With the laptop, we can use some tools to retrieve some information from the system. It is useful to use because we can generate some graphics after analysing the information of the database.

With the laptop, we can create an *SSH* connection to the system to manage its processes. We can use a *SSH* connection to run processes of the database or to control processes related to the backend. This remote connection allows the administrator of the system to manage it directly.

The user can use several operating systems. The *SSH* connection is independent from the operating system.

This laptop should include the following software programs:

- *3T Studio*. This software is used to query the database. Also, it allows to export the information of the database in *JSON*, excel or *CSV*.
- *Putty or a remote manager for SSH connections*. To start a remote connection to the system.
- *Microsoft Excel or other similar editors*. To interpret the obtained results of the queries.

5.2. Technology

In this part of the document, the technology used in this project is explained. This project looks for the appropriate technology to build the system decreasing its costs associate to hardware components and maintenance.

5.2.1. NoSQL database

In this project, it was decided to use NoSQL technology because of its flexibility. With *NoSQL* databases, we can store all kind of information in a system. When it is necessary to store no structural information, *NoSQL* databases work have a good performance. It is not required to indicate to the system which kind of information is desired to store. NoSQL systems do not mind the structure of the information, they just store it.

Other important feature of the *NoSQL* databases is the possibility to store information in real time. In this project, one the goals was to store during long time information. This system has an amount of humidity sensors, and those sensors measure information about the materials during long periods of time.

Developers can build fast and easy queries. In *NoSQL* systems the information is stored in documents, instead of using tables or table spaces.

The administrator of a *NoSQL* database can build operations to optimize the access to the databases too. This kind of operations are used in *Big Data* systems and use reducing operations.

5.2.2. NodeJS

Using NodeJS in this project it is a good way to manage information from the database system. NodeJS is used in this project as *backend* for a web service.

NodeJS works as a process waiting for incoming requests. When NodeJS receives a petition, it processes it and looks for the information in the database. NodeJS

works on an asynchronous mode, increasing the performance of a backend. On the other hand, debugging the execution flow of a backed in a hard task to do. If we want to control operations that should work on a synchronous way *callbacks* are required. Callbacks turn the execution flow of NodeJS to a synchronous mode.

It has a better performance than *PHP* technologies, which are commonly used by web services. The asynchronous mode creates threads that can serve concurrent petitions. This is one of the reasons to deploy this technology.

Performance is one of the goals of this project. The microcontrollers used in this study are limited because of their architecture and *NodeJS* increases the performance of the system.

5.2.3. Python 2.7.13

Python is one of the most important components in this project. It is used to control the information which comes from the network of the sensors. Part of the *backend* is developed using *Python*. This part of the backend is a simple program which receives information coming from the *Arduino*. The *Arduino Mega* is directly connected to the *Raspberry PI 3* using an *USB* cable creating a serial communication. It was decided to use this protocol because it is a reliable media used by microcontrollers based on embedded systems.

The *Raspberry PI 3* has a process developed in *Python* which is listening to the information coming from the *Serial Port*. *Python* has some libraries to control *Serial* connections. The *Python* program controls the opening, reading and closing operations on the *Serial Port* with a customized *API* we have developed. The *Python* program uses some libraries to store the information transmitted by the *Arduino Mega*. When the *Python* program detects new information in the *Serial Port* it checks the information received and if this information is valid, it stores it in a *NoSQL* database. If the information received is not valid, the program discards the information received. The information sent by the *Arduino Mega* has an special structure. The Python program knows the structure of the messages and it analyzes the messages

trying to parse the information received.

The information sent by the *Arduino Mega* has the structure ***sensor-id # temperature # humidity # date***. As it is possible to see, the information is delimited by the character #.

This program is allocated into the *rc.local* file in the *Raspberry PI 3*. This file allows the *Raspberry PI 3* to execute the script when it is turned on. If the script is not specified on the *rc.local* file we should run it from the Raspberry PI 3 console. We can use the *Raspberry PI 3* console via *SSH* to execute scripts and programs.

It was decided to use *Python 2.7.13* because it is one of the most simplest program languages and because it is efficient and reliable.

API MongoDB

As it was explained before, To control the information allocated in the *MongoDB's database* is required. *Python* has some libraries to manage MongoDB's databases. This project uses *pymongo* library. This library allows to create queries to manage the database. It is possible to do all operations allowed by databases with this library.

The uses this library to inset and query information of the database. The API's webpage brings support to the developer to manage all operations of MongoDB's databases. The documentation of this API is complete and well explained. Also, it supports to install MongoDB systems in Windows, Linux and MacOS.

The API manages databases as JSON files. It is not like in other databases systems, where the information is allocated in table spaces. In MongoDB the information is stored in a document. This document appends information to its ending when is required to store structures or variables. The JSON file allows the database administrator to create operations to optimize the access to the database, like Map Reduce operations. It is the most common operation in MongoDB's system.

The investigation of how to apply map reduce operations in our databases is not carry out, but it is an open point to consider in future studies. These operations could be done if the database grows fast and the performance of the databases is reduced by the size of the database.

API Sensor's readings

This API was developed to control the information sent by the Arduino Mega. This API allows the user to read the information and store it in the MongoDB's database. The API controls reads coming from the serial port. Also, the API controls operations such open and close operations over the MongoDB's database using the API pymongo.

This API was developed using Python 2.7.13 and could be used for other systems that want to implement the same methods used in this project.

This library checks the information received from the network of the sensors and checks if the structure of the information received is valid. If the received information has a valid structure, it stores the information in the database.

This library allows the user of the system to connect N humidity sensors. With this feature the user will not have problems to connect more sensors.

5.2.4. Scripting

In other sections of this document we talked about the use of *Python*, *MongoDB* and NodeJS. There is an other important part in this project, scripting. Scripting is important in this project because it allows to automatize the processes used by this system.

Normally, the user does realize about this function because it is completely transparent. Scripting allows to control the processes related to the management of the database and the processes which use the network of the sensors. Also, it is possible to manage the information of the database using *NodeJS* as *backend*. It runs queries

to retrieve the information of the database

In this project was necessary to use *Linux Shell scripting. Linux powershell* allows the user to control the processes used by the system.

There is a script which controls the processes required to run the measure system on the *Raspberry PI 3*. It checks the status of the system and if there is nothing running it loads all required processes.

This method uses the *rc.local* configuration file of the Linux System. This file allows the system to call all functions we want to execute at the boot up.

5.3. Methods

In this section, the methods used in this project are explained. A description of the methodology followed to develop and manage all necessary functions in the system is brought. At least, it is one of the most important topics in this document, because with the methodology described in the following sections, it is possible to obtain the information required to present our results. After that, we can analyze the results and study the conclusions.

5.3.1. Querying methods used by NoSQL databases

This is one the most important topics of this document. We are using a new technology, which uses different methods to obtain the information stored in our database.

In NoSQL systems, the queries used to manage databases are different from the others used by a normal SQL system. NoSQL systems does not use data tables. In NoSQL systems, a JSON document is used as a database. The information is not stored like in normal databases systems, which changes the way to obtain and to store data.

In this project, there are some steps to follow which are described in the MongoDB's guideline. Here, there are several examples about how to insert, update and query the information of the database.

The language used for this purpose has JSON instructions to obtain the information of the database. There are some special keywords to query the information. These special keywords are described in NoSQL documentation too.

5.3.2. Methods used to retrieve information from the NoSQL database

Like it was described before, the database system is completely different to the standard ones. This database system has a different way to retrieve its information.

In *MongoDB*, is possible to use some techniques to manage the information of the database. These techniques are the following:

- Using normal scripting for queries. This is the most difficult way to manage a *NoSQL* database. It is possible to retrieve the information of the database using a Linux console. If it is required to do it, the user has to execute *mongodb* in the Linux terminal and log in with his credentials.

  On the other hand, there is no assistant to help the user. The only way to obtain information about the commands is typing the option *help* when it is desired to run *mongodb*. A knowledge related to the commands used by *MongoDB* is required.

- Using a *GUI* applications. There are several *frameworks* which allows the user to manage databases and build queries. In this study it was used *3TStudio* (*MongoCheff* in older versions).

  This framework has an assistance that allows the user to create his own queries. The user only needs information about the location of the database and the credentials used by the database.

The second option is recommended. Using this option makes more efficient to manage a database with an assistant. The user can build its own queries and check if it is required, the syntax of them. It is possible to see the information of the database in real time too.

The fist option I think is more complicated to use because the user needs to be experimented with the MongoDB console.

5.3.3. Methods used to obtain graphs with the information of the materials

When the information is stored in the MongoDB's database, if we want to make an analysis of the data that is stored, it is required to export the information.

If we want to export the information of the database, we can use 3TStudio to retrieve the information we want to export. It is possible to retrieve the information related to the obtained results after the execution of a MongoDB query.

3TStudio allows to select the format to export the information. The results are exported using a *.csv* format. This format is useful if it is required to do an analysis using Microsoft tools. We decided to use Microsoft Excel to analyze the information of the database.

Microsoft Excel allows the user to generate graphs and making some calculates of the information that we want to analyze. With Microsoft Excel is easy to obtain the results in graphs. The user only has to select the information to analyze and after that, select the graph desired to interpret the information.

In this study, we made two analysis. The analysis made in this study is the following:

- Analysis of the evolution of the floor. With this analysis we look for information related with the drying process of the floor. This information is displayed

in 4 graphs using the values registered from the sensors used in this study.

The purpose to select this graph is to analyze if the floor produces valid values in the drying process.

- Analysis of the results retrieved by the sensors. This analysis allows the interpreter to get information about the current state of the sensors. The sensors show the values along the time.

  In this analysis it was selected the information registered by the 4 sensors used in this study.

