DEMETER PROJECT

13/03/2021

MANAGING ROBOTS OVER THE INTERNET

DemeteR

"Self-education is, I firmly believe, the only kind of education there is." **Isaac Asimov**, writer and teacher.

Co-funded by the
Erasmus+ Programme
of the European Union

INDEX

UNIT: MANAGING ROBOTS OVER THE INTERNET

Introduction

In this introduction we are going to discuss the methodological and motivational principles to be followed in this unit.

To work on this unit, the following methodology is proposed:

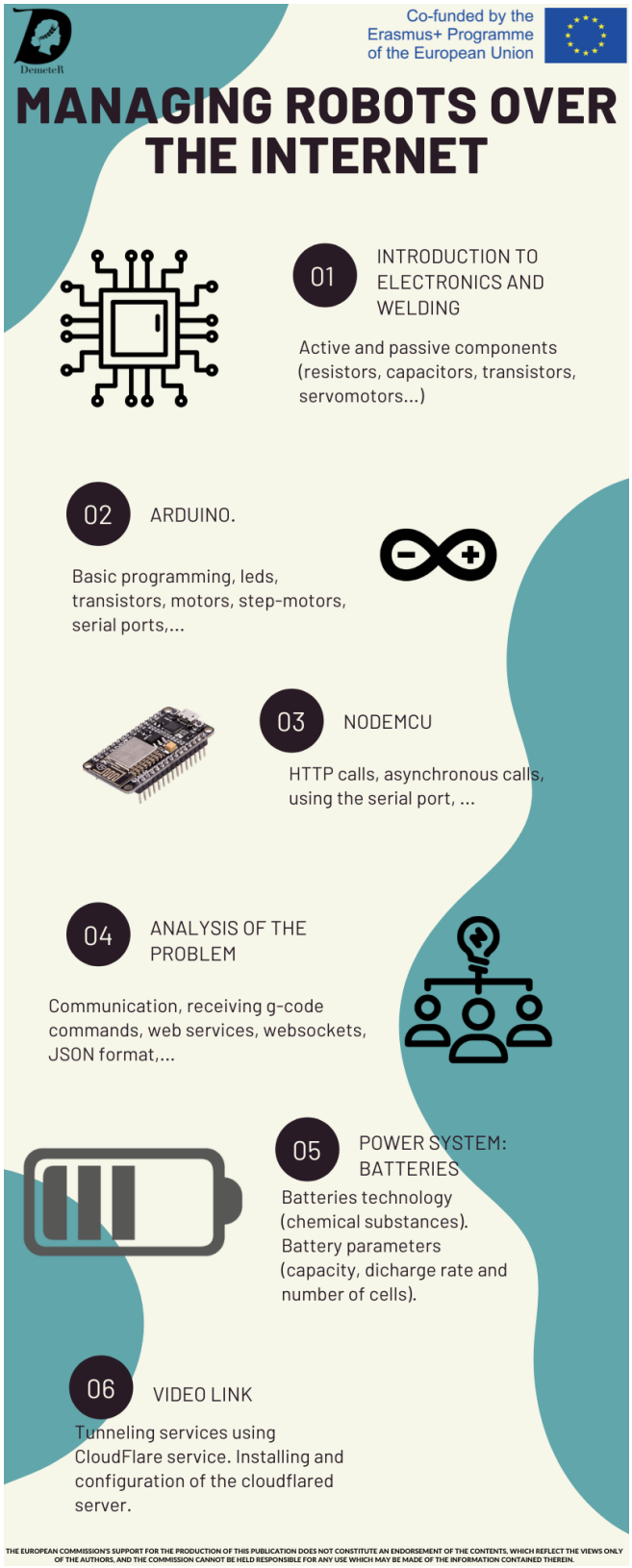
- Exact knowledge of the objectives pursued as a motivating element for the student.
- Provide the mathematical, physical or other elements necessary for understanding the proposed content.
- Didactically it will be accompanied by abundant figures, diagrams, and utility programs that synthesize the procedures that are explained textually.
- They will be accompanied by abundant references and documentation, as support for what has been studied and to relate it to other topics in order to indicate to the students the way forward to tackle problems that are solved in this subject.
- Give the approach of "what is each concept studied for" that helps motivation and the introduction of the student in the real world, which will be very useful when the student has to do real practices in companies.
- Connection of the theory explained with real examples.

In the DEMETER project we try to mobilize student motivation. It is one of our main objectives. The motivation of the students in the classroom can be improved with a work scheme that includes the following elements:

- The prior interests of the students must be known.
- It is convenient to foster a constructive critical spirit about technological activity and the various commercial proposals that can be found in the market.
- It is positive and necessary to help students to make responsible decisions in the selection of technological products.
- Group activities are encouraged by carrying out team practices, projects in which several students can collaborate, etc.
- They should be helped to select and properly handle technical documentation and advertising information.

- In addition, an interdisciplinary approach can be taken from which the student can benefit, providing an integrated unit to the set of knowledge they possess, thus improving the interrelation between the different modules of the training cycle.
- Relationship with the other subjects of the cycle since part of the subject will be given in a network (Local Area Networks). And in part it will be programmed (Fundamentals of Programming).
- Relationship with Human Sciences through the design of databases for access to information, Books, Bibliographies, etc.
- Discuss current affairs and technologies in class (for example, robots, artificial intelligence, etc.).

To take into account: Before this unit, students have already worked on loops and conditional programming statements because they have previously done the **Learn2program** unit. This unit is in the IT module of the DEMETER project.



Methodology

This unit has tried to include the maximum number of skills / abilities possible so that the result is as enriching for the student.

In our methodology we focus not only on the final result but also on the process followed until reaching that result. It is important that the student learn an efficient discipline in solving complex problems. The result must be able to reproduce with other problems and other circumstances. Why do we put so much emphasis on the process and not on the result?:

- Because the student have to work cooperatively
- They have to develop their skills
- They have to improve their knowledge
- They have to improve their way of working

Once the teacher have the skills and the unit scheduled...:

- **The teacher will be the right person to make it happen**. He/she will have to use all his/her skills as a teacher so that the result will be as expected.
- **We need the right group of students to make it happen**. The grouping of students will be very important so that the class environment is good and the acquisition of effective knowledge.
- We will need **other ingredients** like motivation and patience.

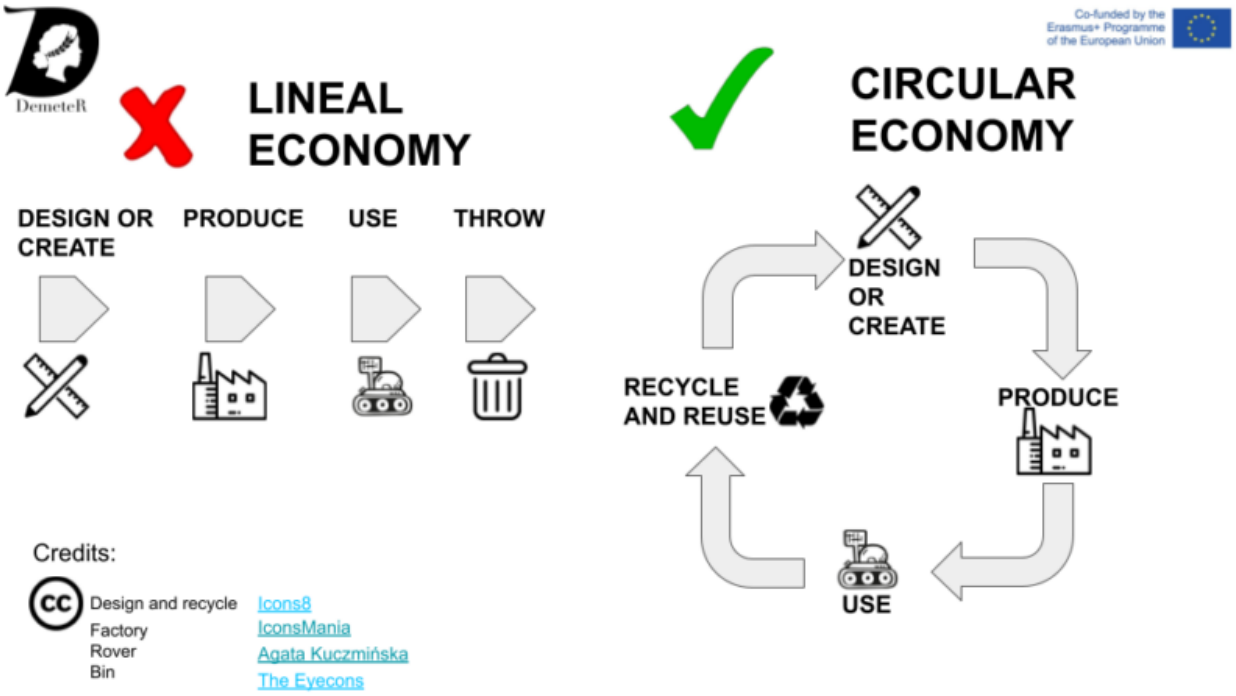
Integration of key competences

In this unit, teachers have tried at least to include the following key competencies:

- **Communicative competence**. Communication between students not only in their mother tongue but in a foreign language such as English is basic. In DEMETER, groups of students of different nationalities have been created to work together on parts of the project. In addition, all communication has always been preceded by a group presentation.

- **Mathematical, science and technology competence.** This DEMETER project is basically a technical project. In it, the use and development of scientific methodology and knowledge is necessary.
- **Digital competence**. DEMETER students use new technologies as the basis for communication, learning, analysis, production of results, etc.
- **Learn to learn**. Working through projects based on thematic learning allows students to improve their organization and collaborative skills.
- **Social and civic competences**. In the DEMETER project many ethical dilemmas have been debated and through many activities the student will understand and rethink concepts such as justice, human rights, solidarity, etc.
- **Initiative and entrepreneurial spirit**. The student has to be responsible for their own learning and has to make decisions regarding problem solving. The student will have to manage and plan in order to solve the problems that the DEMETER project is posing.

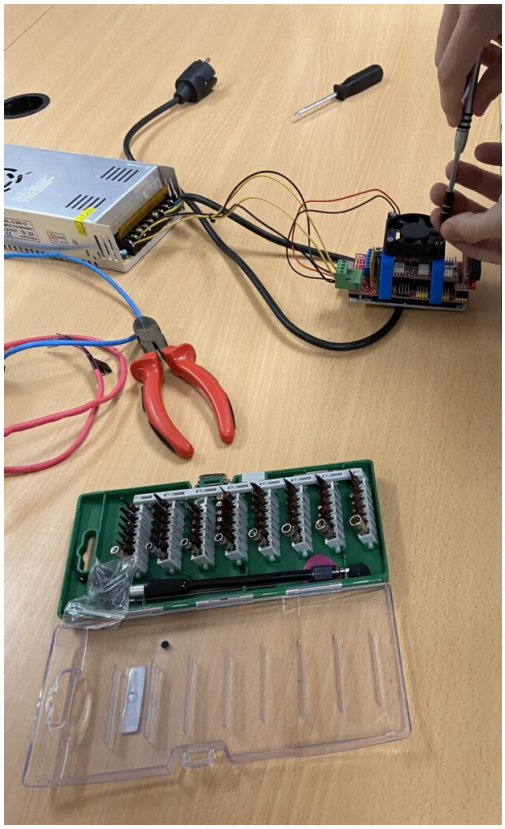
Circular economy. Why is it important in our project?



In a linear economy, devices are designed and mass produced and after use they are disposed of generating waste. Many electronic devices such as a smartphone can be made with more than five hundred different substances, which generates a lot of pollution. In the circular economy, those components that can be reused in other devices are reused. When a device ends its useful life, many of its parts can be useful and can have a second life. With this type of economy the effect of programmed obsolescence is neutralized.

The circular economy in our project is a concept that helps us not only to make our students aware that we must recycle and that products can have a second life, but it also allows us to better understand technology and interact with it.

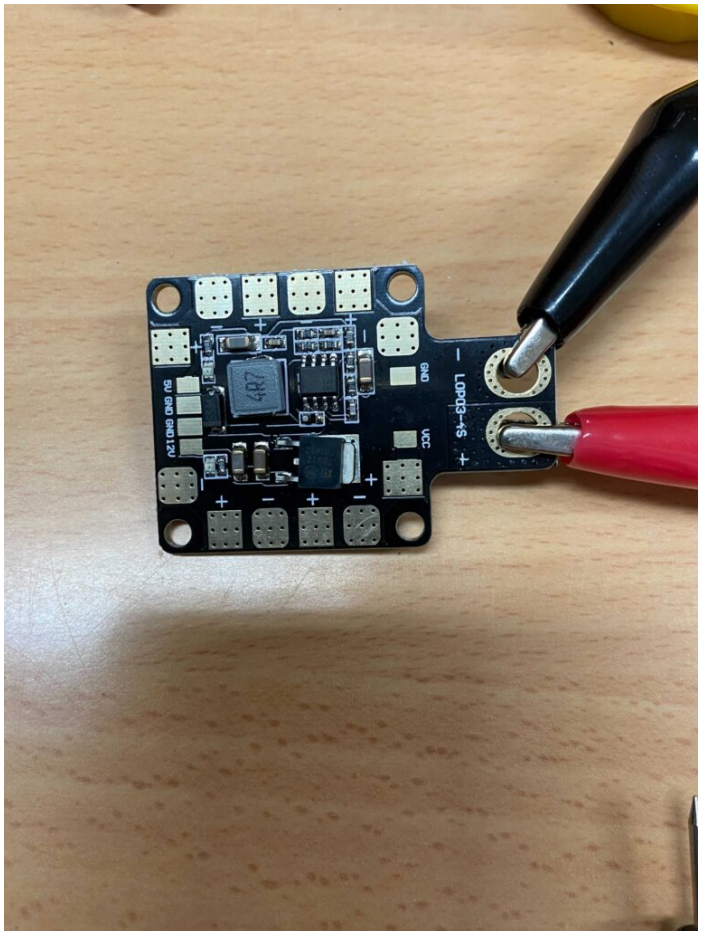
If all we do is buy new components and assemble or buy all the components already assembled, our students do not have a thorough understanding of how they work. For example, in our robotic arm we have reused parts of 3D printers that were no longer in operation due to wear and tear and lack of replacement parts. The stepper motors of a 3D printer can serve not only to move the robotic arm but to move the rover itself. In this way, the student selects the part, studies its operation in depth and can design and use it without having to buy new parts or use external resources that make test models much more expensive.



Picture: Using the electronic parts of a 3D printer to use them into our rover

Understanding the operation of technology allows the student to open his mind and think about the possibilities that certain devices have.

Following the principles of Immanuel Kant, our knowledge cannot exceed the limitations of the mind and the senses, therefore, we do not know how things are themselves, but we know how we experience them. Apart from the philosophical discussions, we take into account that the most important thing in the construction of knowledge is experimentation and it is for this fact that the concept of circular economy is so important.



An example of a circular economy established in one of the cities of the project coordinating team is garbage recycling. Garbage serves the city of Västerås to generate all the necessary energy in it. They do this at a negative cost. They not only earn money by burning garbage without generating waste and pollution, but they also generate energy and improve the quality of life in the city.

Preventing the emission of greenhouse gasses and protecting the environment is one of the objectives of the European Union and it is therefore that awareness at the academic level is so important. If this fact is also being used to generate knowledge and improve the employability of our students, it can be said that this is a good practice.

Teacher Reflections

This is a fairly long unit which may take an academic year to complete with students.

As can be seen, students learn concepts about electronics and Arduino from scratch, and once these concepts have been learned and worked on, they can begin to carry out more complex tasks such as programming devices and complex systems such as a robot.

In this unit the student will work on many skills such as 3D printing, programming, electronics, proxy configuration, etc.

Given the complexity of this unit, it must be said that it is not a unit to be worked on entirely with all students. Perhaps it is better to work this unit in its entirety with the most motivated and most advantaged students and work on a lesson or activity with the rest of the students.

Having worked on this unit with the students, the feedback I have received from them has been very positive. The knowledge acquired by the students in this unit will serve them for their professional future and I believe that they will greatly improve their employability.

Lesson 1. Introduction to electronics.

This lesson is a theoretical lesson in which students will learn the principles of electronics and electrical parameters. Below is the content of this lesson:

- Electronic components.
    - Passive components.
        - The resistance
        - The capacitor
        - The transformer
        - The fuse
    - Active components.

        - The cell or battery
        - The transistor
        - The diode
        - The integrated circuits
        - The microprocessor
- Electrical parameters.
    - The voltage
    - The intensity
    - The resistance
    - The power
    - Difference between direct current and alternating current.
    - Static energy.
    - The multimeter.

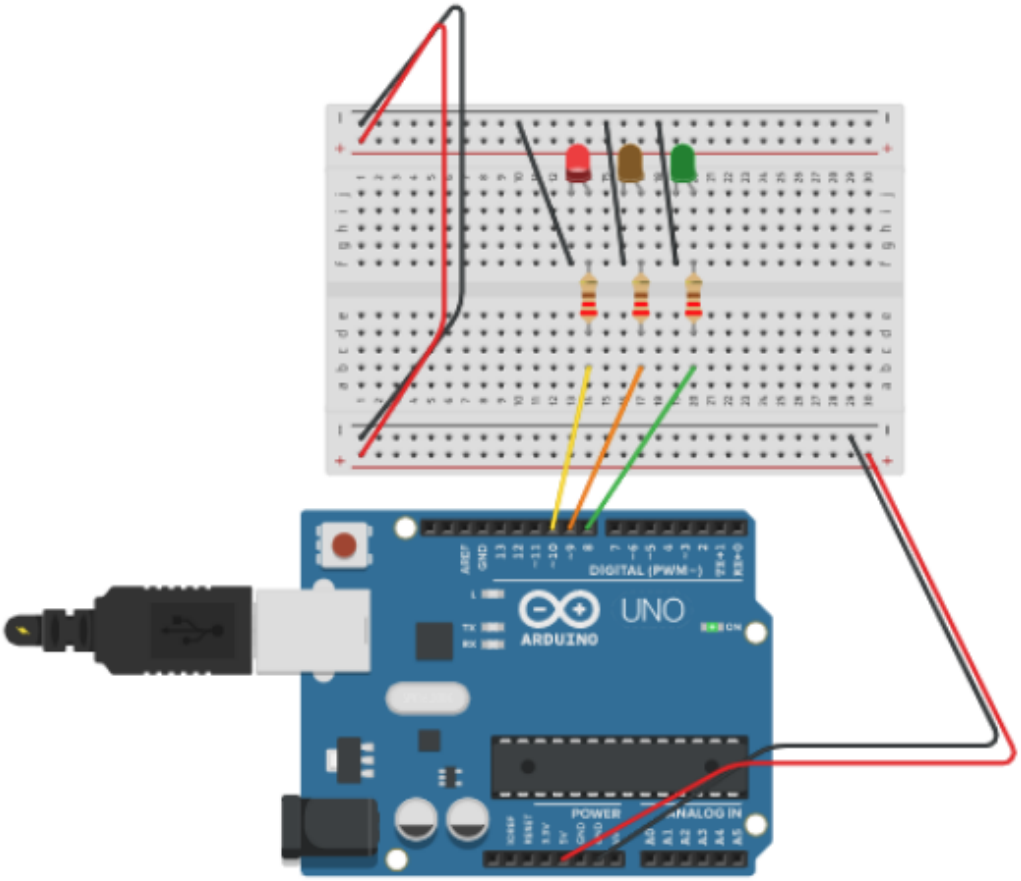
It is important for students to learn these concepts and to know how to efficiently handle tools such as the multimeter.

Lesson 2. Introduction to Arduino.

In this lesson, multiple concepts will be worked on, the first one will be to explain what free hardware consists of and also the reasons for working with Open Source projects.

The first thing we are going to study about Arduino boards is what is called a pinout. The set of inputs and outputs that the microcontroller has. The students are going to do a series of practices starting with the simplest one, which is flashing a LED.

