CR-ÆSIR

TEAM DESCRIPTION MATERIALS ROBOCUP INTERNATIONAL 2024

RAPIDLY MANUFACTURED ROBOT CHALLENGE

CR-ÆSIR Monterrey, Nuevo León, México

TEAM DESCRIPTION MATERIALS

Logistical information

Team name: CR-ÆSIR

Organization: CIDEB - Universidad Autónoma de Nuevo León

Country: México.

Contact person: Erick Federico Sánchez Flores

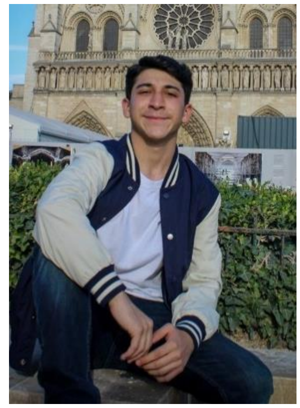
Email: **ericksanchezflores@hotmail.com**

Mentors

Dr. Erick Sánchez

Héctor Barba

Members



Odin Francisco Haro Valdés

Team Leader

and Mechanical Designer

Moisés Eduardo Rivera Flores

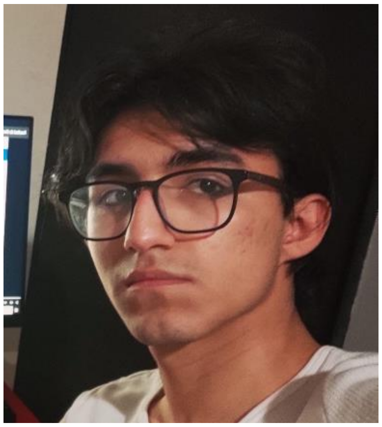
Mechanical Designer

Eduardo Didier Aguilar Alvarez

Programmer

Jesús Alberto Matas Loya

Mechanical Designer



Sergio Alejandro Mondragón Ortiz

Programmer

Aarón Pinales Ibarra

Programmer

1. Introduction:

1.1 Background

CR-ÆSIR is a team of 6 high school students and 2 mentors, our goal is to represent Mexico and demonstrate that we can do a good representation in the RMRC world championship 2024 as we did in the last RoboCup 2023 in which we won the first place in the category.

CR-AESIR old team in the last RoboCup 2023

CR-ÆSIR forms part of the CIDEB International school from the Universidad Autónoma de Nuevo León (UANL), the focus of the organization is to participate and demonstrate our robotic knowledge applied in different sectors, in this case in real life situations, such as the RoboCup "Rapidly Manufactured Robot Challenge" category.

Our robot was made using 3D printed parts, this was because 3D printing offers several benefits such as: allowing for the creation of complex designs that may not be possible using traditional manufacturing methods, the cost per part its quite low and the amount of CAD models that are available makes it easier to develop and adapt already created parts while also sharing our own original models. We believe on the importance of having those resources available and the open-source vision that the competition as a way to preserve the knowledge and tested methods and techniques to complete the challenges presented.

2. System description

2.1 Hardware

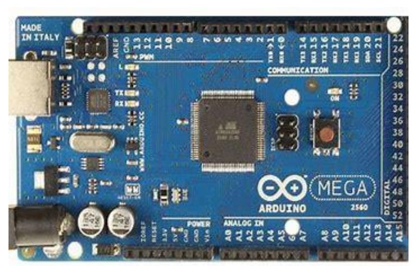
2.1.1 Processing unit

2.1.1.1 Raspberry Pi 5

The Raspberry Pi 5 is a single board computer used in the robot. It is mostly in charge of the communication between the computer of the operator and the Arduino mega 2560 and other processes like the gathering of information from the robot state and environment.

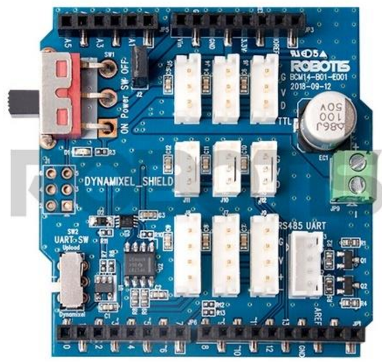
2.1.1.2 Arduino Mega 2560

This microcontroller board is used for sending the instructions to the actuators of the robot. Some examples are the servos and direct current motors.