5.3.4. Methods based on compaction

5.3.5. Methods used to calibrate sensors

This section explains how is the process selected to calibrate the measure of the sensors. This process, registered the values obtained with the sensors at different conditions. These conditions are the following:

- Measures obtained after allocating the sensors in water. This method registeres the lowest value measured by the sensors. The lowest value links to the lowest voltage.
- Measure obtained after allocating the sensors in a dry place. This method registeres the highest value measured by the sensors. The highest value links to the highest voltage.
- Measure obtained after allocating the sensors in different floors with different values of humidity. This method measures diverse values of floors in different conditions.

  When the measures of the soils are done, an oven to heat the soils is required. This technique allows to obtain the precise value of humidity of the different floors analyzed. This process takes a day to produce the expected values.

When the results are produced, the information, which refers to the measures, is linked to the time inverted to dry a specific soil. Also, the percentage of humidity registered when the drying process is finished refers to the value measured by the sensors before drying the soil. These results define how is the behavior of the sensors when they are measuring a specific material. The chosen process allows to know if the selected sensors used to measure the soils are appropriate.

After using the 3 techniques described before, the results produced by the sensors are compared to the values registered by the oven used to dry the materials. With this results, a comparison is done to obtain an average of the results. This average is used to produce precise measures of soils.

5.4. Resources

In this section, the resources used to biuld the system used in this study are explained. The resources are sorted out into two big groups, hardware components and software components. Those two groups are relevant to this project, because with them it is possible to make this study with its results.

5.4.1. Hardware

This group contains the components which form part of the infrastructure of the Information System. The infrastructure is relevant, because it allows to build the Information System and makes easier to monitor the information registered by the sensors. It allows to create a data store to manage the information stored on it.

Microcontrollers

The microcontrollers are the intelligent part of the system. They have features to manage the information retrieved by the sensors and store it into the database. In this project, it was decided to use two microcontrollers:

- *Arduino Mega*. This device allows the system to register the information coming from the network of the sensors. The *Arduino Mega* obtains the information,

packages it, and sends it to the *Raspberry PI 3* using serial communication. The *Arduino Mega* is not an inteligent hardware, it obtains the information and sends it to other microcontroller. It has no complex processing.

- *Raspberry PI 3*. This device is the brain of the system. It gets the information which comes from the *Arduino Mega*. When it receives the information, it analyzes it, it checks if the information is well formed, and after that, it stores the correct information into the database. This processing is managed by the backend running a Python program.

  It has the database too. The database managed by the *Raspberry PI 3* is a *NoSQL* database. The *Raspberry PI 3* has installed *MongoDB* which controls the database.

  The *Raspberry PI 3* has another backend that is used to manage the petitions of the users. The users want to query information and the *Raspberry PI 3* manages their petitions using a *NodeJS* backend. The *NodeJS* backend queries the information requested by the users and sends it, giving to the user the values stored in the database.

Sensors

These are one of the most important parts of this project. The sensors are able to get all the information related to the materials which are monitored. The sensors have some electronic features that allow them to monitor soils.

In this project, the sensors used for the study are simple and have a low cost. The price of these sensors is around 0.80€. and 1€. It is possible obtain them via internet. The model used to get the information of the system is *FC-28*. This model is formed by 2 parts. One of those parts is the device that has the electronics to measure the values of the floor, and the other part, is an electrical component which converts the information retrieved by the sensor in numbers.

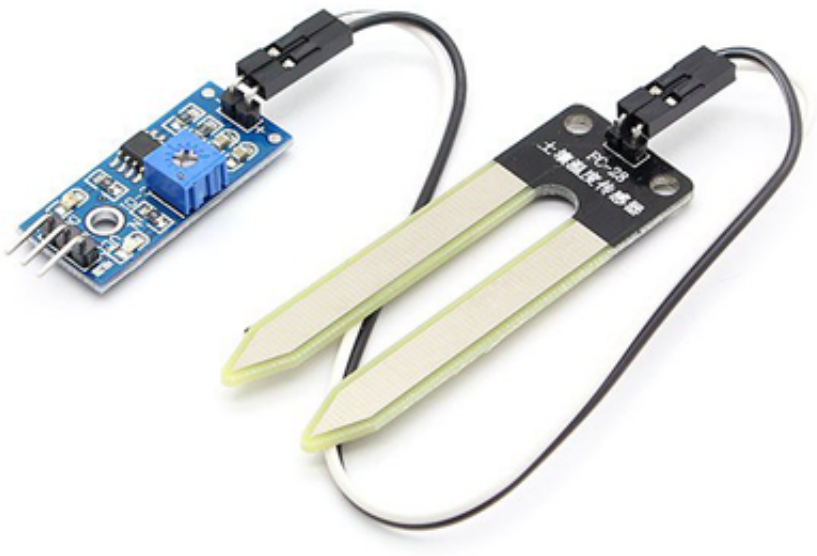

Fig. 5.4: FC-28 humidity sensor

The information treated by the rectifier uses a numerical structure. This structured has relation with the voltage detected by the sensors. These numerical values has a range between 1024 and 0. The value 1024 corresponds to the driest value measured. The value 0 is the lowest, corresponding to the wettest media analyzed.

The connection used is simple. The device which gets the information of the media is connected directly to the rectifier using two cables. The rectifier is connected to the *Arduino Mega* using three cables. The cables used to measure the materials are the following:

- Positive cable. Provides the positive voltage of the power supply.
- Ground cable. Provides the negative voltage of the power supply.
- Data cable. Used to read the values of the measured material.

In this project, it was decided to use four sensors with these conditions to build the prototype. The system supports up to twelve sensors in case it is required to measure more materials.

Network devices

This section is dedicated to explain the network components which form part of the system. These components provide communication to all the devices which are connected to the system.

These network devices provide connectivity to the system to query and manage the information stored on it. The network components used in this project are described in the following points:

- *Ethernet cable*. This component is used to communicate the *Raspberry PI 3* with the router of the system. The model of the cable used in this project is a Ethernet cable category *5E*. We decided to use this connection to connect the *Raspberry PI 3* to the router because it is a stable media. Another option is to use a *Wi-Fi* adapter, but *WiFi* communication it is not reliable and it can fail if there are interferences in the media.

- *Router Livebox 2*. This router is used to create a network between the *Raspberry PI 3* and the users to read information of the *Raspberry PI 3*. This router allows the user to establish a wireless communication too. The user can use a mobile device to connect to the system.

- *USB cable*. This media is used to communicate the *Arduino Mega* with the *Raspberry PI 3*. We decided to use this communication because it is reliable, stable and easy to manage.

The main feature of these components is their reliability, stability and their easy way to manage. It is easy to replace them if it is required. They are common in systems and it is easy to find them.

5.4.2. Software

This section is dedicated to the Software components which form part of the Information System. The software components are the intelligent of the designed system. They allow the user to manipulate the information monitored by the network of the sensors.

They bring to the user useful features. These features are the value of the project. All the functions developed follow the requirements defined by the users of the system.

Upgrades are supported by the Information System, the software components allow the developer to design new solutions based on the initial state of this system.

Also, developers can improve the system, because the system is scalable. The technology selected for this study provides this feature.

One of the epics in Sofware Engineering is the possibility to design customized and reusable systems.

3T Studio

3T Studio is the software used as assistant of queries for the *MongoDB's* database. This tool is available in the Internet and it is possible to download it for free purposes. This tool is available for *Windows*, *Linux* and *MacOs* operative systems.

This tool allows the user to manage the information of the database. The user can build queries to retrieve the desire information of the database. The information is displayed into several options, providing to the user the possibility to show the results with lists of components, *JSON* files or displaying the information in tables (e.g. *SQL* systems).

This tool has to be connected to a *MongoDB* system. It is required to give the program all the information related to the connection used by the database. The

connection to the database is configurable and there are several options to create it:

1. Over *SSH* tunnel. Based on *SSH* protocol. Provides a higher level of security to the connection in case it is required direct communication to the database.

2. Direct communication to the database. Simplest way to connect to the dabase but sacrifices security concepts.

3. Communication via proxy authentication. Complex but adds a new layer to the security of the system. It is used in huge systems where a higher level of security is required.

3T Studio has an assistant to configure the desired connection to the database. In our use cases, the direct communication to the *Database Manager* is the chosen option bacause of its easy way to communicate to the database.

On the other hand, the user can export the information of the MongoDB database in several formats. It is possible to use *JSON*, *CSVs* and excel formats. This project uses *CSV* format because it is the easiest way to automate and manage it with Microsoft Excel.

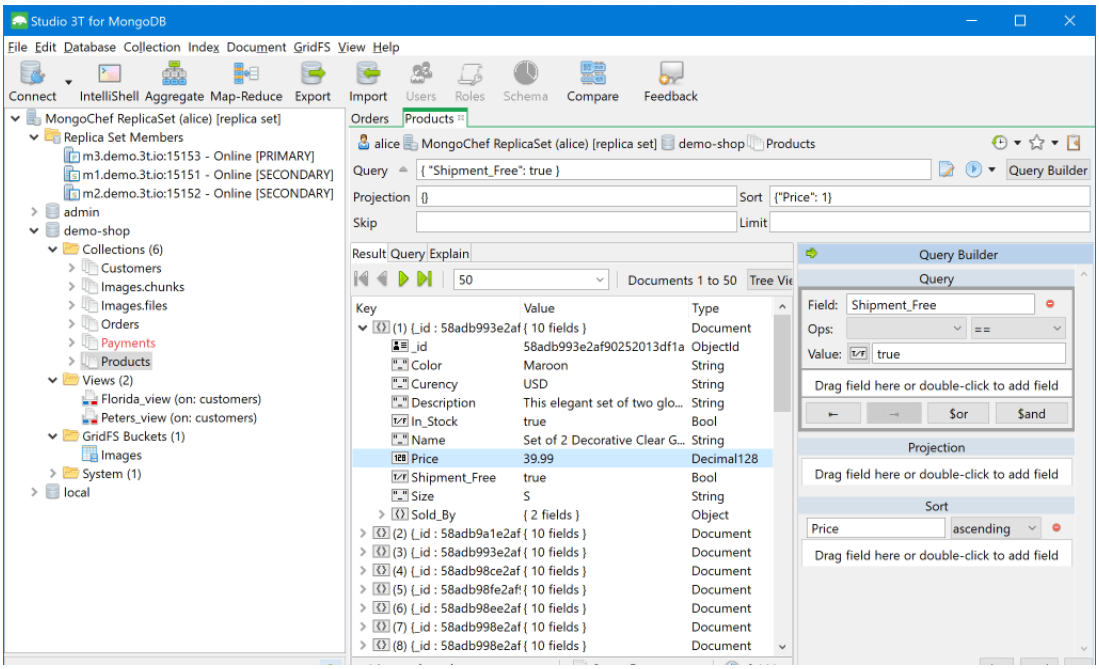

Fig. 5.5: 3T Studio

Microsoft Excel

Microsoft Excel is one of the tools used to display the results measured by the system. *Microsoft Excel* allows to load *CSV* files exported from the *NoSQL* database.

