Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Ground-based drone for security and control

Degree: Grado en Ingeniería de Computadores (Plan 2014)

Academic course: 2022-2023

Author: Sergio Aranda Lizano

Tutor: Alejandro Leo Ramírez

0. Index

Picture index

1. Objective

We have noticed that there are not a lot of ground-based drones in the world and even less in the security field. We also notice that the security field has not improve a lot in years, they still use the same fixed cameras with wired connections, or they are changing to smart home security systems that still use fixed cameras and sensors to function[1][2], using security workers to wander around the areas of the workplace or warehouses. Our objective is to design a system that can be used in a lot of different workplaces based on ground drones. This will be achieved with three tools: the device (ground-based drone), the Android application and the computer application.

This system will be able to cover the security of a lot of different workplaces, focusing on taking care of the hard points to see in the security cameras or patrol areas that are too big to be supervise with cameras and too expensive to have human security. In this regard, we want to develop a system that is cheap enough that every small company should be able to afford it, this can improve the global security of warehouses and small industrial places.

This project can be also used for population security having the drones wondering around the cities inside the dangerous areas without having to lose human lives in those areas. The police forces could buy the system and use it as a support tool for their operations and to patrol the cities as an easy-to-use tool to have an overview of the situation of the city and their locals; this would help to have more information and have the human patrols move towards a more responsive force rather than just random patrols around the areas.

To reach this objective, we will develop a full hardware and software this will join to form a full system.

2. Summary

Nowadays, security is one of the key features requested by every individual, which make it easier to invest and develop in new answers to the security systems. Almost every thief and security company know that the old security systems such as cameras may not always work to prevent the robbery and destruction of personal resources.

In this document we are going to explain the development of this project assign by the subject Final Degree Project (FDP), in which we have developed a ground-based drone with an Android application and a computer-based application, all focused on responsiveness and hardware-based commands.

The document was divided in the next chapters:

1. Introduction and first approach to the problem: In this chapter we are going to explain the meaning behind this project, which problems have we identified and which we are going to try to solve.
2. State of the art: This part is focused on showing the development and research other engineers and researchers have made on this topic, and after analyzing the answers provided to the security field we study if the project is viable to develop.
3. Requisites and specification: This chapter studies which functions and characteristics should the product have, we also study the specification of the project; this chapter focus on showing that the functions and characteristics of the project are made based on accessibility and easy to use systems.
4. Methodology: This section of the document focuses on the methodology of work and development followed in the development of this project.
5. Resources used: This chapter of the document shows the comparison between the resources used on the project and different alternatives. We also explain why we choose those resources and not the others.
6. Components of the systems: This part is focused on the specification of the components used in the system, the services needed, the libraries used during development, and other resources used as a whole.
7. Milestones and tasks: This section of the document list the milestones and task to be accomplished to be able to finish the project and makes a small description each.
8. Design of the system: This chapter explains how the system works and how is the design process we have followed to achieve the task and milestones showed in the section before this one.
9. User manual: This part shows a detailed manual of how to use the system, with pictures and explanations for each step in the process and easy to understand explanations.
10. Problems find during development and how they were solved: This chapter shows the description of the problems find during development and how they were solved.
11. Project budget: This section provides the final cost of the project and time cost of the project with the difference between the first draft and the last draft at the end of the project.
12. Environmental conscience and ethical responsibility: This chapter shows what is the ethical responsibility of this project and what is the environmental impact of the project.
13. Conclusion: This part of the document shows the conclusions reached at the end of the project and what we think about the project.
14. Future ways: This chapter will explain new paths of development and research on the topic of the project that could be followed in the future.

3. First approach to the problem

Urban security, understood as the absence of a serious threat with regards to criminality and the subjective perception of protection, today depends on various structural and local factors which we are going to explain next.

Urbanization worldwide has reached more than half of the world's population[3] and has become one of the first structural aspects that influence cities and their security. Since this trend will increase as shown in the reference before, criminality will be mainly urban and increasingly polymorphous, complex, and difficult to contain through the spontaneous social control which characterizes rural areas and small towns. This evolution forces us to reinvent the co-production of security in new contexts, with the participation of both state and local actors such as civil society.

In conclusion, city security is already an important issue and we as society are failing at targeting what we should improve to have secure cities and workplaces, this includes different types of new technology that improves security, such as flying drones, new camera systems.

This project was started to be able to show that there are new technologies in the field that focus on improving security around the world. As shown before, criminal activity has become more complex and harder to target in the megalopolises, this calls for a new round of improvement of the police force and the development of new technologies able to reach different standards that the human police force cannot achieve without more expenses and investment.

Our project is a ground-based drone with cameras and a low profile. Usually, cameras are fixed and connected via a hard wire which make it easy to the criminals to pinpoint where they are and dodge these cameras, and it is also easy to break the wire and have the fixed camera stop working. Moreover, the local forces and government budget for security cannot afford to put a few cameras per street in a city, with this type of system the government does not have to acquire a lot of drones, they can be used to secure the most insecure parts of the cities and also as a reinforcement of the fixed cameras in the rest of the places. If this technology is combined with flying drones and smart video data analytics, the cities will restore some of its security that they do not have nowadays.

But this project is not only useful for security in the cities, being a wireless drone make it as useful as possible to use in the industrial and agricultural field as well. Because it can achieve security footage from long distances without having to spend human resources.

There are a few characteristics that proves this kind of project useful, some of them are:

- The governments are trying to find new forms of security now that they have seen the percentage of criminal raise year by year[4].
- The policy and law forces have found that human security is not enough to stop modern criminal activity[5].

And the biggest characteristic of this project is that there are not low-cost commercial answers to this problem and we have not found more than 3 papers about this.

4. State of the art

In this next chapter, we are going to study the different papers and articles released that have some similarity to this project, because usually they have a similar objective or show some aspect that support the development of this project, such as power consumption, human adaptation, and acceptance of systems like this.

After researching and reading multiple papers we found that there are some studies about drones but not many of them really focus on real application purposes, we only found the drone that was used to harvest and a few small prototypes for IoT security.

We have found that energy demands of drone-based transportation is like the ground-based transportation even being more energy expensive to use flying drone transportation, which proves that using ground-based drones is more energy efficient that using flying ones[6].

In this demand of new systems, we have found that a lot off development and research tried to find uses for drones in management of global environments, such as temperature mapping and other environmental causes [7].

Other projects and research show us that the rate of success with the ground-based drones is high, such as the sweet pepper harvesting robot[8]. With this paper we display that using ground-based drones for real purposes and automatization is possible.

The society has already accepted the use of automated systems and human-robot interaction, as shown in [9] both females and males have a assess the trustworthiness of the robots above 5/7, which means that they trust them equals to a human counterpart.

Other researchers have already explained the utility and usefulness of ground-based security surveillance, the vast majority focus on ground-based IoT for new security[10]. With this researchers have found that the first challenge they had with IoT security was the privacy and security, there has been different approaches to answer this problem, such as creating a new model[11].

Different researchers have found the need for new security robot based either for workplaces or for public places. We have seen that using advance AI vision processing and ground-based drones it is possible to achieve a higher standard of security in modern society[12].

We have found that other researchers are worried about the use of drones and robots in the workplace and how to improve the security of the workplace and workers with the use of drones and artificial intelligence [13].

The new generation technology can support all these new systems based of microprocessors and microcomputers new development and improvement. New microprocessors are able to support new input and higher bandwidth without needing to create new advance architectures between different processors [14].

We found a few prototypes which show the need for a better prototype that really shows what ground-based system can do, both prototypes follow the same principle that we want to follow in the project, low-cost hardware to prove that ground-drones can be develop [15] [16] .

5. Requisites and specification

In this chapter we are going to explain the different requirements and specification for this project. These requirements have been divided into functional requirements and development requirements.

5.1 Functional requirements

As shown in last chapters, there are new modern problems in security and criminal activity, the biggest one is that nowadays criminal activity seems not easy trackable, and it needs a big number of human resources to follow its activity. In this project we wanted to cover as much problems as possible and solve them.

So, the functional requirements that have been decided are:

- Provide a wireless ground-based autonomous vehicle, drone that can be controlled with a low profile and easy to build hardware and low-cost.
- Provide a wireless video feed for the user to watch in a safe place, instead of being in the field surrounded by danger.
- Provide a new system for society to improve citizens security, by using smart analysis in the video feed.
- Provide a green answer to security by using electric and electronic devices, trying to reduce as much e-waist as possible.
- Provide an easy-to-use application for the workers to use without having to have any sort of unique knowledge or training to be able to use it.

5.2 Development requirements

In this chapter we are going to show the development requirements that we have found needed to finish this project. In this chapter we are going to show task that will be complemented by the Gantt diagram and the milestones in the new chapters.

1. Study of tools and features before development:

    - Learn how to program an ESP32 and an ESP82 with micro–USB B connection using the Arduino suit or similar development environments.
    - Learn how to connect the ESP32 to the WiFi connection using the ESP WiFi library.
    - Learn how to connect one ESP32 to another ESP32 trough WiFi or wireless connection using the ESP WiFi library.
    - Learn how to produce a video feed with the ESP32-CAM by using some library or programming it myself.
    - Learn how to create a web server with the video feed by using some library or programming it myself.
    - Learn how to turn on a sensor trough the ESP32 with HTTPS petitions, being able to translate them to usable data with their own library.
    - Learn how to receive HTTPS petitions trough WiFi with the ESP32 by using its own library.
    - Learn how to send HTTPS petitions trough WiFI with the ESP32 by using its own library.
    - Learn how to communicate with HTTPS trough the Android application or the PC application by using some library.

- Learn how to get the WiFi status in the PC application and Android application.
- Learn how to create the user graphical interface in both the PC application and the Android application.
- Learn how to send HTTPS petitions trough WiFi in the PC application and the Android application.
- Learn how to get the user input in PC application and process it.
- Learn how to get the user input in Android application and process it.

2. Software development of the hardware:

   - Development of the function inside the ESP32-CAM that provides the video feed using the camera that is included with the ESP32-CAM.
   - Development of the function inside the ESP32 - CAM web server to show the video feed and the configuration.
   - Development of the function inside the ESP32-CAM that provides the WiFi connection.
   - Development of the function inside the ESP82 that provides the WiFi connection.
   - Development of the function inside the ESP82 that provides the HTTPS receptor and the translation of the data to motor orders.
   - Development of the function inside the ESP82 that gets the data translated to motor orders and send the order to the motors.

3. Software development of the PC application:

   - Development of the graphical interface of the main page.
   - Development of the graphical interface of the configuration page.
   - Development of the graphical interface of the control page.
   - Development of the function that gets the input of the user after pressing the control button.
   - Development of the function that translate the user input and send it as an HTTPS petition.
   - Development of the function that access the video feed web server and shows it on the control and configuration page.

4. Software development of the Android application:

   - Development of the graphical interface of the login page.
   - Development of the graphical interface of the main page.
   - Development of the graphical interface of the configuration page.
   - Development of the graphical interface of the control page.
   - Development of the function that gets the WiFi status and suggest which WiFi connection the user should have to be able to control the drone.
   - Development of the function that gets the user input.
   - Development of the function that translate the user input and send it as an HTTPS petition.

- Development of the function that access the video feed web server and shows it on the control and configuration page.

6. Methodology

In this chapter we are going to show which methodology to follow and why did we choose it. We analyzed different types of software development methodologies, and we decided to choose an Agile methodology.

6.1 Agile methodology

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders [17] .

Figure 1. This picture describes the process of agile methodology sprints [18]

We choose the tutor of this FPD (Final Project Degree) as a client to show the progress every week. We could have not chosen him and just follow the methodology without any client, but it really did not feel like we were doing it right until we started showing our weekly progress to our tutor and we used the feedback to improve and upgrade the system we were working on.

Advantages of Agile methodology

- Fast development cycles
- Flexible timeline
- Version control

Disadvantages of Agile methodology

- Harder to manage than traditional methodology
- Needs cooperation from the client and whole development team

This methodology was chosen because of the fast development cycles and the flexible time to test and change features, it was essential to be able to change parts of the project on the go without having to compromise the rest of the development process.

In the methodology we also focused on version control as shown in the methodology to be able to have any change uploaded to a server so if we had any problem, we could rollback to previous versions of the code, and using version control would give us that power. We used GitHub as a

repository, which we choose for a variety of reasons, but the most important ones were that we believe in the open-source philosophy projects and we wanted to share our project to everyone following the open-source ideal. We also wanted the project to be as cheap as possible and GitHub is a free to use repository. In this repository we have uploaded all the code of the project and also this document and the videos that show the system working[19].

The other methodologies that we studied were:

6.2 Waterfall development method

Waterfall is considered to be the most traditional software development method. The waterfall method is a rigid linear model that consists of sequential phases (requirements, design, implementation, verification, maintenance) focusing on distinct goals. Each phase must be 100% complete before the next phase can start. There's usually no process for going back to modify the project or direction [20].

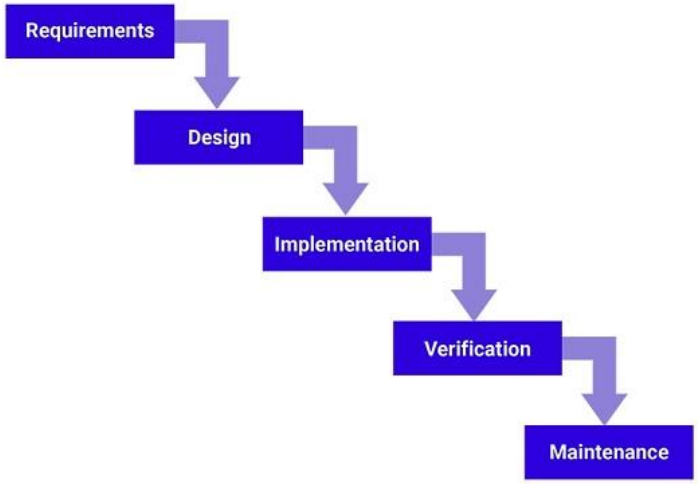

Figure 2. Design showing the waterfall process [21]

Advantages of Waterfall process

- The linear nature of the waterfall development method makes it easy to understand and manage.
- Projects with clear objectives and stable requirements can best use the waterfall method.
- Less experienced project managers and project teams, as well as teams whose composition changes frequently, may benefit the most from using the waterfall development methodology.

Disadvantages of Waterfall process

- The waterfall development method is often slow and costly due to its rigid structure and tight controls.
- These drawbacks can lead waterfall method users to explore other software development methodologies.

6.3 Prototype methodology

The Prototype Methodology is the software development process that allows developers to create only the prototype of the solution to demonstrate its functionality to the clients. Make all the necessary modifications before developing the actual application using this methodology. The best feature of this software development methodology is that it solves a plethora of issues that often occur in a traditional waterfall model [22].

Advantages of Prototype Model

- Show the prototype to the client to have a clear understanding and complete 'feel' of the functionality developed in the software. It ensures a greater level of customer satisfaction and comfort.
- Identify the scope of the refinement and accordingly accommodate new changes in the given requirements.
- Significantly reduce the risk of failure using this method and identify the potential risks at an early stage and moderation steps can be taken quickly.
- The communication between the software development team and the client makes a very good and conducive environment during a project.
- It helps in requirement gathering and requirement analysis when there is a lack of required documents.

Disadvantages of Prototype Model

- Prototyping is usually done at the cost of the developer, so it should be done using minimal resources otherwise the organization's development cost stretches too much.
- Customers sometimes demand the actual product to be delivered soon after seeing an early prototype.
- The clients have too much involvement which is not always aligned with the software developer.
- It does not appreciate too many modifications in the project as it easily disturbs the existing workflow of the entire software development process.
- Customers may not be satisfied or interested in the product after seeing the initial prototype.

7. Resources used

In this chapter we are going to explain the resources that have been used. We will show the different options that are available for the same type of development, and we will explain why we choose our options instead of the others. In this chapter, things such as libraries, packages and other software components will be also included.

7.1 Development environments

The development environments that have been used in this development are, they are going to be explain by the language used.

7.1.1 Python

First, we are going to explain which IDE we choose to use for developing in Python and the other alternatives that are in the market and why we did not choose them.

7.1.1.1 PyCharm

Figure 3. PyCharm icon[23]

PyCharm is a dedicated Python Integrated Development Environment (IDE), providing a wide range of essential tools for Pythons developers, tightly integrated to create a convenient environment for productive Python, web and data science development [24].

This IDE was chosen because it has a free-to-use version and a pay-to-use version, both having the same features except from some complex data science packages, however, in this case this did not matter. This environment has a deeply integrated control version feature, which is able to find your .git folder inside the project and makes the commits and push commands automatically without any extra configuration. Thanks to this environment we developed the computer application without any sort of environment problems.

7.1.1.2 Alternatives to PyCharm

It was not that hard to find the environment we wanted to work with, because the text editors need a lot of configurations to be able to integrate them with the control version and other libraries. With those out of the picture, we picked PyCharm because it has everything already built in, and if we wanted to use other IDEs we would have had to install all the needed plug-ins to work with Python.

7.1.1.2.1 Visual studio Code

Figure 4. Visual studio Code icon[25]

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity) [26].

7.1.1.2.2 Atom

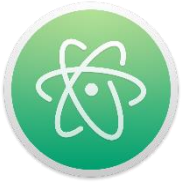

Figure 5. Atom.io icon[27]

Atom is a free and open-source text and source code editor developed by GitHub. Its developers call it a "hackable text editor for the 21st Century". Atom enables users to install third-party packages and themes to customize the features and looks of the editor, so you can set it up according to your preferences and with ease[28] .

7.1.1.2.3 Eclipse

Figure 6. Eclipse icon[29]

Eclipse is an integrated development environment. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse provides a marketplace client to find, install, and vote for new plug-ins from their vast ecosystem of providers [30].

7.1.2 Arduino

First, we are going to explain which IDE we choose to use for developing in Arduino and the other alternatives that are in the market and why we did not choose them.

7.1.2.1 Arduino IDE

Figure 7. Arduino IDE icon [31]

The Arduino Integrated Development Environment or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus, this variety of tools provides an easy and fast development environment; it also is able to identify and work with a lot of different hardware and tools. It also can monitor the serial bus to watch the communications between the IDE and the hardware, with such tool the user can identify some problems before deployment. It connects to the Arduino hardware to upload programs and communicate with them[32].

7.1.2.2 Alternatives to Arduino IDE

In this case the decision of which environment to choose was hard, in one hand, we had an easy-to-use environment, the own IDE of Arduino and in the other hand we have text-editors with more powerful tools.

In the end we decided to use the Arduino IDE, because we were going to use the control version for the full project, so we did not need the use of powerful text-editors for each part. Also, it was the easiest to set up and use out of the box.

7.1.2.2.1 Atom.io + PlatformIO

We have already explained the use of Atom.io and its characteristics, now we focus on Platform IO, to be able to work in Arduino you need to use this plug-in. PlatformIO is a cross-platform, cross-architecture, multiple frameworks, professional tool for embedded systems engineers and for software developers who write applications for embedded products.

7.1.2.2.2 Eclipse for Arduino

We have already explained the use of Eclipse and its characteristics, now we focus on the need of an external plug-in to be able to develop Arduino in this IDE, with the out of the box experience the user cannot develop Arduino in Eclipse.

7.1.2.2.3 Visual Studio Code

We have already explained the use of Visual Studio Code and its characteristics, such as being a powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. Because of it being a source code editor, the user does not need to add any plug-in to develop Arduino but there are some to make the developer experience easier.

7.1.3 Android

First, we are going to explain which IDE we choose to use for developing in Android and the other alternatives that are in the market and why we did not choose them.

7.1.3.1 Android Studio

Figure 8. Android Studio icon[33]

Android Studio is the official integrated development environment for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and GitHub integration[34].

7.1.3.2 Alternatives to Android Studio

We choose Android Studio because we already had used the environment and were used to the way Android studio works. Also, the answer to the question, what to choose for someone that has never develop for Android is usually the same, Android Studio. Because the only Windows and Linux native app for Android development is Android Studio, the other environments will need the Android plug-in to work with Android development, which make it harder to configure and to work around.