2.1.1.3 DYNAMIXEL Shield

This shield is used by the Arduino Mega 2560 to facilitate TTL communication with the DYNAMIXEL servos used in the robot.



2.1.1.4 IBT_2 43 A Drivers

The new robot has different motors, direct current motors, so that change generated new troubles, that were solved with these motor drivers. They supply the need for more resistance of amperage and the challenge of changing the direction of the motors.

2.1.2 Sensors

2.1.2.1 QR

It is an added function for the main camera (Logitech C920), that allows to read QR, and display the written text or open links in the browser. For this we use a library called “pyzbar.pyzbar” in Python.

2.1.2.2 Motion (tracking)

The tracking is done in the computer that is connected to the robot, the image is captured with the main camera, the Logitech C920, and through an option in the interface, you can change the camera mode to the tracking mode, in that mode the image goes through a series of filters which are:

- It finds the differences between two images captured by the camera.

- It goes through a grayscale.

- Remove noise from the image.

- It increases the contrast of different pixels.

- It marks the contours.

- Find motion points and draw a frame in the original image.

For all this we use different functions in the OpenCV library.

2.1.2.3 Hazmat (danger signals)

The camera captures the image for the hazmat, the hazmat is a mode of the camera in the interface, when activating it the information of the hazmat will be shown on the screen, the whole process will be carried out in the computer with which it is controlled, for this the library “tensorflow” is used, to create and train a neural network.

2.1.2.4 Colour

For color the image is taken from the Logitech C920, the interface has a mode for color detection, where through a few lines of code, shows on the screen the color, RGB and hexadecimal code of a pixel in the middle of the screen.

2.1.3 Camera

Our only camera is a Logitech C920 HD, and this camera is for viewing the claw, the front of the robot, and both sides. In addition, it contains all the sensors, such as: QR, motion, hazmat, and color. It is located on the upper arm.

2.1.4 Actuators

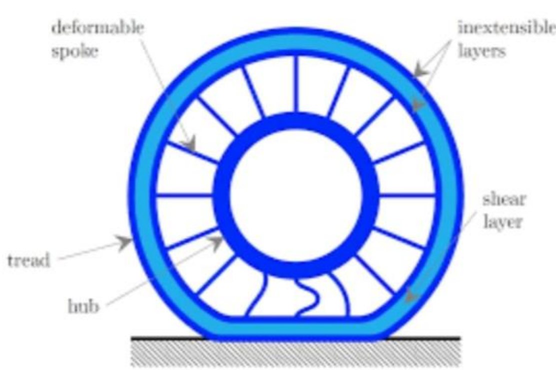
In this robot we used 2 Dynamixel AX-18A and 2 Dynamixel AX-12A for the arm, 2 Dynamixel AX-12A for the flipper and 4 DC Motors Pololu 37D 131:1.

2.2 Structure

All the structure was designed by our team in Fusion 360 by Autodesk and then printed in a Prusa MK4 using PLA filament sponsored by La Plasticueva. In another hand, we use TPU filament for printing the tires of the robot.

2.2.1 Tires

The first attempt was trying out air tires since these can take the shape of different terrains which can help the robot drive through easily. Nevertheless, there was a problem. The size of the tires were too big for the robot. After some research, the airless tires were considered. These tires, as have a thread as the outermost layer. This thread has a pattern texture that prevents that the tire to slip in the ground. The shear layer (which is 2mm long), used to connect the inextensible layer that surrounds it. Under the inextensible layer there are the deformable spokes, which bend accordingly to shape as the surface that the tire passes through. Lastly, the bub, which is the limit of the deformable spoke.



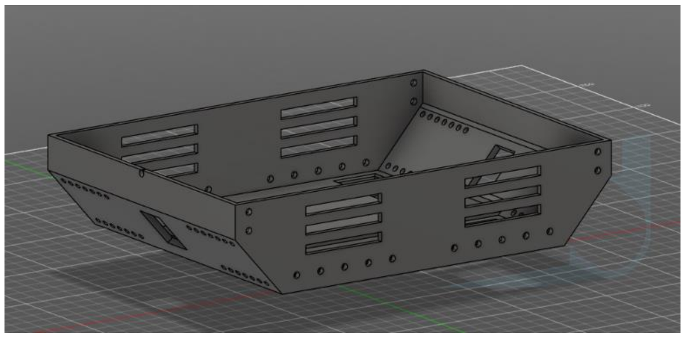
These air tires were decided best for the robot so two different airless tires were designed: a soft and a hard. Both were 3D printed with TPU filament, making the tire more flexible.