Microsoft Excel provides features to apply formulas to the monitored soils. The theoretical values of the measured materials are loaded. The formulas arrange the results obtained with the oven used to calibrate the sensors.

The calculates generate an average of the exported information. To analyze the behavior of the soils it is required to select the desired results. The timestamp of the sensor provides a filter to sort the results of the report.

When the information is selected, it is required to select the desire chart to represent the results. The generated chart allows the user to compare the produces results with other studies or ideal models.

The graphs provided by *Microsoft Excel* allow us to determine if the study is congruent.

PyCharm

PyCharm is a development framework which allows the developer to write *Python* scripts. It was decided to use PyCharm because of its special features to develop *Python* applications. Also, it manages the required packages imported by the generated scripts.

PyCharm is available over the Internet and it is a free framework to use. It is a stark software and it fits to the needs of the developer. The most important features of this framework are listed as it follows:

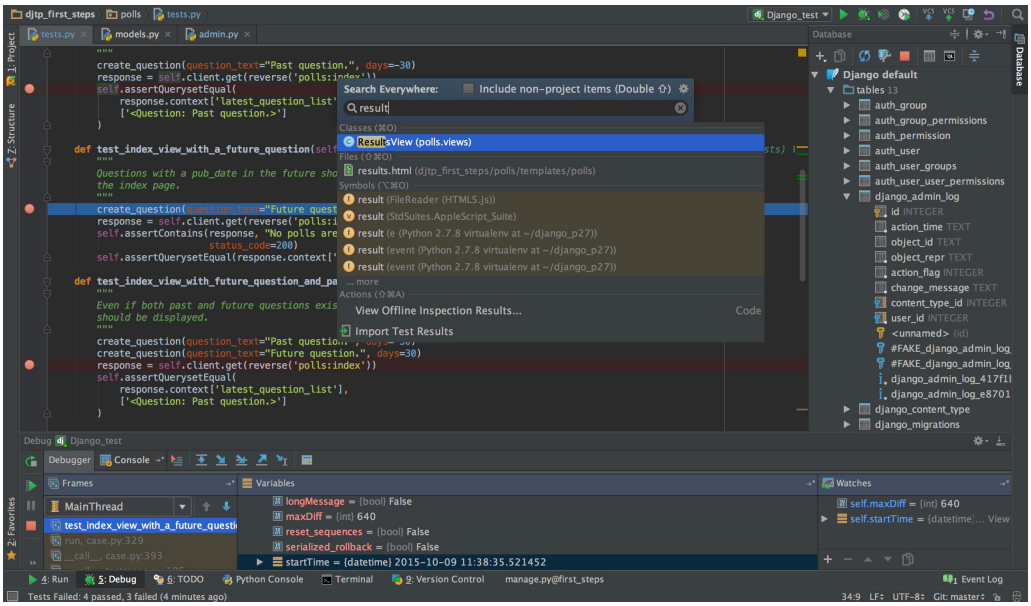

Fig. 5.6: PyCharm Graphycal Interface

- ***Supports several Python's interpreters***. *PyCharm* has support to use an amount of interpreters. It supports *IronPython* and official *Python*.

- ***It supports all Python's versions***. Using different versions of Python is not risk. It allows the developer to load all the required libraries. It has an assistant

to search libraries too.

- ***It has a good assistant for development***. It has a powerful assistant. The assistant has a fair performance when it is required to display the documentation of the methods and libraries imported by the developer.

- ***Shortcuts and development from scratch***. The time invested to develop applications with python is low. Shortcuts and templates are available for the developer.

- ***Community version***. It provides a free to use version with useful tools. Also, there is a professional version which provides more features.

- ***Multiplatform***. *PyCharm* is compatible with Linux, Windows and MacOS operative systems.

Arduino IDE

Arduino IDE is the framework use to upload code to Arduino Boards. This tool is provided by the manufacturer, *Arduino*.

Arduino lets the developer to control all kind of sensors. This IDE allows the user to install the controllers needed to work with all available Arduino's boards. It has an assistant to install all the required libraries and drivers for boards.

The development with this *IDE* from scratch is possible, it is not required wide development skills. The programs are written in *Arduino* using *C* language. *C* is on a low level layer where the developer controls the functions of the sensors.

This *IDE* is available for all operative systems. The framework is uploaded to the official site of *Arduino's* webpage and it is free to use.

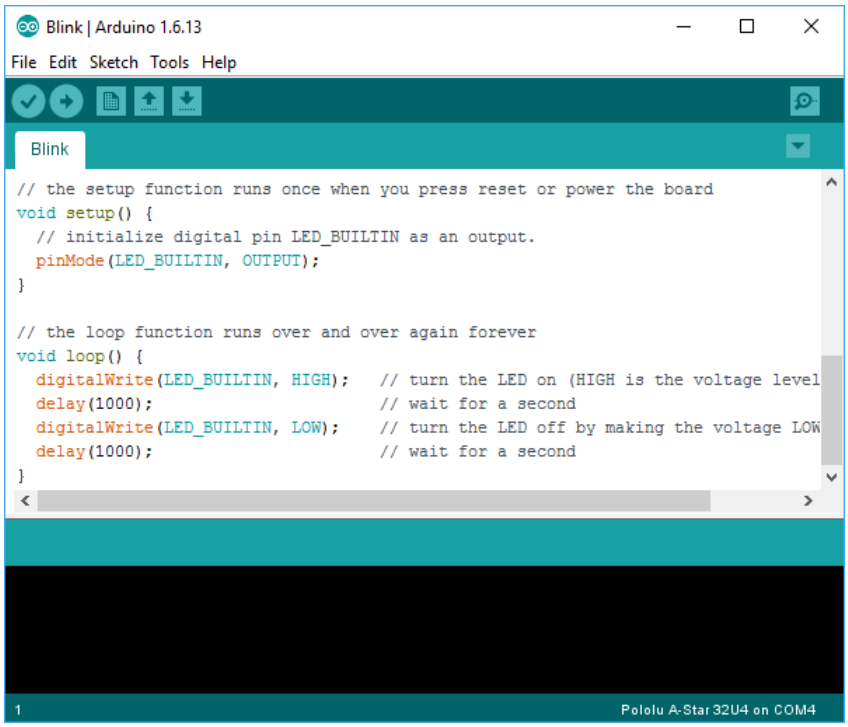

Fig. 5.7: Arduino IDE

Arduino provides to the developer simple instructions and tools to debug programs. It also has a serial console to debug the information measured by the sensors.

Notepad++

This tool is not mandatory for development but it is recommended. *Notepad++* lets the developer to program in all kind of programming languages to create software components.

There are other tools to replace this software such as *Sublime*, *Geany* or other open source applications. It was decided to use this program because of its simplicity. Also, there is an initial knowledge about this tool.

Notepad++ was used to develop the subsystem related to the *NodeJS backend*.

Filezilla

Filezilla is a program which allows the user to transfer files between two different devices using file transfer protocols. *Filezilla* was used to transfer all the software components to the *Raspberry PI 3* instead of typing instructions from the command line.

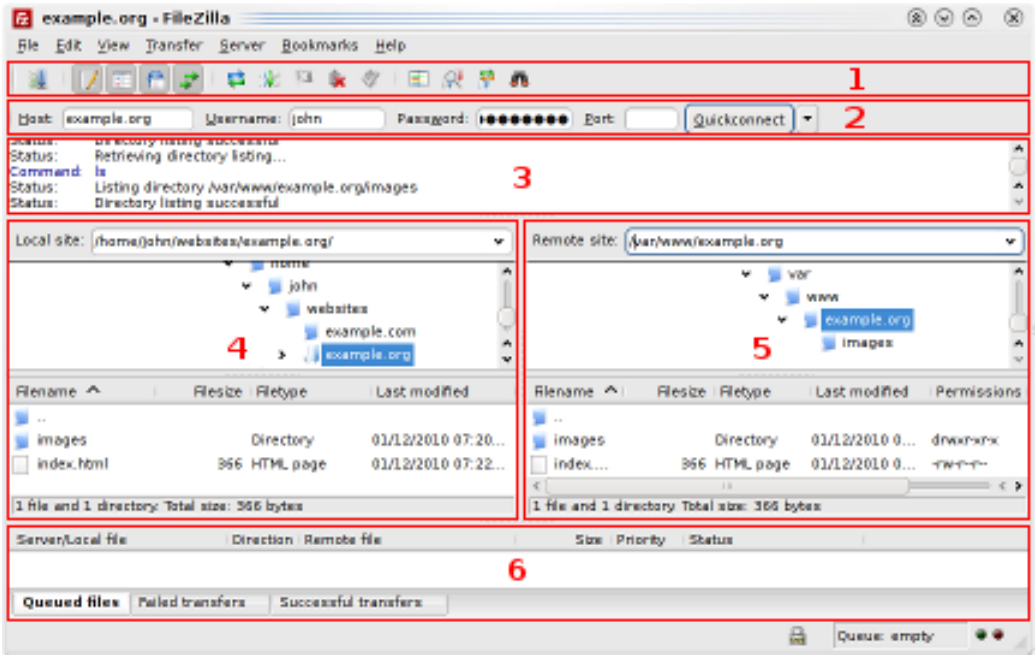

Fig. 5.8: Filezilla's Graphycal Interface

As it was mentioned before, there are other tools which provide similar features. It was decided to work with *Filezilla* because it is multi-platform.