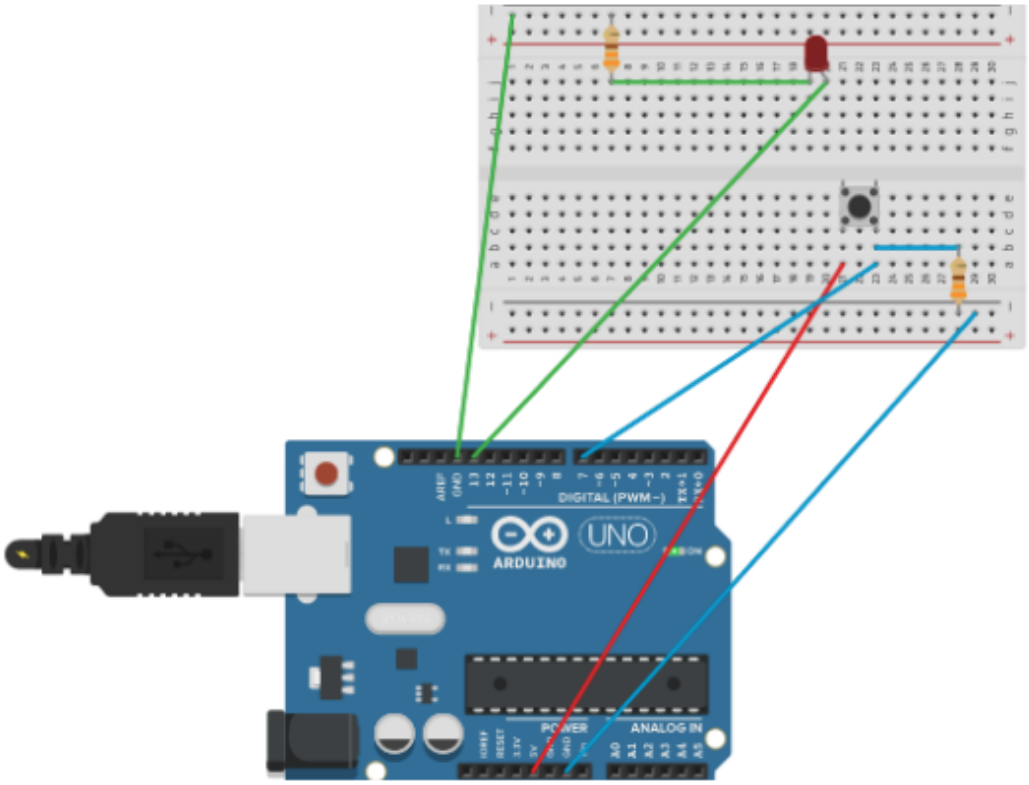
A second exercise will be the realization of a traffic light. In this exercise reading is also used but there is already an added complication which is programming. Various routines must be performed in order to display the traffic light colors properly.



The code for the previous traffic light is shown below:

```
void setup()
{
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
}

void loop()
{
  // ROJO 3S
  digitalWrite(10, HIGH);
  delay(3000); // Wait for 3000 millisecond(s)
  digitalWrite(10, LOW);

  // NARANJA PARPADEA 5 VECES
  digitalWrite(9, HIGH);
  delay(300); // Wait for 500 millisecond(s)
  digitalWrite(9, LOW);
  delay(300); // Wait for 500 millisecond(s)

  digitalWrite(9, HIGH);
  delay(300); // Wait for 500 millisecond(s)
  digitalWrite(9, LOW);
  delay(300); // Wait for 500 millisecond(s)

  digitalWrite(9, HIGH);
  delay(300); // Wait for 500 millisecond(s)
  digitalWrite(9, LOW);
  delay(300); // Wait for 500 millisecond(s)

  digitalWrite(9, HIGH);
  delay(300); // Wait for 500 millisecond(s)
  digitalWrite(9, LOW);
  delay(300); // Wait for 500 millisecond(s)

  digitalWrite(9, HIGH);
  delay(300); // Wait for 500 millisecond(s)
  digitalWrite(9, LOW);
  delay(300); // Wait for 500 millisecond(s)


  // VERDE
  digitalWrite(8, HIGH);
  delay(3000); // Wait for 3000 millisecond(s)
  digitalWrite(8, LOW);



}
```

In a third exercise, a button with an LED will be programmed. The exercise consists in controlling the lighting of a LED when the button is pressed. The hardware to be used is simple since you will only need a LED, a button and a pair of resistors.



The code from the previous exercise is shown below:

```
void setup(){
 pinMode(13,OUTPUT);
 pinMode(7,INPUT);
}
void loop() {
  int val= digitalRead(7);

  if (val == HIGH){
    digitalWrite(13, HIGH);
    delay(100);

  }else{
    digitalWrite(13,LOW);
    delay(100);
  }


}
```

To learn Arduino, students have several options, one of them very important is to make real assemblies and the second is to have a simulator to be able to practice with an infinite number of Arduino elements and to learn better programming. Without simulators, learning is much slower because you have to do a real assembly which takes time and has a cost.

Simulators like Tinkercad have zero cost, so using a simulator of this type will help us learn much faster.

Lesson 3. Learning electronic welding.

One of the first steps students need to take is learning the fundamentals of component welding.

To understand how electronic components can be welded, a motivating exercise has been designed which is the creation of a 3D cube.

Below is a list of the components that we are going to use:

- 27 LEDs
- 3 NPN transistors (2N3904 or also 2N2222). In our case we have used the 2N2222.
- 3 10 kΩ
- resistors 9 220 Ω resistors
- 1 Breadboard
- 1 Arduino UNO
- Connection cables

This cube is made up of 27 LEDs which must be soldered and connected together.

To make the cube, you must use the knowledge acquired in previous lessons, such as resistors, LEDs, transistors, cables, etc.

The advantage of this exercise is that the student has to know each and every one of the components and also has to make multiple welds with a minimum of resistance to withstand the practice. The student will have to do more than 50 soldering with LEDs, with which once the cube is finished, the student will be prepared to carry out any type of soldering of an electronic component.

The advantage of this exercise is that you not only learn soldering but you also have to do programming in Arduino to put a lighting pattern to light the LEDs.

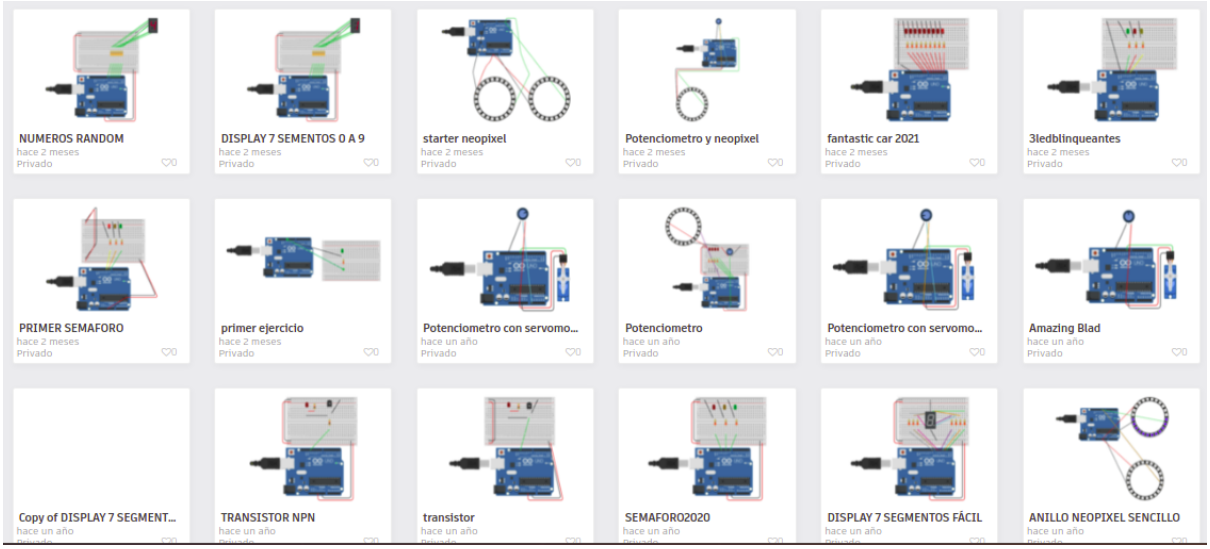
It is a complete exercise in which not only knowledge about electronic components, soldering, connectivity and arduino programming are covered.

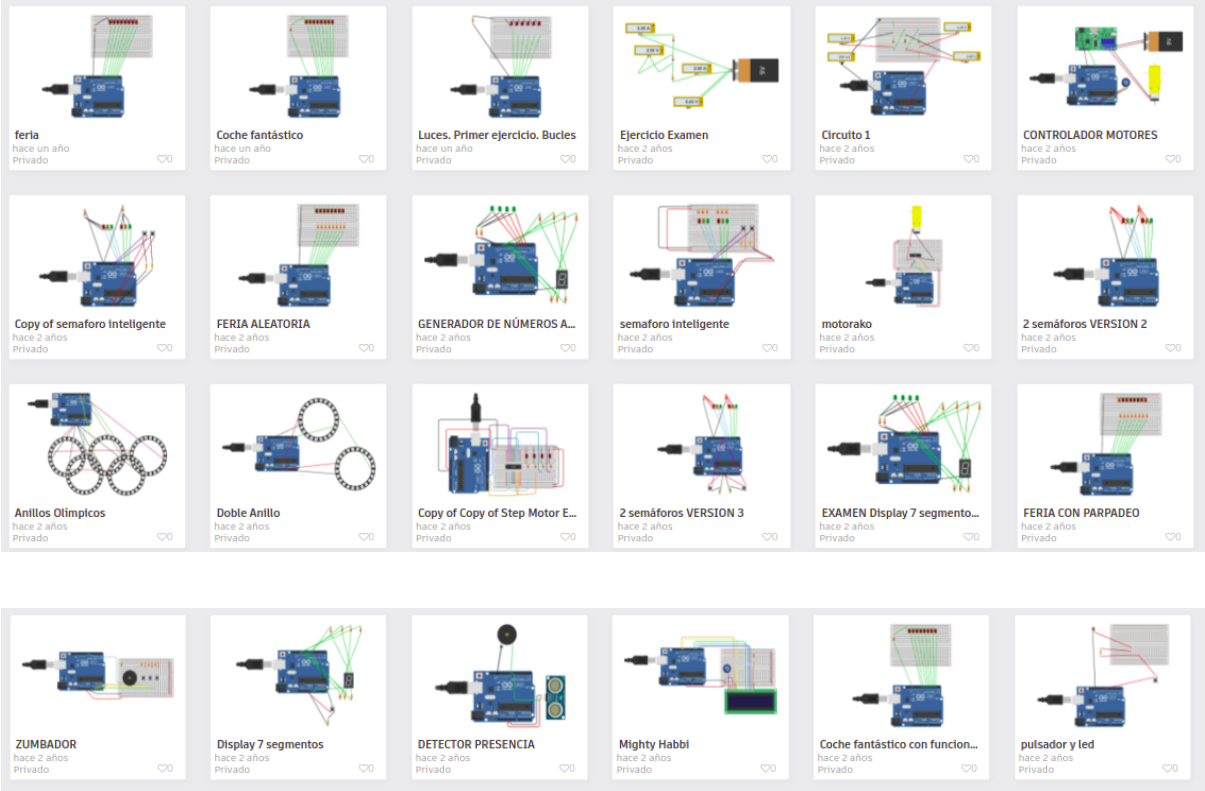
Lesson 4. Programmable electronics. Arduino.

In this lesson we will try to work the maximum number of electronic components with the simulator and we will also introduce students to Arduino programming using loops, functions, etc.

Students have already worked on loops and conditional programming statements because they have previously done the **Learn2program** unit.

Below are some of the simulated assemblies that we have been working on during this lesson and some will be discussed in depth:





Description of a specific example: Control of a servomotor with a potentiometer.

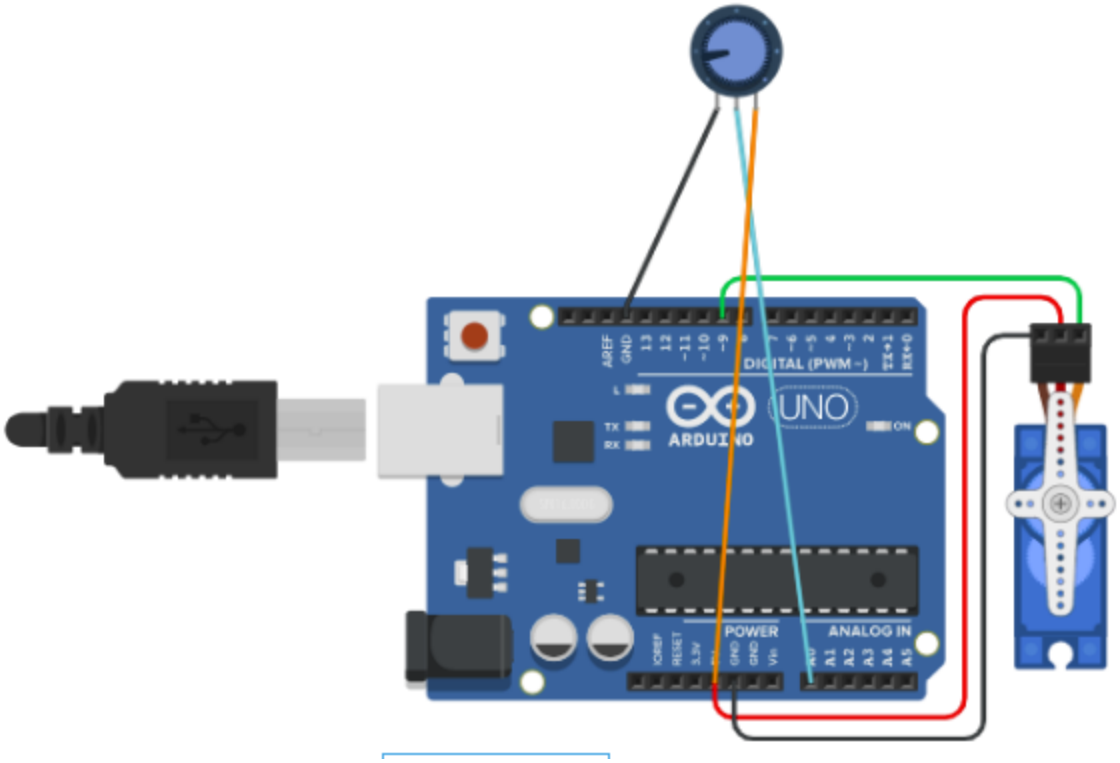
This is one of the exercises that is oriented to our project since we use servo motors which are very similar or have some similarity with the stepper motors that we have used for the robotic arm and the wheel drive motors.

The objective of this exercise is for the student to be able to move the servomotor with specific movements. To carry out this practice, the first thing the student has to know is that it is a potentiometer and also that it is a servomotor before mounting and programming.

- A potentiometer is a type of resistor whose resistance varies when you turn a knob.

- A microservomotor is a motor whose position can be controlled by a microcontroller. In our case we are going to use an Arduino UNO to control the servo motor.

Next the students will have to make an assembly like the one shown below in the image:



Later, this assembly will have to be programmed in such a way that the student will have to connect the servomotor to a port that will be controlled by the code.

The values of the servomotor are going to be received in an analog way in a port of Arduino.

The arduino will receive a value between zero and 1023 which must be coded to degrees instead of analog values, which will require a conversion or analyze this value to see if it is in a specific range.

The next step will be to send the instructions to the servo motor to establish the position in which you want to rotate it.

Below is the code that the students have generated:

```
#include <Servo.h> // Incluimos la biblioteca Servo

Servo Myservo;
// Definimos los servos que vamos a utilizar

//int Mypotenciometro = 0;
// Pin usado para conectar el potenciómeto

int valor;
// Esta variable definirá la posición del servo

void setup() {

  Myservo.attach(9);
  // Definimos el pines de señal para el servo


}
void loop() {
  valor = analogRead(A0);
  // leemos el valor del potenciometro (valor entre 0 y 1023)

  if (valor <= 330){
    Myservo.write(0);
  }
  if (valor > 300 && valor < 700){
    Myservo.write(90);
  }
  if (valor>700){
    Myservo.write(180);
  }
  /*
  Si el valor del potenciometro es 0 - 330 se queda a 0
  Si es 330 700 gira 90 grados
  Si es mayor 700 gira 180 grados
  */



  delay(10);
  // Esperamos para reiniciar el bucle
}
```

Description of a concrete example: Control of an LED with an NPN transistor.

A transistor is an electronic element used in many devices. For example, a processor has millions of them.

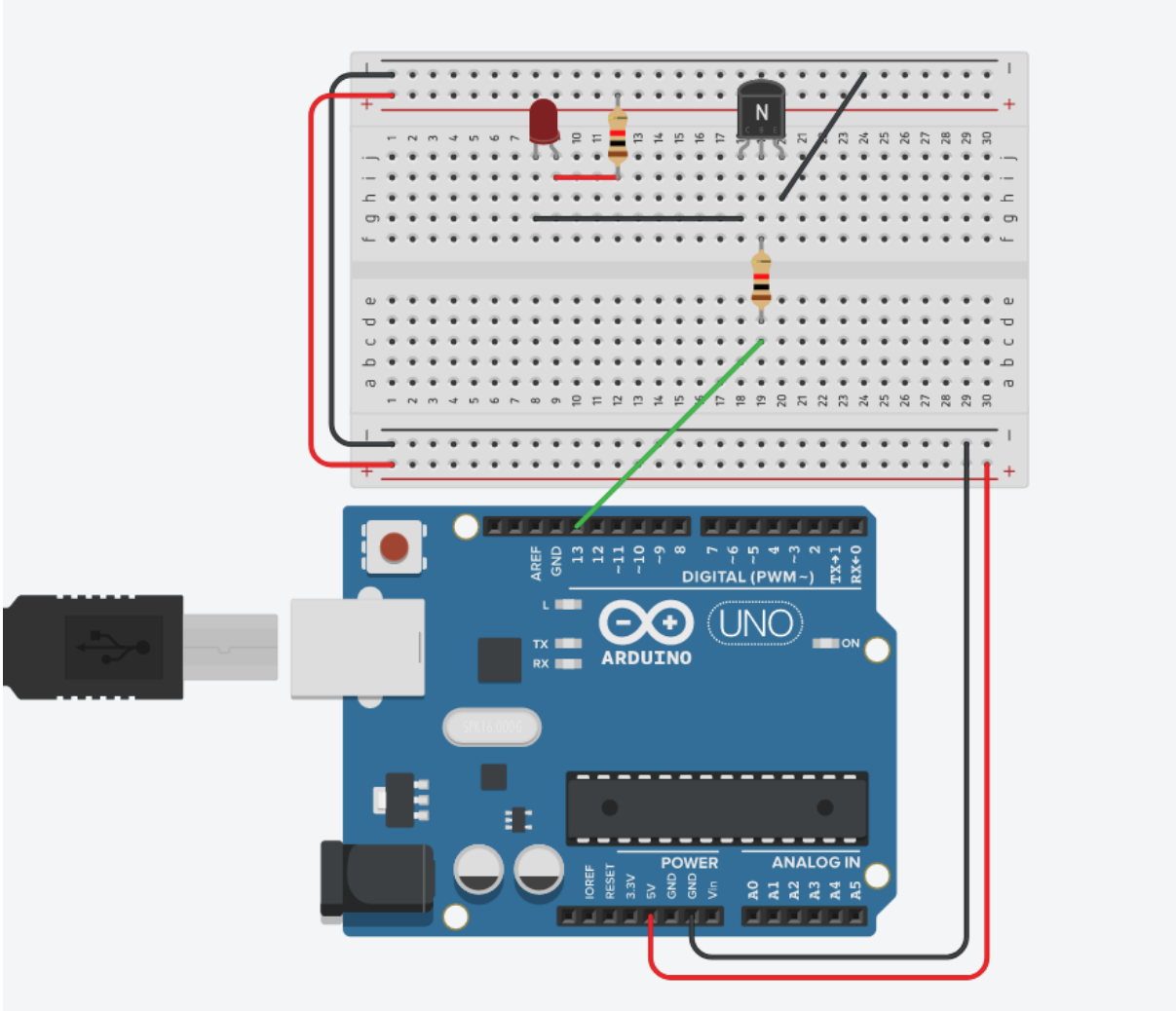
The operation of a transistor is simple, it is used to display an output signal in response to an input signal.

Some of its functions are that of an amplifier, oscillator, rectifier or even as a switch. We in our project are going to use NPN transistors that means Negative, Positive, Negative.

These transistors are widely used in electronics and especially in Arduino projects.

The project that is presented to the students is the following. Students are asked to control a LED by using an NPN transistor.

The assembly of the transistor and the rest of components is shown below:



The code used for this assembly is as follows:

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

As you can see, the base pin is connected to port number 13 of the Arduino and this port is the one that controls the flow of current between the emitter and the collector of the transistor.

With this simple example, students can learn the operation of a transistor in a simple and practical way.

Description of a concrete example: Control of a direct current motor.