7.1.3.2.1 Intellij IDEA

Figure 9. Intellij IDEA icon[35]

IntelliJ IDEA is an Integrated Development Environment for JVM languages designed to maximize developer productivity. It does the routine and repetitive tasks for you by providing clever code

completion, static code analysis, and refactoring, and lets you focus on the bright side of software development, making it not only productive but also an enjoyable experience[36]. To be able to develop for Android in this IDE, you will need the Android plug-in.

7.1.3.2.2 Visual studio

Figure 10. Visual Studio icon[37]

Microsoft Visual Studio is an IDE made by Microsoft and used for different types of software development such as computer programs, websites, web apps, web services, and mobile apps. It contains completion tools, compilers, and other features to facilitate the software development process[38]. To be able to develop for Android in this IDE, you will need the Android plug-in.

7.1.3.2.3 Eclipse

Figure 11. Eclipse icon [29]

We have already explained the use of Eclipse and its characteristics, now we focus on the need of an external plug-in to be able to develop Android in this IDE, with the out of the box experience the user cannot develop Android in Eclipse.

7.2 Other development tools

In this part of the document, we are going to explain which other tools we have used in the development of the project and why we choose them instead of other similar tools in the market.

7.2.1 Text editor and reader

We wanted to use a text editor and reader to improve our usefulness outside the IDE, a lot of times we would find ourselves trying to know which file has each code or class and we would have to open the IDE to see what was inside the file instead of just using a text editor to read the code inside.

7.2.1 Notepad++

Figure 12. Notepad++ icon [39]

Notepad++ is a text and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window.

Based on the powerful editing component Scintilla[40], Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment[41].

7.2.2 Alternatives to Notepad++

We choose Notepad++ because it is easy to use and we already had it, it also does not change the file extension neither the application that launch each file extension which makes it easy to use without having to change or configure anything in your development environment, it is also able to identify the type of code inside and mark the special words such as class, public, private. It was also the faster to load and work out of the box between all the text editors mention in this chapter.

The decision here is difficult if the user do not have one already, the user should look deeper into it. Atom, Visual Studio code and Notepad++ are free to use, between them the difference moves between UI, how to import new packages and plug-ins and community support.

7.2.2.1 Visual studio code

We have already explained the use of Visual Studio Code and its characteristics, such as being a powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. Because of it being a source code editor, the user does not need to add any plug-in to develop software but there are some to make the developer experience easier.

7.2.2.2 Sublime text

Figure 13. Sublime text icon[42]

Sublime Text is a commercial source code editor. It natively supports many programming languages and markup languages. Users can expand its functionality with plugins, typically community-built and maintained under free-software licenses. To facilitate plugins, Sublime Text features a Python API[43].

7.2.2.3 Atom

We have already explained the use of Atom.io and its characteristics as a source code editor before.

7.2.2 API Management

7.2.2.1 Postman

Figure 14. Postman icon[44]

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs faster.

Easily store, catalog, and collaborate around all your API artifacts on one central platform. Postman can store and manage API specifications, documentation, workflow recipes, test cases and results, metrics, and everything else related to APIs. The Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle[45].

7.2.2.2 Alternatives to Postman

We choose Postman because it is the only application with an easy-to-use UI that let you use only what you want without having to spend hours learning the technology. It also gives you the opportunity to send HTTPS and HTTP messages using the application UI through a connection.

7.2.2.2.1 Amazon API gateway

Figure 15. Amazon API Gateway icon[46]

One of the most widely used applications for streamlining administrations is Amazon API Gateway. Its operation is very flexible and dependable. Amazon API provides a fantastic setup and security for hosting REST APIs that comply with HIPAA regulations. External settings, such as portable or web apps, and back-end services, such as RDS, are easily integrated with AWS API Gateway. The finest features are ease of use and dependability. The capacity to record and report information comes in second. Allows several sites to use the same API with the API Gateway[47].

7.2.2.2.2 MuleSoft Anypoint Platform

Figure 16. MuleSoft icon[48]

This is a very good technology to integrate two systems with a security layer or a validation layer in between, this tool is very versatile and allows different types of integrations, it also provides a graphical and programmatic view of the data transformation, It also provides development tools that facilitate API design, allows us to create API prototypes and get feedback immediately using the platform, it is its potential with integration connectors that can facilitate the development of each process, it is also one of the tools It almost cuts development time in half, It is very useful for rapid application development, It is very useful for the software designer, It supports many modeling languages, It is the best platform for the implementation, management and monitoring of the application[49].

7.2.2.2.3 Azure API Management

Figure 17. Azure API Management icon[50]

Azure API Management is a reliable, secure and scalable way to publish, consume and manage API's running on Microsoft Azure platform. Azure API Management provides all essential tools required for an end-to-end management of API's. It ensures optimal performance of the API's, tracks and enforces usage, authentication, and more[51].

7.3 languages

In this part we are going to explain the different languages used to develop each part of the project and why those and not others.

7.3.1 Java(Android)

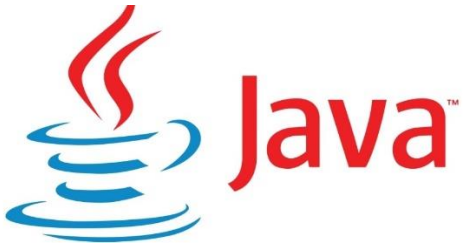

Figure 18. Java icon[52]

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc. Java is based on the virtual machine Java, which make it harder for the language to use the OS power unless you use specific packages for it [53].

7.3.1.1 Alternatives to Java

Between Java and Kotlin we decided to choose Java because it is the main language for Android and the one with the most documentation online and both work together, so if we needed to optimize the app, we could always use a bit of Kotlin inside, and there are other languages that can be ported to Android but are not native, so performance will always take a hit if using those.

7.3.1.1.1 Kotlin

Figure 19. Kotlin icon[54]

Kotlin is an open-source, statically-typed programming language that supports both object-oriented and functional programming. Kotlin provides similar syntax and concepts from other languages, including C#, Java, and Scala, among many others. Kotlin does not aim to be unique instead, it draws inspiration from decades of language development. It exists in variants that target the JVM (Kotlin/JVM), JavaScript (Kotlin/JS), and native code (Kotlin/Native)[55].

7.3.2 C and C++.

Figure 20. C icon[56]

Figure 21. C++ icon[57]

C is a high-level and general-purpose programming language that is ideal for developing firmware or portable applications. Originally intended for writing system software, C was developed at Bell Labs by Dennis Ritchie for the Unix Operating System in the early 1970s.

Ranked among the most widely used languages, C has a compiler for most computer systems and has influenced many popular languages, notably C++.

C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming. C++ is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features[58].

We used both on the Arduino development because it is the languages you can use in the Arduino IDE, both are easy to use if you know what you are doing and hard to use if you are new to the coding world.

7.3.2.1 Alternatives to C and C++

Although we finally choose C and C++ because the easy way to code an Arduino or similar microcontroller, there are other powerful options out there like avr-ada and AMforth that can push optimization to a new level.

7.3.2.1.1 AMForth

Figure 22. AMForth icon[59]

AmForth is an easily extendible command interpreter for the Atmel AVR8 Atmega micro controller family and some variants of the TI MSP430. The RISC-V CPU (32bit) is currently being worked on. It has a turnkey feature for embedded use too.

AmForth runs completely on the controller. It does not need additional hardware. It makes no restrictions for hardware extensions that can be connected to the controller. The default command prompt is in a serial terminal.

It is based on two types of data structures: dictionaries and stacks. A Forth system (that's the terminology because it is more than just an interpreter or compiler) swallows source code consisting of definitions and compiles them into words. When you call some word from the source file or from an interactive session with the system, it is executed. Executing a word means that the system mainly follows pointers to code and keeps addresses on a return stack. At the lowest level, these pointers point to assembly implementations of basic words[60].

7.3.2.1.2 AVR-Ada

The AVR-Ada project provides the gcc based Ada compiler GNAT for the AVR 8-bit microcontrollers. This includes: the Ada compiler, the Ada run time system and the AVR support library, documentation and some sample programs.

Ada is a structured, statically typed, imperative, and object-oriented high-level programming language, extended from Pascal and other languages. It has built-in language support for design by contract, extremely strong typing, explicit concurrency, tasks, synchronous message passing, protected objects, and non-determinism. Ada improves code safety and maintainability by using the compiler to find errors in favor of runtime errors[61].

7.3.3 Python

Figure 23. Python icon[62]

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python is used for server-side web development, software development, mathematics, and system scripting, and is popular for Rapid Application Development and as a scripting or glue language to tie existing components because of its high-level, built-in data structures, dynamic typing, and dynamic binding. Program maintenance costs are reduced with Python due to the easily learned syntax and emphasis on readability. Additionally, Python's support of modules and packages facilitates modular programs and reuse of code[63].

7.3.3.1 Alternatives for Python

In this case we used Python for its fast development time and easy to prototype features, if the project were to get to end users, we probably change to C and C++ for more speed and security features.

7.3.3.1.1 C and C++

C and C++ share some characteristics with Python, other than being a hard statically typed programming language. C and C++ are able to communicate this the OS easier than Python or other languages in the market.

7.3.3.1.2 Java

Java share some characteristics with Python, other than being a hard statically typed programming language. Java runs on a virtual machine by default and Python is an interpreted language, which make them work for different types of applications and solutions.

7.3.3.1.3 Golang

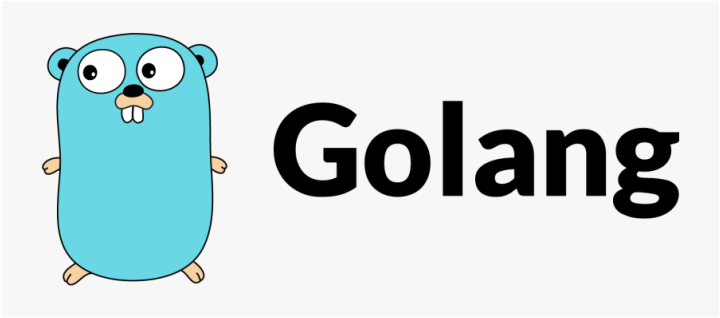

Figure 24. Golang icon[64]

Go is a statically typed, compiled programming language designed at Google. Go is syntactically similar to C, but with memory safety, garbage collection, structural typing, and CSP-style concurrency. The language is often referred to as Golang because of its former domain name, golang.org, but the proper name is Go[65].

7.4 Other resources

7.4.1 Control version

The control version that has been used for this project is git, based in GitHub. We have used it to upload the code every time we develop a new version of it, GitHub is a free web service that lets you use their servers to control version your files, such as code, documents. In this repository we have uploaded all the code of the project and also this document and the videos that show the system working[19]. We wanted to have an open-source mentality, and provide our project to everyone to study.

7.4.1.1 Git

Figure 25. Git icon[66]

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

Git was originally for development of the Linux kernel, with other kernel developers contributing to its initial development. As with most other distributed version control systems, and unlike most client–server systems, every Git directory on every computer is a full-fledged repository with complete history and full version-tracking abilities, independent of network access or a

central server. Git is free and open-source software distributed under the GPL-2.0-only license[67].

7.4.1.2 Alternatives to git

7.4.1.2.1 Subversion

Figure 26. Subversion icon[68]

Apache Subversion (often abbreviated SVN, after its command name svn) is a software versioning and revision control system distributed as open source under the Apache License. Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly compatible successor to the widely used Concurrent Versions System (CVS)[69].

7.5 Libraries

In this chapter we are going to see which libraries we have used to develop the system.

7.5.1 Android

We are going to explain which libraries we have used to develop some functionalities inside the Android application.

7.5.1.1 Volley

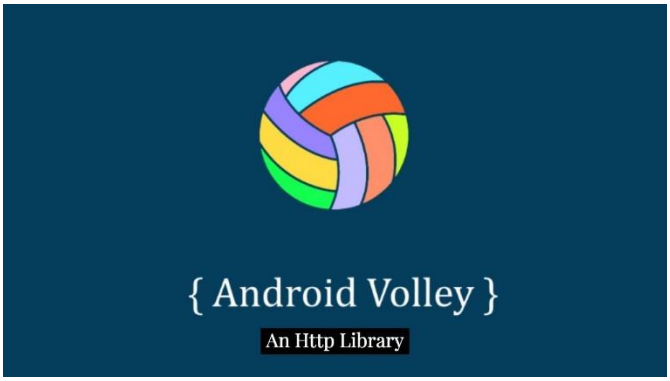

Figure 27. Android Volley icon[70]

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley achieves faster speed with the following benefits: automatic scheduling of network requests, multiple concurrent network connections, transparent disk and memory response caching with standard HTTP cache coherence. It also has support for request prioritization and cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel. Ease of customization, for example, for retry and backoff, strong ordering

that makes it easy to correctly populate your UI with data fetched asynchronously from the network and debugging and tracing tools[71].

This library has been used to make HTTP to connect to the camera web server and for sending the joystick data to the ESP8266 to be able to move the motor and other hardware.

7.5.1.2 Jackandphantom joystick

This library lets you create a joystick with some customization for Android. There are two views in the joystick, the inner circle view,this is small a circle in the joystick so attributes related to this view will using innerCircle in prefix in XML; and the outer circle view, this is the big circle in the joystick so all attributes related to this will using outerCircle in prefix in XML[72].

This library has been used to create an UI joystick and to be able to get the data of the position of the joystick by using the two attributes before mention.

7.5.1.3 Net

Classes that help with network access, beyond the normal java.net.* APIs, it has a lot of functionalities, but we are going to explain a few that we have used inside the project. The Network Suggestion is used to provide a WiFi network for consideration when auto-connecting to networks. Apps can provide a list of such networks to the platform using WifiManager. Which is the main API for managing all aspects of WiFi connectivity.

This library deals with several categories of items such as: the list of configured networks, this list can be viewed and updated, and attributes of individual entries can be modified. The currently active Wi-Fi network, if any. Connectivity can be established or torn down, and dynamic information about the state of the network can be queried. The results of access point scans, containing enough information to make decisions about what access point to connect to and it also defines the names of various intent actions that are broadcast upon any sort of change in Wi-Fi state[73] .

This library has been used to connect the app to the WiFi and to get the connection info, such as the Mac address, of the strength of the connection.

7.5.1.4 Content

Contains classes for accessing and publishing data on a device. It includes three main categories of APIs: content sharing, for sharing content between application components. Package management, for accessing information about an Android package, including information about its activities, permissions, services, signatures, and providers and resource management for retrieving resource data associated with an application, such as strings, drawable, media, and device configuration details[74].

This library has been used to set the view of the UI and other UI elements and UI functionalities.

7.5.1.5 Os

Provides basic operating system services, message passing, and inter-process communication on the device. Information about the current build, extracted from system and other types of system information[75]. This library has been used to be able to compile and execute the code inside Android, also it is used to bundle all the code.

7.5.1.6 Widget

The widget package contains (mostly visual) UI elements to use on your application screen. Library with UI components for Android is mostly used in apps to speed up the design and development of user interface elements and widgets in Android apps. These libraries offer all kinds of functionalities like image zooming, circular images, material design, progress bars and other quick improvements to the user interface[76]. This library has been used to generate and make the UI elements of the application, such as text, images, drawable, media.

7.5.1.7 Webkit

If a developer wants to add browser functionality to an application, the developer can include the Webkit library and essentially embed a browser within the app to do things like render web pages and execute JavaScript[77]. This library has been used to generate a web view that shows the camera feed of the hardware.

7.5.2 PC

7.5.2.1 PyQt5

Figure 28. PyQt5 icon[78]

Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development.

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android. PyQt5 may also be embedded in C++ based applications to allow users of those applications to configure or enhance the functionality of those applications[79].

This library has been used to generate all the UI elements such as windows, text. It also takes care of the keyboard; with it we can take the input to control the hardware.

7.5.2.2 Requests

Figure 29. Request icon[80]

Requests is an elegant and simple HTTP library for Python. Requests allows you to send HTTP/1.1 requests extremely easily. There is no need to manually add query strings to your URLs, or to form-encode your POST data. Keep-alive and HTTP connection pooling are 100% automatic, thanks to urllib3.

Request has a lot of features, and we are going to mention a few that we have used or find useful. It has keep-alive and connection pooling, international domains and URLs, sessions with cookie persistence. Browser-style SSL verification, automatic content decoding, basic authentication, elegant key value cookies. Automatic decompression, Unicode response bodies, HTTPS proxy support, multipart file uploads, streaming downloads, connection timeouts and chunked requests. It also supports .netrc [81]. This library has been used to send HTTP post requests; in them we send the data that is needed for the hardware.

7.5.2.3 Urllib3

Figure 30. Urllib3 icon[82]

Urllib3 is a powerful, user-friendly HTTP client for Python. Urllib3 brings many critical features that are missing from the Python standard libraries such as: thread safety, connection pooling, client-side SSL/TLS verification. File uploads with multipart encoding, helpers for retrying requests and dealing with HTTP redirects. Support for gzip, deflate, and brotli encoding. Proxy support for HTTP and SOCKS and 100% test coverage[83]. This library is used to get the status of the different web services.

7.5.3 Arduino

7.5.3.1 WiFi.h

The Wi-Fi API provides support for the 802.11b/g/n protocol driver. This API includes: station mode (STA mode or Wi-Fi client mode), ESP32 connects to an access point. AP mode (aka Soft-AP mode or Access Point mode), devices connect to the ESP32. Security modes (WPA, WPA2,

WEP, etc.) and scanning for access points[84]. This library is used for generating the station mode, and for the web server.

7.5.3.2 esp_camera.h

This library is used for the camera web server, we are using an ESP32-CAM, and this library is the one for it[85].

7.5.3.3 ESP8266WiFi.h

The WiFi library for ESP8266 has been developed based on ESP8266 SDK, using the naming conventions and overall functionality philosophy of the Arduino WiFi library. Over time, the wealth of Wi-Fi features ported from ESP8266 SDK to ESP8266/Arduino outgrew Arduino WiFi library and it became apparent that we would need to provide separate documentation on what is new and extra[86]. This library is used to connect to the access point and to be able to communicate through WiFi.

7.5.3.4 ESP8266WebServer.h

The WebServer is a simple web server that knows how to handle HTTP requests such as GET and POST and can only support one simultaneous client[87]. This library is used for receiving the HTTP petitions of the PC or phone app.

8. Components

In this chapter we are going to show which component are in our system.

In the next diagram (Figure 31) we can see the hardware used in the system.

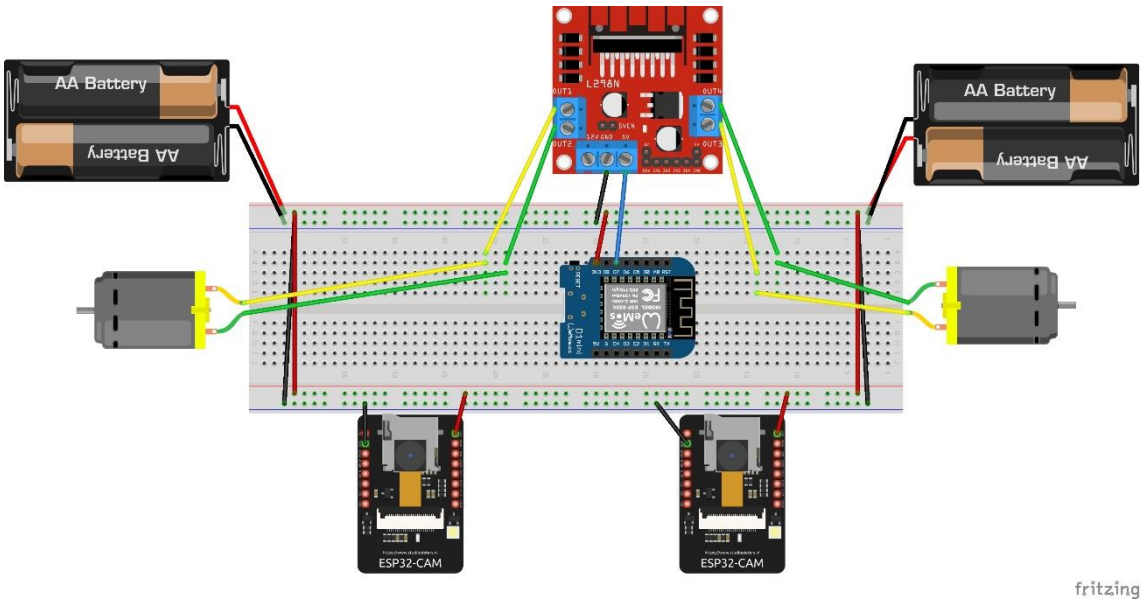

Figure 31. Simple hardware design

8.1 ESP32-CAM

Figure 32. ESP32-CAM photo

Esp32-CAM is a low-cost ESP32-based development board with onboard camera, small in size[88]. It is an ideal solution for IoT application, prototypes constructions and DIY projects.