As the tires were tested the team found main characteristics that the tires must have to archive the highest performance, so a third tire was designed: a soft-hard airless tire that is sufficiently big and rough to stay hard on the ground, but soft when touching an irregular area.

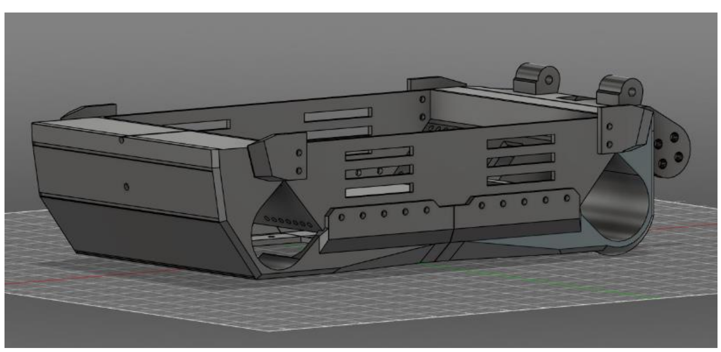
2.2.2 Chassis

Talkin about the chassis, it is divided into three distinct parts, each fulfilling vital functions for its performance in the field.

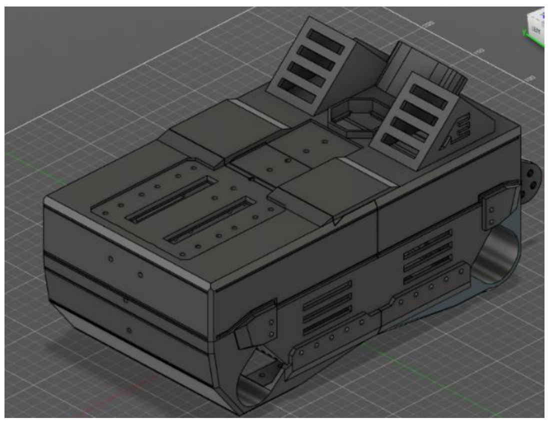
The first part of the chassis, referred to as “the main frame”, resembles a basket and plays an essential role in housing the electronic components necessary for the robot's operation. This section provides secure and accessible accommodation for the electronic components. Additionally, it serves as the assembly point for the robot's motors, ensuring a sturdy and stable structure.



The second part of the chassis, an additional piece designed to cover the motors, plays a crucial role in giving the robot a distinctive tank-like shape that sets it apart from its predecessors. This unique configuration provides protection for the motors and serves as a base to mount the flipper at the rear. Furthermore, this section of the chassis includes a space intended to accommodate the robot's batteries, such as the LiPo battery and two power banks, thus ensuring efficient weight distribution and continuous power supply during competition.



Finally, the third part of the chassis is the lid which has been meticulously designed with attention to detail. This lid features two fan couplings to maintain optimal temperature within the chassis during operation. Also, at the lid's rear, a cable outlet designed for Ethernet communication used in the robot was incorporated. Strategically placed holes in the lid allow for the integration of additional components, such as the arm, armrest, and power on/off switch for the robot, providing a functional and accessible interface for operators.



The decision to upgrade the chassis was driven by a need for enhanced performance and versatility in response to the evolving challenges of the Rescue Robotics Competition. The new chassis design offers several key improvements over its predecessor, primarily in its ability to provide better protection for the internal components and accommodate additional features crucial for effective operation in rescue scenarios. With a more robust structure and optimized layout, the new chassis facilitates easier integration of components, such as the flipper mechanism and arm, while also improving overall stability and maneuverability. These enhancements ensure that the robot is better equipped to navigate various terrains and obstacles encountered during rescue missions, ultimately increasing its efficiency and effectiveness in fulfilling its intended purpose.

2.2.3 Flipper

This part of the robot serves two primary purposes: firstly, in the event of the robot tipping over, the flipper mechanism aids in returning it to its upright position, and secondly, it facilitates climbing on high surfaces with the assistance of the arm.