DC motors are used profusely with Arduino because they are cheap and quite simple to use. To run a motor, all you have to do is apply voltage to it.

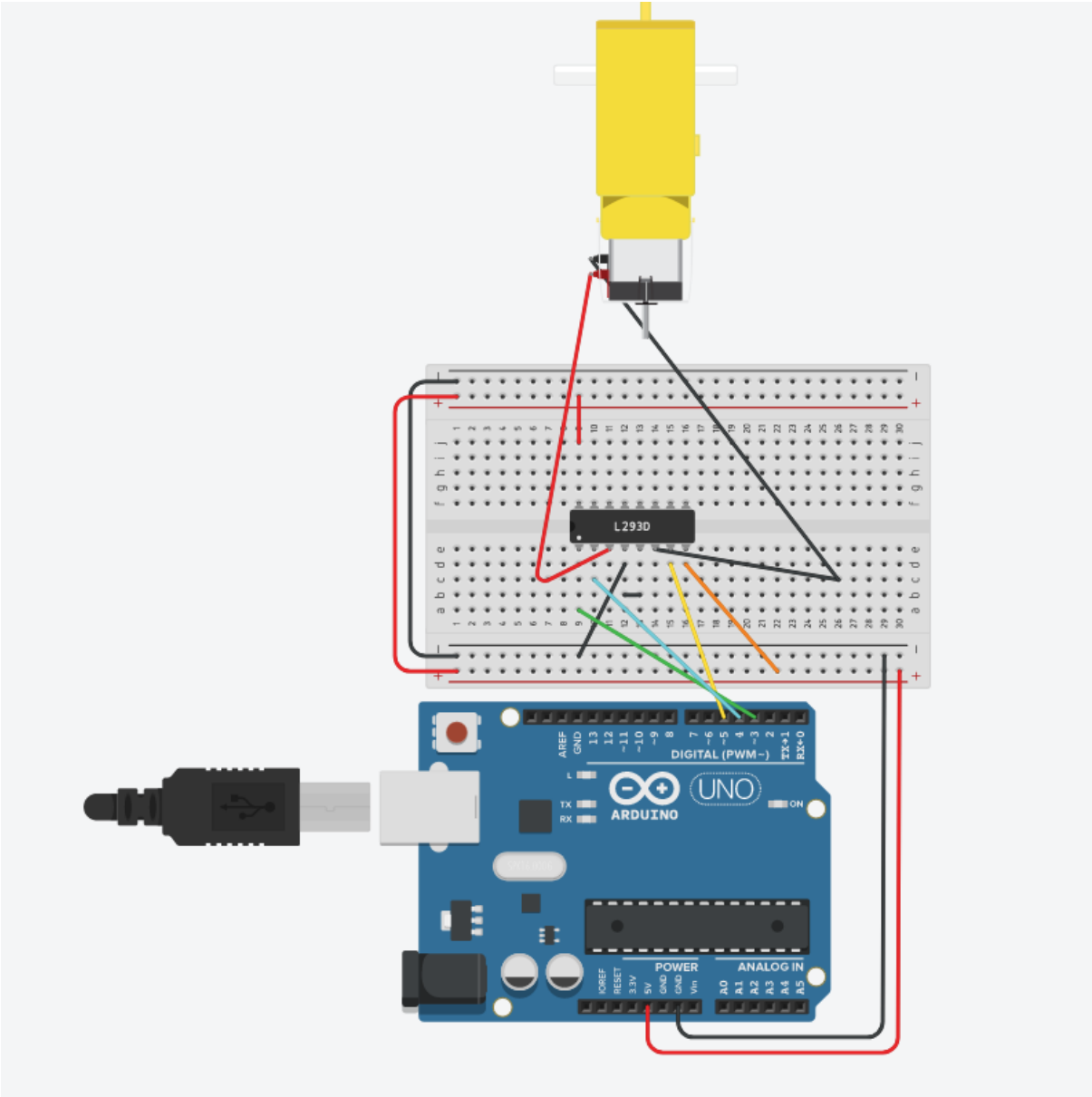
It is usual to use gears or another type of system to move wheels or another type of system.

These motors are usually controlled with a specific driver. In our case, we are going to use the L293D driver. This Driver will allow us to control both the direction and the starting and stopping of the motor.

The only problem that this driver has, which is quite inexpensive, is that it cannot control motors with a lot of power since it has an amperage limitation.

In the case that we need to control a more powerful motor, we will simply have to use a more suitable driver for these needs.

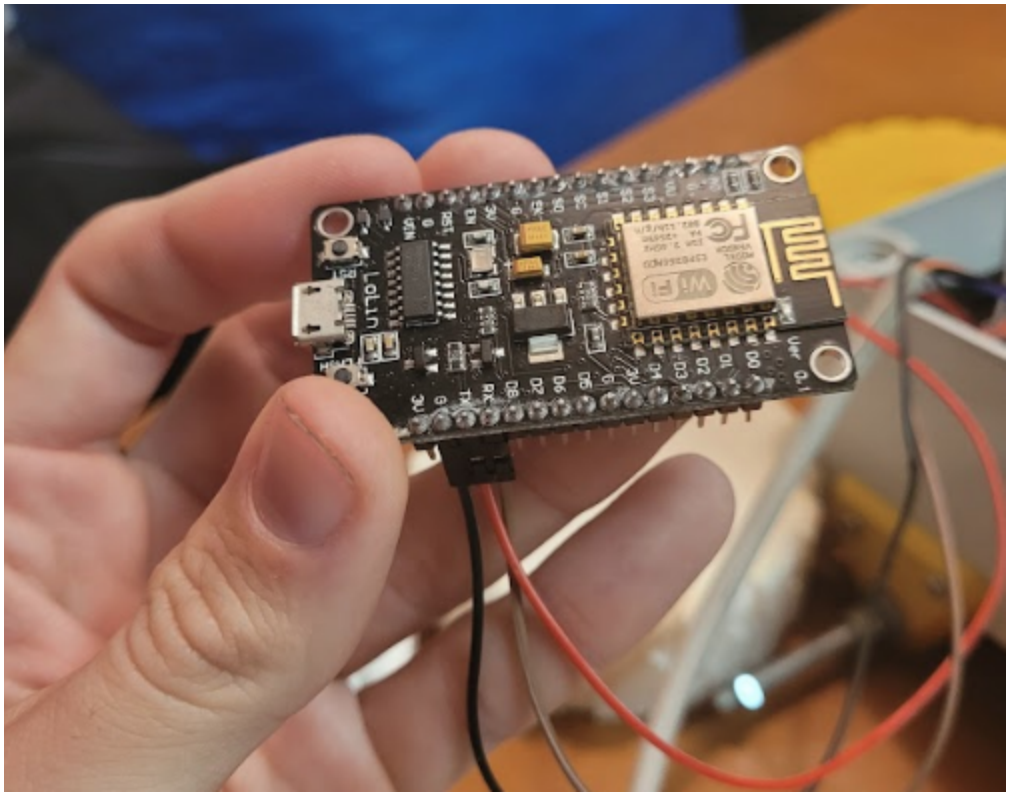
Below we show the assembly of our exercise:



The code associated with this assembly is presented below:

```
#define ENABLE 3
#define DIRB 4
#define DIRA 5

void setup() {
  //int i = 0;
  pinMode(ENABLE,OUTPUT);
  pinMode(DIRA,OUTPUT);
  pinMode(DIRB,OUTPUT);

  digitalWrite(ENABLE,HIGH);

  for (int i=0;i<5;i++) {
    digitalWrite(DIRA,HIGH);
    digitalWrite(DIRB,LOW);
    delay(1000);

    digitalWrite(DIRA,LOW);
    digitalWrite(DIRB,HIGH);
    delay(1000);

  }
  digitalWrite(ENABLE,LOW);

}
void loop(){

}
```

Lesson 5. Programming in NodeMCU

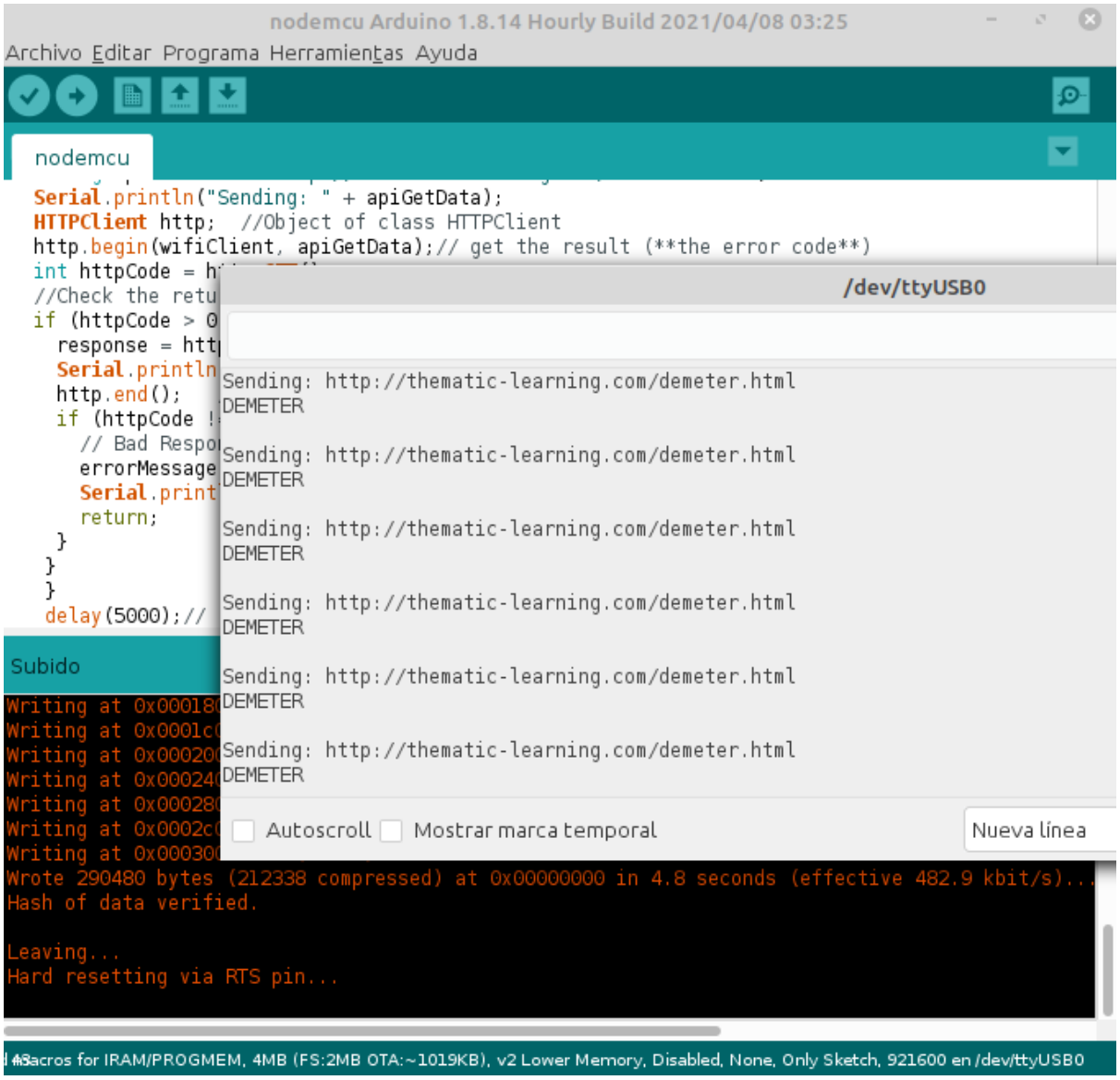


Two exercises have been established for programming with this device. In the first basic one, all that students had to do was read a file located on our web server. The file contained the name of our project.

The objective of the exercise was to read the content of the file and display it through the serial port.

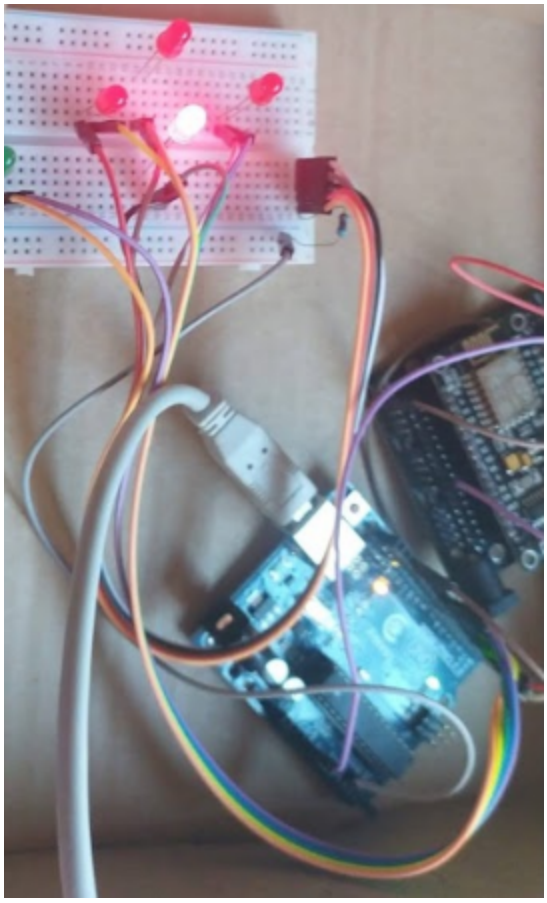
If the student managed to perform the exercise successfully, the exercise was considered correct.

The code and execution of one of our students is shown below:



Later, we included a more complex exercise. The exercise consisted of reading the content of a file located on our website and depending on its content, lighting a series of LEDs. By means of another web page the content of the file could be modified and in this way the Led's could shine according to the information that was encoded by the web page.

The objective was for the LEDs to indicate the movement forwards, backwards, left, right, move the arm: the GCode commands that the web file had (LEFT, RIGHT, FORWARD, REAR and ARM).



Lesson 6. Stepper motor programming.

For the movement of the arm, specific motors called Nema 17 have been chosen. These motors have a characteristic: their dimension is 1.7 × 1.7 inches. These motors are the most used in 3D printers. More used than any other stepper motor.

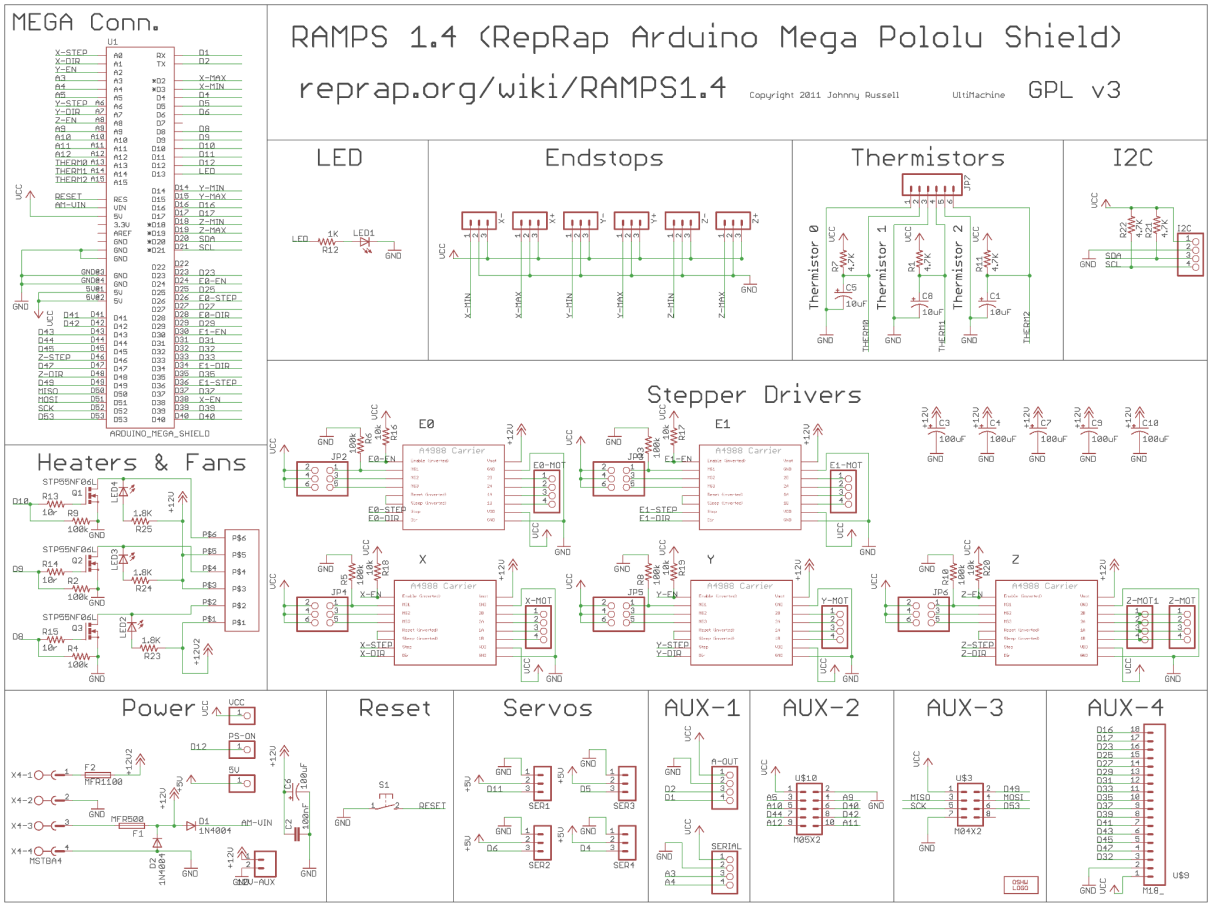
These motors are chosen not only because they are inexpensive but are also ideal for controlling precision movements such as that of a mechanical arm or a robot.

These motors can rotate their axis one 1,8° for each step, which would take 200 steps to make a complete revolution. The internal coils work with a current of 2 A at 4 V.

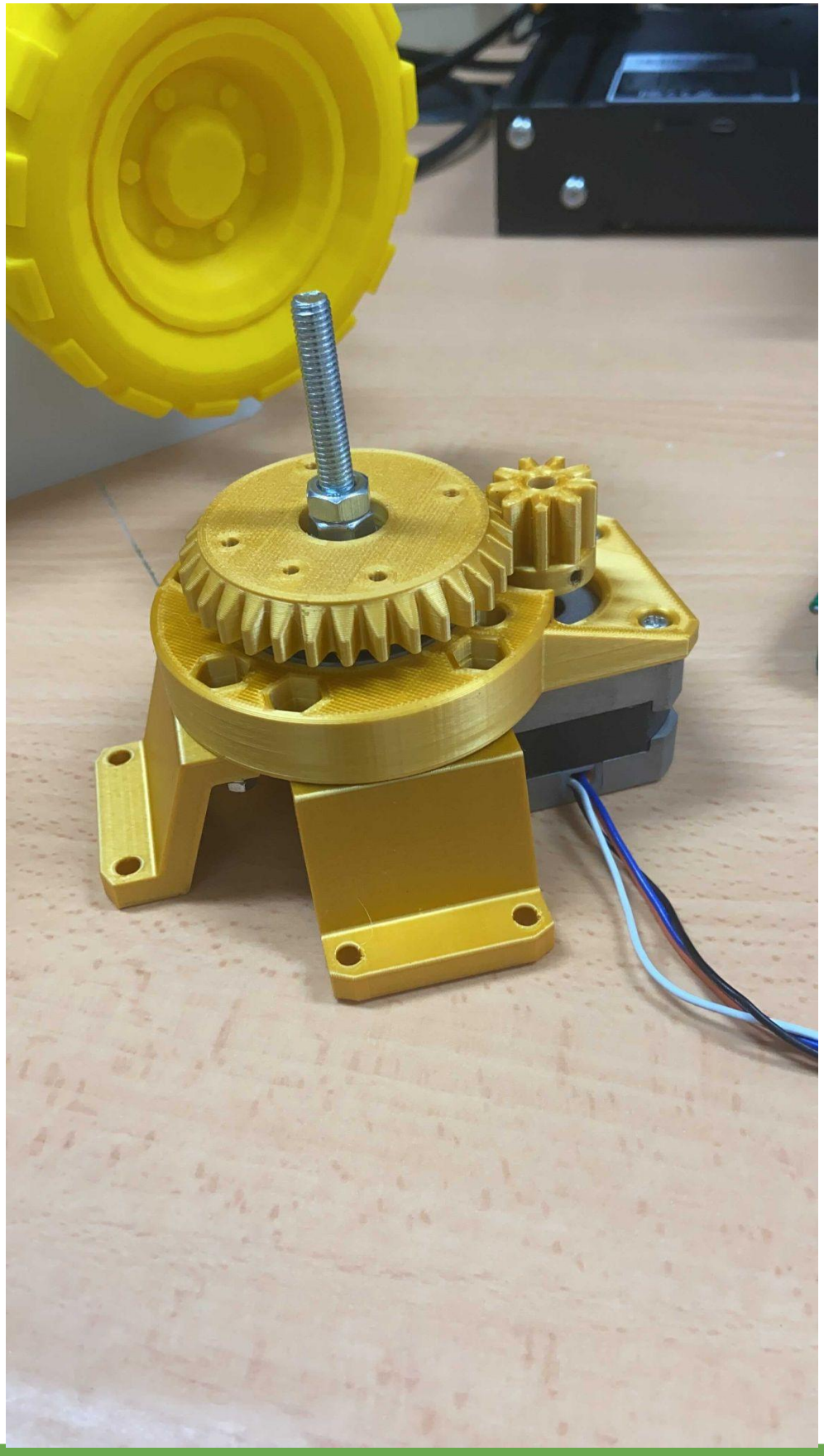
One of the characteristics of this kind of motors is its robustness, reliability and power.