It is used in the project to get video feed and provide it to a webpage to be taken by our application, inside this webpage we can configure all the details and options of the video feed, such as: resolution, detail, contrast and other types of quality-of-life options.

The board integrates WiFi, traditional Bluetooth and low power BLE, with 2 high-performance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80Mhz to 240Mhz.

Fully compliant with WiFi 802.11b/g/n/e/I and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices.

In the next pictures (Figure 33 and Figure 34) we can see how the system works with 3.3V and 5V.

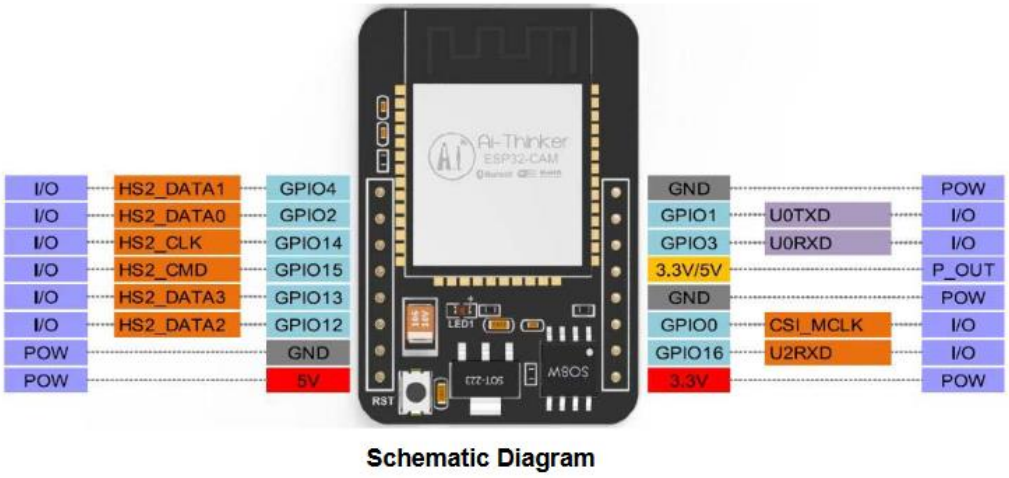

Figure 33. ESP32-CAM GPIO schematics

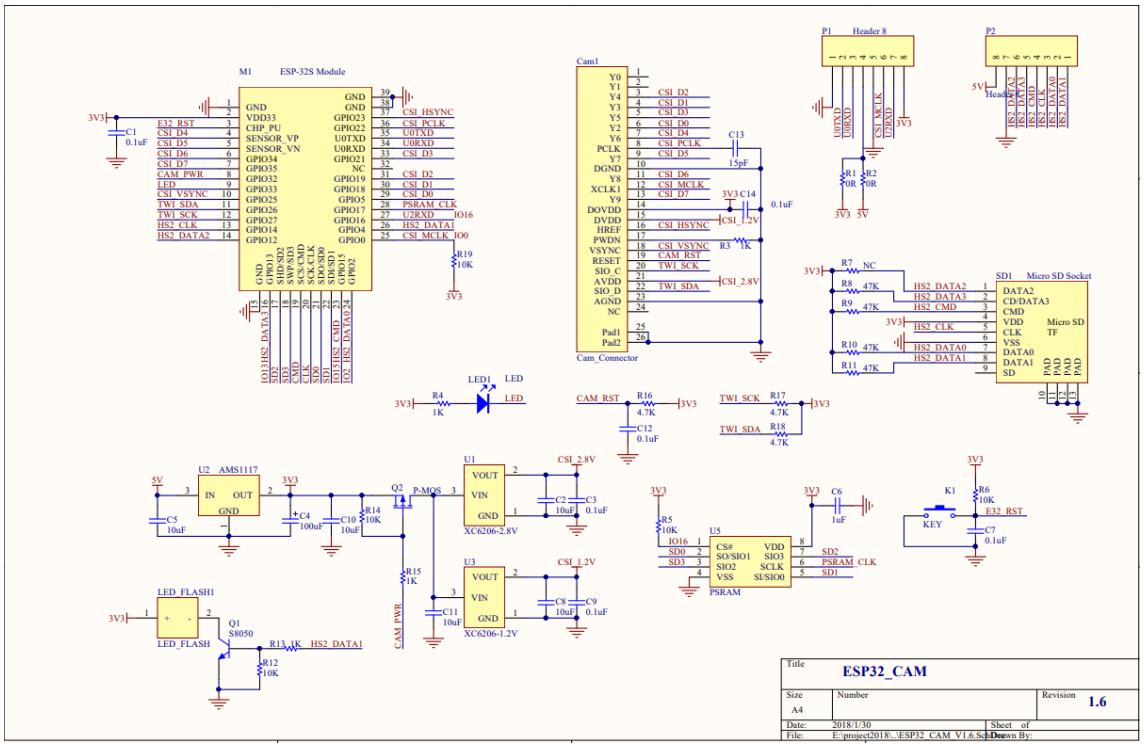

Figure 34. ESP32-CAM electronic schematics

8.2 L298N

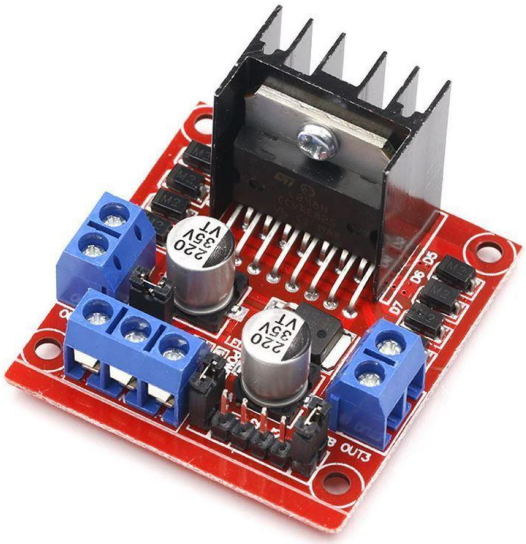

Figure 35. L298N photograph

The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages[89]. It is a high voltage, high current dual full-bridge driver de-signed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional Supply input is provided so that the logic work at a lower voltage. We are going to use the L298N to control two 3-6V DC motors and to receive the signal needed to know the direction of the motors sent by the ESP8266.

Its schematic diagram is as shown in Figure 36, it shows how the system process the voltage talked before.

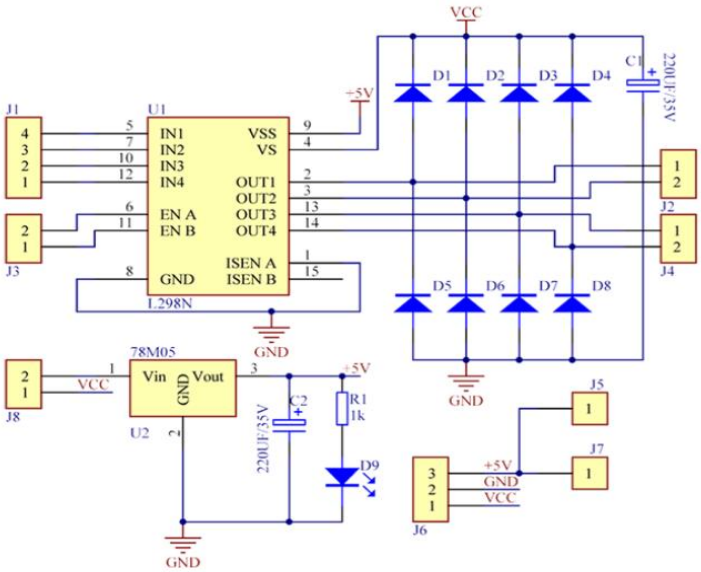

Figure 36. L298N schematics

8.3 Wemos D1 Mini

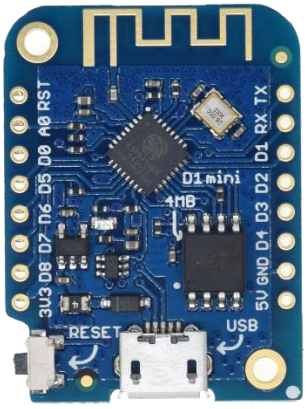

Figure 37. Wemos D1 Mini photograph

Wemos D1 Mini is a development platform is similar to Arduino, specially oriented to IoT[90]. The Wemos D1 Mini board has an ESP8266 as core, it also has integrated a USB-serial converter TTL CH340G and a micro-USB connector needed to program and communicate with. Wemos D1 Mini is design to work on top of a protoboard or to be welded in a board. It has a 3,3-voltage regulator, this let the board to work directly from the micro-USB port or by the 5V and GND pins. The GPIO work as 3,3 V so if you need to connect to 5V systems you will need to use 3,3 to 5 V conversors. The SoC ESP8266 of Espressif Systems is a chip developed for the new needs of the interconnected world, it has a powerful microcontroller with a 32bits architecture and Wi-Fi connectivity. The ESP platform let you code with different languages such as Arduino, Lua, MicroPython , C/C++, Scratch. We are going the Wemos D1 Mini to receive the signal from the PC or the Android application and send it to the L298N to control the DC motors to move the drone. In the Figure 38 and Figure 39 we can see the GPIO schematic and the electronic schematic, they show how the system works with 3.3V and 5V and how it can be connected through micro-USB.

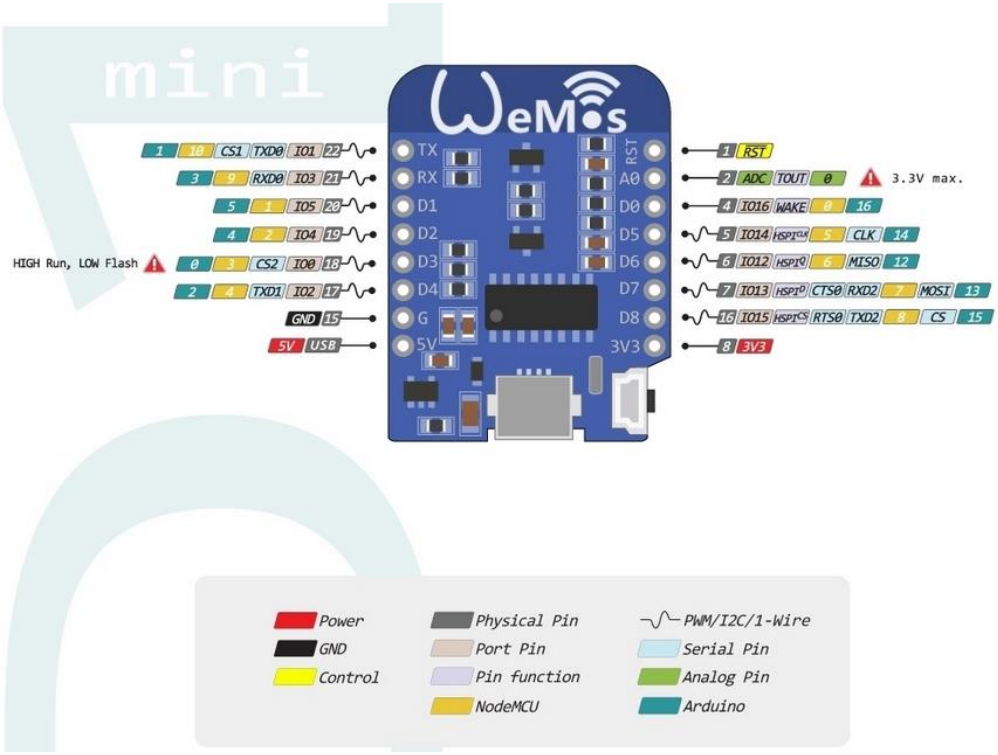

Figure 38. Wemos D1 Mini GPIO schematic

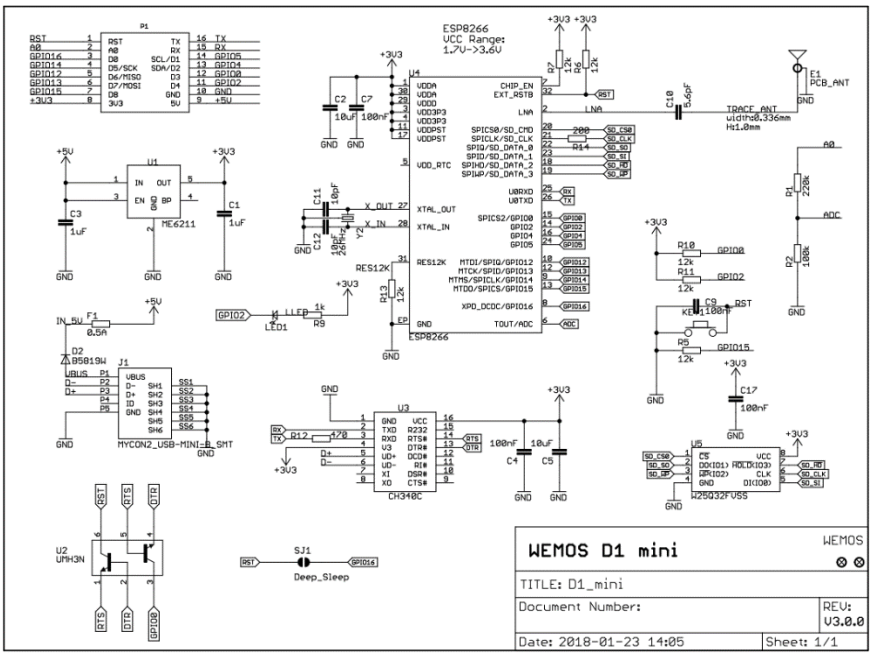

Figure 39. Electronic schematic of the Wemos D1 Mini [91]

8.4 3-6V DC motor

3-6V DC motor is a rotary motor which can convert the direct current into mechanical energy or convert mechanical energy into DC power. We use it to move the whole system, it is connected with the L298N and the esp8266 to get the signal and move the system.

Figure 40. Photograph of the 3-6V DC motor

8.5 Connectivity

Figure 41. Simple diagram of the connectivity

All the systems are connected through WiFi, the ESP32-CAM will raise a webserver and the PC application, and the Android application will access that webserver to get the video feed and the configuration of the camera. The control of the drone is made by sending HTTP petitions to the ESP8266, both the applications and the ESP8266 have to be connected to the same WiFi connection. We use this type of connection because we wanted to have a wireless system that was able to send video feedback and other types of data, and WiFi was the cheapest of all. The other main options available for this type of system are Bluetooth[92] or radio-controlled. Bluetooth has a sorter range than WiFi, and we wanted our system to be able to travel some distance without losing connection, this made us not choose Bluetooth. Radio-controlled systems are mid-long-range systems, but they are more expensive than WiFi if you want to have a video feed or resource-intensive communications.

9. Milestones

We have followed a work plan based on milestones, which must be completed to be considered fulfilled. These milestones have been divided in smaller tasks to make tracking the development of the project easier.

9.1 Milestone 1: Previous study of the system

- Task 1: System requirements.
- Task 2: Planification of tasks and times.
- Task 3: Study of tools for Android development.
- Task 4: Study of tools for PC development.
- Task 5: Study and research of the system hardware.
- Task 6: Other studies (state of the art, tools used for writing and making diagrams).

9.2 Milestone 2: Design of the system

- Task 7: Design of the system architecture.
- Task 8: Design of the Android application interface.
- Task 9: Design of the PC application interface.
- Task 10: Design of the Android application functionality.
- Task 11: Design of the PC application functionality.
- Task 12: Design of the hardware functionality.
- Task 13: Design of the live camera feed functionality.
- Task 14: Design of the connection between applications.
- Task 15: Other designs.

9.3 Milestone 3: Development of the system

- Task 16: Development of the Android application interface.
- Task 17: Development of the PC application interface.
- Task 18: Development of the Android application functionality.
- Task 19: Development of the PC application functionality.
- Task 20: Development of the hardware application functionality.
- Task 21: Quality assurance.
- Task 22: Testing the hardware.

9.4 Milestone 4: Meetings and other tasks

- Task 23: Meetings with the tutor.
- Task 24: Writing of this document.

Thanks to this list the tracking of the task to accomplish is easier, making it simple to choose between them. The client was presented with the milestones and task, and he gave the confirmation that the task was correct before going with the next, following the agile methodology. We had weekly meetings with the client to communicate the status of the tasks and we used tools such as Trello and Teams to be able to have communication with the client; and have the appointments with the client. We wanted to follow an open-source mentality so everything about this project is in this repository [19] it also helps following the agile methodology.

We are going to explain a bit more of what do we mean with each milestone and the work that needs to be put in these:

- Milestone 1 - Previous study of the system: Trying to reduce the time spent in development, we first defined the requisites that we had, we planned the time we had for each task and analyzed different tools (IDE, languages,...). The most part of the study is inside the 'state of the art 'and 'resources used' chapters.
- Milestone 2 - Design of the system: To be able to have an easier development process we decided to make designs for each part of the system, this allowed us to see clearly the objectives of the system and the way to achieve them. We made a general design of the system, and after we designed each application user interface, then we finished by designing every functionality that was going to be in the system.
- Milestone 3 - Development and quality assurance: In this milestone we had to follow the design previously made. We first created the user interface for each platform, and after that we added the different functions, everything was uploaded to the GitHub repository [19]. We ended by programming the hardware. After all this process, we decided that a lot of testing was needed so we decided to follow a Quality & Assurance protocol, trying to test everything and achieve a 100% coverage of the tests. We also tested the hardware integration with the prototype.

10. Design of the system

In this chapter we are going to talk about the design of the system, which requisites had to be essential and how the components of the system are connected between them.

10.1 First designs

We made some design at the beginning of the development to have some idea of what UI elements we wanted to have in the software and the organization of the project. These two diagrams that are included after this paragraph (Figure 42 and 43) show the first idea we had about the Android application UI and the PC application UI. The only difference with the end design and UI is that we changed the video stream button that only shows the video feed because we already get the video feed in the other two windows.

The main idea of the design is to be easy to follow and to access every part of the application, this means that we wanted to have as less clicks by the user as possible. To get to the control of the drone, the user only needs one click in the control button and to get to the configuration of the camera feed, the user only needs to click the configuration button.