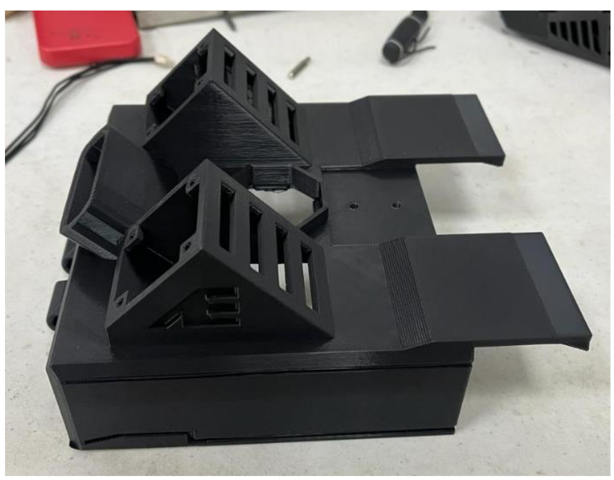
The flipper assembly comprises five distinct components. Beginning with the flipper grip, it is securely affixed and screwed into the designated holes beneath the chassis hinge. This grip houses the connection point for the Dynamixel AX-12A servo motors. The second and third components are integral to the rotational movement controlled by the servo motors. These pieces are interconnected in the center via three screws and at the top with a screw crossing the tire's axis. The fourth component is the tire crafted using TPU filament and employing a similar technology to that of the robot's tires. However, it features a specific pattern for increased rigidity, preventing deformation when under the force exerted by the servo motor. It's noteworthy that this piece incorporates three bearings internally to facilitate tire rotation. Lastly, the fifth component is an axle that acts as a linkage between the second and third pieces and the tire.

2.2.4 Cover

The cover has two purposes. The first is that it helps us to cover and protect the hardware inside the chassis. The second is the connection between the chassis and the arm.

In the front part there are several holes, where the shoulder of the arm is screwed. These holes are useful at the moment of moving the shoulder if we want the arm to be more forward or more backward depending on the test.

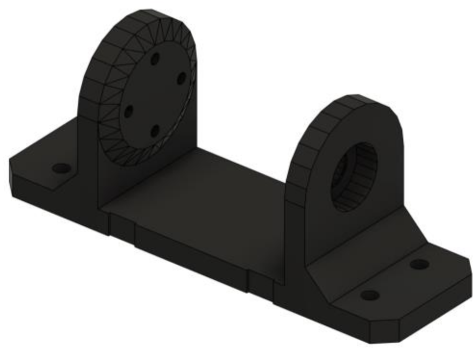
Continuing with the second part of the cover are our two mini fans that helps us to keep the hardware from overheating. Next to the mini fans is an outlet where the ethernet cable comes out of the chassis to be connected to the operator's computer.



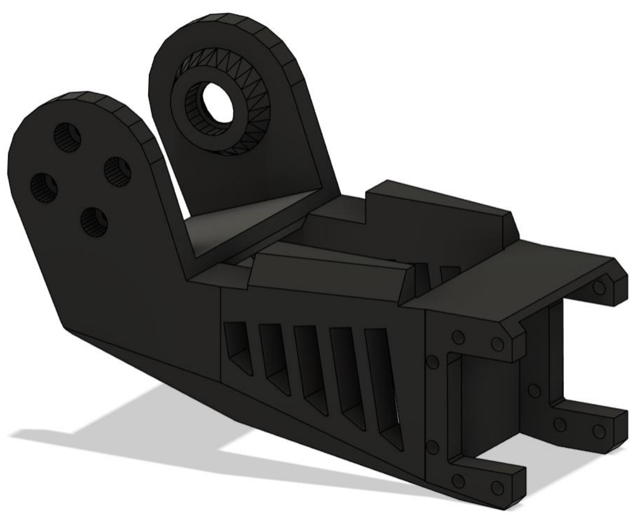
2.2.5 Arm

The arm is one of the most important parts of the robot because with it, we can do many things in the different arenas, it can move the camera to suit the operator, you can see backwards, you can zoom in to see things closer, among other things. The arm is divided into 4 servo motors:

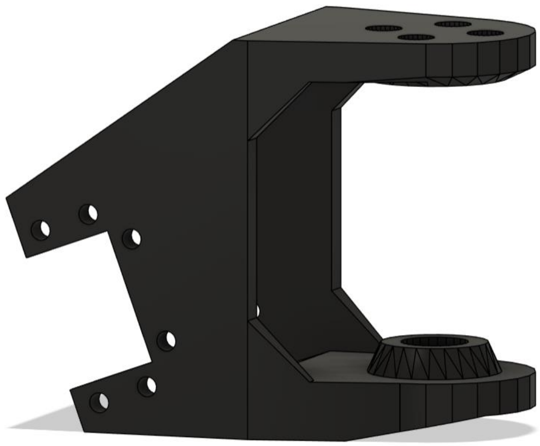
1. Shoulder servo: This servo is in charge of the forward and backward movement of the arm, a Dynamixel AX-18A is used to have more torque in this movement. This servo can be placed further forward or backward on the cover if desired.



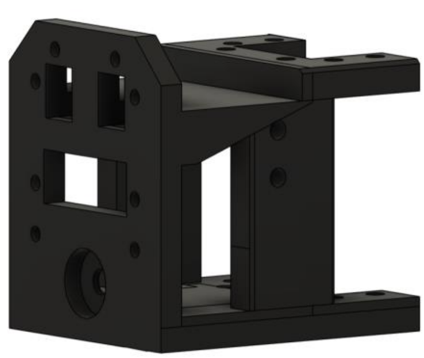
2. Servo elbow: this servo is in charge of moving the arm up or down, a Dynamixel AX-18A is used to have more torque in this movement.



3. Servo neck: this servo is in charge of the movement to the sides of the arm, called "snake movement", this helps us to have a better view of our surroundings. It is controlled by a Dynamixel Ax-12A.



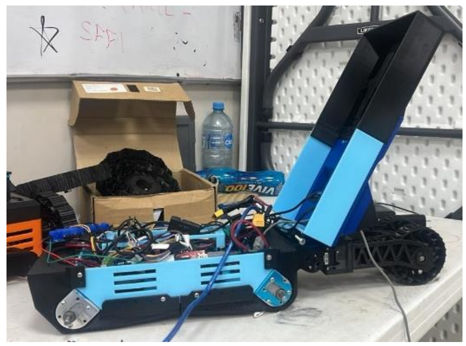
4. Servo claw: This last servo is responsible for opening and closing the claw by a mechanism in which there are two extensions that are screwed to a Dynamixel AX-12A which when closed makes the movement of a rail containing 4 claws, printed with TPU filament, using the technology of the tires molding to any surface that is taken, two of these claws have two extensions that will help us in the precision test.



Finally, the arm has a piece called "camera holder" which helps us to ensure that the camera is well positioned at the time of passing through the tracks.

2.3 Assembly of the robot

Being 3D printed we can modify if we want the robot to be light or not, since we modify the filling that each piece will have and among other properties of the robot. The only disadvantage we found was time, since 3D printing is not a fast way to get things done and that was an obstacle in the robot's production.



2.4 Software

2.4.1 Operative system

Raspberry PI OS: basic operating system for the Raspberry Pi 5

Windows 11: operating system used to host the user interface.

2.4.2 Programming languages

Python: The main programming language used in general. The user interface and Raspberry Pi's code were done with this language.

C++: Used on the Arduino Mega 2650 to control and get feedback from the robot's actuators and some sensors.

2.4.3 Arduino Mega's code

2.4.3.1 Purpose and Functionality

This code controls all the actuators and gives motion to the robot. There are 2 different types of motors that we use, servos and direct current motors. It has a doble serial communication between the Raspberry Pi 5 and the Dynamixel shield. One is for receiving instructions from the host and the other is for sending instructions to the servos. The motion of the dc motors is used to move the robot to one place to another, while the servos are used for support as in the arm or the flipper.

The code has a class that creates an instance for each dc motor. These instances have methods for stopping and moving the motor in a clockwise and counterclockwise direction.

The servos are managed differently.

2.4.3.2 Development and Creation

Firstly, the code uses the <DynamixelShield> library. This library is made for these servos and the TTL communication they need. These servos are controlled by giving the specific position that you want the servo to be. In the code is a method that the library includes, ".writeControlTableItem()" that helps to send any instruction the servo could admit, such as the mode, the speed, the position, among others. The servos code is based on this function.