Filezilla allows the user to connect to devices using network protocols like *FTP* or *SFTP*. All the required transfers were done over *SFTP* protocol. This protocol requests authentication.

With *Filezilla* the user has access to the main file system of the connected device. The access levels are defined by the permissions of the users.

Putty

This software is used to manage remote connections to hosts. This program is required for *Windows* systems. Windows does not support remote connections to hosts using the *cmd* terminal.

With *Putty*, a *Windows'* user can create a *SSH* or a *Telnet* connection to a remote host. It also supports serial communications. All the remote connections to the Raspberry PI are established via *SSH* communication. This protocol is reliable and secure.

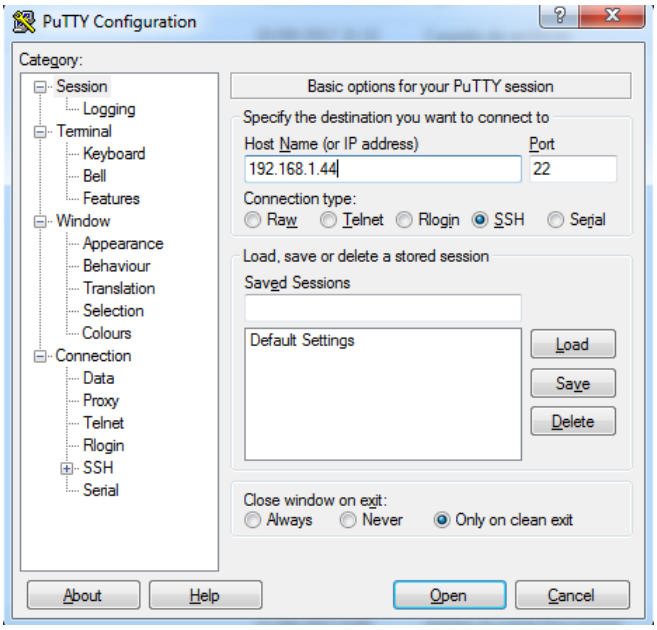

Fig. 5.9: Putty Software

Putty is available from its main webpage. It is a free program and it is simple to use.

MacOS and *Linux* operative systems do not need this software. These platforms support remote connections from the command line.

Chapter 6

Results

The following section is dedicated to expose the different results obtained after using the developed *Measure System*. The validation of the system is determined by the obtained results. In case the results are valid, the system could be applied to future studies.

There is comparison of the results obtained with the ideal model used by this study. The ideal model is the base to determine if the developed system is applicable to future studies. This section explains the selected use case to test the system. With the obtained information it is possible to decide if it is affordable to use the developed system in future studies.

6.1. Studied cases

The purpose of this project is to find a solution that can be used to study the properties of soils. The solution has to obtain accurate measures, manageable and affordable to apply it to new use cases.

This section describes a real use case used to test the system. The chosen use case allow us to determine if the solution obtains realistic values or not. This use case has an amount of values based on time which allow to analyze if the results are valid or not.

The variations of the measures produced by the sensors are important too. These variations are critical to decide if the used sensors are appropriated for the system.

There are lot of factors which will influence in the produced results. The results could change if we consider the following factors:

- Inclination of the substratum. The humidity will change depending on how tilted is the substratum. Depending on the area we are analyzing, the measures could change.
- Current state of the sensors. The sensors could be damaged if they are connected for a long time. It is not the same to use a new sensor to use a degraded

sensor which was connected to the system since months.

- Type of substratum used in the study. Depending on the substratum we are using in the study, the results could change getting different values. Not all the substrates have the same drying behaviour. It is relevant for the study because the produced values could change.
- Environmental temperature. If there are changes in the environmental temperatures the results can vary. If the environmental temperature is constant the results will follow the ideal model used for the study.

To do this study, it was designed a prototype with different soils. This prototype has filters made by other soils like gravel or soils with different granularity properties. The prototype simulates an inclined terrain where the drying process changes depending on the concentration of the water when the terrain wets.

6.2. Analysis of the obtained results

After studying the different use cases that could be presented in the system, the obtained results are studied to decide if the system can be used for future studies where materials are involved.

To do this study, a representation of the information is requited. This representation has to be done with graphs that can allow us to determine if the measures are reliable or not.

The results obtained have to be compared with results obtained by accurate systems. The results obtained by other systems have to be represented in graphs too. To do this comparison, our graphs are built with the information monitored by the system. These graphs are done with Microsoft Excel. To build these graphs, a selection of the monitored information is required. This selection includes information where the changes associated to the humidity of the soil are presented. The evolution of the humidity of the soil depends on the time where the measure was done.

When the selection is done, it is required to select the desired graph to represent the selected information. The chosen graph allows to compare the results with other studies or measures.

The following picture represents the ideal dried graphic used as reference. This graph represents how the vales can change in an ideal study.

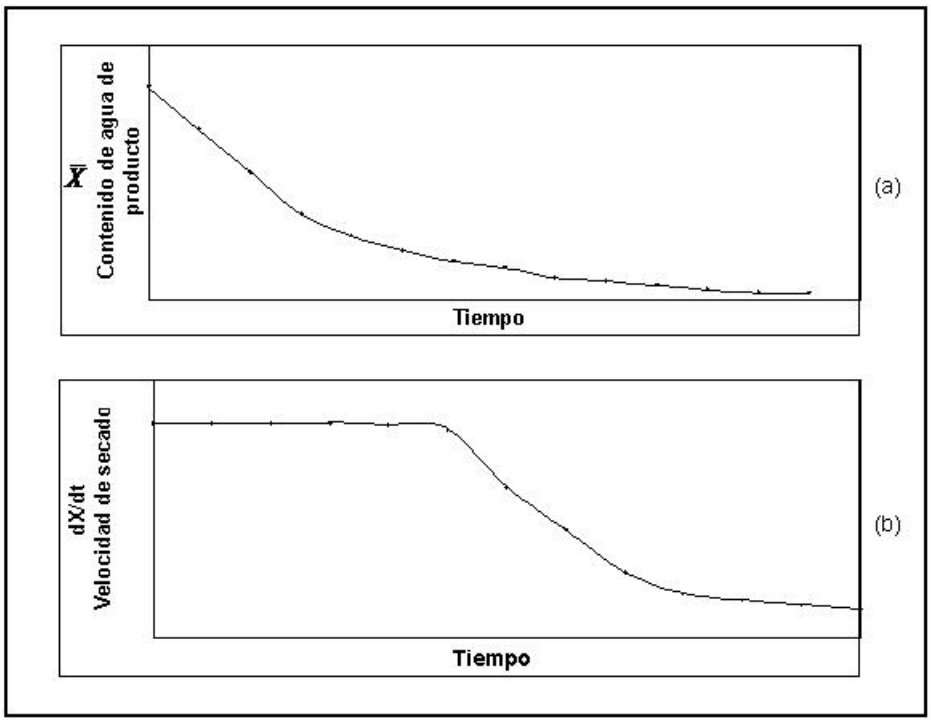

Fig. 6.1: Ideal dried graphic

As it is possible to see, this picture allows us to know how is the evolution of the measure when the time changes.

In the following picture, it is showed the results obtained after using the developed system.

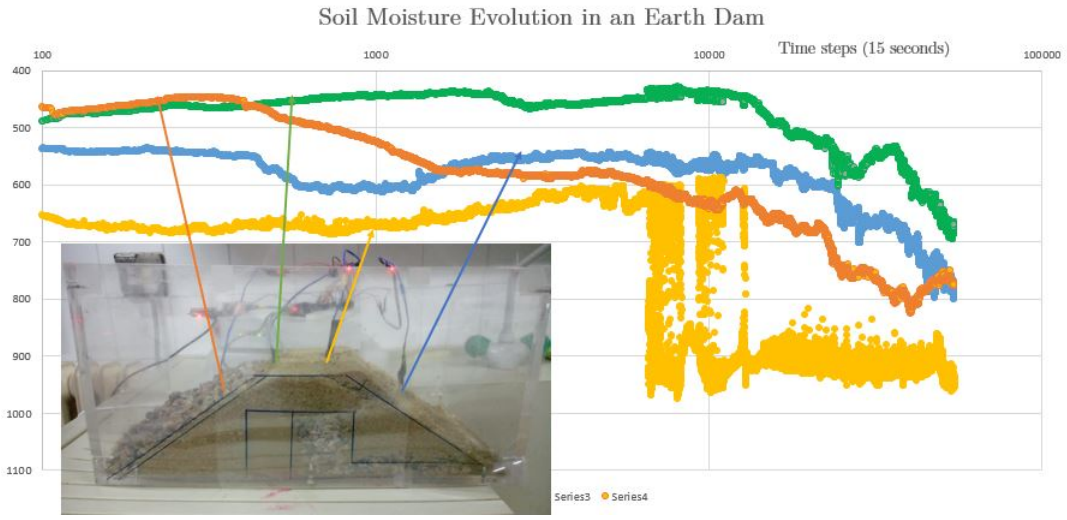

Fig. 6.2: Soil Moisture Evolution in an Earth Dam

This graph represents the evolution of the humidity of the soil during the time. There are four humidity sensors connected to the chosen soil. These sensors are allocated in different positions to obtain different values depending on the inclination. The prototype represents a real use case, where the inclination is considered and affects to the measured values. Depending on the inclination, the humidity changes. Higher values are tied to the driest values of the soil, and lower values represent the higher humidity values. The evolution of the humidity of the soil depends on time. Also, the chosen prototype has an amount of soils. These soils are the following:

- Soil with gravel.
- Soil with big granularity.
- Soil with filter.
- Normal soil.