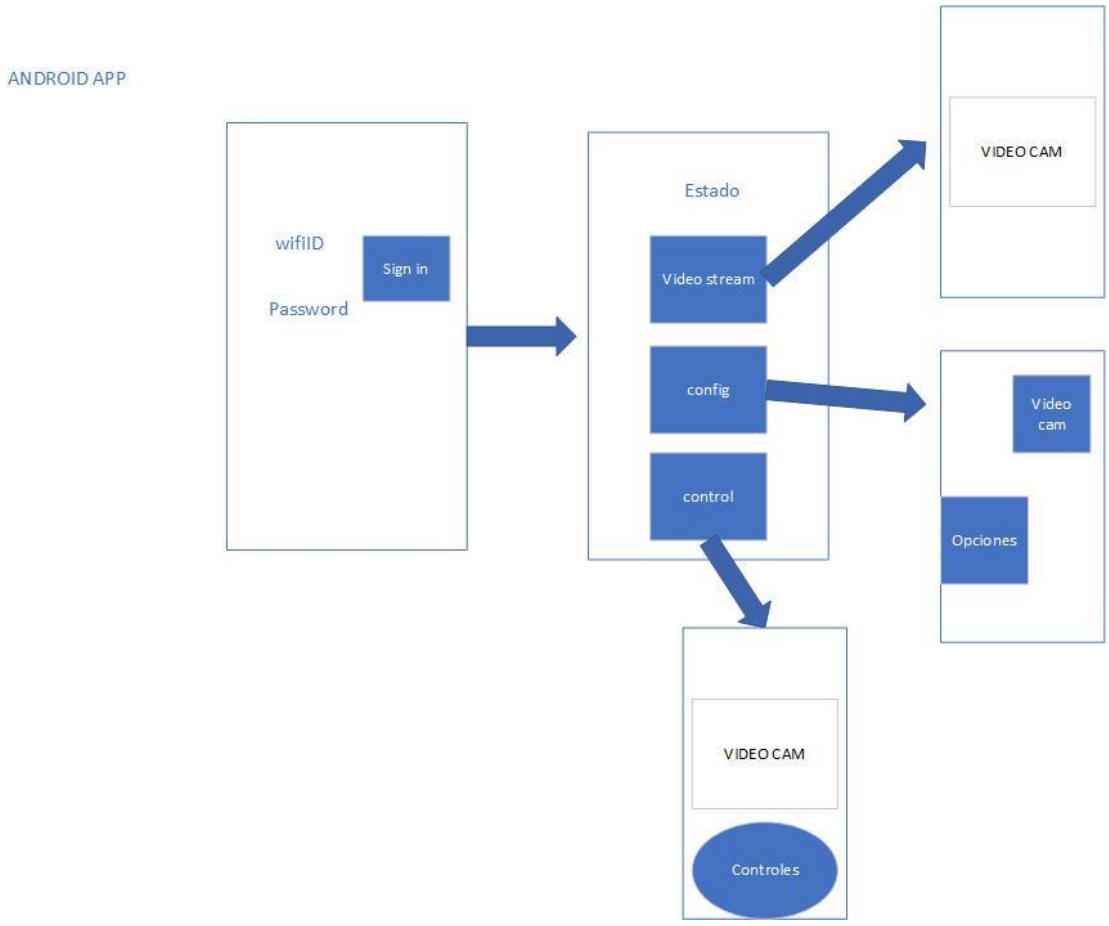

Figure 42. Simple diagram of the Android application UI

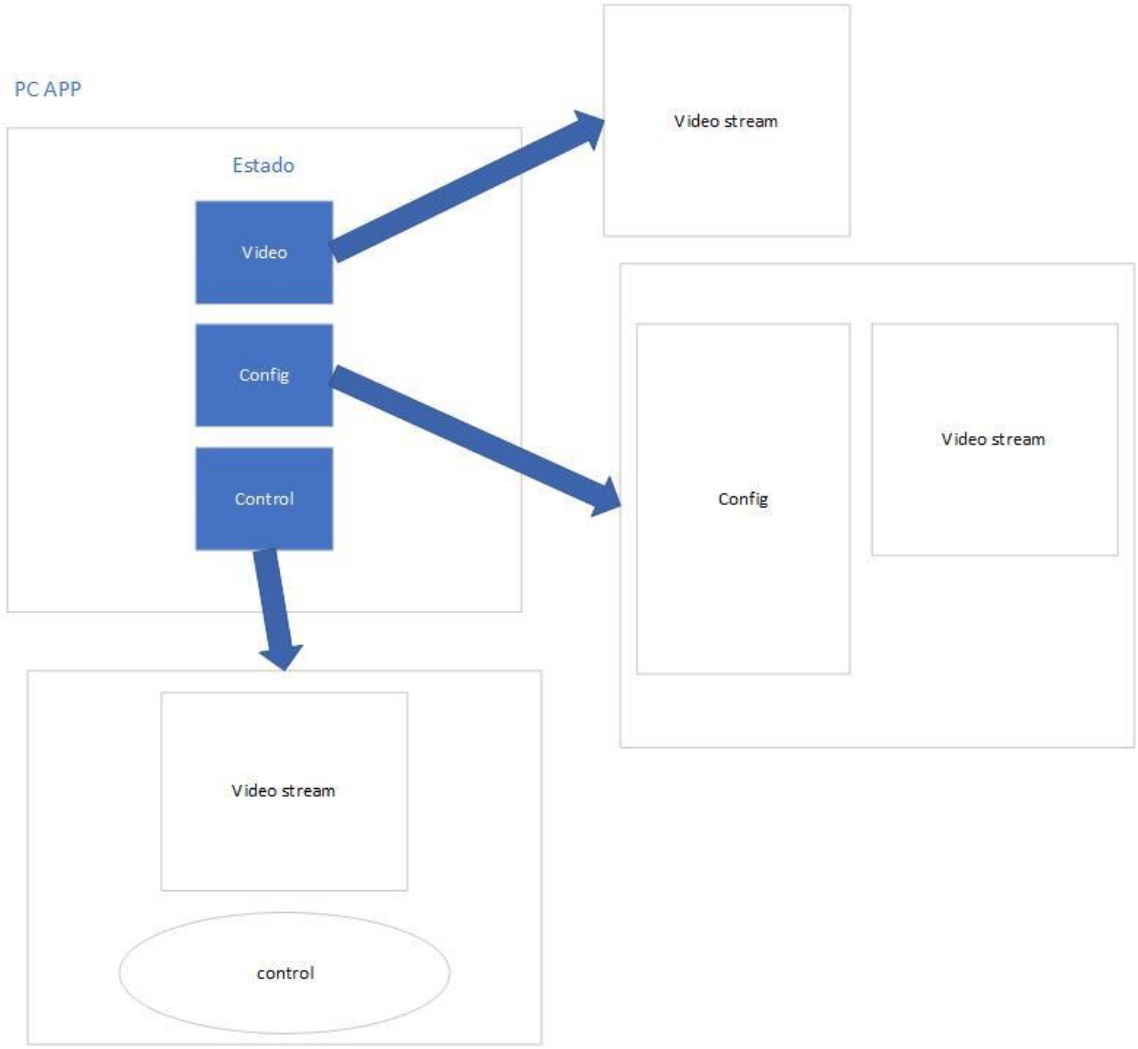

Figure 43. Simple diagram of the PC application UI

10.2 Use case diagram

A use case diagram was drawn to show the main function of the system that the user has access to and is able to choose between, this use case diagram can be found in the Figure 44.

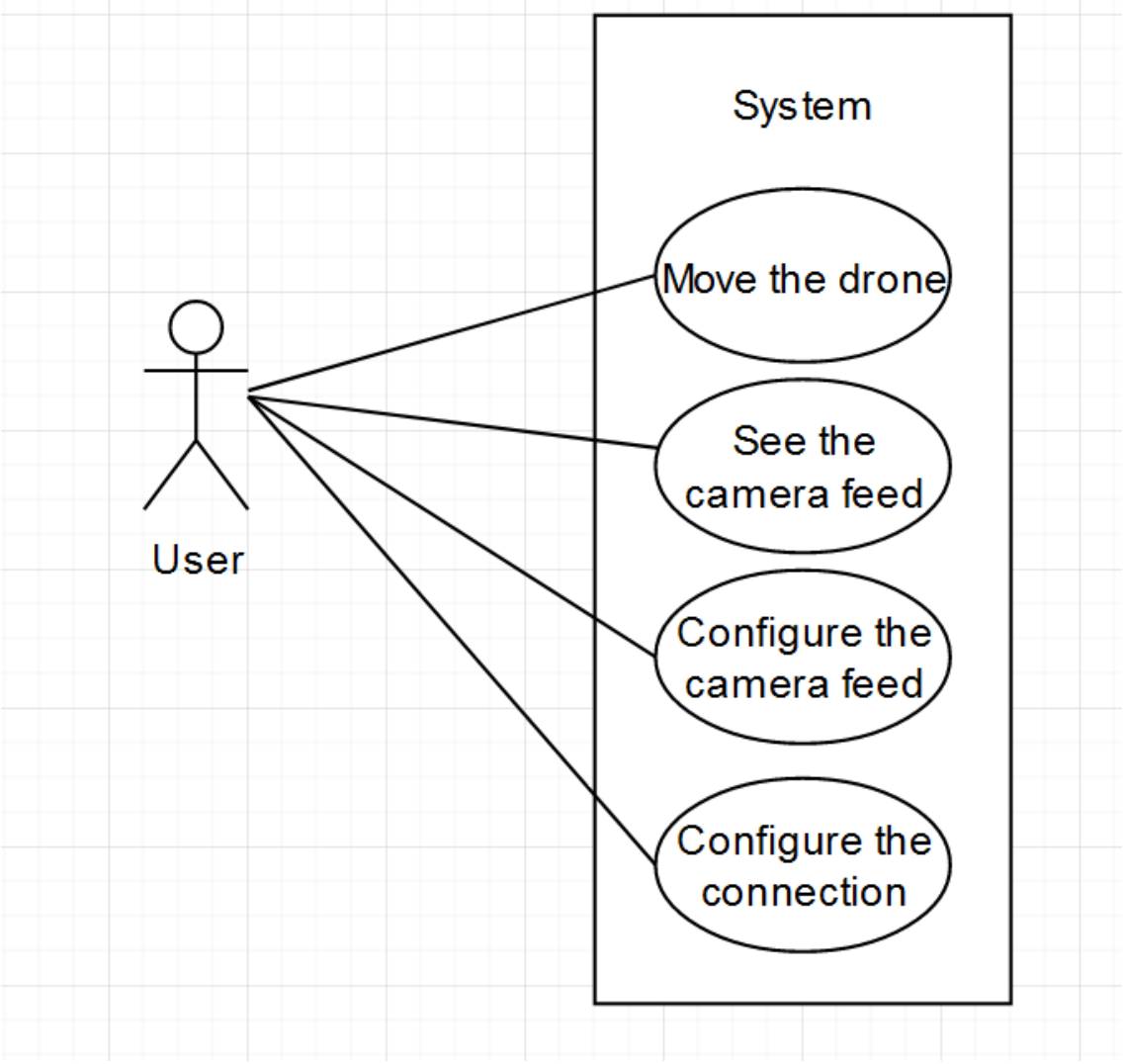

Figure 44. Use case diagram

This system gives the user the ability to be able to move the drone, watch the video feed and configurate the system and video feed. The user will be able to change the WiFi connection (you can change it to whatever you want if the ESP32 is able to detect that network only works on 2,4Ghz range) and the video feed configuration that will be explained on the 'user manual' chapter.

10.3 Class diagrams

Before starting to code, this project needed some organization and UML usage. We wanted to make a class diagram for each part of the project, either hardware or software related. We also added which packages and libraries the software uses; both are represented as packages in the diagrams to be easier to understand without having UML knowledge. We also wanted to have class diagrams to be able to follow the development process and upload all the needed code to the repository following the open-source mentality [19].

10.3.1 PC class diagram

We have used PyQt5 to make the UI as told before in the 'resources used', and urllib3 and request for sending the data to the hardware. We have a main class that invokes the three different windows, the RealMainWindow will be the first window the user sees, and it will work towards the other two windows. The config window will show the configuration options and the control window will send the input to the hardware. The Utility class has been created to be able to take apart the UI elements and the code only elements such as calculations, status checks, and other types of software elements.

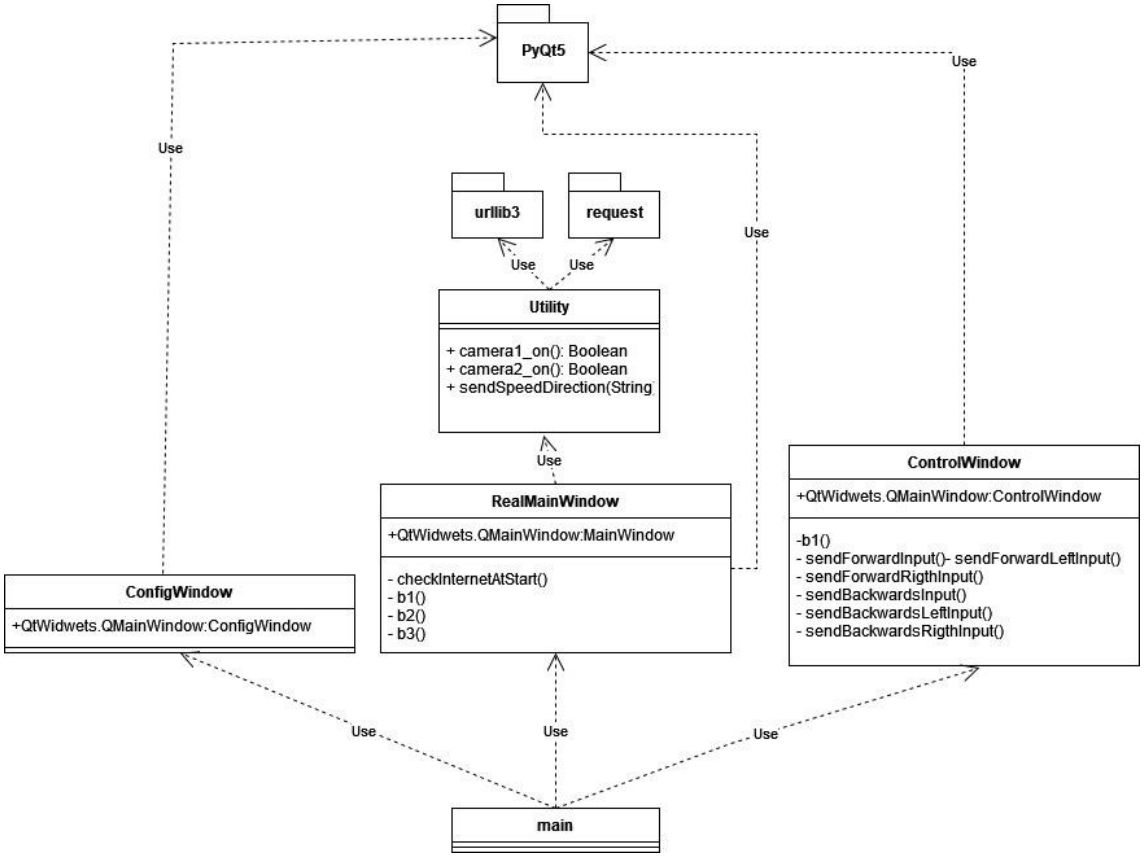

Figure 45. PC class diagram

10.3.2 Hardware class diagrams

10.3.2.1 Camera class diagram

We have used a specific library to use the ESP32-CAM camera, and the usual ESP32 WiFi library.

The cameraWebServer class has a few methods that will start the camera and the web server, and you will have the ssid and the password.

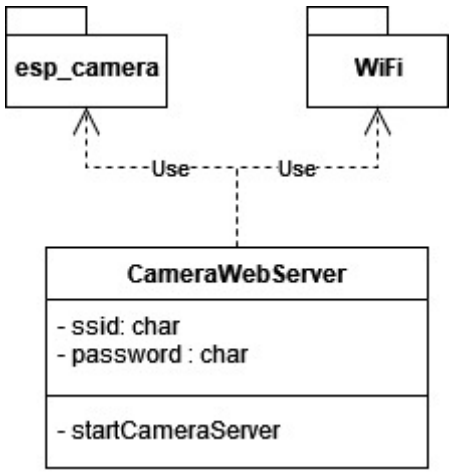

Figure 46. Camera class diagram

10.3.2.2 Control class diagram:

We have used the usual ES8266 WiFi and web server libraries, the ESP32 control motor class has a few more methods that will work towards getting the input from the other applications, change it from raw data to process data and move the DC motors.

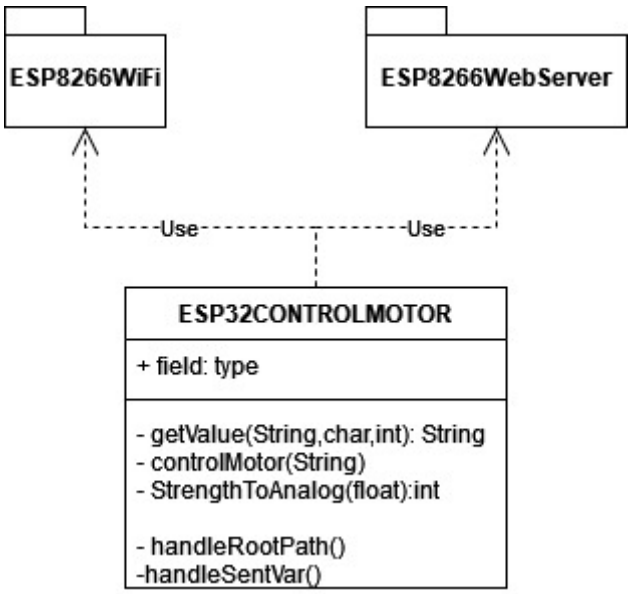

Figure 47. Control class diagram

10.3.3 Android application class diagram

The Android application diagram is the more complex out of all the diagrams in this chapter, we have used a few classes to have different UI elements all separated from each other, we have used the common Android, Androidx and java.util libraries and the only exceptional package jackandphantom.joystick.

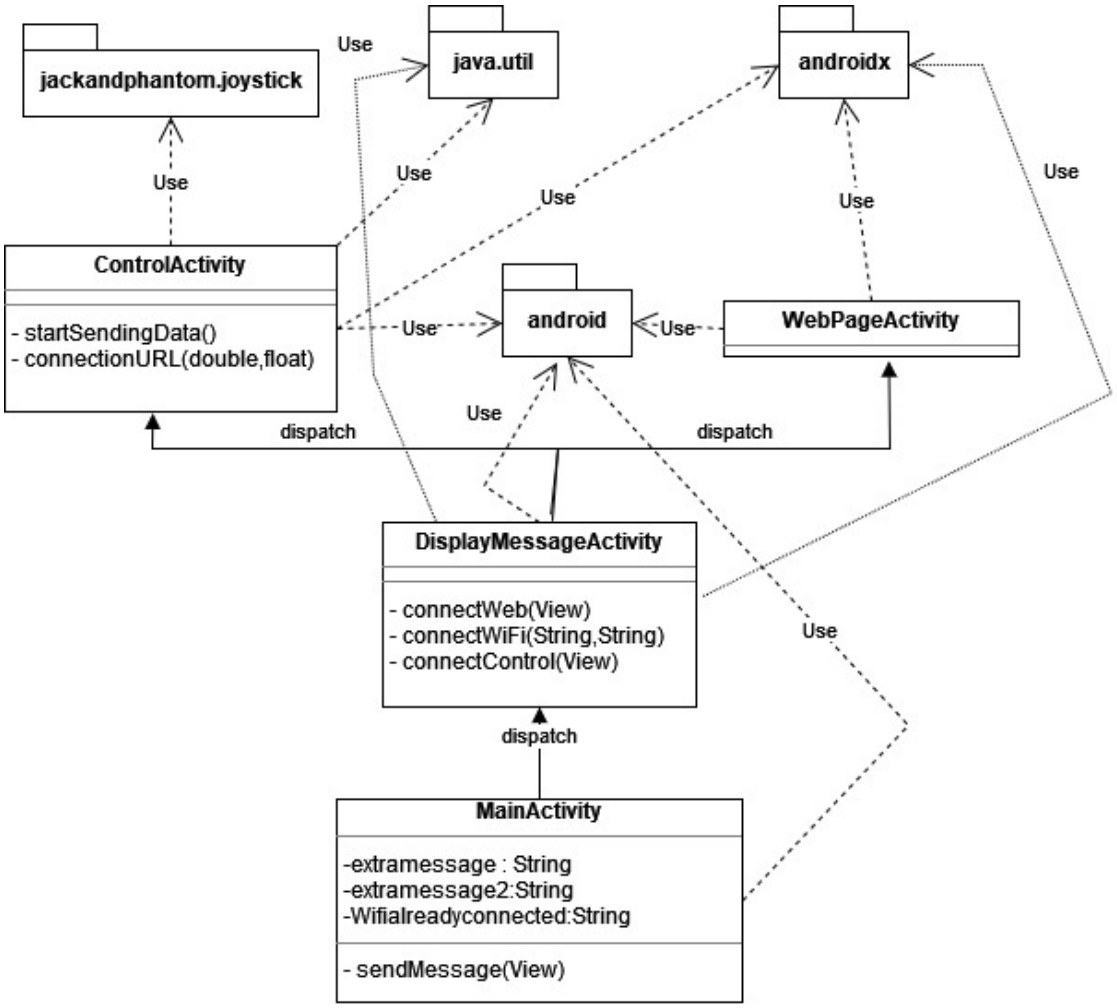

Figure 48. Android application class diagram

10.4 Hardware assembly

In the next pictures, we show the initial assembly to test all the hardware we have introduce in the earlier part of this document. In this first three pictures we can see the general assembly of the control of the DC motors, the ESP8266 is powered by USB, the ESP8266 send the signals to the L298N and receives the HTTP from the PC and Android application. Then the DC motors are connected to the L298N. In the next picture the ESP32-CAM is powered by USB and is already working and showing the video feed in the application. The last pictures show the connections in more detail, if needed for a new assembly.