In the other hand, there is a class for the dc motors, so each motor has its own instance. This class includes some methods such as moveC(), moveCC()and stop (), that are used to control them. By this way, the code is organized and effective.

2.4.4 Raspberry Pi code

2.4.4.1 Purpose and Functionality

The Raspberry Pi connects the user interface in the computer and the Arduino Mega 2560. It creates two servers, one that receives direct keyboard input and the other one that sends the information received from the Arduino and sensors. These two processes are simultaneously running, without interrupting each other.

The script that receives the keyboard input decodes the message and creates a new one with specific instructions for the Arduino Mega. This is how the instructions are given: the user presses a key, then the key is decoded by the raspberry pi, a new instruction is created according to the pressed key and finally this instruction is sent and received by the Arduino Mega.

The script that sends messages works like the previous one. First, the information is gathered by the sensors and the Arduino. Then it is received by the raspberry, where depending on the information an instruction could be activated. Finally, the information is interpreted by the interface.

2.4.4.2 Development and Creation

These scripts are developed from two libraries: “socket” and “serrialpy”.

Socket is used for receiving and sending information between the raspberry pi and the laptop from the user. This is done by ethernet connection, this is how the messages are sent.

Seralpy is like the previous library, with the exception that is used by the raspberry pi to send instructions to the Arduino Mega. This is done by UART communication where a module is used by Arduino, because of the need of having 2 serial monitors.

2.4.4.3 Future developments.

The improvements we plan to implement, from writing this document to the competition are:

-Implementation of more sensors like a thermal camera and a LIDAR,

- The recording of the instructions there were and the repetition of them.

-More actuators like a speaker and lights.

2.4.5 User Interface

2.4.5.1 Purpose and Functionality

The User Interface is divided into 3 different sections: the camera, the keyboard, and the message receiver.

It has the purpose of letting the user interact directly with the robot communicating to the Raspberry Pi via Internet/Ethernet.

In the keyboard input section, it reads the global input of your computer and sends it in a literal way to the Raspberry to interpret. The message receiver does pretty much the same as it receives the messages directly from the Arduino and are interpreted over there.

The camera section of the code is the most complex out of the three, since it reads the camera connected through the UART cable, it is here where different sensors are activated at will. The tracking, QR read, Hazmat, and Color are all programmed here and can be activated at the operators will using the number pad.

QR and Hazmat libraries were made using external libraries, for the camera feed, tracking and Color we used OpenCV library as a rudimentary interface for us to watch.

2.4.5.2 Development and Creation

This User Interface was created to avoid using the wireless keyboard connected to the Raspberry Pi, so that we could use internet/ethernet. But as we included the camera and realized we could not transmit it without a bad framerate and optimization issues on the Rasp, we opted for the Interface to do the work in terms of detection and data interpretation.

This interface is still in development and not yet finished, it was shaped to fulfill the immediate needs of the robot while we were developing it.

2.4.5.3 Future developments

Some of the improvements we plan to implement are as follows, keep note that among other things that are not listed here may also be added:

-Create a specific window for everything instead of the default camera feed that the OpenCV library offers.

-Have a specific library for keyboard detection that does not read the global keyboard (also that does not get detected by the antivirus).

-Add more sensor compatibility as the Raspberry Pi's code get implemented with new code

-Make messages more readable for the pilot.

3. Application

3.1 Set up and pack up our robot and operator station.

Inside the sturdy briefcase, we have taken additional measures to safeguard the robots' components. We added custom cut foam inserts that fit each piece, including the chassis, arm with servos and the 4 wheels. This ensures that they remain firmly in place and minimizes the risk of damage during transit.

The operator station, consists of a laptop, and additional equipment, is protected within a padded laptop bag or dedicated equipment case for enhanced safety. These specialized containers provide reinforced protection against impacts and help shield our equipment from potential damage. Inside, cable organizers and pouches keep cables, adapters, and peripherals such as the mouse neatly organized reducing the risk of tangling or damage during transit. To protect the second monitor, it is carefully wrapped in foam padding or bubble wrap.