The behaviour of the drying process depends on the type of the used soil. Soils with filters dry faster than soils without filters. Also, the granularity of the soil influences on the drying process. Soils with more granularity dry faster than soils with

few granularity.

The prototype has four humidity sensors, which are represented with four different colours. The values associated to the used sensors are described as follows:

- Orange sensor. This sensor dries faster than the other sensors although it is located next to the water. The soil measured by this sensor has a bigger granularity which allows a faster drying process.
- Green sensor. The core of the prototype is measured by this sensor. This core maintains the humidity better than the other measured parts although it wets slowlier than the other soils.
- Yellow sensor. The variations represented by this sensor are normal. This soil has a filter composed by gravel. The filter changes the humidity of the sensor drastically.
- Blue sensor. The soil measured by this sensor dries slowlier than the other sensors because it has no filter to speed-up the drying process.

It is possible to see that the represented graph is similar to the one used as model of study. The time slot used to represent the measures of the system is minor than the one used by the ideal model.

The comparison of the results of the ideal model with the studied model shows the validity of the designed solution. The developed solution is applicable to future studies where it is required to monitor humidity values. The curves of the drying process obtained with the developed system are similar to the graphs obtained by reliable systems.

Chapter 7

Conclusions and future lines

This chapter is dedicated to present the different conclusions of the study. Also, there is an explanation of the different future lines of this study which are opened to improve the developed system.

7.1. Conclusions

In this section the main conclusions of our study are presented. These conclusions refer to the objectives described in the first chapter of this document:

1. *Apply Informatics Engineering knowledge to solve real life problems.* The developed system looks for a solution for the explained problem. The applied knowledge solves the problem defined by the user, designing a system which is able to measure environmental variables related to soils.

2. *Estimate the domain of the problem to be solved.* The exposed problem is delimited and has a defined domain. It is affordable in terms of time and resources. The time invested to solve the problem is realistic. The resources provided by the University are relevant to design a solution for the commented problem.

3. *Consider and analyse different technological alternatives.* It was required to study several technologies to consider which one is the correct one to designed the purposed *Informtion System.* It was possible to analyze and decide the appropriated technology to solve the problem.

4. *Design Information Systems useful for the automation of process.* The discussed problem had an inconvenient related to handwork. The measures were done using slow techniques which require to invest time and energy. The designed system, automates the measurement process working with low consumption sensors. Also, the hardware used by the system is based on low energy microcontrollers.

5. *The price of a sensor does not determine its validity for the project.* It was proven that using a low cost sensor does not imply that the information obtained cannot be valid and useful.

Using statistical analysis tools allows us to obtain acceptable results.

6. *The use of new technologies in the field of Computer Engineering are completely applicable to other Engineering fields.*

   Using NoSQL databases and free hardware provides us solutions that can be applied to other engineering areas. With NoSQL databases we can monitor huge amounts of information that can be analysed.

   Using free hardware allows the creation of Information Systems that can be adjusted to other user needs.

7. *Flexibility is quality too.* Developing an Information System like the system that is referenced in this document, allows the user to realize that his needs can be covered and also, the user can adapt his future needs to an Information System that it is continuously evolving. The quality of a product is determined by the final user.

8. *The requirements of the users are the Computer Engineer's business.* Bringing informatics solutions to users with requirements that are evolving continuously is the real market niche for all Informatics Engineers. Requirements do not have to remain static during all times. It is normal that the bigger projects are developed when the users have requirements that are continuously evolving.

7.2. Future lines

After the analysis of the conclusions, the possible future lines are presented:

1. ***Increasing the computational resources of the Information System that has been designed***. Like every *Information System*, it is always possible to obtain a higher number or computational resources. In this work, it was developed an *Information System* with a computational core based on *Raspberry PI 3*. The *Raspberry PI 3* is a simple processing module with limited specifications. If needed, it is possible to upgrade the system with a more powerful

hardware. Some of the specifications that could be improved are the following:

- *A more powerful processor.* The processor of the Raspberry PI 3 has four cores based on an ARM architecture. ARM architecture is characterized by processors suitable for mobile devices with low energy consumption. It is recommended to use a device with a processor with an architecture based on 8086 processors. These processors are more powerful and can do more complex calculations.
- *More RAM memory.* The device used in the prototype has 1GB of RAM memory. Despite its RAM memory the device is capable of dealing with low volumes of information. It is able to query and retrieve information from the used database, storing ambient information. The problem might arise in the future when the data will grow to higher volumes of information (like it happens in systems used in *BigData*) and the system would have slower response than expected. To avoid this problem, it would be necessary to increase the RAM memory.

2. ***Data redundancy***. This requirement was not deployed in this prototype. It would be necessary to create a redundancy in the database with the information of the ambient variables to avoid a possible loss of information if the devices used for storage fail. to do so, it is viable to use a storage system that allows the replication of information in the database.

3. ***Replace sensors***. In this prototype low cost sensors that have a limited lifetime were used. To avoid the constant replacement of the sensors it would be necessary to use sensors with better quality. These sensors will have a longer lifetime than the sensors used in this prototype.

4. ***Using Map Reduce algorithms***. The use of these algorithms could optimize the time spent by queries when it is needed to retrieve the information stored in the system. These algorithms are very common in *BigData*.

5. ***Development of the Front End***. The font end of the graphical interface was not developed in this study. Consider this feature is reachable future line.

6. ***Improve web back-end***. The back-end of this project is developed but with limited features. Considering this feature will open new features for the future system.

Chapter A

Appendix

A.1. User manual

This section is dedicated to explain most common actions presented in the system. In the following points it will explained this information:

- How to reboot the full system.
- How to manage MongoDB.
- How to access remotely to the system.

Also, it will be explained more details we consider which are important in the system.

A.1.1. How to reboot the full system

A completely reboot should be likely established. For that purpose, we need to consider we have different subsystems that can lead to executing problems in the full system. Those subsystems are the following:

- Network system.
- Database system.
- Sensor system.
- Computing system.

In the following section how to manage with this systems and operation is explained.

Restart network system

At first, we are using a system which manages heavy information. In some situations, the system should support a higher number of petitions to queries to the database or intensive transmission of information that can generates failures in the network architecture.

In this case, a restart of the whole network is sometimes required. To do it, it is necessary to know the different elements which composed each subsystem. The elements we have in our system are the following:

- Router.
- Raspberry PI.
- Different devices connected to the router:
  - Smartphones.
  - Computer.
  - Sensors.

The router is the network brain of our system due to it control all information related to the network. It will allow to stablish the among communication with all the devices of the system.

If we have to restart the network, the steps are:

- Press the power on button of the router.
- Wait some seconds before plug in it again.
- Press the power on button of the router again.
- Wait two minutes until the devices are connected again to the router.

If everything is correct, the router will allow the communication between the different devices of the network again. As the configuration is done by the router, user does not have to care about the configuration of the different devices.

Restart the database

When a restart of the database is required, we need to control the Raspberry Pi which can be done following different options:

- Remote access to the Raspberry Pi:
  - Via Linux terminal.
  - Via Mac terminal.
  - Via Putty. This software is available to download it, and it is really useful for Windows' users.
- Connecting peripherical devices to the Raspberry Pi:
  - Monitor.
  - Keyboard.
  - Mouse (it is not mandatory but maybe it is useful to have one).

In the following sections it will be explained how to create a remote connection to the Raspberry Pi and how to access with different peripherical devices to the Raspberry Pi.

Restart the sensor system and the Raspberry PI

This is the most common way to restart the system. Usually, it is only needed to restart the Raspberry Pi with the Arduino which is connected to the Raspberry Pi.

To do it, next, it is required to follow these steps:

1. Disconnect the Raspberry Pi from the current.
2. Disconnect the Arduino from the Raspberry Pi.
3. Wait some seconds.
4. Connect again the Arduino to the Raspberry Pi.

5. Plug in the current of the Raspberry Pi.

6. Wait two minutes until the Raspberry Pi finishes its booting operations.

After applying these steps, we ensure the system has been reset.

A.1.2. How to manage MongoDB

To manage databases, it is required to know the status of them. MongoDB is a database system which allows us to store different kind of information. In this section, it will be explained how to know different information about our MongoDB database system.

See the current version of MongoDB

In case we a working with a *Linux* system, the steps after opening a terminal are:

sudo mongod –version

This command will show the current version used by MongoDB. This operation is useful to know which *API*[**?**] (*Application Programming Interface*) to use to store, query or maintain the different information we have stored in our database.

It is recommended to use MongoDB version up to 2.6. If we are under 2.6 version, it is recommended to update it to 2.6 or greater versions.

Check mongod service

In case the system suffers a problem related to a restart or a power off of our system, the services can go down. This is typically a problem we have when we are managing Information Systems.

If we need to check if our MongoDB database system is working, we need to follow these steps:

- Open a Linux terminal.
- Get administrator privileges using *sudo su*.
- Run *ps -ax | grep mongo*.

With these commands, it would be possible to see if MongoDB is running in our system. If we see something like the information described in the following image, it will mean MongoDB is running in our system.