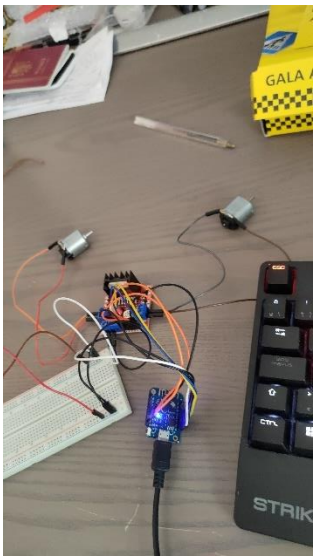

Figure 49. Global test assembly without camera

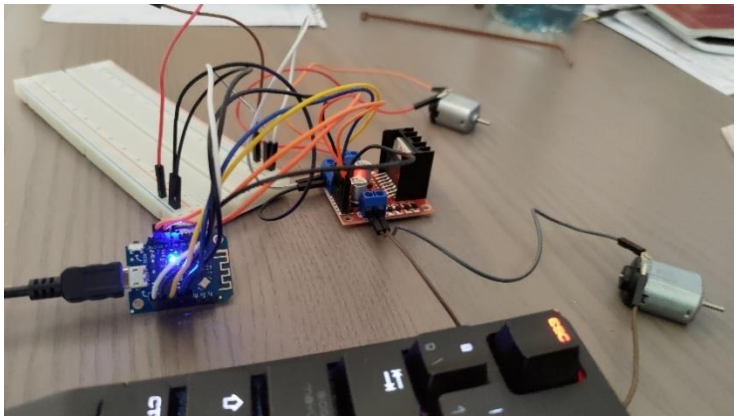

Figure 50. Global assembly without camera

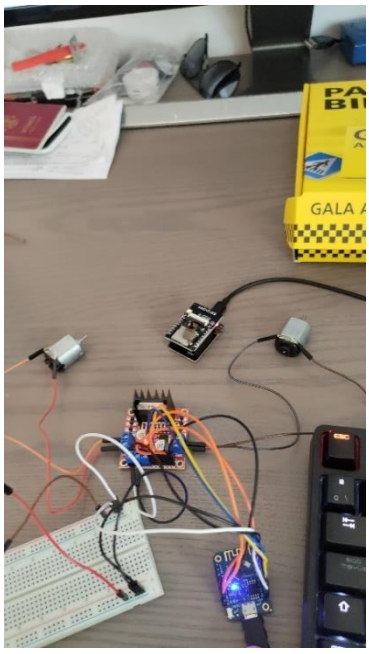

Figure 51. Global assembly with camera

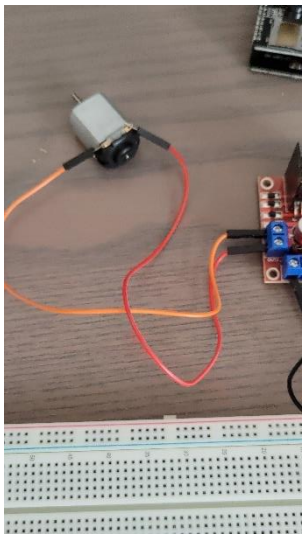

Figure 52. DC motor assembly

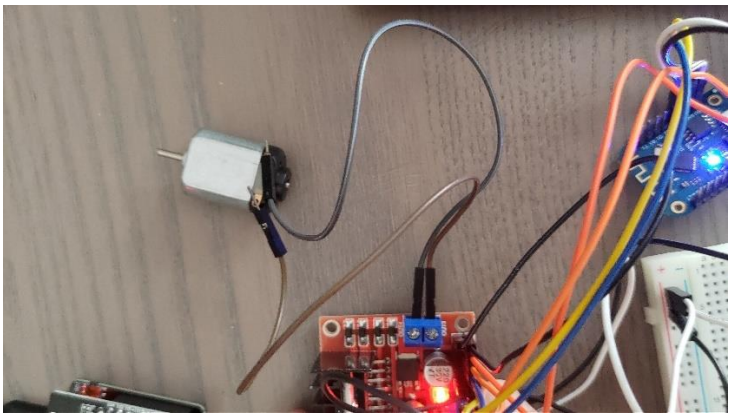

Figure 53. DC motor assembly 2

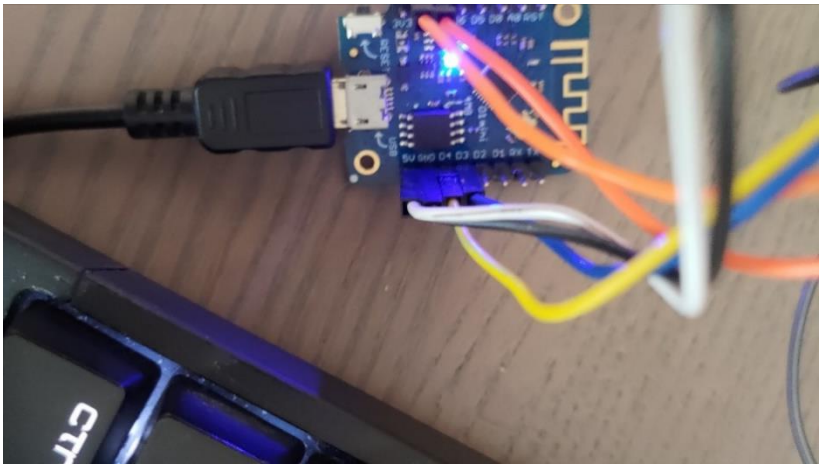

Figure 54. ESP8266 close-up connections

Figure 55. ESP8266 close-up connections 2

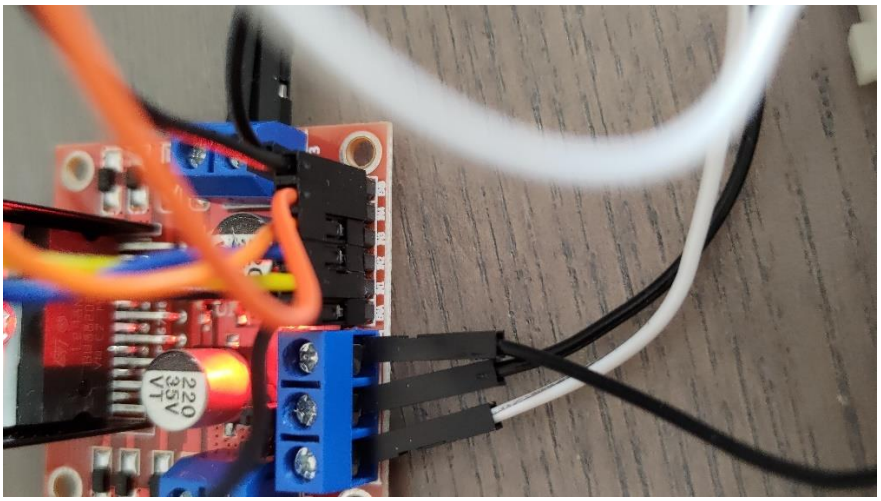

Figure 56. L298N close-up connections

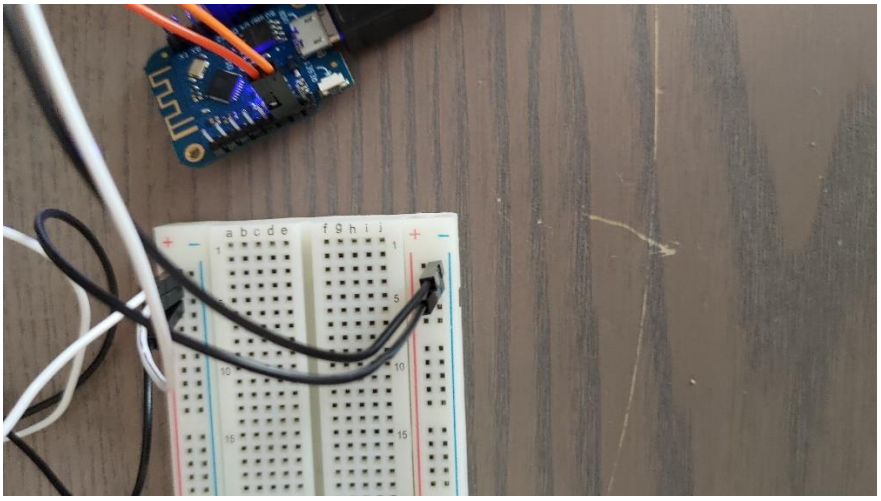

Figure 57. Power connections close-up

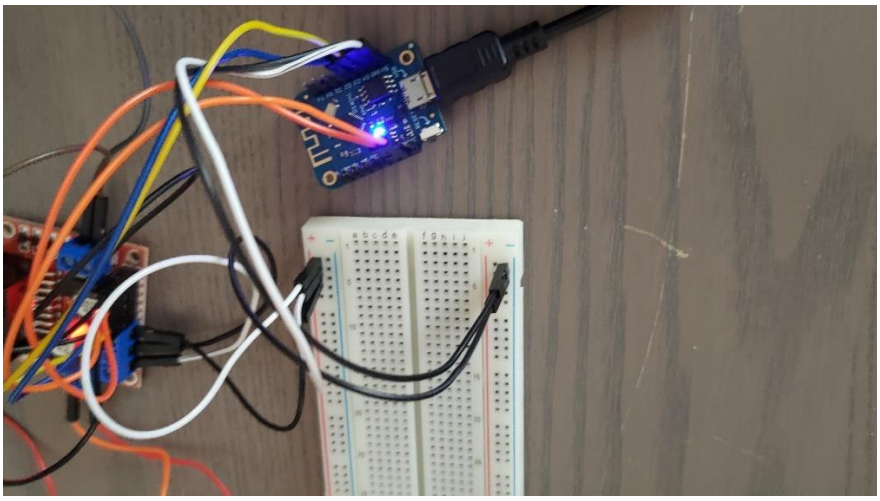

Figure 58. Power connections close-up 2

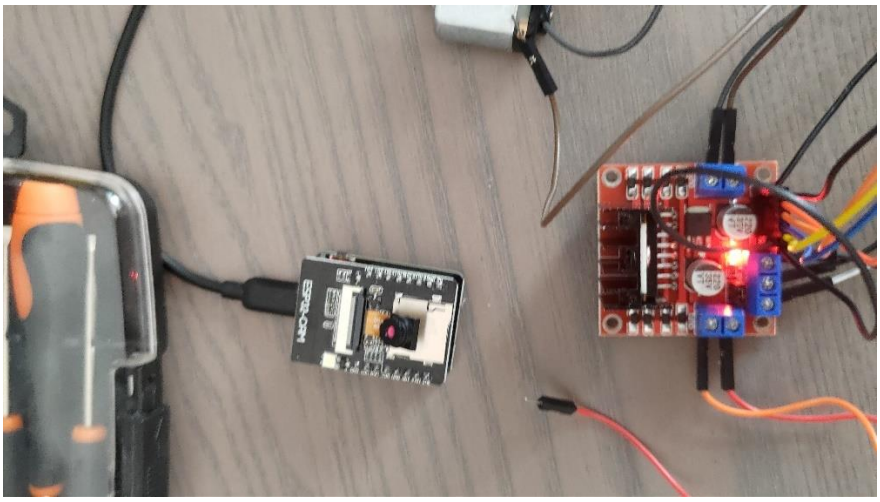

Figure 59. Camera connection close-up

11. User Manual

11.1 PC Application

First, we are going to make a checklist of everything needed before the start of the application and point some important information needed.

- You have to make sure that the PC is connected to the same WiFi connection because the application will try to connect to the cameras directly without asking the user.
- To get the video feed working either on the PC application or the Android application the user first needs to go to the configuration window and press the start stream button. Or access the IP address set by the developer if they have access to it and press the button through a website.
- The main window is going to still be open when the other windows are working, the window you selected will be focused and the main window will hide behind.

In the main window you will be able to find the connection status, which is marked by the red arrow. To refresh the connection and see if there has been changes the user needs to press the Restart connection button. There 4 status:

- Connected, when both cameras are working.
- Camera 2 offline, check camera connection. It appears when the camera 2 is offline and camera 1 is online.
- Camera 1 offline, check camera connection. It appears when camera 1 is offline and camera 2 is online.
- Both cameras offline, check WiFi connection, when both cameras are offline.

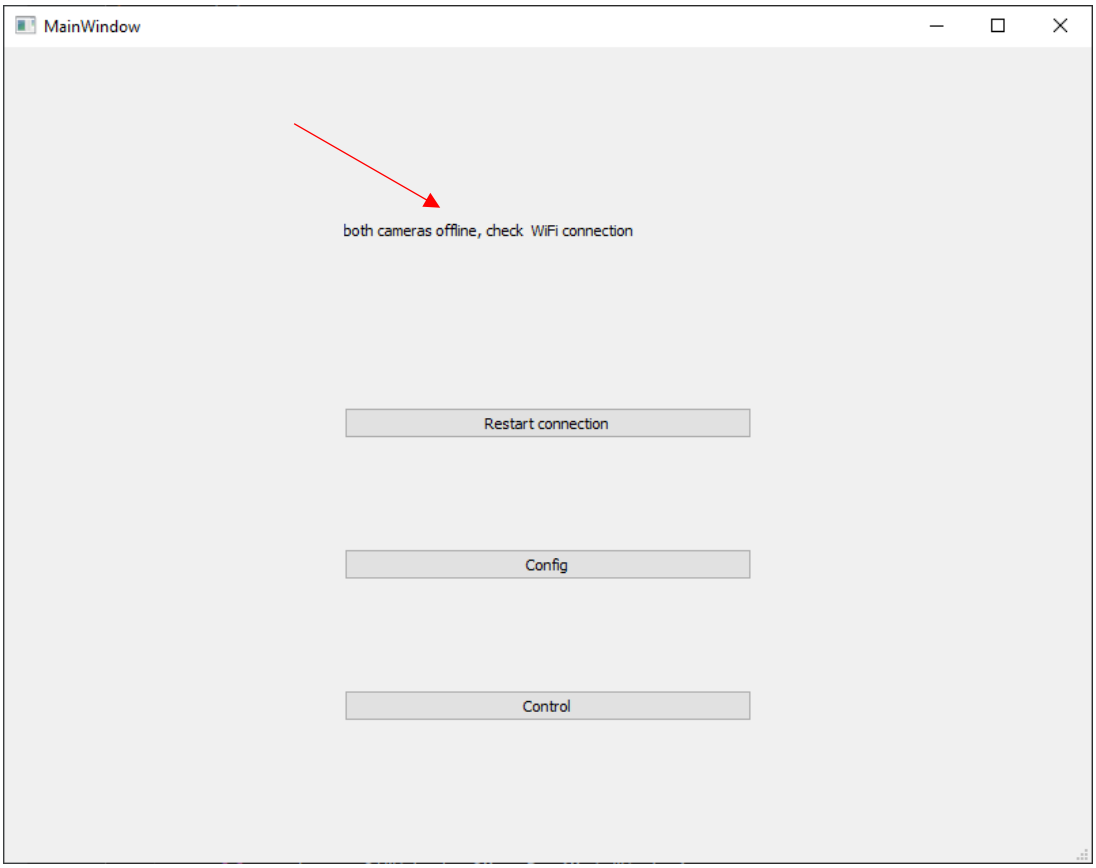

Figure 60. PC Main Window

11.1.1 Configuration

To configurate the video feed, you will press the config button (Figure 61), this will open the configuration window (Figure 63).

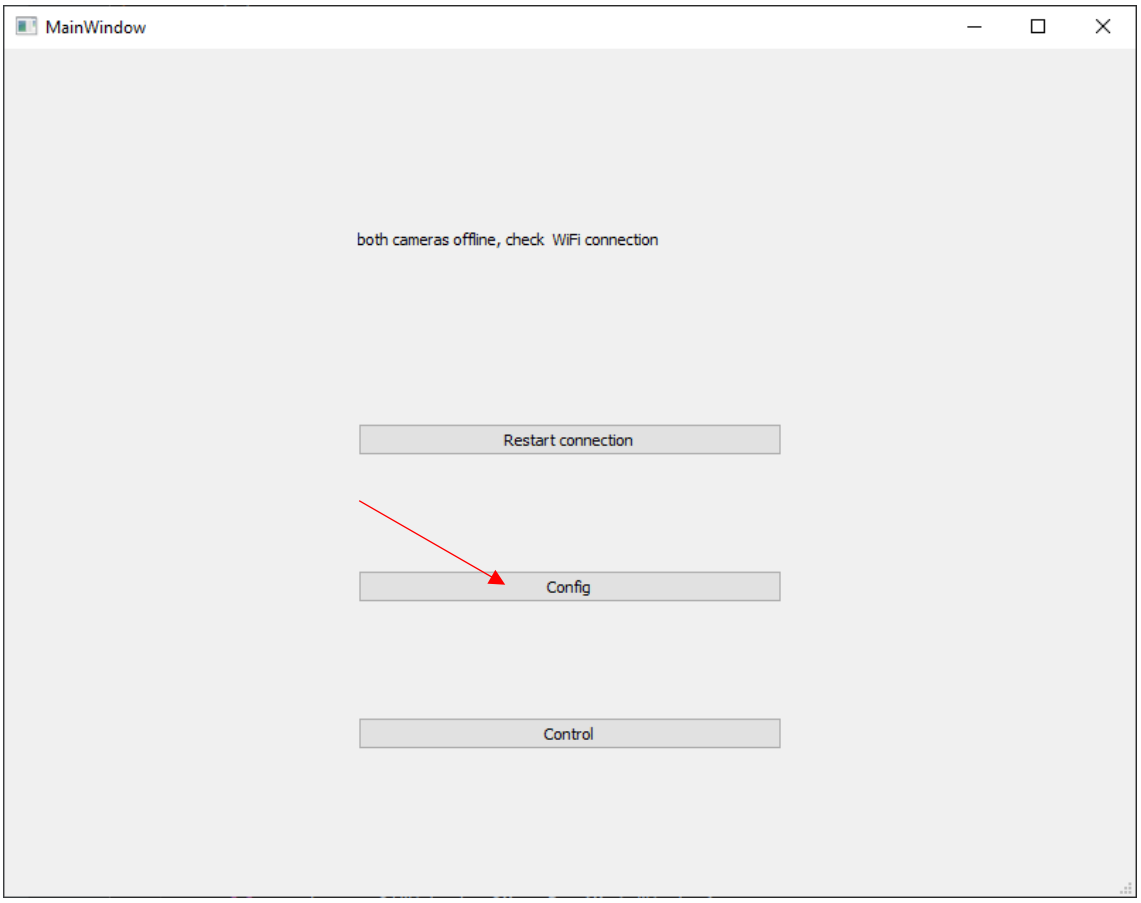

Figure 61. PC Main Window 2

This picture shows the configuration window (Figure 63), in it you can change a lot of options, which we are going to explain:

11.1.1.1 Video quality

- Resolution: In this part we can choose the different resolution of the camera, the maximum resolution of the camera is 1600x1200, but we can drop it down to 400x296 to get a more fluid video feed if needed.
- Quality: The quality scroll set the percentage of the resolution that is rendered, it means that if you set it to 10, it will render the resolution to the 100% of the resolution. If you set it 1600x1200 it will render it to 1600x1200, if you set it to the minimum, it will down render it to 160x120 and it will be more pixelated.
- Brightness: This scroll sets the brightness of the video feed.
- Contrast: This scroll sets the contrast of the video feed.
- Saturation: This scroll sets the saturation of the video feed.