3.2 Mission strategy

Our strategy for the competition is to capitalize as much as possible on the advantages of our robot. For example, on inclined terrain, we plan to use our flipper skillfully to facilitate climbing. Our robot's wheels are designed to adapt to a variety of surfaces, which will allow us to move with agility and efficiency. In addition, we have incorporated a unique feature: the ability to rotate the camera using a mechanism in the neck of the robot, giving us a versatile view without having to reorient the entire robot.

3.3 Experiments and testing

In the following link you can find the different tests that we made with our robot:

https://drive.google.com/file/d/17TN-m_0nJqBosby3ZLb2A-jcgEaFptKp/view?usp=sharing

3.4 How are the strengths of our team relevant to applications in the field?

As we said in the beginning of this TDM, each member of the team, including the mentors, had a specific role for the build of the robot, this was useful since the communication of the team made the process practical, and at the same time we were effective in the robot creation. Here is explained the role of each member:

Odin is the leader of the team. He makes sure that the team doesn't have any missing material so everyone can work. Also, he designed the chassis, the top of the chassis and different aesthetic things so the robot looks pretty.

Eduardo is one of our three programmers; he was in charge of programming the movement of the robot. His specific developments are the movement of the Dynamixel AX-18A motors of the arm and the movement of the DC motors of the wheels.

Sergio is the second programmer of the team. He was in charge of the programming of the sensors, including tracking, hazmat, QR, quality, and the other ones.

Aaron is the third programmer of the team. He was in charge of the human-robot communication and the interface.

Moisés is one of our designers. He was in charge of designing the arm and the claw, designing the clothes of the team, and getting the sponsors. Also, he manages the social media accounts of the team.

Alberto is one of our designers. He oversaw the design of the brain which is inside of the chasis and the creation of the circuits that we used to practice with the robot.

4. Conclusion

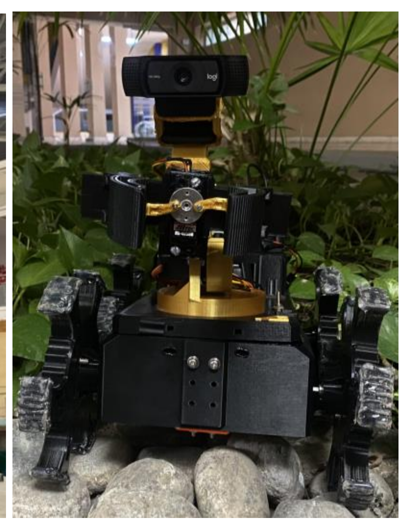
Over the past year, we have experienced a constant evolution, where continuous learning has been our best partner.

We have been committed to adapting to our budget in a creative way, always looking for innovative ways to improve our capabilities without compromising quality. Despite our achievements, we recognize that there is still plenty of opportunity for growth and improvement. We have set ourselves the challenge of exceeding our own expectations, and for the upcoming pre-RoboCup TDM, we are determined to present to you a robot that is significantly improved over last year. Although we won the RoboCup in 2023 with our previous robot, we are committed to exceed that achievement. We are focused on merging our accumulated learning with innovative ideas to build a robot that is even more competent, efficient and versatile on the playing field. With this promise in mind, we are confident that we can reach new heights and defend our title with exceptional performance in the next competition.

4.1 Our robots over the last year

We have created several versions of our robot over the last two years:

Starting from the robot which led us to win the first place in the TMR 2023 (Mexican Robotics Tournament).



Then the robot who which led us to win the RoboCup 2023 last year.

Finally the robot that we are going to present in the TMR 2024 and that we are going to improve for the RoboCup 2024.

5. Appendix

5.1 Table outlining components and estimated cost of our robot.

Components	Unit Price (USD)	Cuantity	Total Price (USD)
Dynamixel AX-18A	$ 109.90	2	$ 219.8
Dynamixel AX-12	$ 49.90	4	$ 199.6
DC Motors Pololu 37D 131:1	$ 30.31	4	$ 121.24
Kilograms of PLA Filament	$ 30.21	10 kg	$ 302.1
Kilograms of TPU Filament	$ 30.21	1 kg	$ 30.21
Driver IBT2 43 A	$ 12.91	4	$ 51.64
Raspberry Pi 5	$ 64.89	1	$ 64.89
Arduino Mega	$ 45.46	1	$ 45.46
Dynamixel Shield	$ 23.90	1	$ 23.90
Screw Set	$ 33.77	1	$ 33.77
Total cost			$ 1091.81 usd