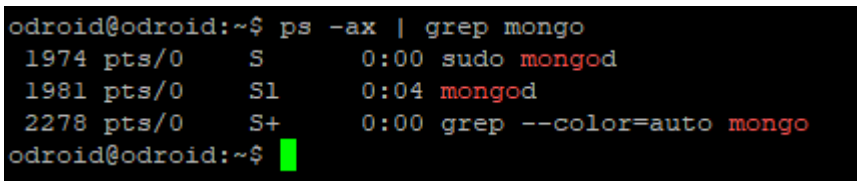

Fig. A.1: How to display MongoDB service when it is running

As we can see in the picture displayed before, there are some services running with the name **mongo**. The last one is irrelevant for us, because it is the result obtained after executing *ps -ax | grep mongo*. The information which is important for us is the different lines displayed before, *sudo mongod and mongod*. These lines are telling us that MongoDB is running properly in our system.

The steps described the MongoDB status. There are several ways to check. For example, after the installation of MongoDB using the package manager of Linux, the system will create a new entire into */etc/init.d* and in services which will allow us to identify if the service is running. This entry is name as service. Linux has lot of services, and it is possible to query if there are services running in our machine.

If we have created a service entry in our system, it is possible to query the status of the service using sudo service [name of the service we want to query]. To know if MongoDB is running, we can type *sudo service mongod status*. If the service is installed in our machine, it will be possible to see the following information.

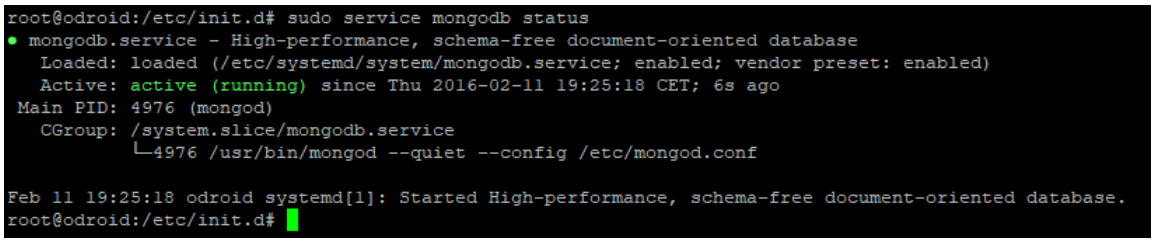

Fig. A.2: MongoDB's service is active

To display this information, it is needed to type in the Linux console *sudo service mongodb status*.

As can be seen, the server is running, showing its *pid* (identification associated to MongoDB's thread) and its executable (allocated in */usr/bin/mongod*).

Start MongoDB service

This operation is usually useful when there is a problem on the system (for example, MongoDB has not started at the beginning of the booting process) or when there is no configuration to run it automatically.

To do it, it is necessary to type in the console *sudo service mongodb start*. With this command we will tell *mongodb* it has to run.

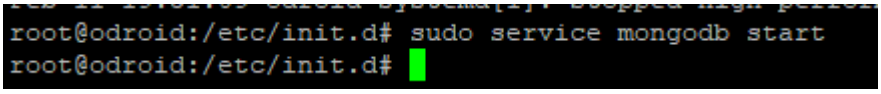

Fig. A.3: Start MongoDB's service

To check if MongoDB is running, it is necessary to type *sudo service mongod status*. If we noticed the system running after executing this command, we do not have to be worried about the startup of the service.

Stop MongoDB service

Sometimes, it is not only need to start the daemon process. There are several situations where we want to stop the database server, for example, if there is a query that has blocked the database. In these cases, it is useful to have control of MongoDB's service.

To do it, we just only need to type *sudo service mongod stop*. It is useful to check if the service has stopped or not. To do it, we just have to type *sudo service mongod status*. If we see some information like not running, then we can ensure the service has stopped properly.

In the following picture there is an example of the execution of this last command.

```
root@odroid:/etc/init.d# sudo service mongodb status
● mongodb.service - High-performance, schema-free document-oriented database
   Loaded: loaded (/etc/systemd/system/mongodb.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2016-02-11 19:37:13 CET; 1s ago
  Process: 5379 ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf (code=exited, status=0/SUCCESS)
 Main PID: 5379 (code=exited, status=0/SUCCESS)

Feb 11 19:33:43 odroid systemd[1]: Started High-performance, schema-free document-oriented database.
Feb 11 19:37:13 odroid systemd[1]: Stopping High-performance, schema-free document-oriented database...
Feb 11 19:37:13 odroid systemd[1]: Stopped High-performance, schema-free document-oriented database.
root@odroid:/etc/init.d#
```

Fig. A.4: Stop MongoDB's service

How to manage MongoDB with no services installed in the system

In some situations, it is possible to see that MongoDB has not a service architecture defined.

When there is an installation process in Linux, the system will create different link files, executable files and configuration files. Some of the files created in the installation process will be copied to the folder */usr* and the files related to configurations will be placed in */etc*.

MongoDB will create all the configuration files in */etc*. Here, it is possible to define the different parameters needed to start running the system like:

- Path of the executable file to start the database system.
- Path of the different log files where it is possible to see the debugging status of the database system or problems related to configuration.
- Path of the folder which is going to be used to store the different databases used by MongoDB.
- Configuration of the user.
- Configuration of the listen port used by MongoDB. This port is used to allow possible incoming connections to the database.
- Mode of execution of MongoDB. The different modes we have available are the following:
  - Verbose mode. This mode allows the user to see the output generated by the full system. It is possible to see the incoming connections, problems occurred during the execution of the database system, timeouts, logouts, logins, etc.
  - Silent mode. This mode does not display information about the execution of the database system. This is the default mode used by the main program.
- Configuration used to authenticate the different user connected to the database.
- Allowed IP addresses to access to the database.

All this information is stored in */etc* in one file named **mongod.conf**. It is possible to edit this file if it is needed to do it, but it is necessary to be careful if we apply any modifications because it is possible after those modifications the system will not run correctly.

There is another important folder that is needed to mention in this section. This folder is *init.d* and it is stored in */etc*. This folder will contain all the different executable shell scripts needed by the system. Obviously, MongoDB will create some

files in this folder. The most important file to consider is *mongodb*. It is possible to control all the operations done by MongoDB with this file. The possible operations are start, stop, restart and display status. Those operations are executed as follows:

- Type in the console *sudo /etc/init.d/mongodb start* to start MongoDB. After the execution of this command, MongoDB must run in the background.
- Type in the console *sudo /etc/init.d/mongodb stop* to stop MongoDB. This command does the same operations described in the service section of MongoDB.
- Type in the console *sudo /etc/init.d/mongodb restart* to restart MongoDB.
- Type *sudo /etc/init.d/mongodb status* to show the status of MongoDB. The information displayed will exactly the same if we execute *sudo service mongod status*.

This knowledge is required to manage a MongoDB database system.

It is important to know that all the commands described in this section needs the sentence *sudo*. In all Linux system, if the user wants to modify some features related to configuration of some programs, it is needed to use it. This capability allows to control the access to certain files of folder that could be damaged by some users.

Check MongoDB network connection

As it was explained before, sometimes it is necessary to check if MongoDB is running in the system or, for example, to reboot it in case we have problems when we are storing information in our database. In some cases, MongoDB is running although it is not possible to access the databases. The common reason is the rejection of incoming connections. When we have this problem, it is necessary to check if MongoDB has its port reachable. To do it, it is necessary to execute the following command.

sudo netstat -l | grep 27017

With this command, we will ask *Raspbian* if there are any processes using this port. If it is not displayed any information after executing this command, then we can be ensured the system is not allowing incoming connections to the databases we have.

If we have this problem, then we should restart again MongoDB as it was explained as before.

A.2. Solution to frequent problems

This section of the user manual is dedicated to explain how to solve common problems we can find when we are working with MongoDB.

It is often to have problems related to network communication, storage or problems related to how to manage the MongoDB service installed in our machine.

In the following subsections it will be explained how to deal with all these problems.

A.2.1. Remote access to Raspberry Pi

As we explained before, there are several tools to allow remote access to the Raspberry Pi. The Raspberry Pi allows by default demote management, and this is one of the most powerful capabilities it has. Thanks to remote access we can manage the full system.

Then, it would be explained how to connect to Raspberry Pi using *Putty* due to is the most extended protocol for remote connection for Windows' users.

To do it, first it is needed to download this software from its webpage, after we have downloaded it, we have to open **Putty**.

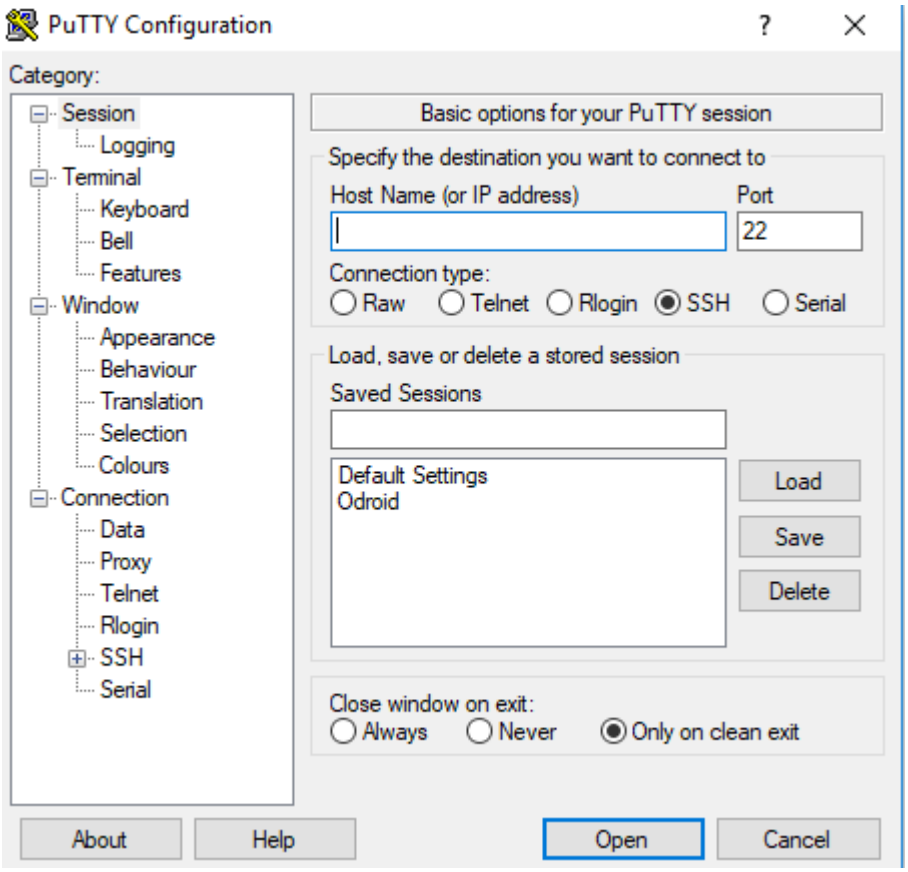

Fig. A.5: Putty's interface

As it is possible to see in the picture, we have some fields we have to complete. The different fields we have are the following:

1. Host name or IP address. This field has to be fielded with the IP address used by the Raspberry PI. In our case, the IP address of our Raspberry Pi will have the structure *192.168.1.X* (the X has to be replaced by the last number of the IP address used by the Raspberry PI). By default, the Raspberry Pi uses *192.168.1.50*. If the user does not know the network address used by the Raspberry Pi, it is needed to read the section How to see Raspberry Pi's network address.

2. Port. It is not needed to modify this port because the Raspberry Pi uses by default the port *22* for **SSH** connections.

After filling this information, we have to press Open. When we have pressed Open it will be displayed a console. This console will have the following structure.

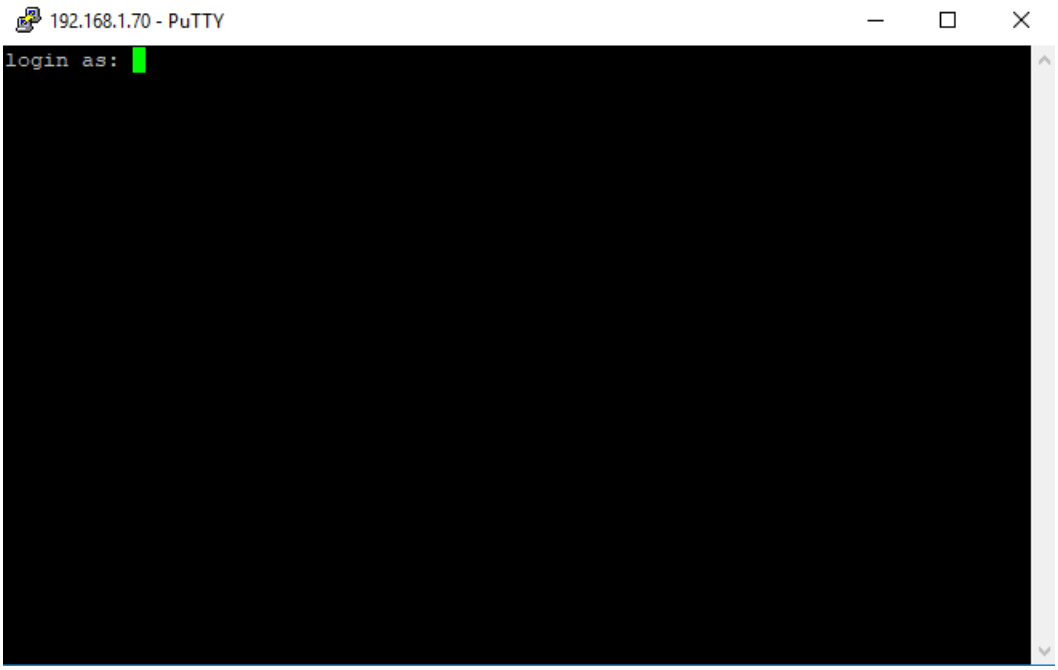

Fig. A.6: Loggin into Raspberry Pi

Here, Raspberry Pi requires the username of the system. In this step we have to type *pi* (it is important to type it in lowercase).

Next, Raspberry Pi will ask us about the password. In the following picture it will be displayed the information related to the password request.

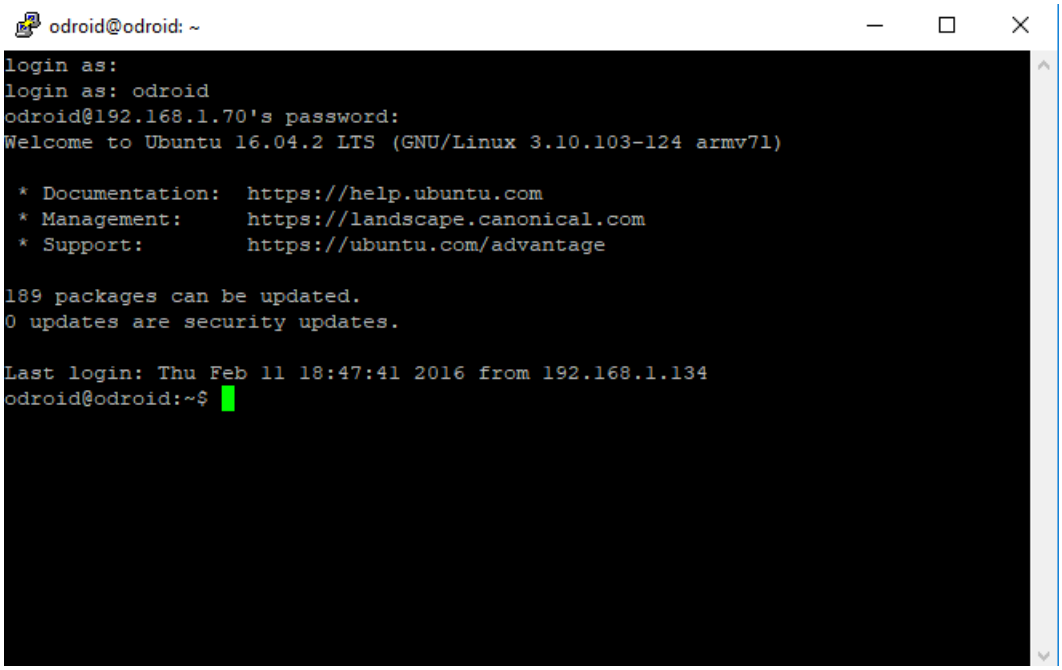

Fig. A.7: Loggin into Raspberry Pi with its password

Now we have to type raspberry (be careful with uppercase and lowercase), and the prompt of the system will be displayed. To restart the database daemon, it is necessary to apply the different steps described in the section Check MongoDB's status.

A.2.2. Moving the measure system

In some situations, it is necessary to turn off the full system because it is needed to do some hardware modifications. To do it, it is important to turn off the different components we have in our system to avoid electrical problems or storage problems (it is possible that after disconnecting some devices in our system, the MongoDB daemon could write corrupt information in the database we are using). To avoid these problems, it is important to do the following steps:

- Turn off the database. This step is not mandatory, but it is useful to know we could do it to be ensure that the daemon is not doing any important stuff. To do it, we need to execute the command: *sudo /etc/init.d/mongod stop*

- Turn off the Raspberry PI. It is really important to do this step if we want to disconnect all the sensors or we want to move the full system. If we do not follow this step, it is possible we can break down the database which is being used by MongoDB or it is possible to generate wrong measures in our database. If we want to turn off the Raspberry PI, it is necessary to execute the command: *sudo shutdown now*
- Disconnect the rest of the device from the electrical supply.

It is important to know that after doing reset explained before, it is necessary to fix the date of the system. This step must be done because our system does not have any connection to Internet to synchronize its date information with any Time Server. To do it, it is necessary to execute the following command.

sudo date -s 'YEAR-MONTH-DAY HOUR:MINUTES:SECONDS'

An example of the execution of this command is for example:

sudo date -s '2018-05-25 18:40:10'

How to see Raspberry Pi's network address

In some cases, the user does not configure a static network address in the Raspberry Pi. Also, this configuration is missing and the Raspberry Pi has enabled *DHCP* configuration. Due to this problem, the user has to look for the device in the network.

When this problem happens, it is possible to check the configuration with the following methods:

- Direct connection to Raspberry Pi. It is possible to have a direct connection to the Raspberry Pi using a monitor, a keyboard and a mouse. If we decide to use this solution, we need to follow these steps:
  - Open a Linux terminal going directly to applications and after seeing Terminal.