11.1.1.2 Effects and other configurations

- Special effect: Negative, grayscale, red tint, green tint, blue tint ,sepia.

- Automatic White Balance (AWB): White balance is how warm or cool the overall colors in your video feed look.
- AWB gain: It change between two modes of automatic white balance, one more towards warm colors and other towards cool colors.
- White Balance (WB) Mode: Sunny, cloudy, office, home. This change the overall white balance of the modes based on the setting the camera is on.
- Automatic Exposure Control (AEC) Sensor: It maintains optimal self-adjusting exposure. This ensures that acquired images fall in a desirable region of the camera's sensitivity range for optimal image luminance.
- AEC DSP: Automatic exposure control by Digital Signal Processor (DSP) instead of by Image Signal Processor.
- AE Level: This scroll sets the level of exposure of the camera.
- Automatic Gain Control (AGC): Automatic gain control maintains optimal gain settings. This ensures that acquired images fall in a desirable region of the camera's sensitivity range for optimal image luminance.
- Gain Ceiling: This scroll sets the gain ceiling of the camera, this set the maximum gain that the AGC will use.
- Black Pixel Correction (BPC): This setting adjusts the black level of the picture or video feed.
- White pixel correction (WPC): This setting adjusts the white level of the picture or video feed.
- Raw Gamma (GMA): The main purpose of the Gamma (GMA) function is to compensate for the non-linear characteristics of the sensor. Raw gamma compensates the image in the RAW domain.
- Lens Correction: Lens corrections help offset imperfections present in nearly every camera image.
- H-Mirror: Horizontal mirror the video feed or picture.
- V-Flip: Vertical flip the video feed or picture.
- Downside EN (DCW): This setting scales automatically the UV channel value with the sensor gain, it also helps with the special filters.

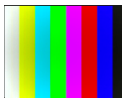

Figure 62. Color bar

- Color Bar: It puts the old color bars that appear on the tv, Figure 62 for reference.
- Face Detection: It starts the software that is able to detect faces, it only works on 500x296 or less resolution because the ESP32-CAM is a low-power microcontroller.
- Face Recognition: It starts the other software that is able to detect faces, it only works on 500x296 or less resolution because the ESP32-CAM is a low-power microcontroller.

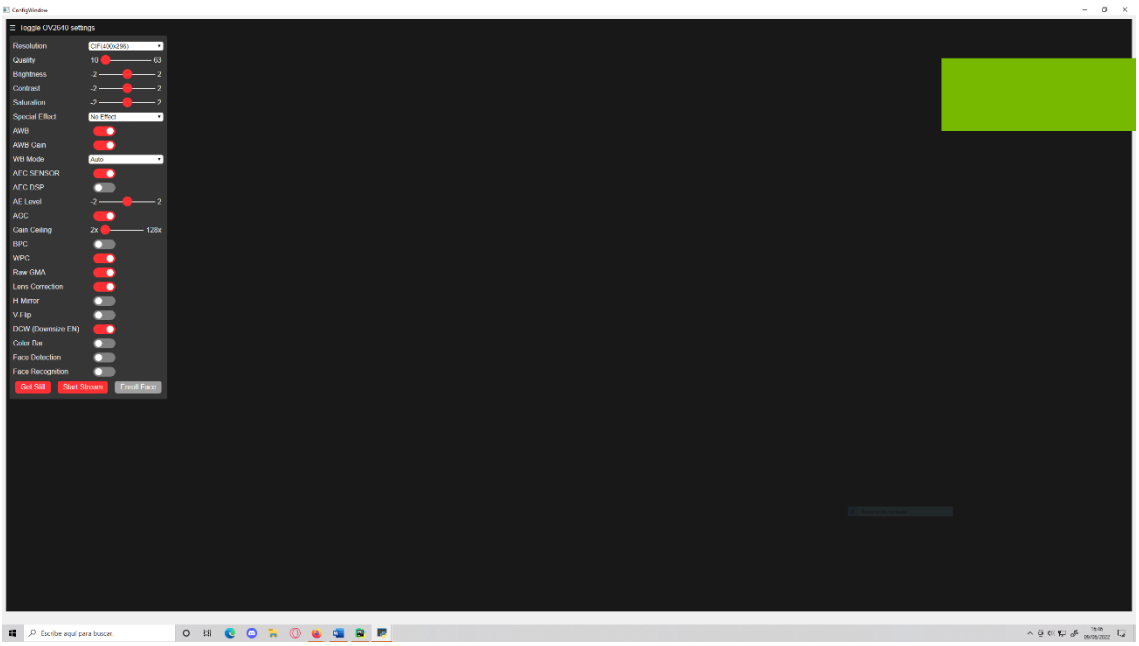

Figure 63. PC Configuration Window

11.1.2 Control

To control the drone, you need to click the control button (Figure 64).

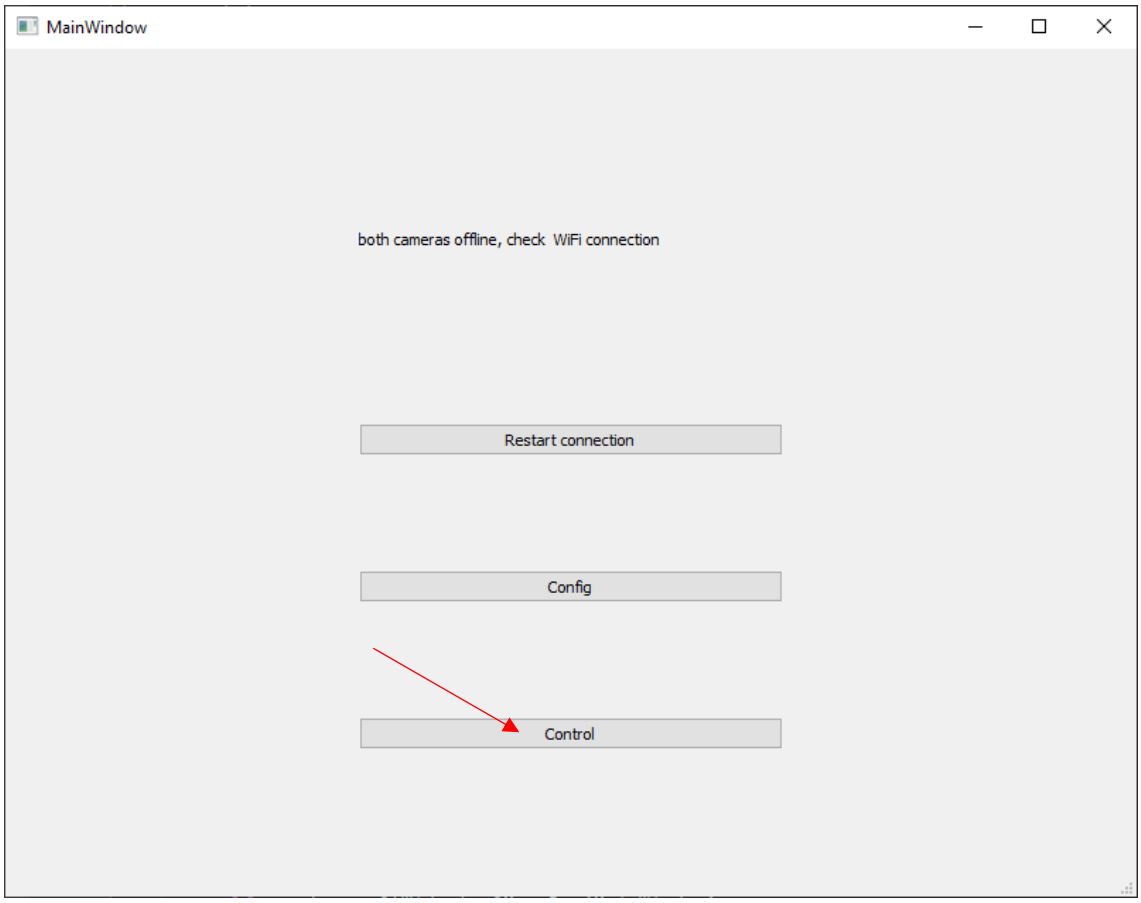

Figure 64. PC Main Window 3

Then you need to click the start button (Figure 65) to start controlling the drone, if you want to stop you just press it again.

Then the commands are by keyboard, W to go forward; Control+W to go forward and to the left; Alt+W to go forward and right. S to go backwards; Control+S to go backwards and to the left and Alt+S to go backwards and to the right. At the same time the user will be able to watch the video feed if it has been started.

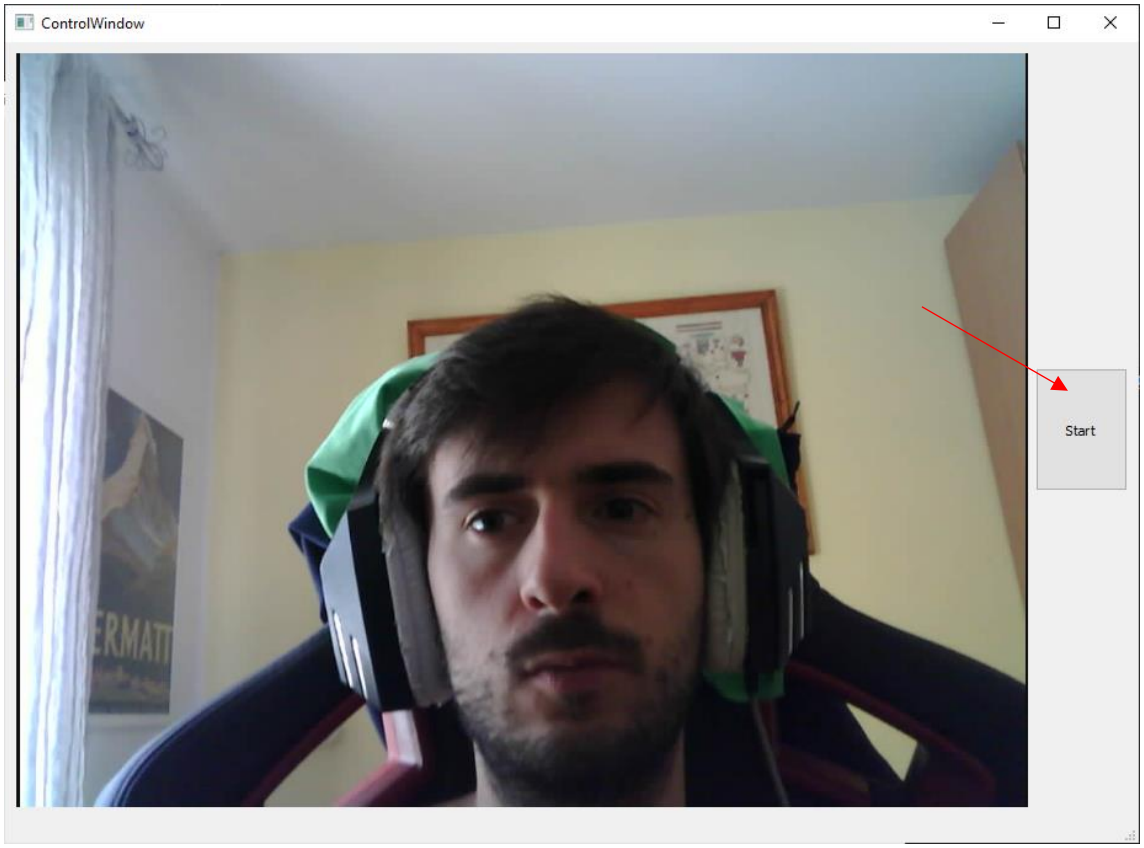

Figure 65. PC Control Window

11.2 Android application

You can connect the Android phone first to the WiFi if you want, if you don't want to, the application will have a window that provides the UI to write the SSID and password of the WiFi to connect to. As seen in the Figure 66, the user have two text fields to insert SSID and password and when the user press the sing in button, the application will connect to that WiFi if the data was introduce right and if there was no other WiFi already connected.

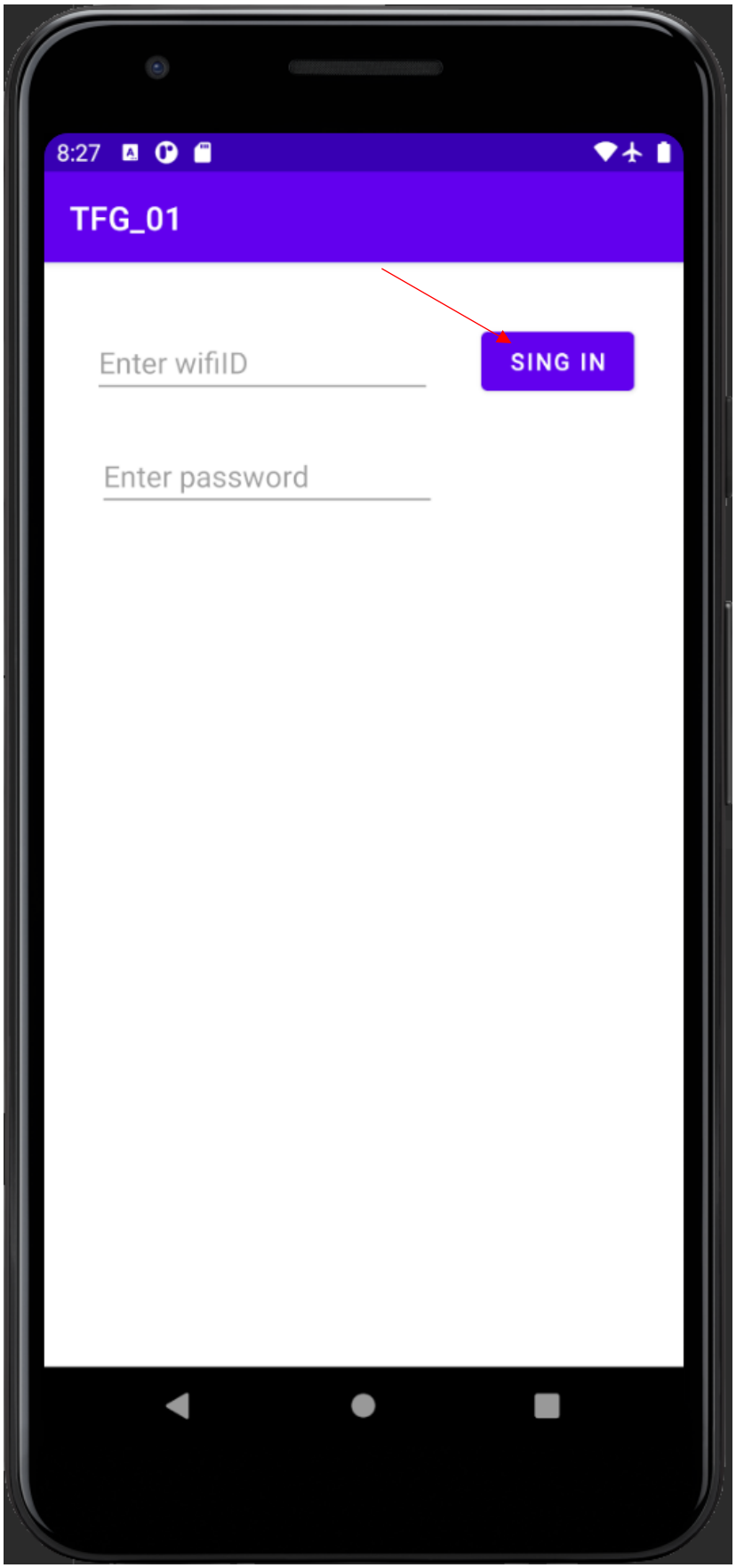

Figure 66. Android sing in window

If you are already connected to the WiFi or you send the connection information, you will have next this screen (Figure 67).

Figure 67. Android Main Window

11.2.1 Configuration

The configuration window let you change the options of the camera feed, to access it you need to click the config button (Figure 68).

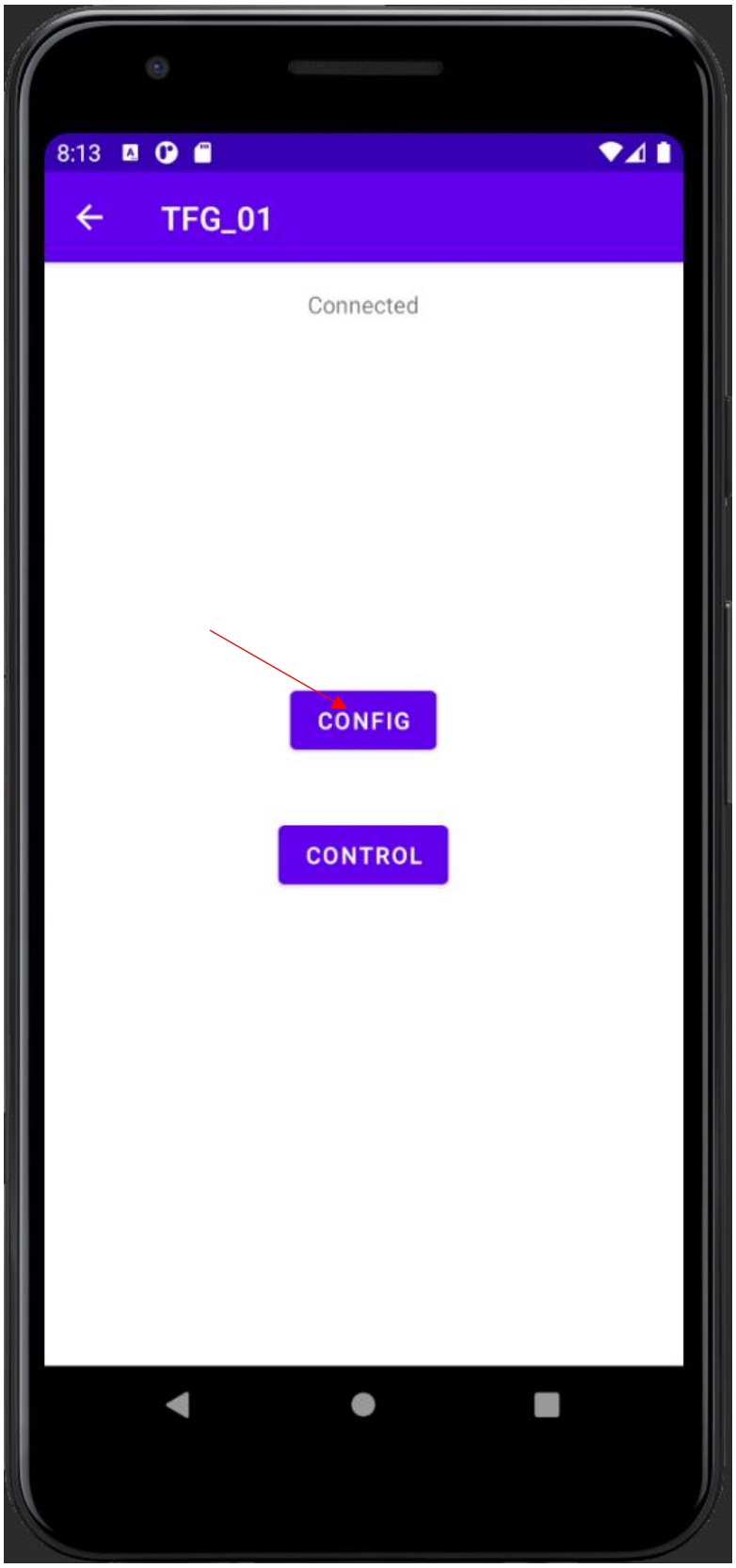

Figure 68. Android Main Window 2

In this section of the application, you have the same configuration options that the one explained earlier in the PC application.