To control these motors, certain drivers are needed. In our case we have used the A4988 drivers which can be integrated into a shield type plate called Ramps version 1.4.

Below is a diagram of the connectivity of this board:



As can be seen in the following image, a nema 17 motor is responsible for the rotating movement of our rover's robotic arm:



One of the practices we have done with the students has been to check if we are able to control a stepper motor with Arduino and the corresponding Drivers.

The objective of this practice was to simulate the sweeping movement of the metal detector that we are going to install in the rover.

Below is the link to a video of the result of the proposed exercise. In this exercise, we have used a nema 17 motor, an A4988 driver to control the motor and a computer power source.

https://youtu.be/6XwvgZ8NhxU

Once we have verified that we are able to successfully move a stepper motor and perform the sweeping movement necessary to move the metal detector sensor. The next step will be to integrate these achievements into our complete system: the rover.

Lesson 7. Creation of the license plate. Graphic design.

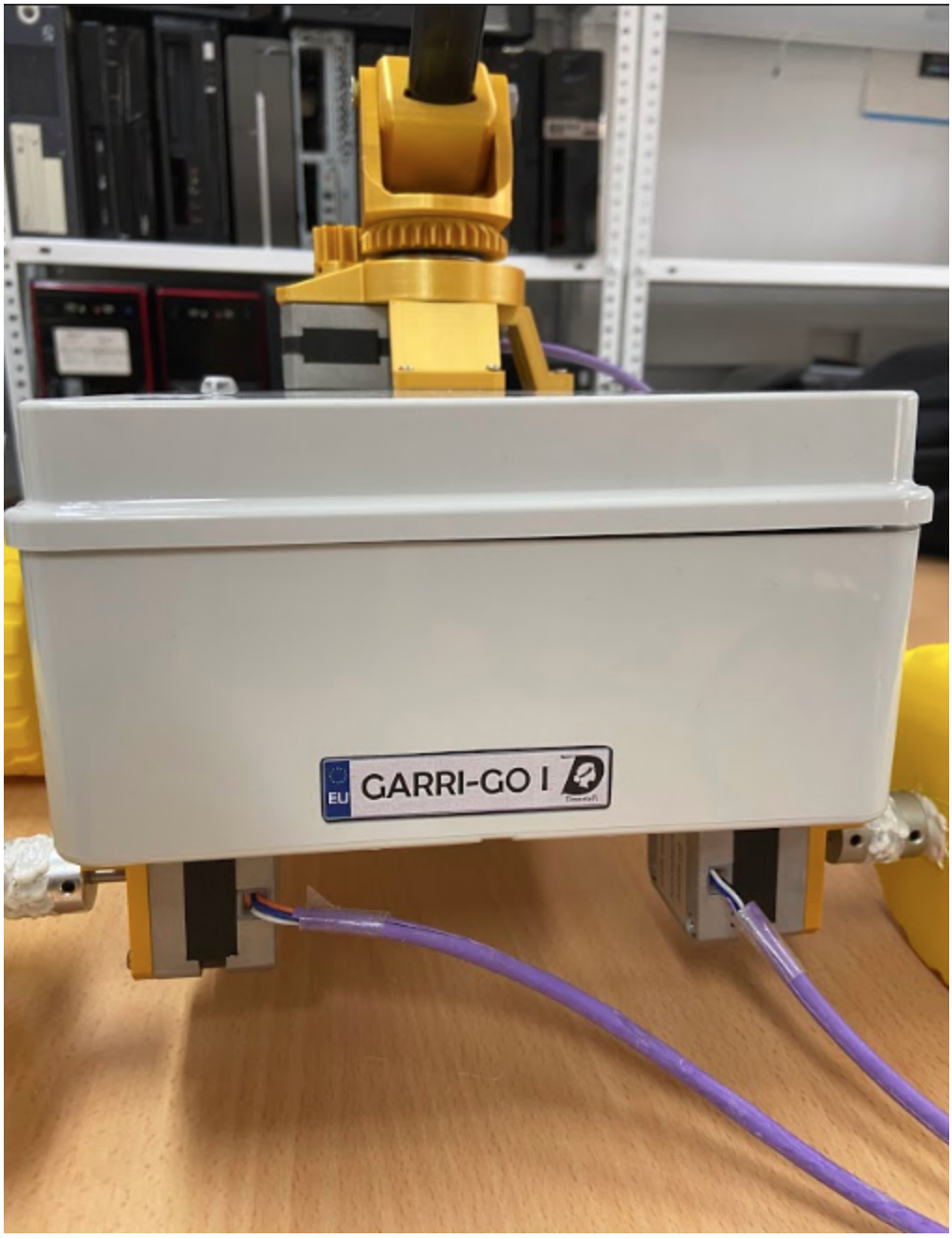
One of the activities that we want to promote in our students is creativity. With the knowledge acquired by our students in the subject of office applications, we proposed that they make a plate for our rover prototype.

There were several designs but the one that we liked the most is the one made by first-year SMR student Jaime Alamillo.

This registration was made to send a message of encouragement to a member of the project who suffered an accident at that time: Marc Garrigou.

Below is a detail of the student's work:

Once the design phase was finished, the design was printed and placed as the plate of our rover:



Lesson 9. Creating a 3D logo

One of the activities we do profusely in the project is three-dimensional printing.

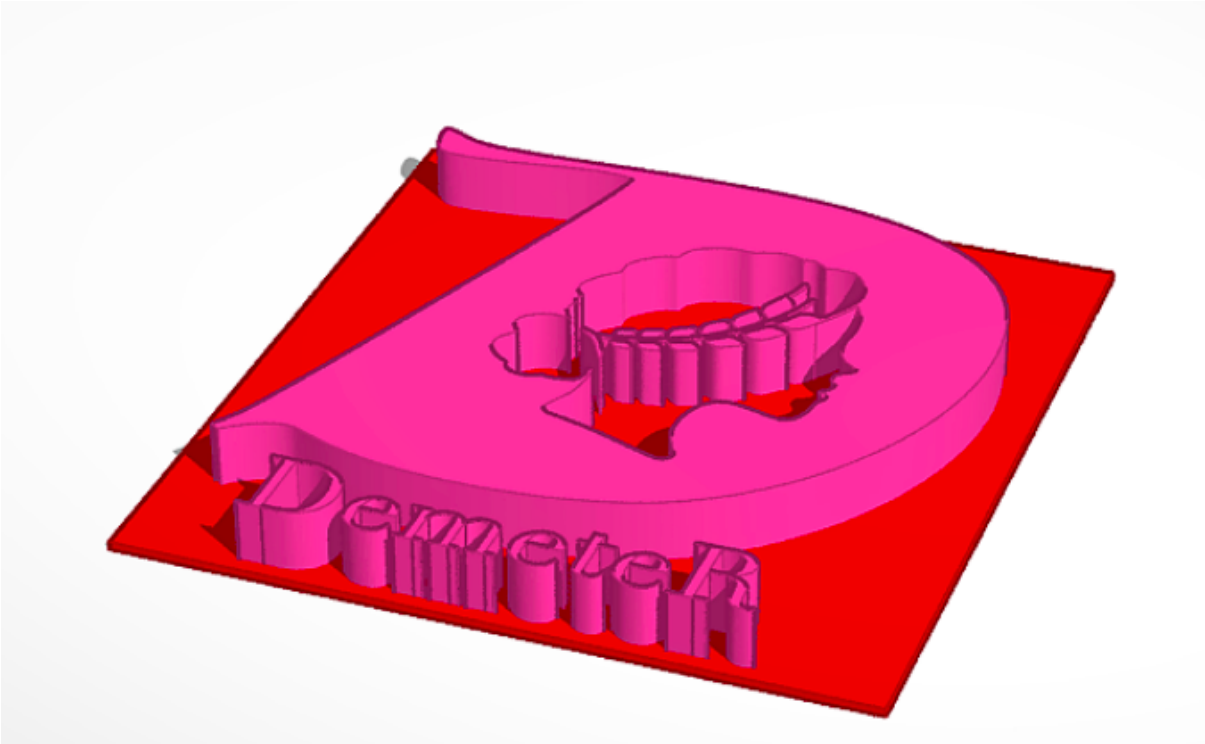
For this exercise, what we have done has been to translate a vector type image to an STL design.

Vector images are images made from geometric objects and therefore can be modified using mathematical algorithms. Therefore vector drawings are images created using mathematical formulas. These images are easily transformed into three-dimensional STL files.

The STL format is the most widely used in the industry to create prototypes. This format uses triangulation to form its models.

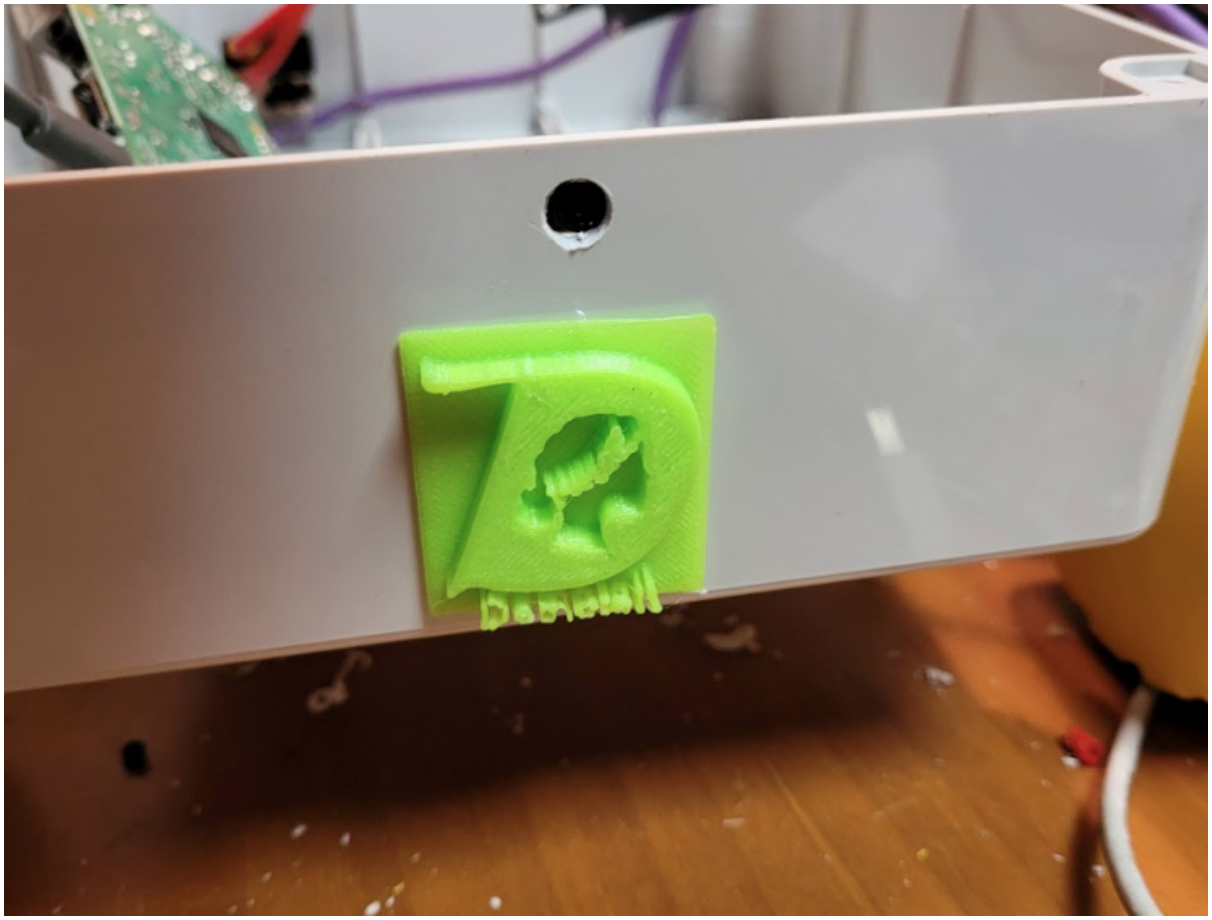
Once we have our image file converted to STL, it is very easy to translate it into a format that the 3D printer understands. These already translated files are called GCode.

The following image shows the result of translating an image file to an STL file:

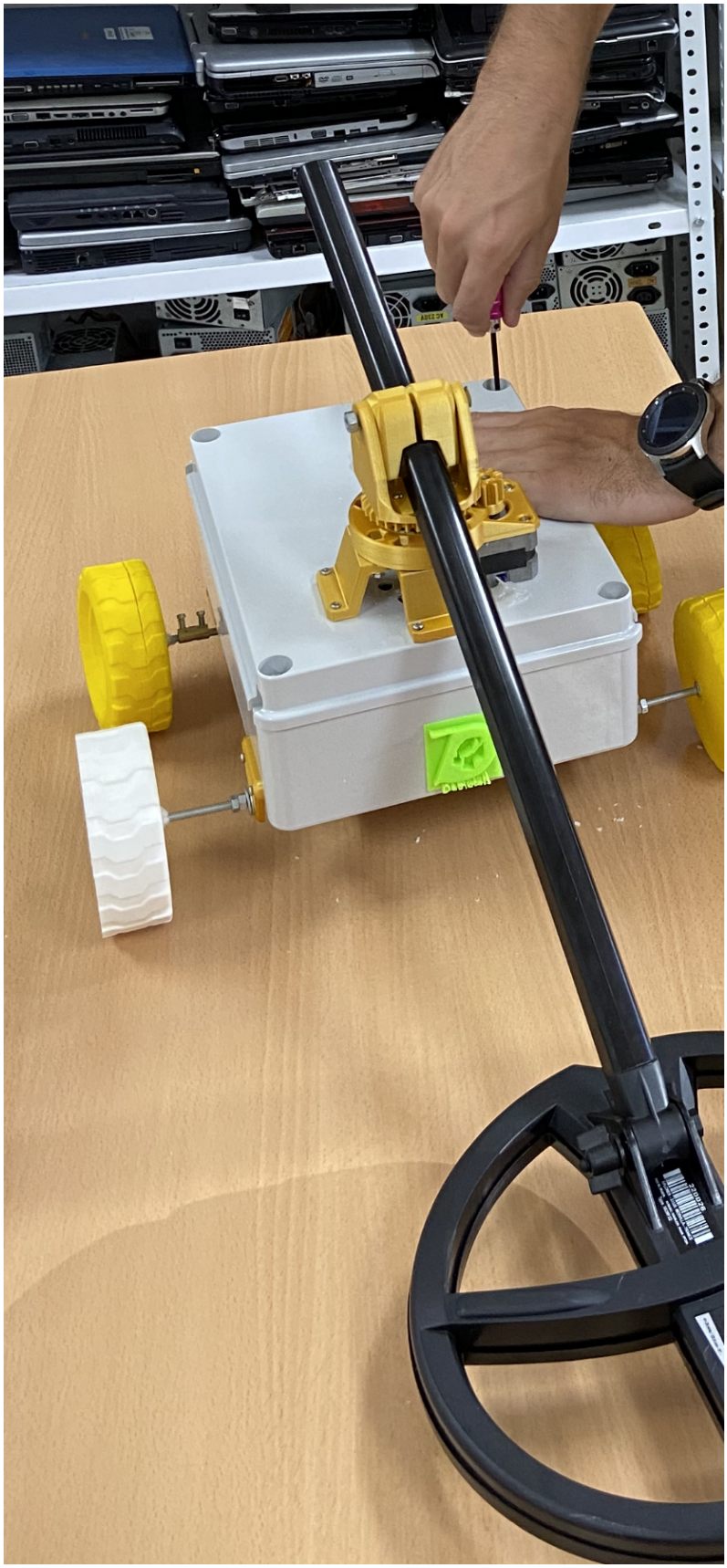


To generate the GCode file we will use either a proprietary program or the Ultimaker cure (open source program).

Once the GCode file has been generated, and it is understandable by the 3D printer, the next step will be to print with the thermoplastic material. The material chosen for this print is PLA. This thermoplastic material is easy to print and has many advantages as it is biodegradable. One of its disadvantages is that it is not suitable for parts that suffer a lot of friction or pressure, which is not our case.



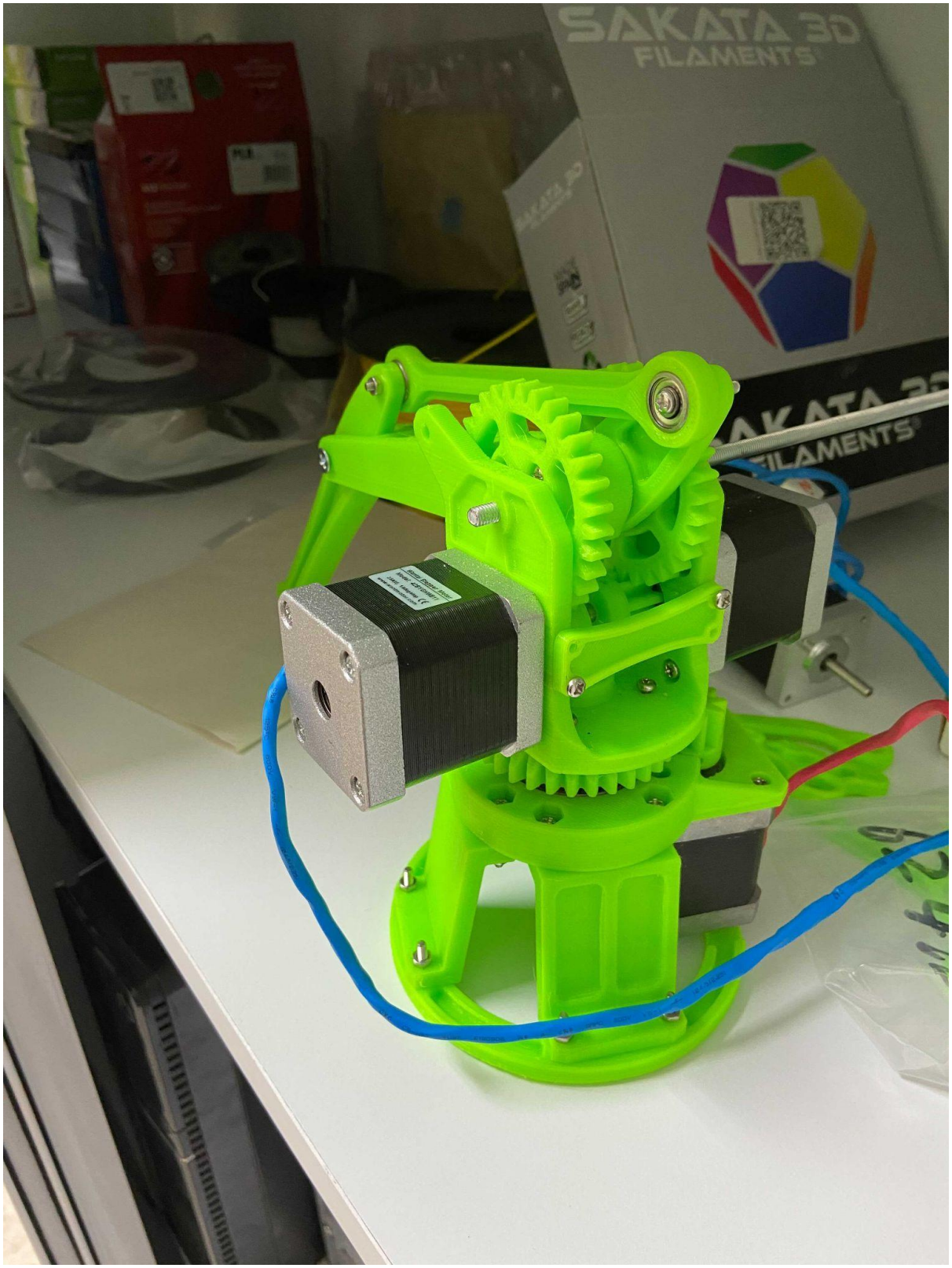
Below is an image of how the logo is printed after all this transformation process:



Lesson 10. Adaptation of the robotic arm.