- When we have opened the Linux terminal, we have to type *ifconfig* in the terminal. The information displayed in the screen is:

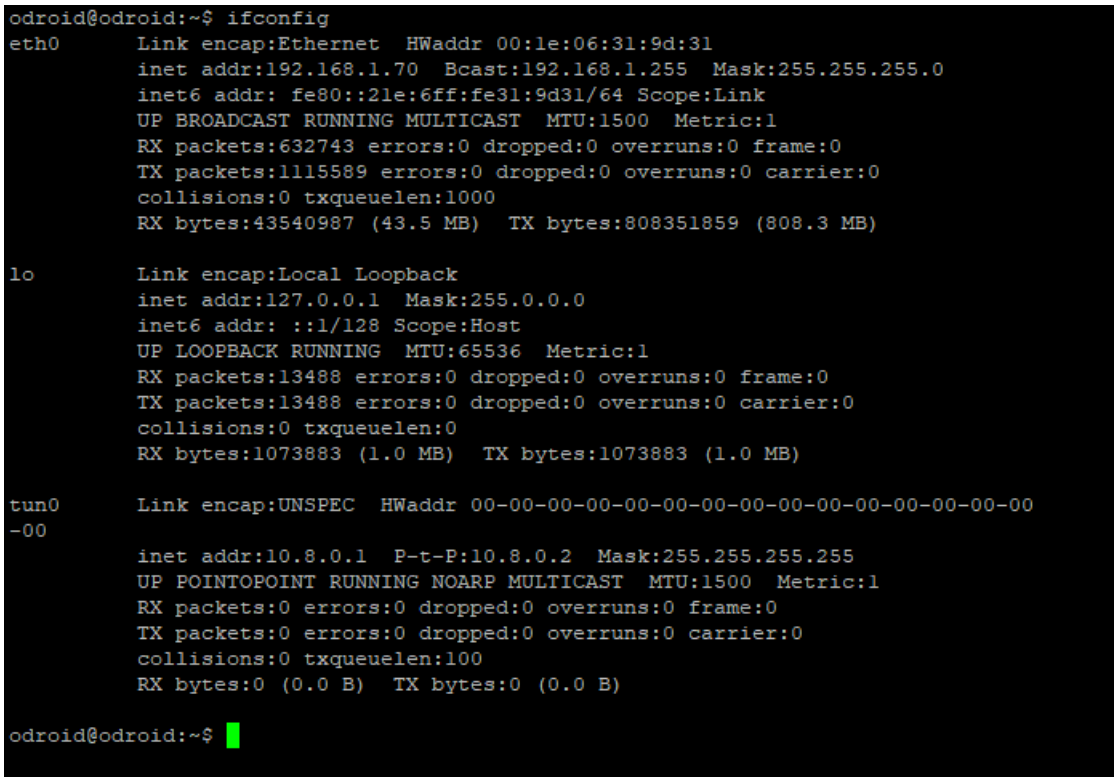

Fig. A.8: Network adapters Raspberry Pi

- As it is possible to see, some information of the Raspberry Pi is shown.
- Next, we need to change the IP address of the interface *eth0*. In this interface it is possible to see its IP address (represented in the picture as *inet addr: 192.168.1.X*). This IP address is the one used by the Raspberry Pi.
- If the user does not know how to log in the Raspberry Pi, please read the section Remote access to Raspberry Pi.

- Check the IP address using the router. This option is recommended if we can not connect to the Raspberry Pi. For that purpose, the next steps are required:

- Open a web browser (for example **Firefox**) and type in the address field **192.168.1.1**.

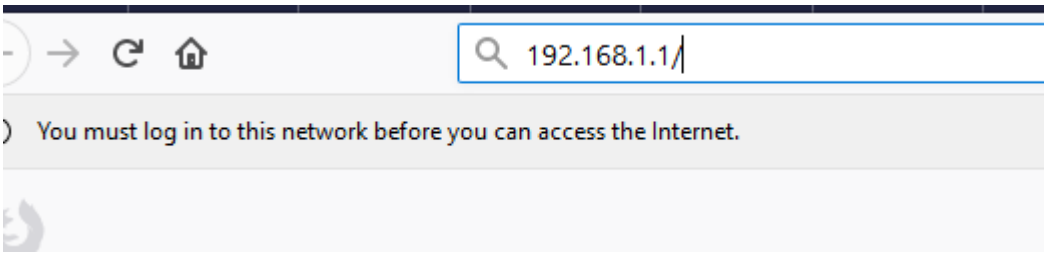

Fig. A.9: IP address of the router

- After entering in the address displayed in the pictured, it will be shown a login dialog. In the login dialog we need to enter user and the password of the router:
  - User: *admin* / Password: *admin*
  - User: *admin* / Password: *1234*
- After typing this information, we will see the information associated to the Raspberry Pi.

How to proceed when the database is corrupt

In case we want to move the system and we do not have knowledge about it, we can generate several problems in our database, as storage problems. When it happens, normally, **MongoDB** cannot run and we will see in **3TStudio** there are no new information in the database. When we have this problem, it is necessary to remove the corrupt information we have in our database. It is important to know that we will lose the information we have stored in our database:

1. The most important thing is that we need to stop MongoDB daemon. To do it we need to execute:

   a) *sudo /etc/init.d/mongod stop*

2. We need to move to the folder where the database is saved. To do it, it is necessary to execute the following commands:

   a) *cd /home/pi*

   b) *cd databases*

3. When we are in the database folder, we need to remove all the information related to the database which is corrupt. To do it, we need to execute:

   a) *sudo rm material**

4. When we have removed all the files related to the database which we were using before, we need to start again MongoDB daemon. To start again MongoDB daemon, we need to execute:

   a) *sudo /etc/init.d/mongod start*

5. Wait until thirty seconds until the database is running and check if it is possible to connect to it using **3TStudio**.

A.3. Limits of the system

In the following section it will be explained the different things we need to consider about our measure system. All these items should carefully consider if we want to do a new modification in the system:

1. If it is necessary to move the full system, it is important to disconnect the different components described in the points explained before.

2. To disconnect the different components we have in the measure system, follow the different tips we have explained before.

3. If it is necessary to modify or to replace some components in the system it is recommended to disconnect the system as it was explained before.

4. It is necessary to extract the information which is stored in the database periodically. The system can store up to ***two Gigabytes*** of information.

5. Do not let the system to reach ***two Gigabytes*** of information in the database. If it happens it is necessary to clean up the database and it will be lost the information stored.

6. If there is any problem in the communication to the Raspberry PI, be ensure that the service of the database is executing in the background.

7. If the Raspberry PI is disconnected from the supply, it is necessary to fix its date information. Remember it has not a Time Server to synchronize its date information.

8. The system has no Internet connection. If it is needed to modify something in the system it is recommended to create a Virtual Machine to download new packages.

9. If MongoDB is updated it is necessary to check the compatibility of new packages downloaded. Also, the API used to store the information in the Database.

10. In the future it is recommended to update MongoDB to a 64bits version. The 32bits version only allows you to store up to two Gigabytes of information. By default, *Raspbian* packages install the 32Bits version in the system, so we have to be careful when we want to install MongoDB in our system.

References and bibliography

[1] *Arduino Mega 2560 Rev3.*

[2] jjtorres, *Hardware para novatos (VII): Arduino ¿qué es y cómo funciona?* Mar. 2014.

[3] A. Ltd, *Leadership – Arm.*

[4] A. Ltd, *A-Profile Architectures.*

[5] *¿Qué es Big Data?* June 2012.

[6] G. PowerData, *Big Data: ¿En qué consiste? Su importancia, desafíos y gobernabilidad.*

[7] *How Applications of Big Data Drive Industries.* Oct. 2015.

[8] *El concepto NoSQL, o cómo almacenar tus datos en una base de datos no relacional.*

[9] *NoSQL vs SQL: Principales diferencias y cuándo elegir cada una de ellas.* Nov. 2015.

[10] *NoSQL Databases Explained.*

[11] *Chapter 1: What Is an Information System? \textbackslashtextbar Information Systems for Business and Beyond.*

[12] *Information system.* Sept. 2017.

[13] "Smart Room Temperature Controller Atmega | NevonProjects."

[14] diy_blokeFollow, "Watering Your Plants With an Attiny Microcontroller."

[15] N. Benn, F. Turlais, V. Clark, M. Jones, and S. Clulow, "An Automated Metrics System to Measure and Improve the Success of Laboratory Automation Implementation," *JALA: Journal of the Association for Laboratory Automation*, vol. 11, pp. 16–22, Feb. 2006.

[16] *Métodos para compactación de suelos.* July 2014.

[17] O. T. d. Tuy, *COMPACTACION DE SUELOS.*

[18] *Atlas de Petrología Sedimentaria - Rocas carbonáticas - Diagénesis - Compactación.*

[19] *Arduino - Reference.*

[20] *Arduino - Home.*

[21] A. Ltd, *Linux and Android Application Development – Arm.*

[22] *Arduino - Software.*

[23] *Reinventando la gestión de datos.*

[24] *The MongoDB 3.4 Manual — MongoDB Manual 3.4.*

[25] *Query Documents — MongoDB Manual 3.4.*

[26] *Tutorial — PyMongo 3.5.1 documentation.*

[27] *pyserial 2.7 : Python Package Index.*

[28] *Installing Python Modules — Python 3.6.2 documentation.*

[29] T. p. developers, *pip: The PyPA recommended tool for installing Python packages.*

[30] *Products.*

[31] *Raspberry Pi.* Sept. 2017.

[32] *Raspberry Pi : Adafruit Industries, Unique & fun DIY electronics and kits.*

[33] *Microcontrollers, what is a microcontroller? 8 bit, 16 bit & 32 bit microcontrollers - Future Electronics.*

[34] M. Verle, *1.1 What are microcontrollers and what are they used for? \textbackslashtextbar Architecture and programming of 8051 MCUs.*

[35] *FrontPage - Raspbian.*

[36] h. online, *Raspbian: Version Stretch auf Basis von Debian 9 veröffentlicht.*

[37] *Node.js Introduction.*

[38] *Build Node.js Apps with VS Code.*

[39] *Node.js.* Aug. 2017.

[40] *¿Que es FrontEnd Y Backend en la programación web? \textbackslashtextbar Ser Programador.es.*

[41] *Todo lo que necesitás saber sobre backend \textbackslashtextbar All you need to know regarding Backend.*

[42] *Medir la humedad del suelo con Arduino e higrómetro FC-28.*

[43] P. Domone and J. Illston, *Construction Materials: Their Nature and Behaviour, Fourth Edition.* CRC Press, June 2010.

[44] *Arduino Soil Moisture Sensor.*

[45] *Effects of soil compaction and forest floor removal on soil microbial properties and N transformations in a boreal forest long-term soil productivity study - ScienceDirect.*

[46] *What is an API? In English, please. – freeCodeCamp.*

[47] *17 important case studies on Big Data.* Aug. 2014.

[48] *Autonomous Driving – five steps to the self-driving car.*