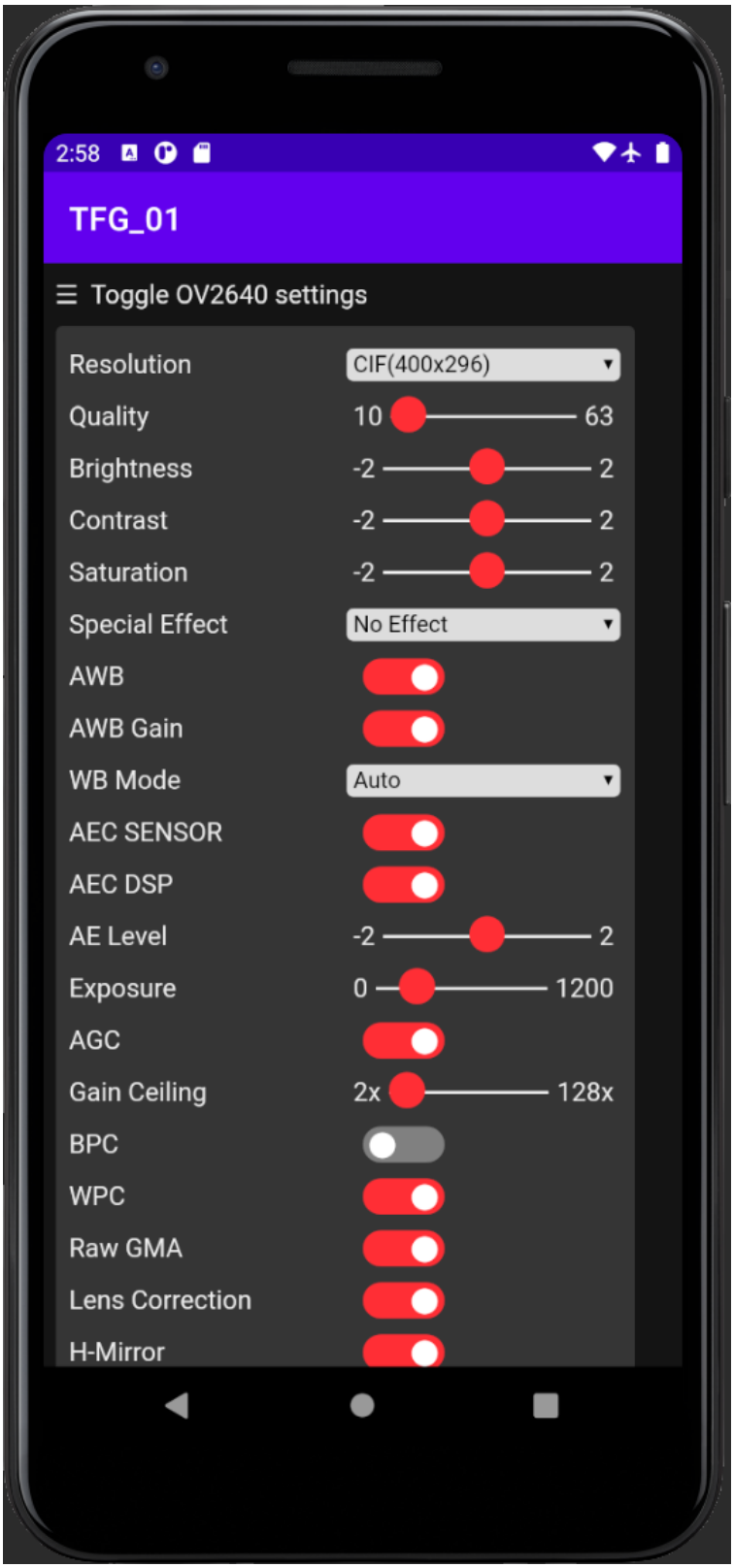

Figure 69. Android Configuration Window

11.2.2 Control

The control window let you get the video feed of the camera and send the signal to the drone to be able to move it, to access it you need to click the control button (Figure 70).

Figure 70. Android Main Window 3

After that you will be able to control the drone by using the joystick on the application while seeing the video feed if it was started (Figure 71), if the user presses the stop button, the drone will stop immediately.

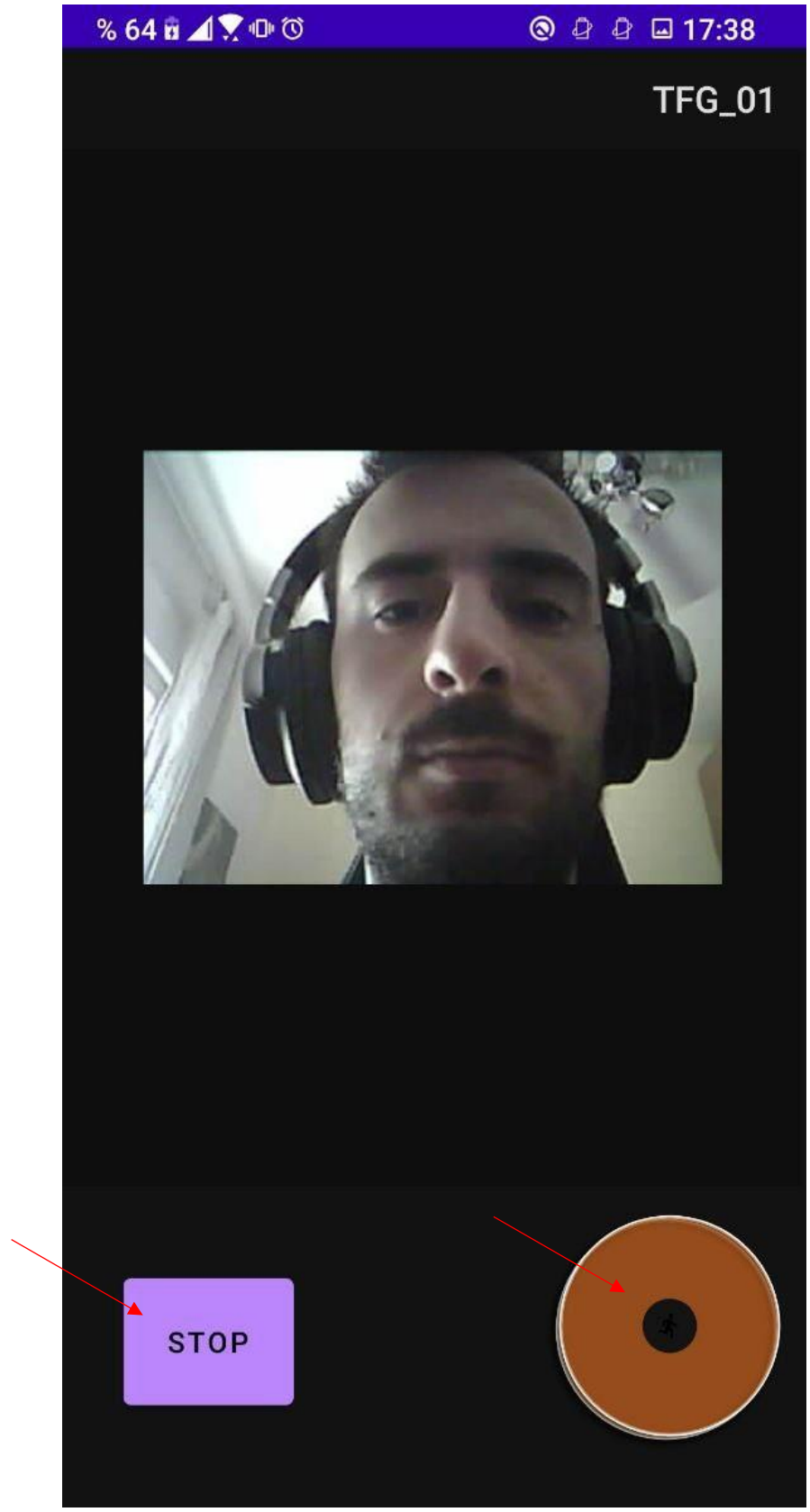

Figure 71. Android Control Window

12. Problems during development and answers

In this chapter we are going to describe the problems we found while developing the project and the answers we found to solve them.

This chapter is going to be divided in four parts: the first one is going to be focused on the hardware problems, the second one is going to pivot around the PC application problems, the third one is going to be focused on the Android application problems and the last one is going to be focus on other types of problems that happened during the development. To show the problems and the answers we provide to them we are going to show code pictures and different screen captures to provide more context to the problems and answers.

12.1 Problems found during the hardware development

As the hardware itself does not have any graphical interface or user interface we only can problems with software related topics or with the quality of the hardware.

12.1.1 Hardware problems

We received some faulty cameras with the ESP32-CAM, and they could not show the image they should be showing. This resulted in having to ask for a refund and having to buy new cameras, after that we could see the image and keep working in the next steps.

12.1.1 Problems related to devices IP

12.1.1.1 ESP8266 IP

At the time of having to receive HTTP petitions it was needed to be raise a WiFi and it was needed to be connected knowing its IP address, the problem that we found when trying that is that in the WiFi library ESP8266 we used this sentence to be able to generate the WiFi connection, but we did not have an IP address to be able to send the HTTP petition only to this device.

```
WiFi.begin(ssid,password);
```

So, we had to find a way of having a fixed IP address ,for this we used the same library with the following instruction, in which you can configure the IP address you want for the device, the gateway and the subnet, achieving to solve this issue.

```
WiFi.config(local_IP,gateway,subnet);
```

12.1.1.2 ESP32-CAM IP

We had the same problem as the ESP82 with the device IP address we needed the IP address, and we did not use the right instruction to configure it, so we needed to do the same as the 12.1.1.1 to solve it.

12.1.2 Problems related to receive the HTTP petitions

One of the most common problems that we had to confront was the control of HTTP petitions and how to get the value needed from it. However, we conclude that we were going to send the data to http://ip/data/ and after that, handle the data through a function to transform the information in a useful way for the hardware to understand it.

```
server.on("/data/",handleSentVar)
```

As shown in the handleSentVar, we receive the data and output it to the controlMotor function to translate it and send it to the motors.

```
void handleSentVar()
{
  if(server.hasArg("plain")==false)
  {
    server.send(200,"text/plain","Data not received");
    return;
  }
  String message = "Body received:\n";
  message += server.arg("plain");
  message += "\n";

  server.send(200, "text/plain", message);
  Serial.println(server.arg("plain"));
  controlMotor(server.arg("plain"));

}
```

12.1.3 Problem related to hardware design and construction

We found a few problems while designing the hardware and the layout of the system, first we had at least two motors to control, but we wanted to control them without having to connect them to different inputs. So, we found a controller make just for that, the L298 controller takes the input from the system and send it to both motors so you do not need to balance the power between both and the speed between them.

12.2 Problems found in the development of the Android application

We found a few problems related with the graphical interface and some functionalities in the development of the Android application, this we are going to explain next.

12.2.1 Problem with WiFi connection

We wanted to be able to connect the application to the WiFi network we wanted, but wifimanager Android library is becoming deprecated, nowadays you should not force the user application to connect to the WiFi we need to use, instead you should ask the user to connect to the WiFi and the user should accept.

Then to make the new connection we needed to use the WiFNetworkSuggestion Android library instead of using the deprecated wifimanager:

```
final WifiNetworkSuggestion suggestion = new Builder().setSsid(message).setWpa2Passphrase(message2).build();

final List<WifiNetworkSuggestion> suggestionList = new ArrayList<WifiNetworkSuggestion>();
suggestionList.add(suggestion);

Context context = getApplicationContext();
final WifiManager wifimanager = (WifiManager) context.getSystemService(Context.WIFI_SERVICE);
final int status = wifimanager.addNetworkSuggestions(suggestionList);
```

12.2.2 Problem with the UI and Android

There is not a native joystick inside the Android API, so we had to find a joystick to be able to input the control of the hardware, we used the jackandphantom joystick.

```
import com.jackandphantom.joystickview.JoyStickView;
```

There is a problem with this joystick: if it goes back to the center of the circle, the joystick does not send that it is on the 0.0 position. So, we had to find an answer to it. We decided to send the 0.0 at the beginning of every instruction. If the joystick is not moving or in center, it will not set the angle or strength and this will make the connection only send 0.0, but if the joystick is moving, we will only send the angle and strength and not the 0.0 at the beginning.

```
JoyStickView joyStick = (JoyStickView) findViewById(R.id.joystick);
joyStick.setOnMoveListener(new JoyStickView.OnMoveListener() {

    @Override
    public void onMove(double angle, float strength) {

        connectionURL(0,0);

        sleep(1);

        connectionURL(angle,strength);

        sleep(100);
```

12.2.3 Problem with accessing the webserver in the application

When accessing the webserver to get the video feed or the configuration webpages, we found that the application reported an error called:”err-cleartext-not-permited”. It means that cleartext such as the webpages that are HTTP instead of HTTPS.

To fix it we needed to add in the manifest an XML with the network-security-config, and in it we could give permission to access some domains with cleartext.

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">192.168.1.184</domain>
        <domain includeSubdomains="true">192.168.1.110</domain>
    </domain-config>
</network-security-config>
```

12.3 Problems with the PC application development

We had a few problems with the PC application development. We used Python as the main language to develop the PC application because it is an easy and fast language to code prototype applications.

12.3.1 Problem with the PC application not getting the input

One of the biggest issues we had with the PC application was that we wanted to get the input of the user by using the keyboard like a normal control. If the user pressed the W key the software would send the order to go forward, if the user pressed the W+A key, the software would send the order to go forward and left, if the user pressed the W+D key, the software would send the order to go forward and right, if the user pressed the S key, the software would send the order to go backwards, if the user pressed the S+A key, the software would send the order to go backwards and left, if the user pressed the S+D key, the software would send the order to go backwards and right. But the Python library(pynput) we used does not recognize more than one key stroke per input, and you could not get more than one key stroke per input, so we could not press multiple keys at the same time.

We managed to fix it with pyqt5 GUI. We used this sentence to manage to get the key sequence of the user, and this allowed us to use a combination of multiple keys at the same time.