As specified in another unit, one of the steps we have done to build our rover is to create a robotic arm.

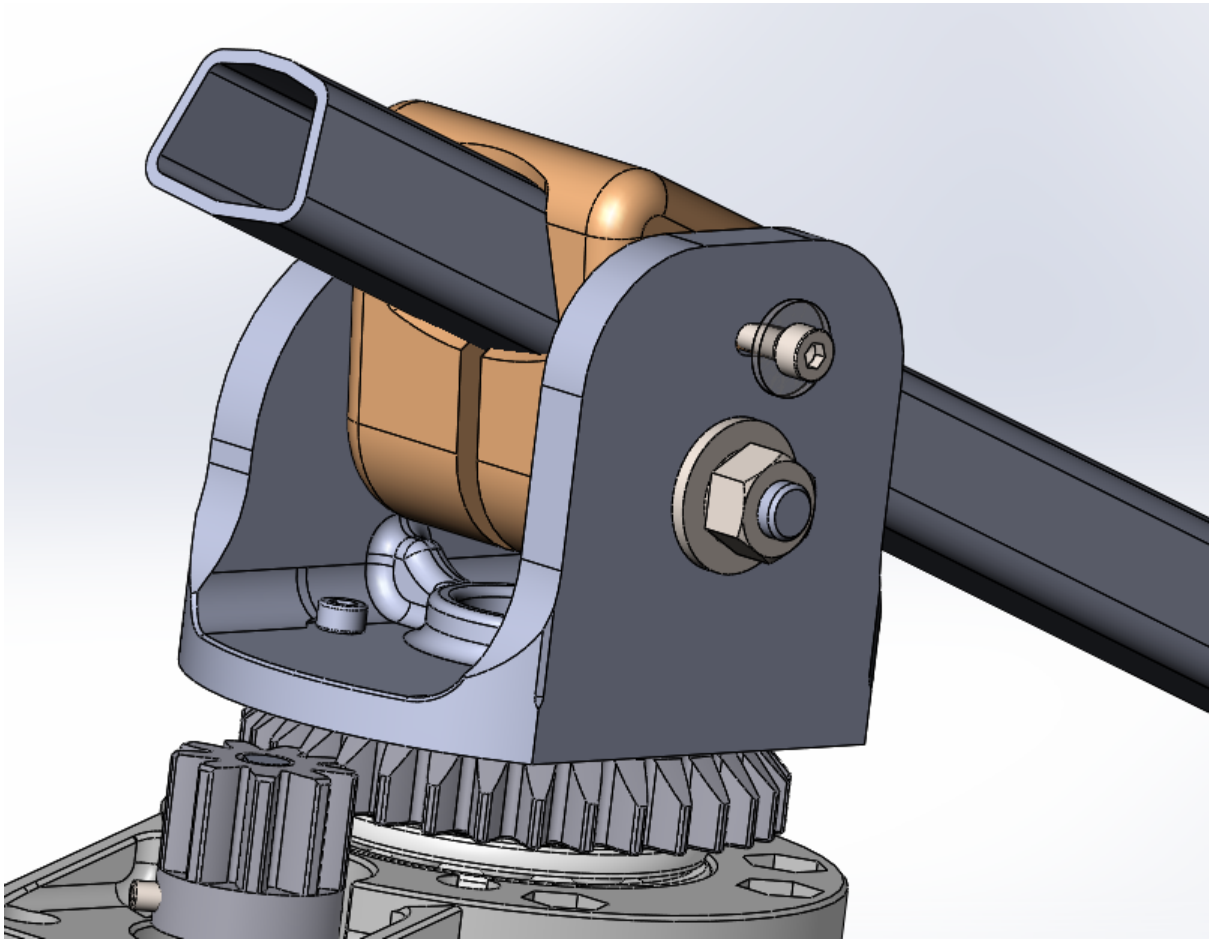
This arm was built with the aim of being able to adapt and transform it for our system.

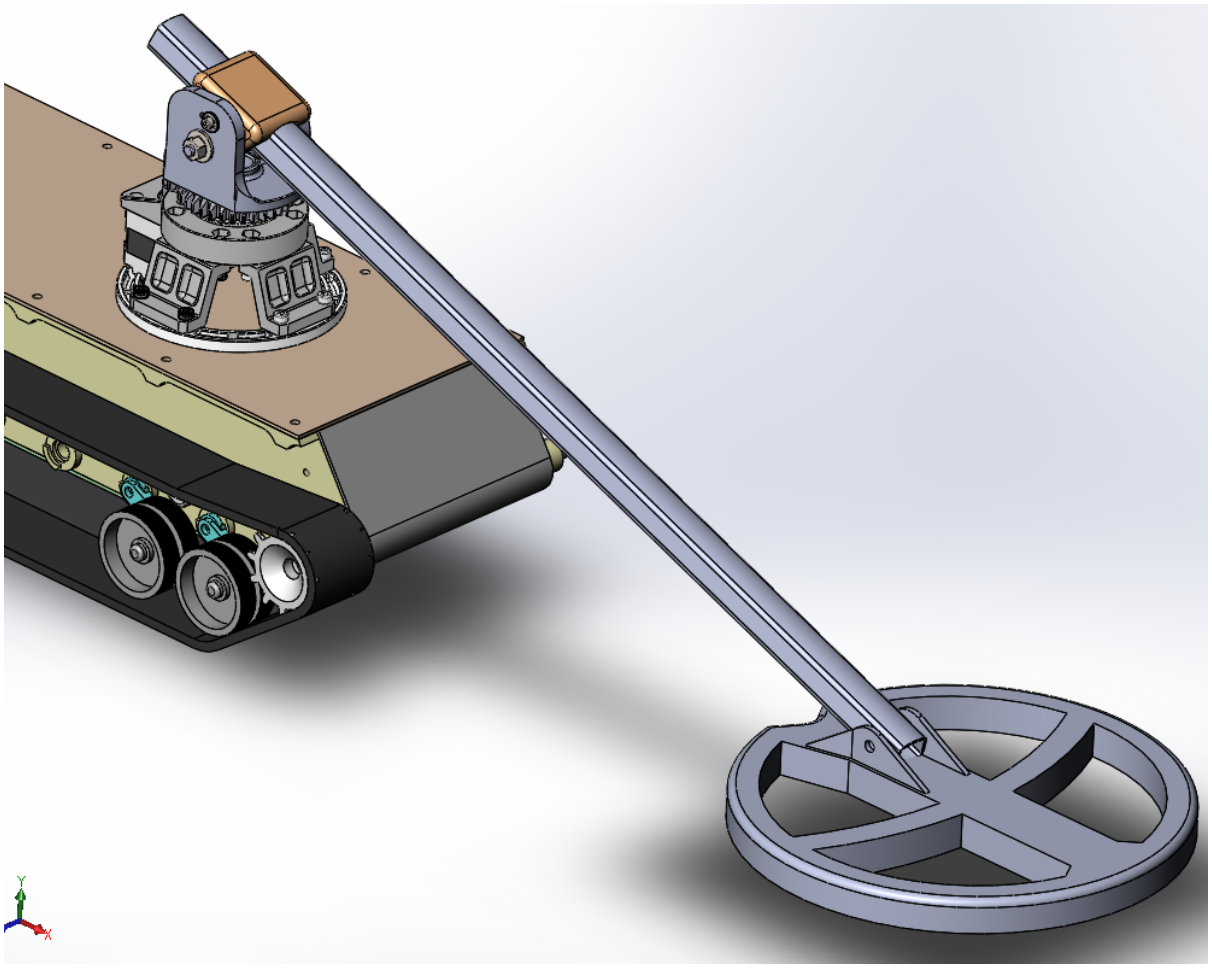


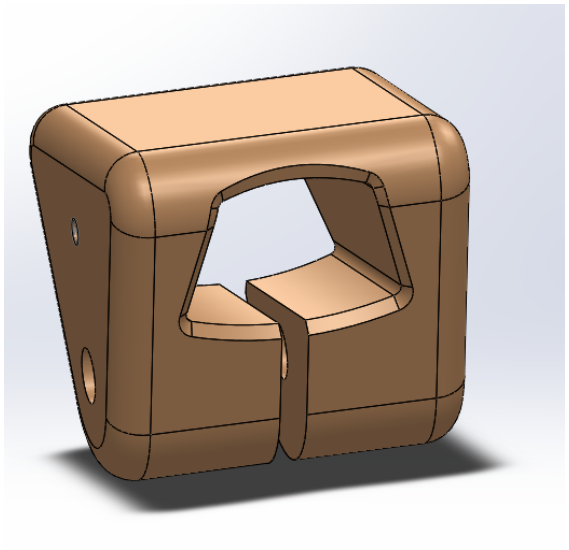
The figure above shows the look of the robotic arm implemented by our students. For the arm that is going to move the sensor we will only need one of the three stepper motors.

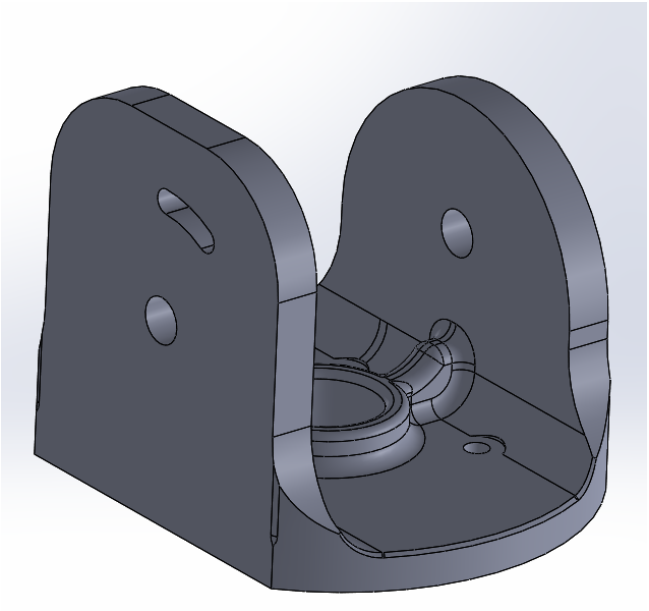
Therefore the Spanish team and the French team work together in a solution to implement in our rover.

The French team that is an expert in industrial parts design, designed the fastening system shown in the following images:



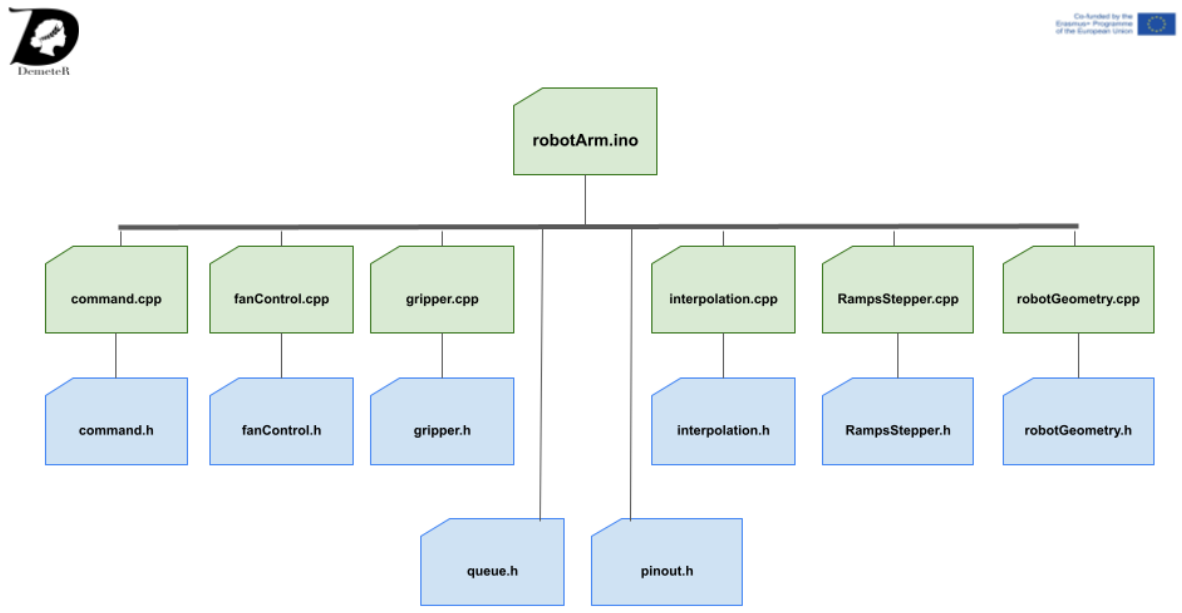




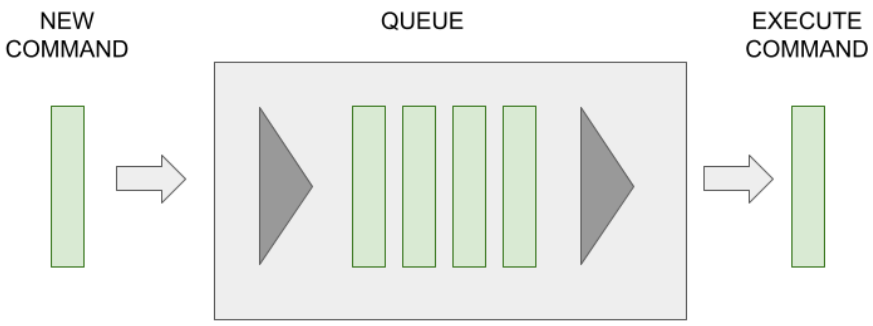


Later, subsequently, what we did was a thorough analysis of the code of the robotic arm in order to understand it in depth and be able to recode those parts for which they could make our arm do the sweeping motion.

Below is one of the diagrams that we generated in the analysis of the robotic arm code. The following image shows a tree structure with the source code (files) of the three-dimensional robotic arm:



The robotic arm uses a FIFO (First In First Out) queue system to store the GCode commands and execute them one by one according to the order of arrival.



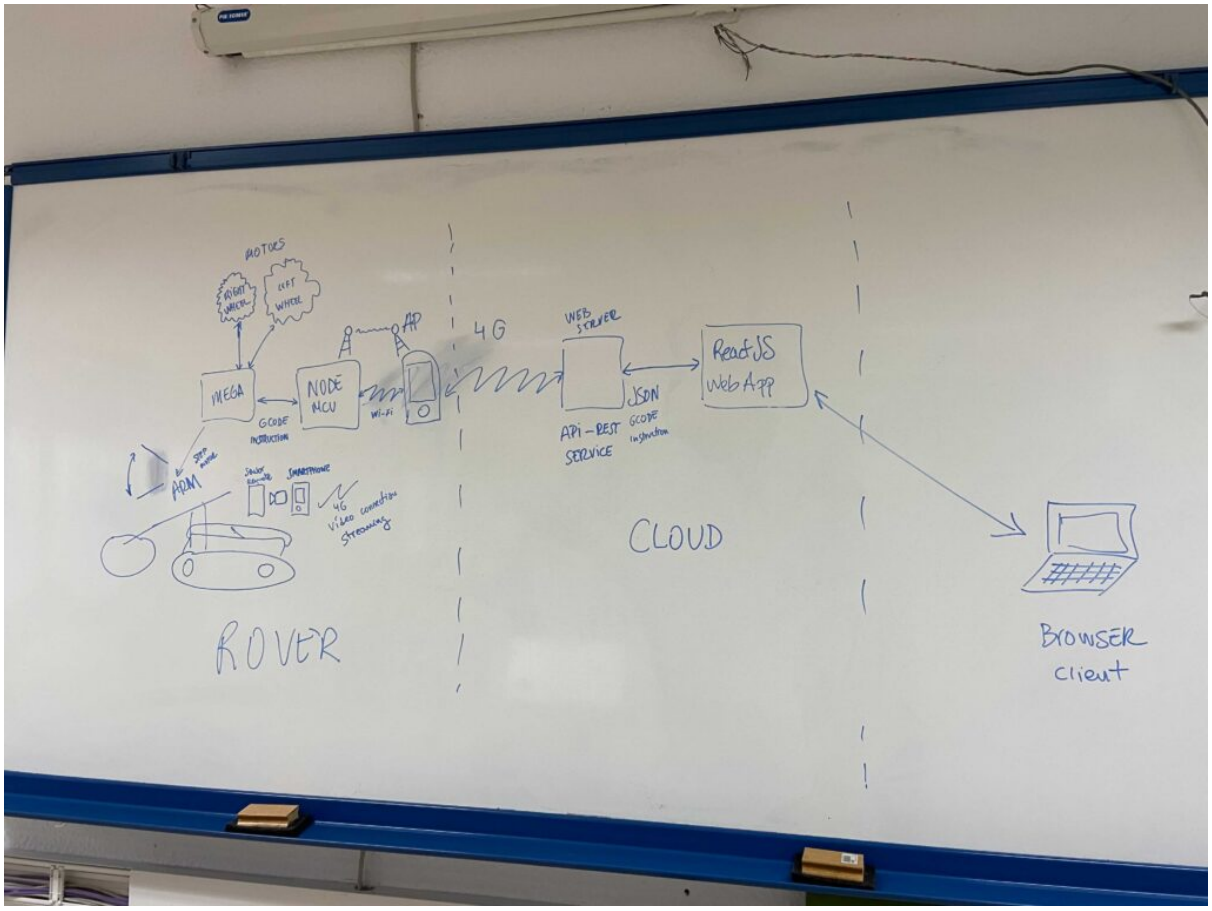
Lesson 11. Brainstorming. Analysis of the problem.

Once we have practically all the pieces of the puzzle to assemble the first prototype, what we have to do is come up with a design that we can implement. To design this prototype we have to base ourselves on the technologies that we already know and master, for this, we have chosen a hybrid system that uses the cloud to manage the control of the rover.

Below is a picture of the first prototype we have designed. The Rover will be controlled through the Internet by a browser that can be located on any device (a smartphone, a laptop, a tablet, etc). For example,we can control a rover located in Croatia from Spain.

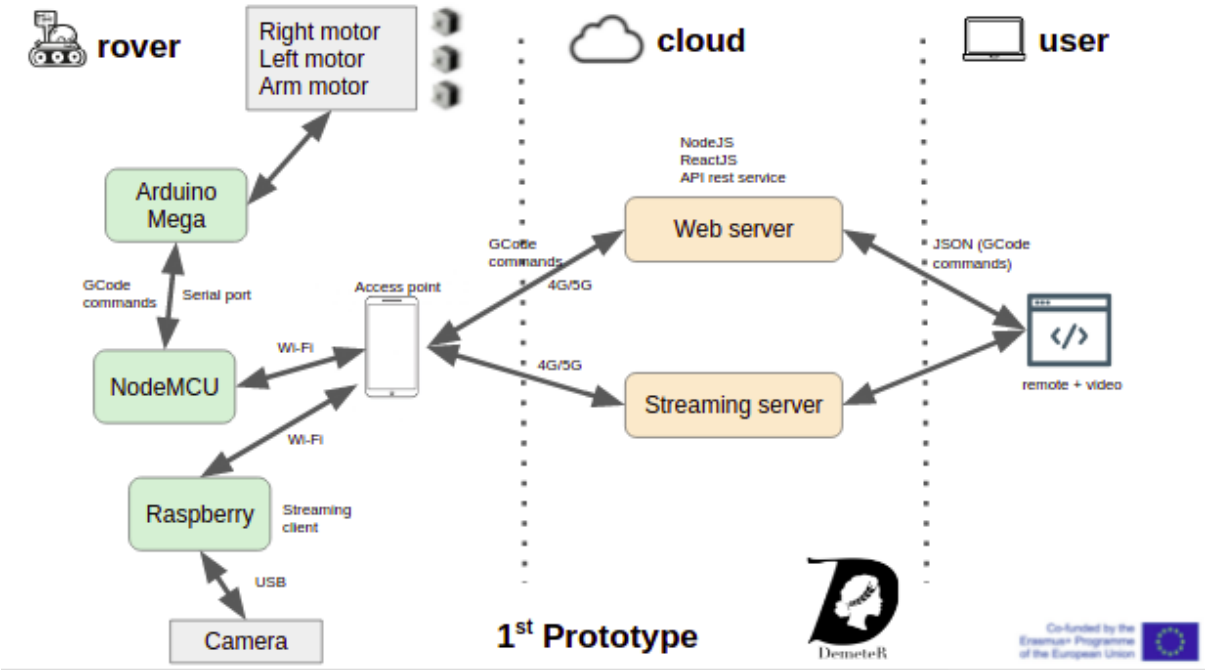
We will also need to have cloud services that can interact with the application that manages the rover and can send information to our device.

Below is an image of one of our brainstorming designs:



For the creation of this prototype we have relied on stepper motors and microcontrollers such as an arduino mega, a nodeMCU, and a raspberry pi (this is more like a small computer).

For communication between the browser and our web services, we have used the JSON format. JSON is a format used by JavaScript that is much lighter than XML. JSON is easy to analyze, interpret and generate by any program. Today JSON has displaced XML as a data interchange format.



To communicate the rover with our cloud services, what we are going to implement is an access point in the rover, which could be a mobile phone. A mobile phone can serve as an access point easily and is easy to place inside the rover.

The advantage of using a mobile phone is that it is self-powered and has enough power and a 4G or 5G connection to provide our rover with Internet.

The microcontroller that we are going to use to connect and receive GCode commands from our web services is a nodeMCU.

The advantage of using this device is that, like the arduino, it is open hardware, so there are many devices and versions on the market with a fairly low price. This board is based on the ESP8266 chip equipped with a Wi-Fi antenna which can connect to a network that is not too far away.

As the access point and the antenna of the nodeMCU will be practically together, we will not have any connectivity problems on that side.

The version that we are going to use of this device is version number three, which is second generation. This version (3.0) is similar to the previous version (2.0) but has some improvements, with which we can enjoy the latest features of the device.

The nodeMCU is going to be connected via serial to the arduino mega to be able to manage both the rover and move the sensor arm.

The raspberry will also be connected to the access point but it will work independently. Various systems can be used for video broadcasting. The video can be redirected directly to the backend of the website using a cloudflare-type service or an intermediate video server (streaming server) can be used.

Lesson 12. The power of the system. The batteries.

Batteries are a basic issue in any system that needs propulsion. Batteries are used in many devices such as mobile computers, drones, etc.

In our rover we need battery propulsion and we have chosen the old batteries that we had from the previous ICAROS project. We need to recycle components since in the DEMETER project we clearly bet on the circular economy.

Below are some of the slides that we are using to show the characteristics of the batteries to our students:

Baterías anticuadas - Nickel Metal-Hydride (Ni-MH)

Parecidas a las de Niquel-Cadmio pero sin el cadmio (que es muy contaminante).

Más capacidad de carga que las Ni-Cd (2 o 3 veces más).

Menos efecto memoria que las Ni-Cd.

Vida 500-2000 ciclos.

Más tasa de autodescarga que las Ni-Cd.

Posteriormente se crearon baterías LSD NiMH de baja autodescarga.

Baterías actuales - Lithium Polymer (Li-Po)

La sal de litio está en un gel (compuesto polimérico).

Fuga más improbable (gel en vez de líquido).

Baterías más flexibles.

Más baratas.

Más ligeras.

Desventaja: Muy inflamables - pueden explotar.

The batteries that we are currently using are lithium polymer batteries, the lithium salt is a polymeric compound in gel form which provides enough power to a device.

Some of the advantages these batteries have is that they are lighter than their predecessors and cheaper because lithium is a very common substance on earth.

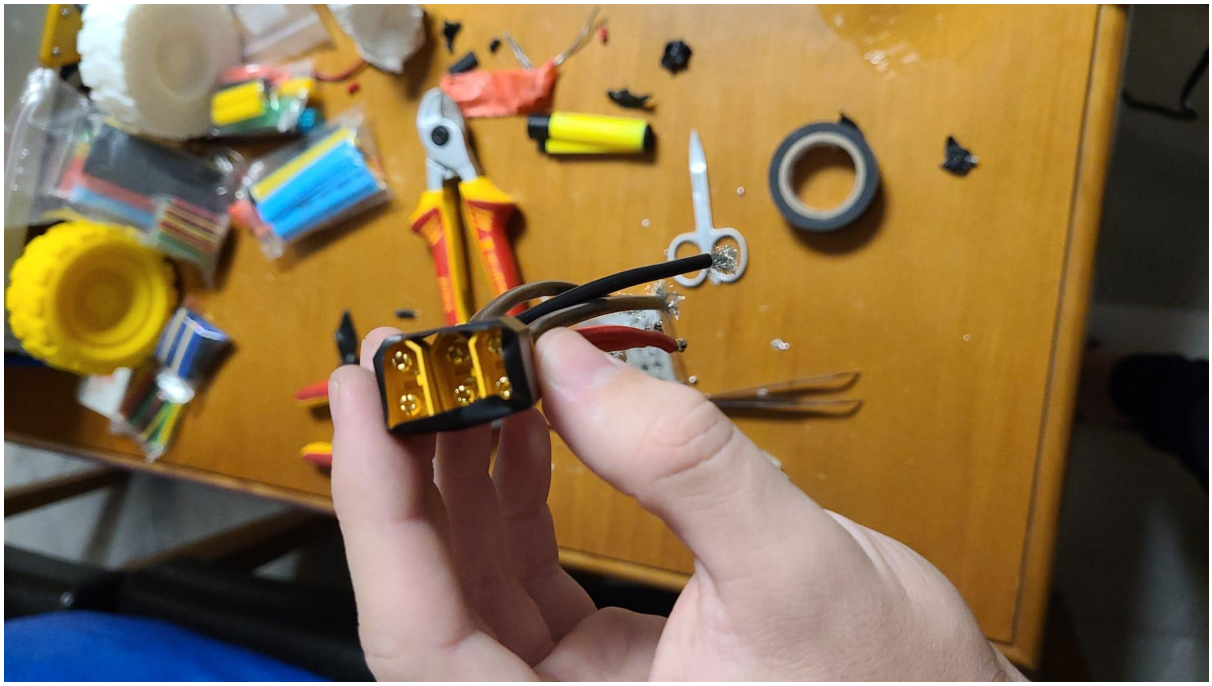
One of the disadvantages of these batteries is that, in certain circumstances they could explode and are highly flammable. That is why transport by plane can become complex due to its dangerousness.

For a first prototype we have chosen, as we have said, the batteries of our previous drones which had three cells, with which they will provide us with 11.1 V.

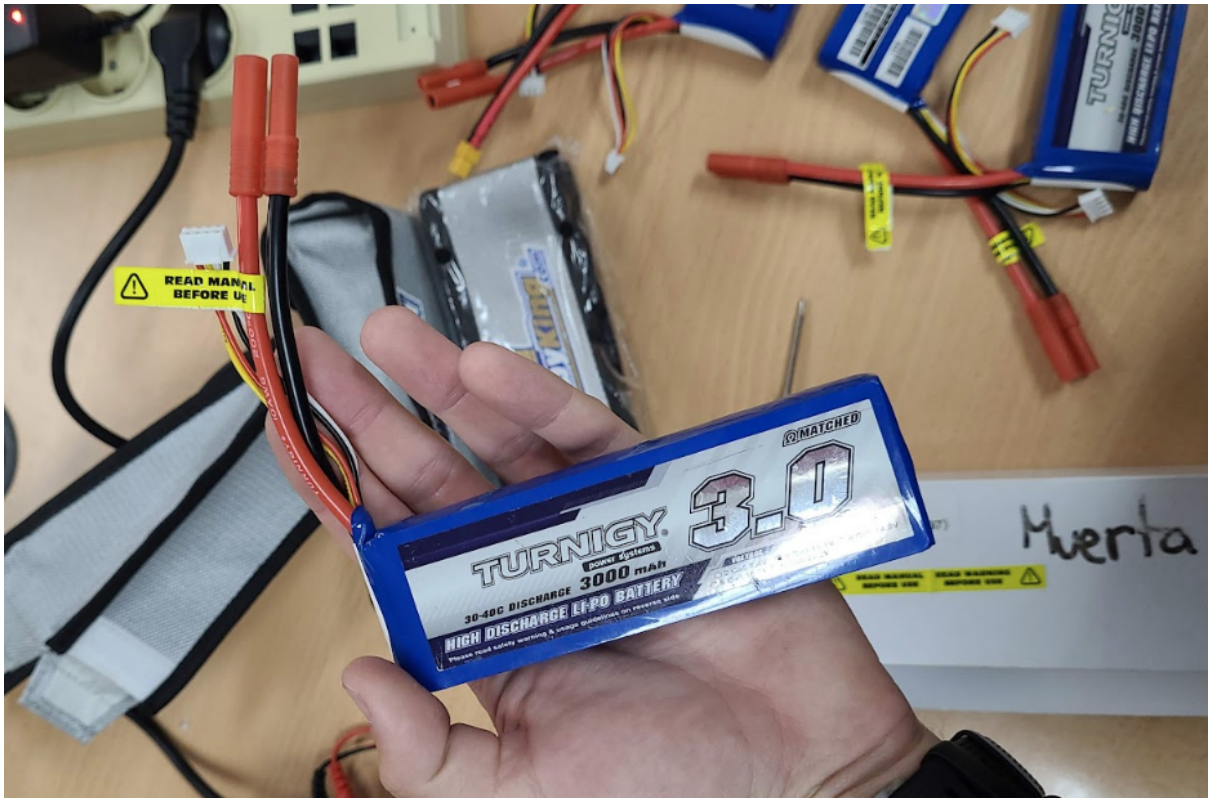
It is important to always also take into account two other parameters. The first is the capacity, which is measured in milliamps/hour and the second is the discharge rate (normally 30C or 40C). The discharge rate is important since the higher the discharge rate, the more power the rover or drone can receive.

To increase the capacity of the system, what we have chosen is to join three batteries in parallel with which we maintain the voltage but triple the capacity of the global battery.

In the following image you can see the assembly that one of our students has made to be able to connect three batteries in parallel using XT60 connectors:

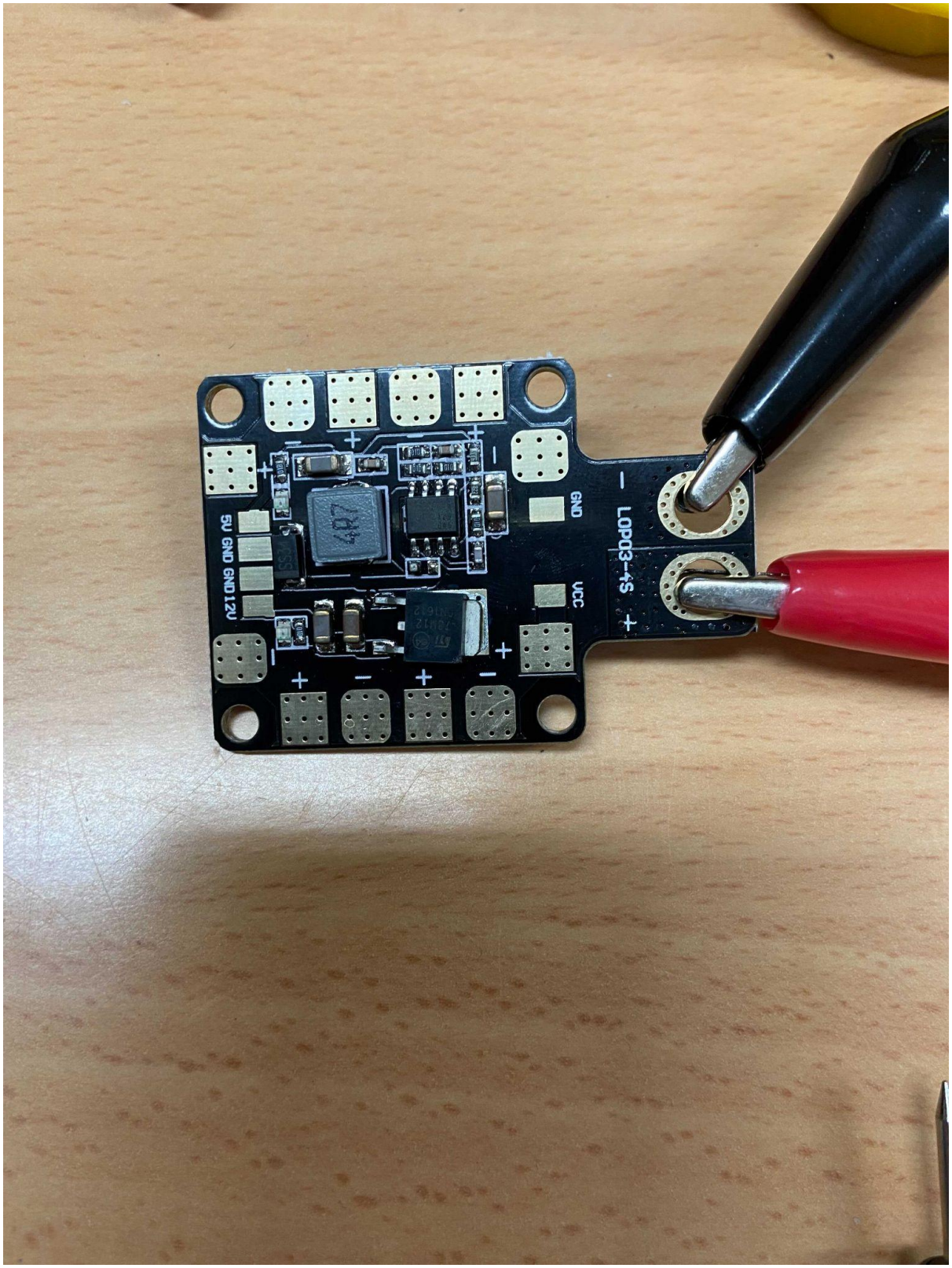


For the connection of the batteries we have used XT60 connectors. They are profusely used in modeling.



These connectors can be found in male or female format. The male connectors can be distinguished in the figure above. The batteries will have a female XT60 connector.

Among the advantages of these connectors is that they have very good connection quality and a fairly low resistance, as well as being perfect for mounting devices up to 65 A, which is much higher than what we are going to handle. In addition, this type of connectors can receive a voltage of up to 500 V in direct current which is also still much higher than the voltage that we are going to handle in our rover.



To supply the power to our system, we have reused one of the power distribution board that we use in the previous ICAROS project for our drones.

This power distribution board offers a voltage of 12 V in the six outputs and 5 V. We have verified that the 5 V voltage does not have enough amperage to power our electronic devices, therefore we will have to do a conversion of the outputs of 12 V to 5 V.

In the beginning we used the electronic circuitry of a car cigarette lighter plug. This device is prepared to transform the 12 volts of the car battery to 5 V to power any electronic device via USB.

Reusing elements and parts of everyday devices is one of the activities promoted in this project. In this way, students stimulate imagination and problem solving using real-life gadgets and devices.

Lesson 13. Programming the GCode command reception system in the rover: nodeMCU

In this phase of the project we decided to program the Rover reception system. For this, a technology called websockets has been used. This technology will allow us to communicate quickly and without delay the website and the Rover.

If you want the code of the receiving system, contact the DEMETER team through our website.

For the connection between the Rover and our web server, a technology called websocket has been used. This type of technology is used when two devices have to be connected and it is required to maintain an open connection between them. In 2008 Michael Carter and Ian Hikson invented this communication system. The advantage of this system is having a persistent full duplex connection.

The objective of our system is to implement a websocket client in such a way that it listens at all times to the server implemented in the cloud. Every time the server detects that the user makes a move, it quickly sends it in GCode format to the client located on the rover.

Lesson 14. Programming the action system in the rover: Arduino Mega .

The application in the arduino mega is based on at least four files of which only the main file will be commented on.

If you want the code of the action system, contact the DEMETER team through our website.

Lesson 15. Emission Of video. Tunneling services with CloudFlare.

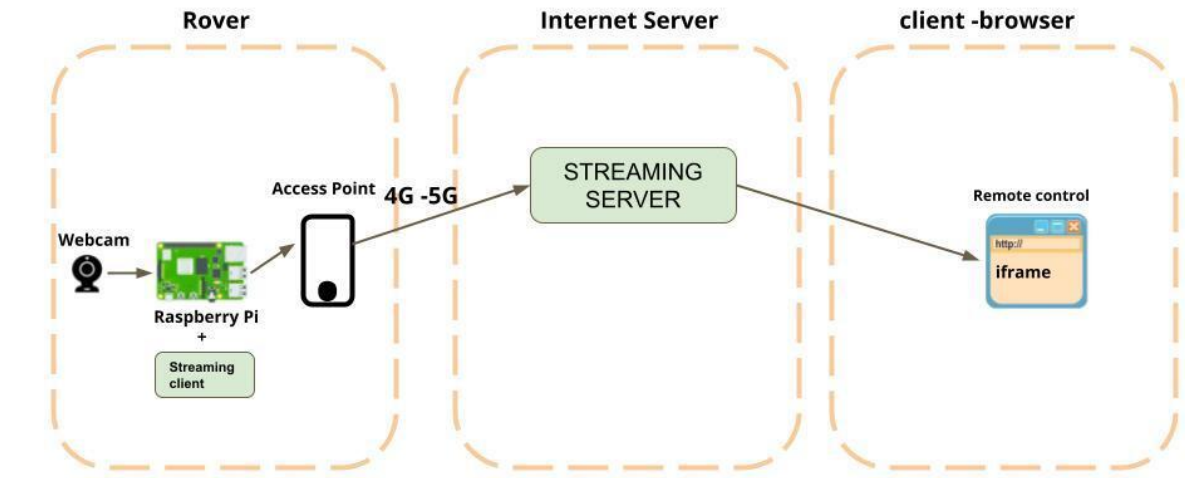
Important: to understand this lesson you need to have knowledge of networking and operating systems.

The main idea is to broadcast the camera connected to the Raspberry Pi over the Internet and receive the broadcast on the rover control web page. Due to security issues and the shortage of IPv4 addresses, the 4G / 5G connection of our mobile phone to the Internet is made through a NAT(Network Address Translation)

NAT allows many users to use the same public IP address. This means that we do not have the possibility of opening ports for streaming broadcasts. In order to do this, we have the need to use a proxy, which tunnels the broadcast service, and publishes it on the Internet using the IP address and an open port of the proxy.

What we are going to do using cloudflare is to use its servers as a proxy, and in this way, each connection made to the URL of the camera broadcasting will be made through its servers, who will be the ones that connect to the Raspberry Pi and interconnect the broadcast with the client. This service provides security against attacks, privacy by not exposing the IP address of the access point, granular access controls, and we can avoid the problem of having a NAT in the network of the access point.

Cofinanciado por el programa Erasmus+ de la Unión Europea



Cloudflare, in its free account plan, allows us to configure it as the domain's DNS server, and we have thousands of features available for free. The most interesting may be:

- It allows us to use it as a CDN, with which we can serve the cached content of our website, increasing speed and taking work off the server.
- Using cloudflare as a proxy, which benefits us in that the site is served from its IP address, not exposing ours (security), it assigns us an https certificate signed by it, adds layers of security (HTST, force HTTPS, only access from certain IPs / countries, etc.)
- A powerful firewall from which we can restrict access to our site and monitor it
- Access control by username / password (gmail, double authentication factor, etc.)
- And a long etcetera, which help us to have granular security control.

Using cloudflare in our domain is very simple, we just have to go to the cloudflare page, register, add the domain to link, and it will provide us with some nameservers, which we must configure in the NS or nameservers option of our domain provider and automatically, Cloudflare It will detect the current entries of our domain and configure them (in case we use a hosting, so that the records of it are automatically added, without having to configure them manually).

We can also acquire a domain with them, which gives us Whois privacy (in the domain records, our contact information will not appear, but generic ones, protecting our privacy) and with a fixed purchase price and the same fixed renewal price (.com domain is usually around 8 $)

The function that we are going to use today is Argo Tunnel, which is based on, lifting the Cloudflare service on a machine (docker, LXC container, real machine or Raspberry Pi), create a tunnel using cloudflare servers in their different countries and expose a service (http / https) through their servers and under their proxy, avoiding exposing ports on our machine, and adding the cloudflare security layer, and its firewall for the subsequent control.

The service is called cloudflared and its installation is very simple, they have packages for different operating systems (Windows, Mac, Linux) and a Docker container ready to deploy. (We have the official documentation on your Github: Https://github.com/cloudflare/cloudflared

We have a step-by-step guide from cloudflare in this webpage: https://developers.cloudflare.com/cloudflare-one/connections / connect-apps / install-and-setup / tunnel-guide.