```
self.forward = QShortcut(QKeySequence("w"),self.MainWindow3)
self.forward.activated.connect(self.sendForwardInput)
```

13. Budget

This chapter will describe the time and economic budget of the project by using different diagrams to explain it.

13.1 Time budget

13.1.1 Gantt diagram of estimated time spent

This picture is going to be in the annex because it needs to be horizontal to be able to understand it, annex 1 (Figure 72). I am going to explain a bit why those times, first we wanted to study the environments and other factors like I explained before in other chapters of the document, for that we gave a few days per each study, because we knew it was not going to take a long time to study each element. We assigned more time to development because we assumed that it was going to take more time to finish it and to test it. During the whole process we had meetings with the teacher, and we were writing this document.

13.1.2 Gantt diagram of real time spent

This picture is also going to be in the annex because it needs to be horizontal to be able to understand it, annex 1 (Figure 73).

13.1.3 Deviation

As shown in the different diagrams, there has been some deviation from the estimated time. The main reason that we have some deviation in the time spent is that we needed to stop working in the development to study other subjects and we needed more time to fix some problems in the development that we did not plan.

13.2 Economic Budget

In this chapter we are going to find the expenses of every part of the project: hardware, services, and engineering cost.

Item	Cost per unit(€)	Unit	Total price(€)
ESP32-CAM	10,30	2	20,59
Wemos D1 Mini	5	1	5
L298N	9,70	1	9,70
Motor DC 3-6V	1	2	2
Battery holder	2	2	4
All			41,29

Concept	Cost per hour	Hours	Total price
Engineer	10	450	4500

The total budget spent in the project was 4541,29€ and as shown in the last table, in this project the majority of the budget was invested in the engineering team that works on the project.

14. Environmental conscience and ethical responsibility

In this chapter we are going to study the impact this development and system can have in society and the environment, on top of other areas such as ethical responsibility and professional responsibility.

The main objective of this project consists of making a ground-based drone that provide more security in the world; this project brings a lot of ethical questions such as:

- Can we leave our security to drones piloted by humans instead of human forces on the field?
- Can we have ground-based drones and flying drones around without any sort of visual indication to be easier to see the drones?
- Is this project and other project like this influence the dismissal of security workers because drones are going to replace them?
- Is society ready to adapt to this paradigm change?

Based in our system, we would still have people in charge of the drones cameras and system, so the drones would not be the only system behind to protect people and improve security, which rules out the idea that our security will be left only on drones hands; on the other hand it is true that we could use AI powered applications to improve the rate of law violation by the drones, but that is out the scope of this project.

We have to think also that this system will probably improve security around the world, which should make it easier for everyone to go outside without being afraid of anything, if this technology is implemented in cities, it will also make it easier for everyone to walk in the city at every time of the day, without having to worry about being attacked or robbed.

In terms of visual and physical security of the drones, maybe we should have to paint them in bright colors or add some sort of sound indication to secure that the drone is not a liability to human security in pos of other sort of security. If the drone is going to be used in an application that does not require to have this system around people, then we do not need to do any extra work.

When the industrial revolution started a lot of workers lost their jobs and other types of works were created at the same time, which each improvement and revolution jobs and human relations change. In this revolution security workers will maybe not be fire because drones are going to replace them, the workers should only need to receive a small course on how to operate the new drones.

With this new improvement security workers should not have to suffer the bad climate when having to patrol around a base or around a warehouse. They may work operating the drones and watching the cameras, which is a work that some security companies already have.

As seen inside the state of the art, some research already shows that society is ready to adapt and trust automated systems to take small decisions, in this case the decisions are taken by the worker that operate the drones. So, we can say that society is ready to have advance security based on drones operated by the workers, which will have the choice to use AI to have a helping hand to make it easier to operate or just work on manual mode.

In professional responsibility we need to ask ourselves other set of questions. Are we following the needed procedures to have a secure system? In every project the developer will try to follow every procedure to have a secure system, because it is going to work with humans and other resources, we need to follow some secure procedures.

14.1 Environmental conscience

To evaluate this fact, we have some studies that shows that ground transport is more efficient than flying transport, but that does not mean that the project is entirely environmentally friendly. We have to consider the environmental cost that have every drone that is produce and the energy that we have to spend to be able to recharge them.

To consider the difference in environmental impact between our system and the security nowadays we would have to make a comparison between the environmental impact by the drone and the human worker controlling it inside a building and the impact of the team of people patrolling around.

This project also helps to reduce the carbon footprint because all our drones are powered by electricity. This will reduce the carbon footprint because the security teams nowadays usually use cars to move around. Some parts of the development and building of our drones are carbon footprint free, but some must have some carbon footprint for now until new methods are developed. The industries that produce electronic boards are carbon footprint free because they use electricity to power their machines, that electricity could be produce with non-renewable technology, which will impact and make it not carbon footprint free. The other part of this process that is not carbon footprint free is the other industries that provide chemicals to be able to produce the electronic boards. To obtain these chemicals within a budget, the industries still use old systems such as carbon powered machines and invasive and toxic chemicals.

And last, we have to acknowledge the elephant in the room, e-waste. It is something that has not been solve to this day, and we can do nothing about it, we tried to develop using small boards and cheap ones that do not contain a lot of metals but even with that, every time an electronic system is developed some e-waste will be created in the future. We as engineers can try to reduce the waste, but some will be produced. Until we have a method to recycle or reuse this boards, we will have to keep working towards the future while thinking of being sustainable.

15. Conclusion

The objective of this project was to make a security advance and provide the user with enough flexibility to have a secure environment with this new technology, we also wanted to develop a low-cost answer with an easy-to-use UI and tools.

We found some development that focused on automatic ground drones to recollect objects or to control environmental variables, we also studied the tools needed to make a development of this characteristics. Two of the most important tasks related to the previous work before development have been:

- Pinpoint of milestones, a task related with the definition of the requirements of the systems. This definition of milestones has been really useful to help at the time of tracking the advance in the project development and it has let us make an estimate of the time spent in every task and time needed to finish the project.
- Planification with agile methodologies. The agile methodologies have let us work closer to the task and milestones without having to worry about the big picture since the beginning of the development. It also gave us the opportunity to test every new sprint before calling it done.

This task reflects that there is a need of planification and structure inside big projects. Design was also a big part of the planification and structure in the project. We could have not finished this project on time without the help of the design to make everything more structured.

We have learned a lot about working with libraries and how not to trust all the libraries documentation until you code it, because half of the problems we found during development were supposed to be solve by the libraries we found in the previous study. But some were not because the libraries documentation was old, or the libraries did not work as the documentation indicated.

We finished the objectives of the project, and we showed that there are new answers to the security problems in cities and warehouses. We were able to accomplish all the objectives of the project, and now we maybe be able to reduce the criminal activities and help security.

16. Future lines

During this document we have shown a project that gives the user two additional tools to work with: the Android application and the PC application. In this chapter we are going to talk about new ideas to expand the project. The new future lines could be:

16.1 Accessibility

We have seen that nowadays accessibility is one of the main points to improve in every application that has been online for some time, and our application could have a new phase with new accessibility features such as the possibility of:

- New colors to focus on the specific parts of the application text.
- Options to increase or decrease the size of the buttons or text.
- Use of pictures or easy to understand images of the features inside the application.

16.2 Functionality

We have thought of adding new features to the application and software, we are going to show some examples of ideas:

- Adding a GPS receptor in the drone to be able to pinpoint the location of the drone in a map.

- Adding a feature to be able to code a path for the drone to follow, that way the drone can patrol with that path.

- Adding a feature to be able to connect the application with other types of drones such as flying drones or other types of ground-based drones. With that you could have the skill to be able to see the video feed of the area with different points of view.

- Make a charging base for the drones, with it if the drones detect that they have low level battery, they can go back to the charging base by their own, this base could have wireless charging or expose copper connection this will make it easier for the drone to connect itself without worrying about the connection.

- Develop a database of the system data to be able to process it later and make detailed explanations of change in the system. For example, add a new sensor to learn about the ground and be able to understand the energy spent in relation with the ground topography.

- Use new systems to send the HTTP petitions without having to make them ourselves, either with automated systems or with new APIs.

- Use newer systems to make the connection HTTPS or at least use new systems or APIs to have more secure messages and communications.

17. References

[1] N. Surantha and W. R. Wicaksono, "Design of Smart Home Security System using Object Recognition and PIR Sensor," *Procedia Computer Science*, vol. 135, pp. 465–472, Jan. 2018, doi: 10.1016/J.PROCS.2018.08.198.

[2] R. Hasan, M. M. Khan, A. Ashek, and I. J. Rumpa, "Microcontroller Based Home Security System with GSM Technology," *Open Journal of Safety Science and Technology*, vol. 05, no. 02, pp. 55–62, 2015, doi: 10.4236/OJSST.2015.52007.

[3] D. Hoornweg and K. Pope, "Population predictions for the world's largest cities in the 21st century," *Environment and Urbanization*, vol. 29, no. 1, pp. 195–216, Apr. 2017, doi: 10.1177/0956247816663557.

[4] "Military expenditure (current USD) | Data." https://data.worldbank.org/indicator/MS.MIL.XPND.CD?end=2020&start=1988&view=chart (accessed Jun. 04, 2022).

[5] "Improving Police Efficiency – Challenges and Opportunities." https://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/2015-r021/index-en.aspx (accessed Jul. 08, 2022).

[6] T. Kirschstein, "Comparison of energy demands of drone-based and ground-based parcel delivery services," *Transportation Research Part D: Transport and Environment*, vol. 78, p. 102209, Jan. 2020, doi: 10.1016/J.TRD.2019.102209.

[7] J. W. Atkins, A. E. L. Stovall, and X. Yang, "Mapping Temperate Forest Phenology Using Tower, UAV, and Ground-Based Sensors," 2020, doi: 10.3390/drones4030056.

[8] B. Arad *et al.*, "Development of a sweet pepper harvesting robot," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1027–1039, Sep. 2020, doi: 10.1002/rob.21937.

[9] D. Gallimore, J. B. Lyons, T. Vo, S. Mahoney, and K. T. Wynne, "Trusting robocop: Gender-based effects on trust of an autonomous robot," *Frontiers in Psychology*, vol. 10, no. MAR, p. 482, 2019, doi: 10.3389/FPSYG.2019.00482/BIBTEX.

[10] D. Gallacher, "Drones to manage the urban environment: Risks, rewards, alternatives," *Journal of Unmanned Vehicle Systems*, vol. 4, no. 2, pp. 115–124, 2016, doi: 10.1139/JUVS-2015-0040/ASSET/IMAGES/LARGE/JUVS-2015-0040FIG2.JPEG.

[11] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "IoT Privacy and Security: Challenges and Solutions," *Applied Sciences 2020, Vol. 10, Page 4102*, vol. 10, no. 12, p. 4102, Jun. 2020, doi: 10.3390/APP10124102.

[12] Institute of Electrical and Electronics Engineers. and IEEE Robotics and Automation Society., *2010 IEEE Workshop On Advanced Robotic and its Social Impacts : IEEE ARSO 2010 : conference proceedings : October 26-28, 2010, International Cooperation Building, KIST Korea.*

[13] C. Zheyuan, M. A. Rahman, H. Tao, Y. Liu, D. Pengxuan, and Z. M. Yaseen, "Need for developing a security robot-based risk management for emerging practices in the workplace using the Advanced Human-Robot Collaboration Model," *Work*, vol. 68, no. 3, pp. 825–834, 2021, doi: 10.3233/WOR-203416.

[14] L. Penghou, C. Haichao, and D. Yanzhe, “Design of Multifunctional Intelligent Security Robot Based on Single Chip Microcomputer,” in *Journal of Physics: Conference Series*, May 2019, vol. 1187, no. 3. doi: 10.1088/1742-6596/1187/3/032017.

[15] Institute of Electrical and Electronics Engineers. Madras Section and Institute of Electrical and Electronics Engineers, *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing (ICCSP) : 28th - 30th July 2020, Melmaruvathur, India.*

[16] “WIRELESS BASED INTELLI INDUSTRIAL SECURITY ROBOT”.

[17] “https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/.”

[18] “What is Agile software development? - K&C.” https://kruschecompany.com/agile-software-development/ (accessed Jun. 04, 2022).

[19] “serarandal/TFG: This is my degree final project.” https://github.com/serarandal/TFG (accessed Jul. 07, 2022).

[20] “Top 4 software development methodologies | Synopsys.” https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/ (accessed Jul. 06, 2022).

[21] “Top 4 software development methodologies | Synopsys.” https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/ (accessed Jun. 04, 2022).

[22] “PROTOTYPE METHODOLOGY - KNEC STUDY MATERIALS, REVISION KITS AND PAST PAPERS.” https://knecnotes.co.ke/2021/07/03/prototype-methodology/ (accessed Jul. 06, 2022).

[23] “PyCharm: the Python IDE for Professional Developers by JetBrains.” https://www.jetbrains.com/pycharm/ (accessed Jun. 10, 2022).

[24] “PyCharm: the Python IDE for Professional Developers by JetBrains.” https://www.jetbrains.com/pycharm/ (accessed Jul. 06, 2022).

[25] “Archivo:Visual Studio Code 1.18 icon.svg - Wikipedia, la enciclopedia libre.” https://es.m.wikipedia.org/wiki/Archivo:Visual_Studio_Code_1.18_icon.svg (accessed Jun. 10, 2022).

[26] “Visual Studio Code - Code Editing. Redefined.” https://code.visualstudio.com/ (accessed Jul. 06, 2022).

[27] “átomo, Editor De Texto, Editor De Código Fuente imagen png - imagen transparente descarga gratuita.” https://www.freepng.es/png-pry2ek/ (accessed Jun. 10, 2022).

[28] “Atom.” https://atom.io/ (accessed Jul. 06, 2022).

[29] “Descargar Eclipse IDE gratis - 2022 Última versión.” https://es.wizcase.com/download/eclipse-ide/ (accessed Jun. 10, 2022).

[30] “Eclipse desktop & web IDEs | The Eclipse Foundation.” https://www.eclipse.org/ide/ (accessed Jul. 06, 2022).

[31] “Arduino - Wikipedia, la enciclopedia libre.” https://es.wikipedia.org/wiki/Arduino (accessed Jun. 10, 2022).

[32] “Software | Arduino.” https://www.arduino.cc/en/software (accessed Jul. 06, 2022).

[33] “Android Studio 3.0: ya disponible la versión estable con soporte para Kotlin y las nuevas APIs de Android.” https://www.xatakandroid.com/programacion-android/android-studio-3-0-ya-disponible-la-version-estable-con-soporte-para-kotlin-y-las-nuevas-apis-de-android (accessed Jun. 10, 2022).

[34] “Download Android Studio & App Tools - Android Developers.” https://developer.android.com/studio (accessed Jul. 06, 2022).

[35] “IntelliJ IDEA - Wikipedia, la enciclopedia libre.” https://es.wikipedia.org/wiki/IntelliJ_IDEA (accessed Jun. 10, 2022).

[36] “IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains.” https://www.jetbrains.com/idea/ (accessed Jul. 07, 2022).

[37] “Microsoft lanza Visual Studio 2019, disponible también para Mac con un nuevo editor.” https://www.eleconomista.es/tecnologia/noticias/9800724/04/19/Microsoft-lanza-Visual-Studio-2019-disponible-tambien-para-Mac-con-un-nuevo-editor.html (accessed Jun. 10, 2022).

[38] “Visual Studio: IDE and Code Editor for Software Developers and Teams.” https://visualstudio.microsoft.com/ (accessed Jul. 07, 2022).

[39] “Notepad++ - Viquipèdia, l’enciclopèdia lliure.” https://ca.wikipedia.org/wiki/Notepad%2B%2B (accessed Jun. 10, 2022).

[40] “Scintilla and SciTE.” https://www.scintilla.org/ (accessed Jul. 06, 2022).

[41] “Notepad++.” https://notepad-plus-plus.org/ (accessed Jul. 06, 2022).

[42] “Sublime Text - Wikipedia, la enciclopedia libre.” https://es.wikipedia.org/wiki/Sublime_Text (accessed Jun. 10, 2022).

[43] “Sublime Text - Text Editing, Done Right.” https://www.sublimetext.com/ (accessed Jul. 06, 2022).

[44] “Postman, realiza peticiones HTTP y prueba tus API REST.” https://www.luisllamas.es/realizar-peticiones-http-con-postman/ (accessed Jun. 10, 2022).

[45] “Postman API Platform | Sign Up for Free.” https://www.postman.com/ (accessed Jul. 06, 2022).

[46] “XpertLab - Website Designing Company Junagadh,Website Development Company Junagadh, Software Development Company Junagadh, Professional IT Project Training Junagadh.” https://xpertlab.com/amazon-api-gateway/ (accessed Jun. 10, 2022).

[47] “Amazon API Gateway | API Management | Amazon Web Services.” https://aws.amazon.com/api-gateway/ (accessed Jul. 07, 2022).

[48] “MuleSoft Anypoint Platform Reviews 2022: Details, Pricing, & Features | G2.” https://www.g2.com/products/mulesoft-anypoint-platform/reviews (accessed Jun. 10, 2022).

[49] “MuleSoft | Integration Platform for Connecting SaaS and Enterprise Applications.” https://www.mulesoft.com/ (accessed Jul. 07, 2022).

[50] “Azure API Managemnet - Jump Start - TechMindFactory.com.” https://techmindfactory.com/Azure-API-Management-Jump-Start/ (accessed Jun. 10, 2022).

[51] “API Management – Manage APIs | Microsoft Azure.” https://azure.microsoft.com/en-us/services/api-management/ (accessed Jul. 07, 2022).

[52] “Cómo actualizar Java en tu ordenador.” https://www.xataka.com/basics/como-actualizar-java-tu-ordenador (accessed Jun. 10, 2022).

[53] “What is Java and why do I need it?” https://www.java.com/en/download/help/whatis_java.html (accessed Jul. 06, 2022).

[54] “ 👨🏻‍🏭 Mi primera aplicación en Kotlin.” https://blog.sosafeapp.com/mi-primer-aplicacion-en-kotlin/ (accessed Jun. 10, 2022).

[55] “Kotlin Programming Language.” https://kotlinlang.org/ (accessed Jul. 07, 2022).

[56] “C Language - Publicaciones | Facebook.” https://es-la.facebook.com/pg/C-Language-293990047842373/posts/ (accessed Jun. 10, 2022).

[57] “C++ - Wikipedia, la enciclopedia libre.” https://es.wikipedia.org/wiki/C%2B%2B (accessed Jun. 10, 2022).

[58] “cplusplus.com - C++.” https://m.cplusplus.com/ (accessed Jul. 07, 2022).

[59] “AmForth — AmForth.” http://amforth.sourceforge.net/ (accessed Jun. 10, 2022).

[60] “AmForth — AmForth.” http://amforth.sourceforge.net/ (accessed Jul. 07, 2022).

[61] “AVR-Ada / Wiki / Home.” https://sourceforge.net/p/avr-ada/wiki/Home/ (accessed Jul. 07, 2022).

[62] “Historia de Python - Wikipedia, la enciclopedia libre.” https://es.wikipedia.org/wiki/Historia_de_Python (accessed Jun. 10, 2022).

[63] “Welcome to Python.org.” https://www.python.org/ (accessed Jul. 07, 2022).

[64] “Golang el lenguaje que me ha cautivado | by Benjamin De la Cruz | Benjamin De la cruz Martinez.” https://benjamin.kimvex.com/golang-el-lenguaje-que-me-ha-cautivado-582596ee2edc (accessed Jun. 10, 2022).

[65] “The Go Programming Language.” https://go.dev/ (accessed Jul. 07, 2022).

[66] “Archivo:Git-logo.svg - Wikipedia, la enciclopedia libre.” https://es.m.wikipedia.org/wiki/Archivo:Git-logo.svg (accessed Jun. 10, 2022).

[67] “Git.” https://git-scm.com/ (accessed Jul. 07, 2022).

[68] “Subversion (software) - Wikipedia, la enciclopedia libre.” https://es.wikipedia.org/wiki/Subversion_%28software%29 (accessed Jun. 10, 2022).

[69] “Apache Subversion.” https://subversion.apache.org/ (accessed Jul. 07, 2022).

[70] “(177) String Request | Android Volley - YouTube.” https://www.youtube.com/watch?v=LNrE0f-V1O4 (accessed Jun. 10, 2022).

[71] “Volley overview | Volley.” https://google.github.io/volley/ (accessed Jul. 07, 2022).

[72] “GitHub - sparrow007/AndroidJoyStickView: This library lets you create joystick with some customization for android.” https://github.com/sparrow007/AndroidJoyStickView (accessed Jul. 07, 2022).

[73] “android.net | Android Developers.” https://developer.android.com/reference/android/net/package-summary (accessed Jul. 07, 2022).

[74] “android.content | Android Developers.” https://developer.android.com/reference/android/content/package-summary (accessed Jul. 07, 2022).

[75] “android.os | Android Developers.” https://developer.android.com/reference/android/os/package-summary (accessed Jul. 07, 2022).

[76] “android.widget | Android Developers.” https://developer.android.com/reference/android/widget/package-summary (accessed Jul. 07, 2022).

[77] “Webkit | Android Developers.” https://developer.android.com/jetpack/androidx/releases/webkit?hl=en-419 (accessed Jul. 07, 2022).

[78] “Download PyQt Logo in SVG Vector or PNG File Format - Logo.wine.” https://www.logo.wine/logo/PyQt (accessed Jun. 10, 2022).

[79] “PyQt5 · PyPI.” https://pypi.org/project/PyQt5/ (accessed Jul. 07, 2022).

[80] “Requests: HTTP for Humans™ — Requests 2.28.0 documentation.” https://requests.readthedocs.io/en/latest/ (accessed Jun. 10, 2022).

[81] “Requests: HTTP for Humans™ — Requests 2.28.1 documentation.” https://requests.readthedocs.io/en/latest/ (accessed Jul. 07, 2022).

[82] “urllib3 1.26.9 documentation.” https://urllib3.readthedocs.io/en/stable/ (accessed Jun. 10, 2022).

[83] “urllib3 1.26.9 documentation.” https://urllib3.readthedocs.io/en/stable/ (accessed Jul. 07, 2022).

[84] “WiFi - Arduino Reference.” https://www.arduino.cc/reference/en/libraries/wifi/ (accessed Jul. 07, 2022).

[85] “esp32-camera/esp_camera.h at master · espressif/esp32-camera · GitHub.” https://github.com/espressif/esp32-camera/blob/master/driver/include/esp_camera.h (accessed Jul. 07, 2022).

[86] “ESP8266WiFi library — ESP8266 Arduino Core documentation.” https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html (accessed Jul. 07, 2022).

[87] “Arduino/ESP8266WebServer.h at master · esp8266/Arduino · GitHub.” https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WebServer/src/ESP8266WebServer.h (accessed Jul. 07, 2022).

[88] “ESP32-CAM Wi-Fi+BT SoC Module V1.0 ESP32-CAM Module Overview,” 2017.

[89] STMicroelectronics, “DUAL FULL-BRIDGE DRIVER,” 2000.

[90] “ESP8266-datasheet.” [Online]. Available: https://www.espressif.com/en/subscribe.

[91] “sch_d1_mini_v3.0.0schematic”.

[92] “Bluetooth - Wikipedia.” https://en.wikipedia.org/wiki/Bluetooth (accessed Jul. 07, 2022).

18. Annex 1 – Big diagrams

18.1 Gantt diagram of estimated time spent

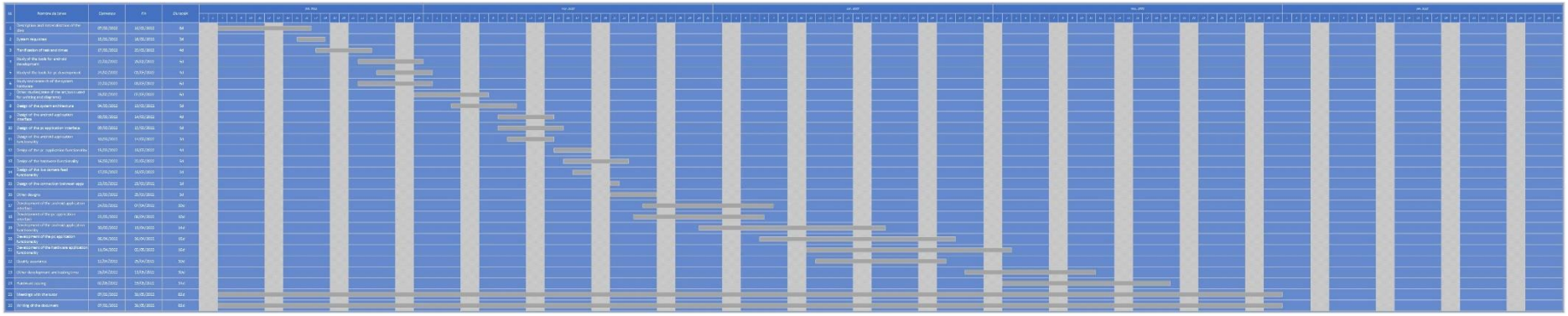

Figure 72. Gantt diagram of estimated time spent

18.2 Gantt diagram of real time spent

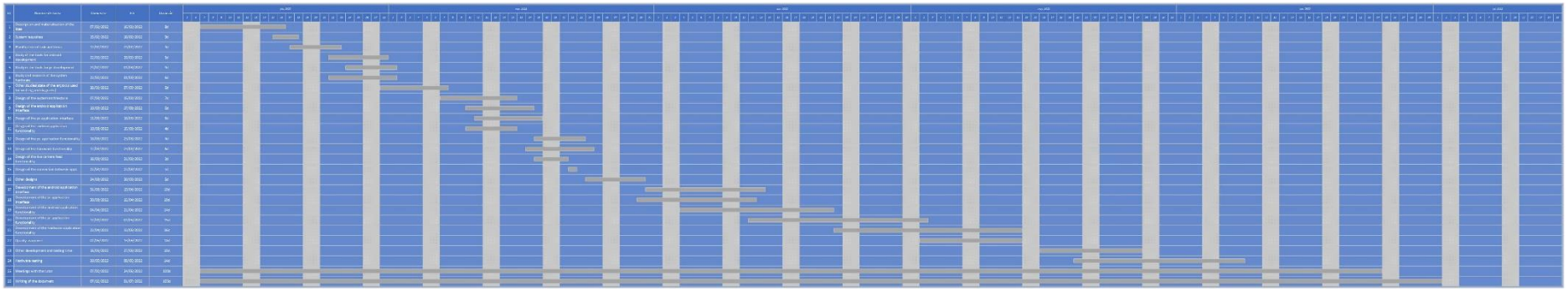

Figure 73. Gantt diagram of real time spent