The installation will be carried out as an example exposing an HTTPS web such as the Proxmox server web administration. This will be adapted depending on the IP address and ports of your video broadcast, since this service is suitable to expose any HTTP or HTTPS service.

Installation of the clouflared server

We are going to perform the installation in an LXC Ubuntu server container, running on Proxmox and being root (we will not use sudo) .

In order to do this, the first thing to accomplish is adding the cloudflare repositories:

```
echo 'deb [signed-by = / usr / share / keyrings / cloudflare-main.gpg]
https://pkg.cloudflare.com/ focal main' | tee
/etc/apt/sources.list.d/cloudflare-main.list
```

```
root@Contenedor-pruebas:~# echo 'deb [signed-by=/usr/share/keyrings/cloudflare-main.gpg] https://pkg
.cloudflare.com/ focal main' | tee /etc/apt/sources.list.d/cloudflare-main.list
deb [signed-by=/usr/share/keyrings/cloudflare-main.gpg] https://pkg.cloudflare.com/ focal main
root@Contenedor-pruebas:~#
```

And then we import the GPG key from the repository:

```
curl https://pkg.cloudflare.com/cloudflare-main.gpg -o
```

```
/usr/share/keyrings/cloudflare-main.gpg
```

```
root@Contenedor-pruebas:~# sudo curl https://pkg.cloudflare.com/cloudflare-main.gpg -o /usr/share/ke
yrings/cloudflare-main.gpg
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   640  100   640    0     0   2229      0 --:--:-- --:--:-- --:--:--  2229
root@Contenedor-pruebas:~#
```

And update the repositories:

```
apt update
```

Now it is time to install the package cloudflared from the newly added repositories.

```
apt install cloudflared We
```

```
root@Contenedor-pruebas:~# apt install cloudflared
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  cloudflared
0 upgraded, 1 newly installed, 0 to remove and 159 not upgraded.
Need to get 17.3 MB of archives.
After this operation, 36.3 MB of additional disk space will be used.
Get:1 https://pkg.cloudflare.com focal/main amd64 cloudflared amd64 2021.11.0 [17.3 MB]
Fetched 17.3 MB in 1s (13.8 MB/s)
Selecting previously unselected package cloudflared.
(Reading database ... 20747 files and directories currently installed.)
Preparing to unpack .../cloudflared_2021.11.0_amd64.deb ...
Unpacking cloudflared (2021.11.0) ...
Setting up cloudflared (2021.11.0) ...
Processing triggers for man-db (2.9.1-1) ...
root@Contenedor-pruebas:~#
```

Now we can download multiple packages, without the need to add the repositories, but by adding the repositories we make sure that the package is updated correctly.

Configuration

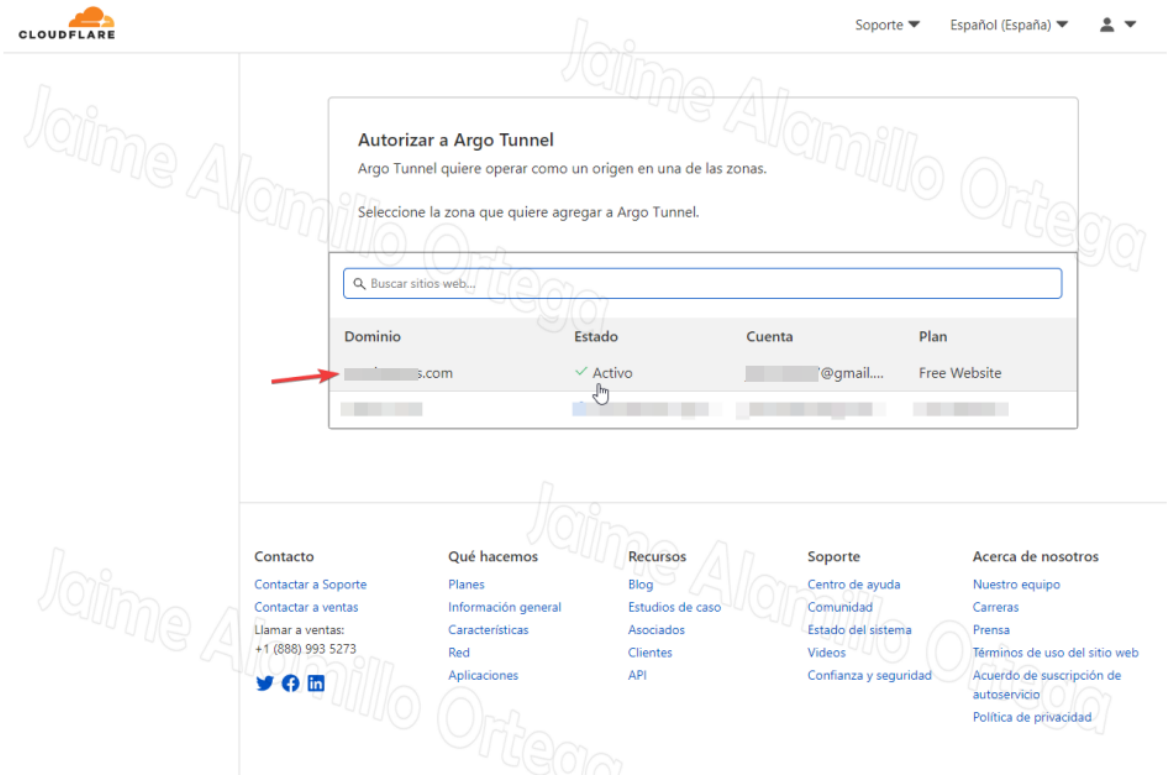
It is time to configure the service. First we must link our account in cloudflare. In order to do this we must execute the following command in a terminal:

```
cloudflared tunnel login
```

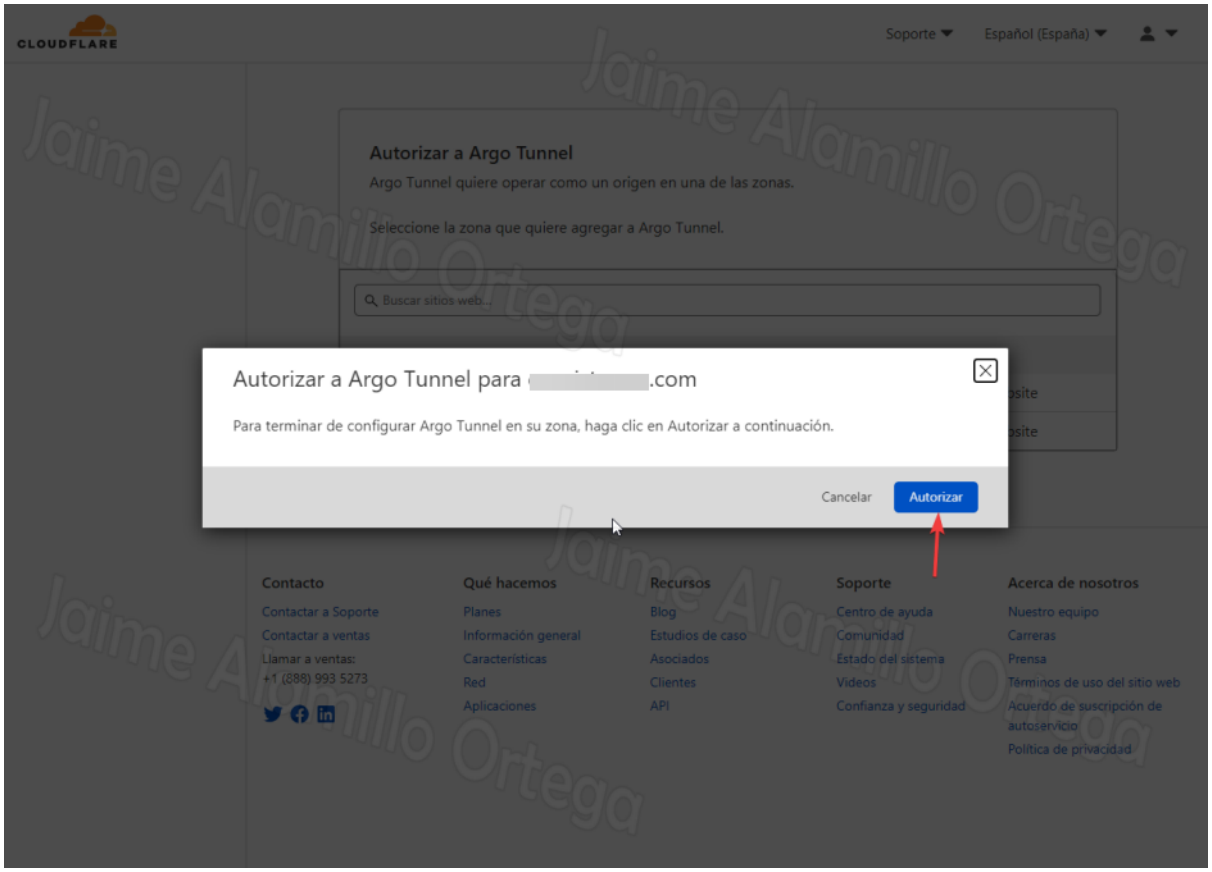
And a url will appear that we must open in the browser and log in with our account in cloudflare.

```
root@Contenedor-pruebas:~# cloudflared tunnel login
Please open the following URL and log in with your Cloudflare account:

https://dash.cloudflare.com/argotunnel?callback=https%3A%2F%2Flogin.cloudflareaccess.org%2Fnbgx8zboB
P5-G1gY93AhkX6M-yBUoS8TxYlG-cLGiCY%3D

Leave cloudflared running to download the cert automatically.
```

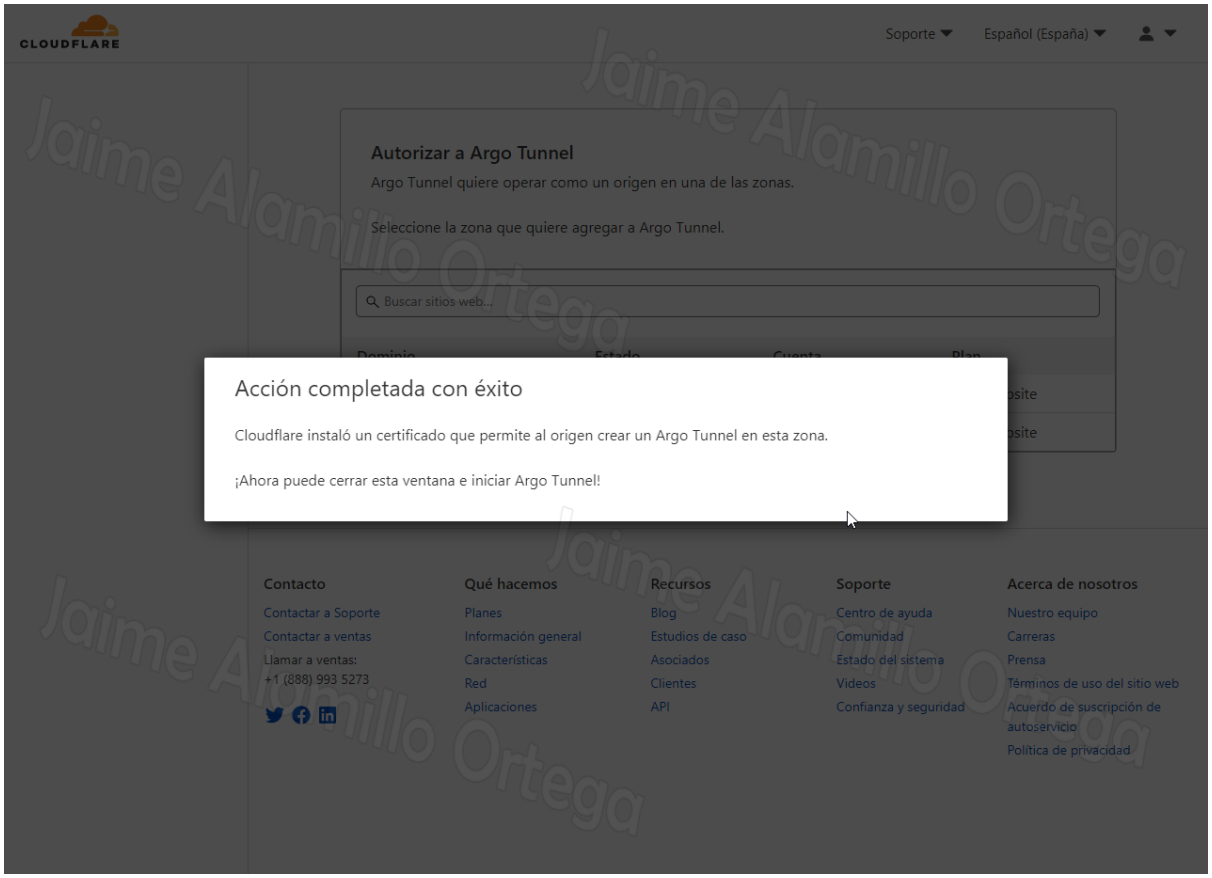
Then, when opening the link, it asks us from which zone the tunnel will operate (zone = domain). We select it.



The system will ask us for confirmation to authorize the cloudflared service to operate in that area. To continue, we click on Authorize.



At this point, the authentication has been successful, and cloudflare has installed a certificate in the cloudflared service that allows you to create tunnels in our area.



When we return to the Ubuntu machine, we can verify that it has been successfully logged in and that the certificate has been created in the location of the user's personal folder (root), in /root/.cloudflared/, in case we want to move it to another server.

```
root@Contenedor-pruebas:~# cloudflared tunnel login
Please open the following URL and log in with your Cloudflare account:

https://dash.cloudflare.com/argotunnel?callback=https%3A%2F%2Flogin.cloudflareaccess.org%2Fnbgx8zboB
P5-G1gY93AhkX6M-yBUoS8TxYlG-cLGiCY%3D

Leave cloudflared running to download the cert automatically.
2021-12-11T13:59:34Z INF Waiting for login...
2021-12-11T14:00:27Z INF Waiting for login...
You have successfully logged in.
If you wish to copy your credentials to a server, they have been saved to:
/root/.cloudflared/cert.pem
root@Contenedor-pruebas:~#
```

Now, it is time to create a tunnel, with the command cloudflared, and the tunnel arguments create "name". The name that we will assign for the test will be tunnelalboran.

```
cloudlfared tunnel create tunnelalboran
```

The system warns that the tunnel has been created correctly, and the credentials are in the path /root/.cloudflared/. These credentials help us to identify this tunnel, and have it saved to build it later.

Also, it indicates the unique identifier of the tunnel, which is the same as the name of the file and is used for the subsequent configuration and access to it.

```
root@Contenedor-pruebas:~# cloudflared tunnel create tunnelalboran
Tunnel credentials written to /root/.cloudflared/a54c9d89-7721-486f-aaff-6a6a1ba14dba.json. cloudfla
red chose this file based on where your origin certificate was found. Keep this file secret. To revo
ke these credentials, delete the tunnel.

Created tunnel tunnelalboran with id a54c9d89-7721-486f-aaff-6a6a1ba14dba
root@Contenedor-pruebas:~#
```

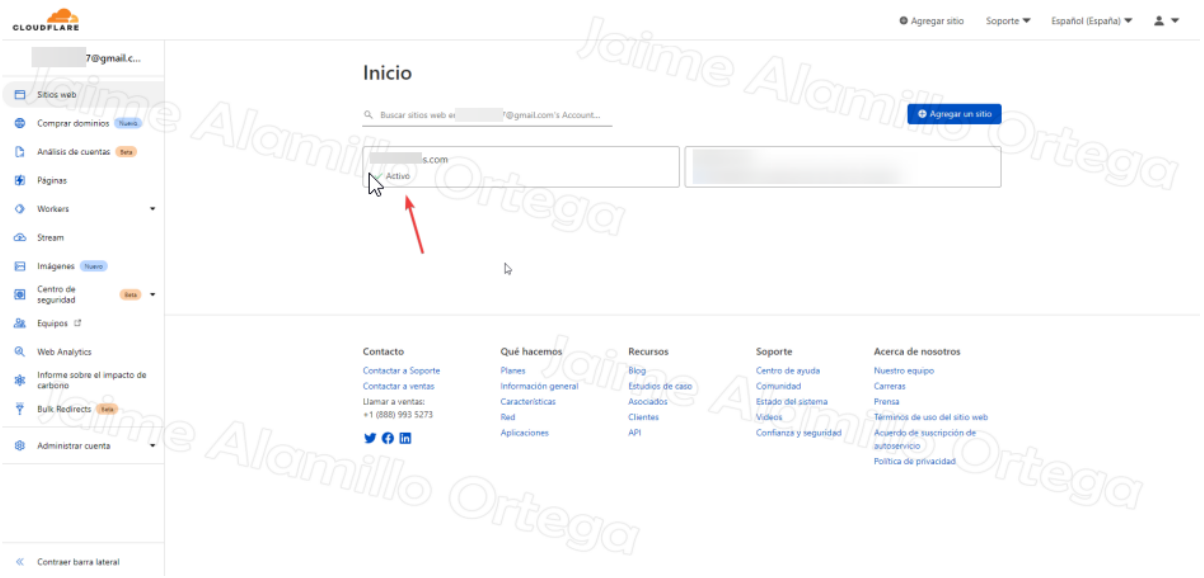
We can make use of the argument list to see the tunnels created in our cloudflare account. We currently have 2, one dedicated to the demeter project, and the one just created for testing.

```
cloudflared tunnel list
```

```
root@Contenedor-pruebas:~# cloudflared tunnel list
You can obtain more detailed information for each tunnel with `cloudflared tunnel info <name/uuid>`
ID                                   NAME         CREATED              CONNECTIONS
                                     demeter      2021-12-09T20:07:32Z
a54c9d89-7721-486f-aaff-6a6a1ba14dba tunnelalboran 2021-12-11T14:05:32Z
root@Contenedor-pruebas:~#
```

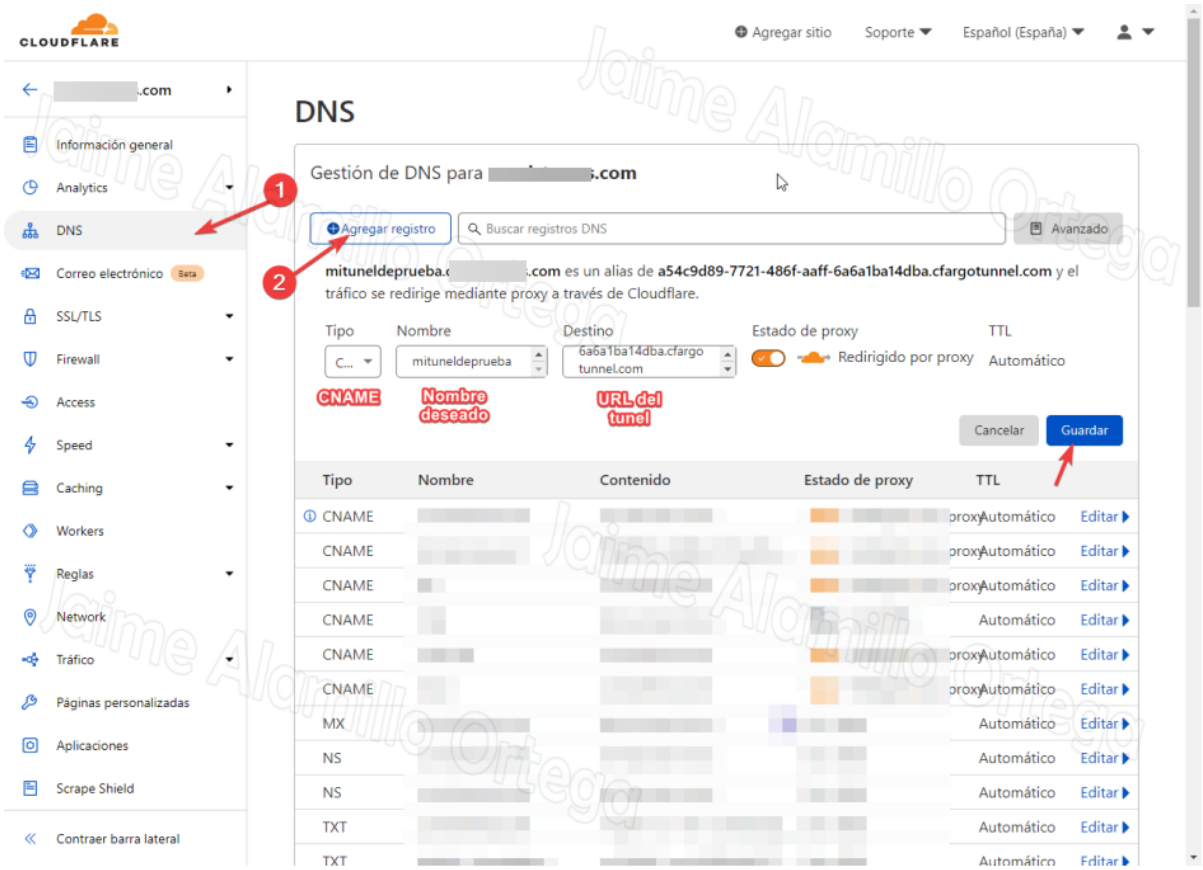
Now, we need to create a CNAME for the tunnel.

The default url of the tunnel is IDDELTUNEL.cfargotunnel.com, in our case, is a54c9d89-7721-486f-aaff-6a6a1ba14dba.cfargotunnel.com . We have go to cloudflare and select the area we are working on.

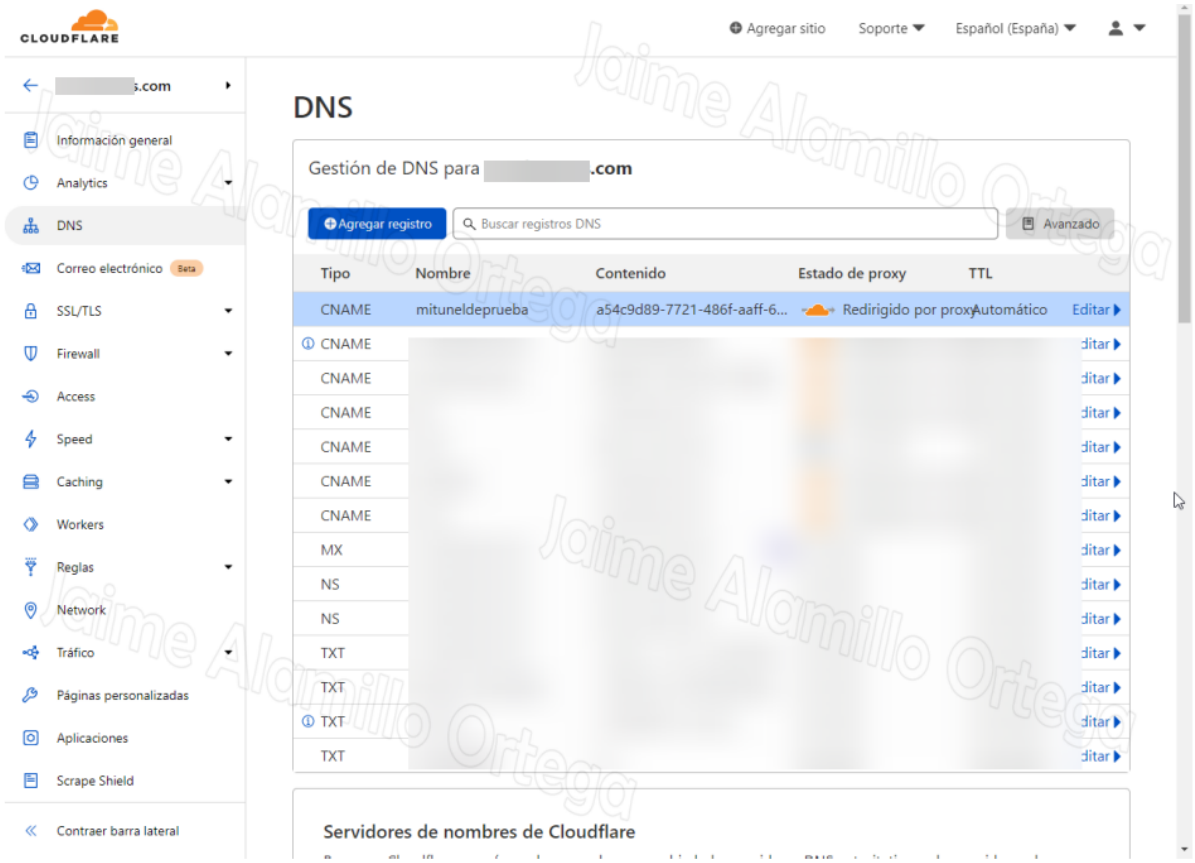


On the side panel, we turn to DNS records. Once in them we click on *add new record*. In the type of record, we select CNAME, in name, we define the name of the subdomain that the tunnel is going to have, in my case, mituneldetest, and in destination, the default url of the tunnel. We leave the checkbox for Redirected by proxy to use the cloudflare proxy to use the benefits of the cloudflare proxy and click on save.

We can verify that it tells us that mydomain.com is an alias of a54c9d89-7721-486f-aaff-6a6a1ba14dba.cfargotunnel.com and the traffic is redirected by proxy through Cloudflare.



And we can check that the registry has been created correctly.



Next, we must create a configuration file where we will define the tunnel together with the ip: port to tunnel. To do this, we can create it in the following paths **~ / .cloudflared**, **~ / .cloudflare-warp**, **~ / cloudflare-warp**, **/ etc / cloudflared**, or **/ usr / local / etc / cloudflared.**

In my case, I will create it in the same path where the certificates and tunnel configuration have been created, in **/root/.cloudflared/** (i.e. **~ / .cloudflared /**). To do this, we use the nano command.

```
nano /root/.cloudflared/config.yml
```

In the file, we must add the following

```
tunnel: nameofltunel
credentials-file: /root/.cloudflared/ID.json
ingress:
 - hostname: cname.domain.com
   service: https: // ip:port
   originRequest:
     noTLSVerify: true
 - service: http_status: 404
```

- In the tunnel line, we will specify the name of the tunnel, in my case, tunnelalboran.
- On the credentials-file line. We specify the path to the file .json the credentials of the tunnel
- create the ingress section to specify the hostname to go through the tunnel
- on the line specify the hostname cname created, ie, subdomain and domain, in my case **mituneldeprueba.midominio.com**
- In the service line, we define the url to tunnel, together with the protocol (http or https) and its port. The protocol must be the one we currently use to access the service, cloudflare automatically tunnels it over https. I am going to tunnel the Proxmox web administration, so my url is **https://10.0.0.5:8006**
- And the line **originRequest**, with the argument **noTLSVerify**: true, specifies, that it does not verify the certificate issued by the service to tunnel (proxmox), since being self-signed, if we do not add that line, it throws us an error since it does not trust the certifying entity and does not lift the tunnel. This line is indifferent if we work on http.
- And finally, we specify that all traffic that does not go to this service, send a 404 error page, that is, it is a general rule.

In our case, we will use the service **https 10.0.0.5** with port **8006**, since it can be a service from the local machine (localhost) or from our local network (IP address of the service).
The content of our file will look like this:

```
tunnel: tunnelalboran
credentials-file:
/root/.cloudflared/a54c9d89-7721-486f-aaff-6a6a1ba14dba.json
ingress:
 - hostname: mituneldepricación.midomain.com
   service: https: // 10.0.0.5:8006
   originRequest:
     noTLSVerify: true
 - service: http_status: 404save
```

Weand close the file

Operation

Once the configuration file is created, we can simply raise the tunnel with the command.

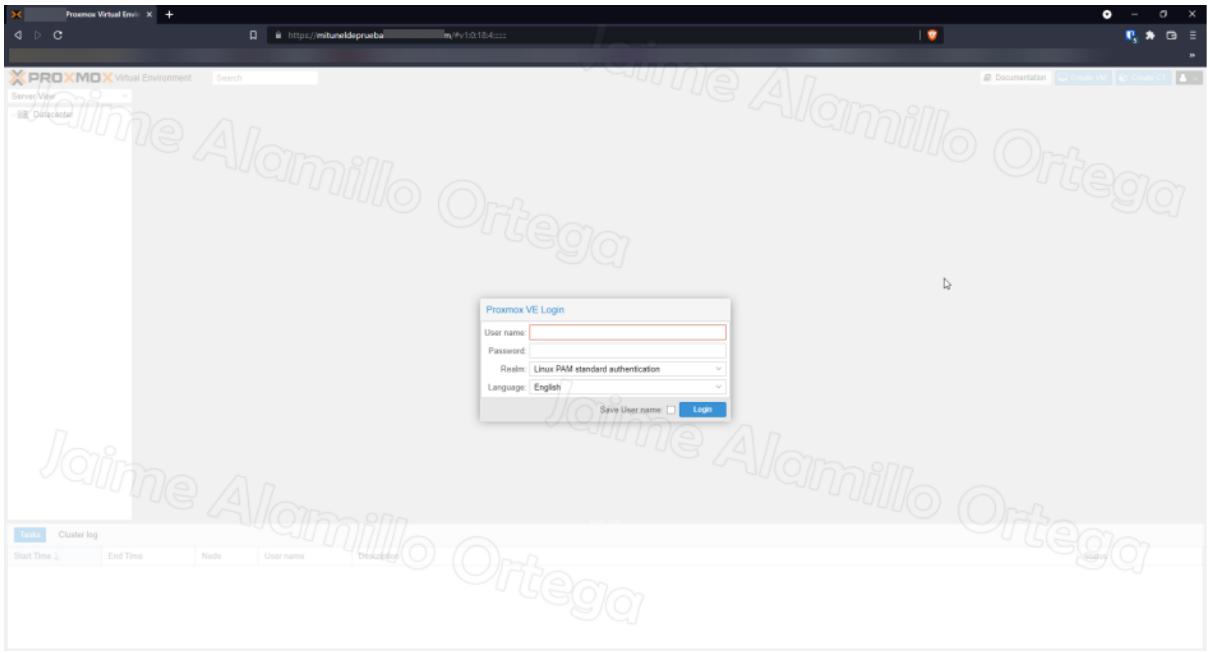
```
cloudflared tunnel run
```

adding the tunnel ID or name later. For my tunnel it would look like this::

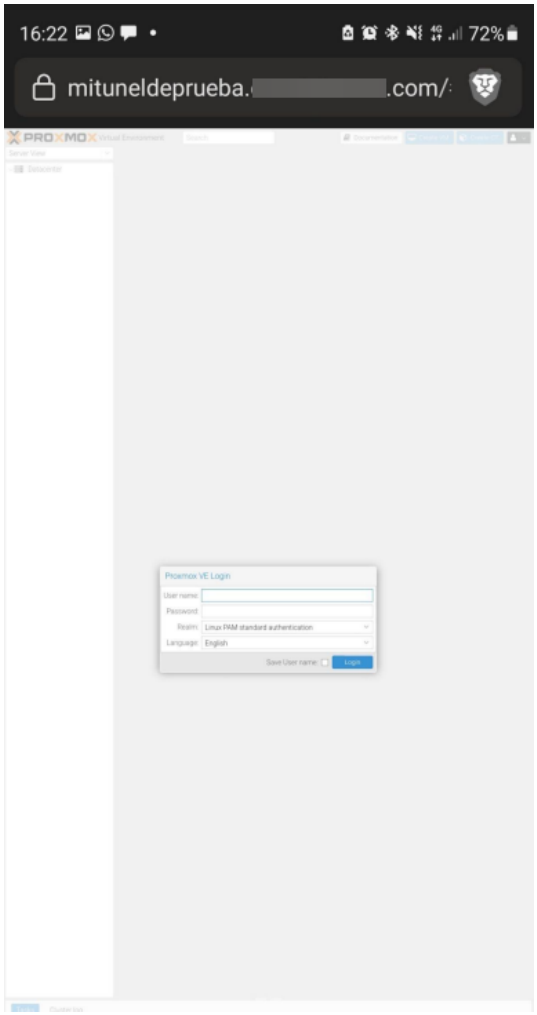
```
cloudflared tunnel run tunnelalboran//mituneldepricación.midominio.com/
```

```
root@Contenedor-pruebas:~# cloudflared tunnel run tunnelalboran
2021-12-11T15:20:30Z INF Starting tunnel tunnelID=a54c9d89-7721-486f-aaff-6a6a1ba14dba
2021-12-11T15:20:30Z INF Version 2021.11.0
2021-12-11T15:20:30Z INF GOOS: linux, GOVersion: devel +a84af465cb Mon Aug 9 10:31:00 2021 -0700, Go
Arch: amd64
2021-12-11T15:20:30Z INF Settings: map[cred-file:/root/.cloudflared/a54c9d89-7721-486f-aaff-6a6a1ba1
4dba.json credentials-file:/root/.cloudflared/a54c9d89-7721-486f-aaff-6a6a1ba14dba.json]
2021-12-11T15:20:30Z INF Generated Connector ID: 8fb9ba26-93ab-4d65-99b1-df733aceba46
2021-12-11T15:20:30Z INF cloudflared will not automatically update if installed by a package manager
.
2021-12-11T15:20:30Z INF Initial protocol http2
2021-12-11T15:20:30Z INF Starting metrics server on 127.0.0.1:45665/metrics
2021-12-11T15:20:31Z INF Connection 2a0b08eb-05d2-41b8-9a9a-17ed706a281e registered connIndex=0 loca
tion=MAD
2021-12-11T15:20:32Z INF Connection bbf3b50c-a5b2-40c1-96f8-6271acdbbe06 registered connIndex=1 loca
tion=MXP
2021-12-11T15:20:33Z INF Connection 685a703a-0e50-494d-a01b-b313fc8bde77 registered connIndex=2 loca
tion=MAD
2021-12-11T15:20:34Z INF Connection e8a8b151-d18b-4242-aff1-f96cd703ba63 registered connIndex=3 loca
tion=MXP
```

And we tried to access the service from the CNAMEAnd we **https**



can verify that it works perfectly.
Now, I am going to perform the test from my mobile phone, working on the 4G network.



Additional

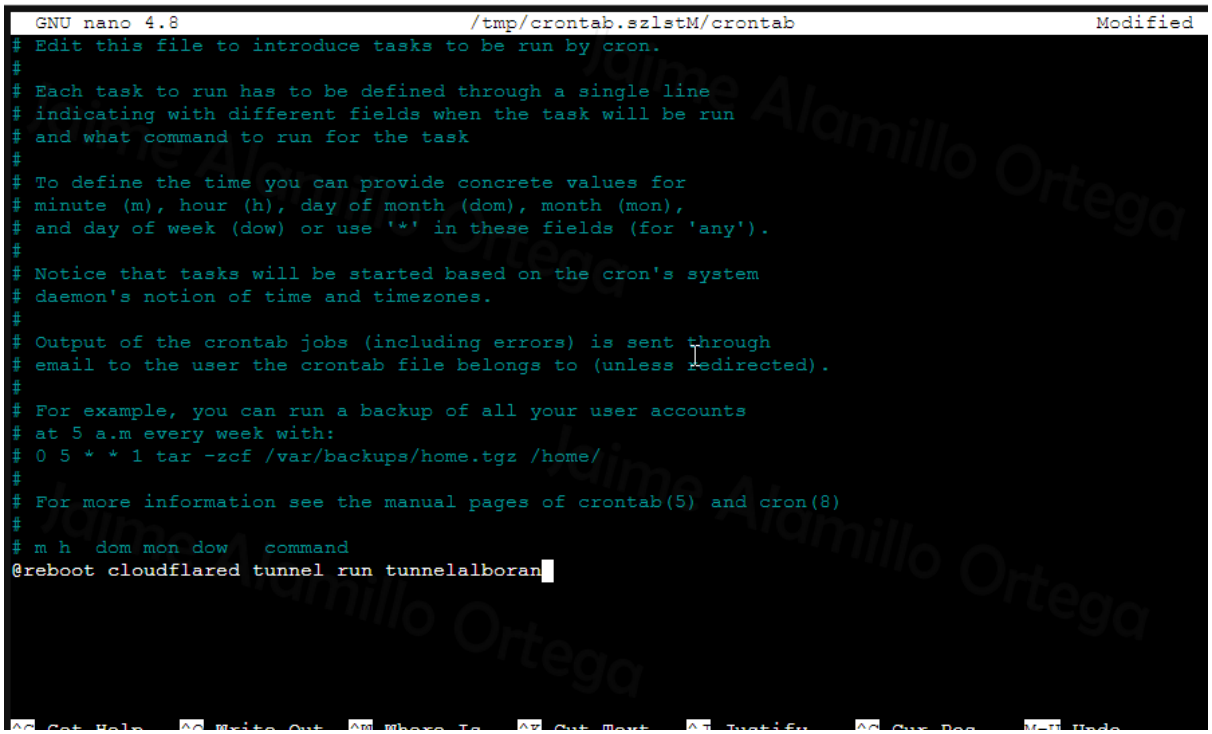
tips We can auto-start a tunnel, adding the command to crontab, so that it starts automatically when the computer starts.
We open the crontab editor.

```
crontab -e
```

We add the following line:

```
@reboot tunnel run tunnelalboransave
```

And we close the file saving the information in it.



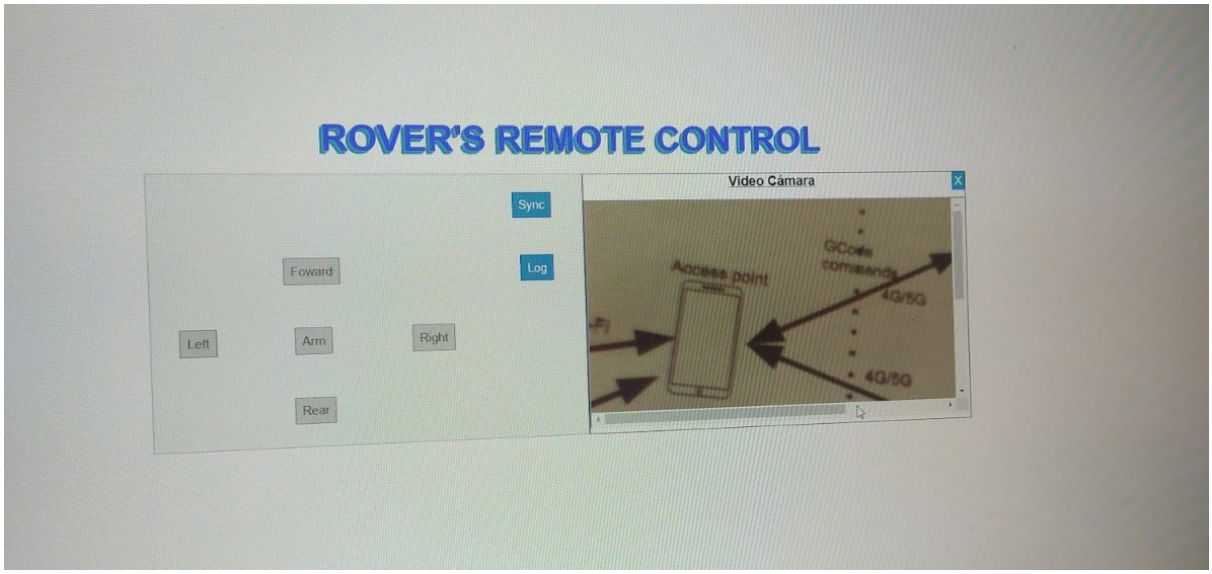
And every time we restart the machine, the tunnel will automatically start.
We can also stop the tunnel with the command

```
cloudflared tunnel cleanup tunnelalboran
```

and eliminate it with the command

```
cloudflared tunnel delete tunnelalboran
```

The following shows the video reception and control system in the browser:



Bibliography

- Ecured.cu. *Modelo de prototipos*. Retrieved from: https://www.ecured.cu/Modelo_de_prototipos. Revised on 20 june 2020.
- UML.org. *Unified Modeling Language*. Retrieved from: http://uml.org/. Revised on 15th of march 2020.
- Reactjs.org. *React A JavaScript library for building user interfaces*. Retrieved from: https://reactjs.org/. Revised on 7th of february 2020.
- Salcescu, C. 2019. *An introduction to the Flux architectural pattern*. Retrieved from: https://www.freecodecamp.org/news/an-introduction-to-the-flux-architectural-pattern-674ea74775c9/. Revised on 12th of april 2020.
- git-scm.com. *GIT--fast-version-control*. Retrieved from: https://git-scm.com/. Revised on 13th of april 2020.
- atom.io. *A hackable text editor for the 21st Century.* Retrieved from: https://atom.io/. . Revised on 18th of february 2020.
- getbootstrap.com. *Build fast, responsive sites with Bootstrap.* Retrieved from: https://getbootstrap.com/. Revised on 17th of january 2020.