Presentación

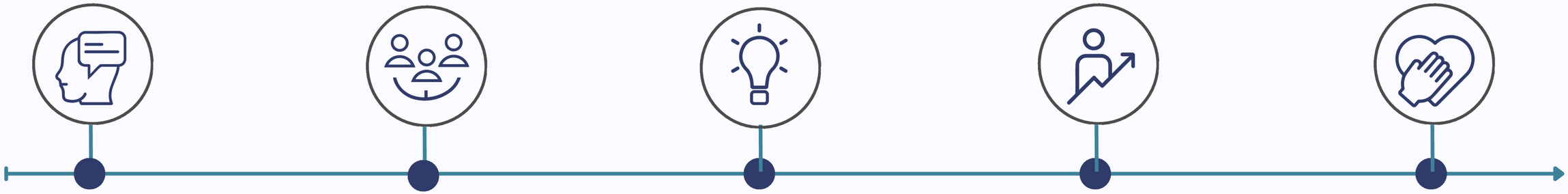
Proyecto Heracross

Índice de **contenidos**

Reintroducción

A continuación procederemos rápidamente a recapitular lo visto en la presentación anterior

Puntos tratados anteriormente



Objetivos

Crear un robot que sea capaz de agarrar una pelota de pingpong y llevarlo a un lugar, aprendiendo como utilizar linux y python ev3

Organización

Se repartieron los roles entre los integrantes de el grupo

Planificación

Se planificó utilizando una carta Gantt

Estimación de costos

Se organizó y estimó los costos que se harían en la creación del proyecto llegando a la cantidad total aproximada de $9.000.000

Progreso

Logramos avanzar con el modelo del robot y con su garra

Análisis y diseño

Requerimientos funcionales

El robot tiene una conexión estable con computadora y se comunica mediante la interfaz gráfica a través de Wi-Fi.
El robot es capaz de moverse hacia adelante, hacia a atrás y también girar gracias al sensor de giro, además de una pinza que puede tomar una pelota de ping pong.

Requerimientos no funcionales

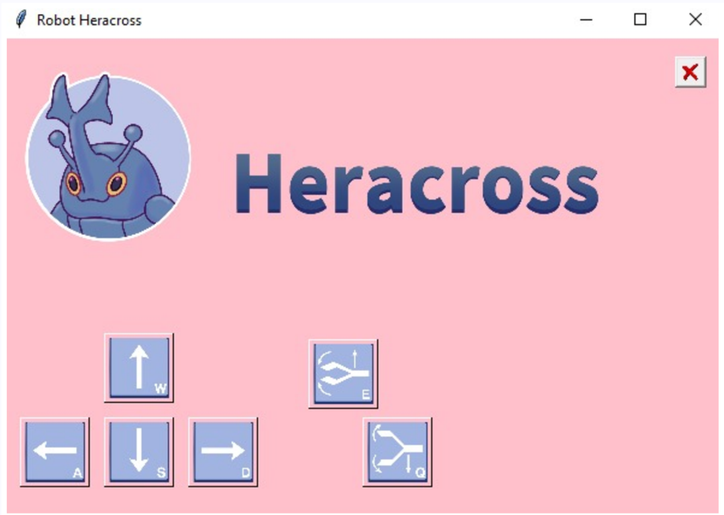
Interfaz gráfica fácil de entender y usar, con los botones de movimiento, también cuenta con botones para usar la garra mecánica.
El robot cuenta con motores que permiten realizar sus movimientos además de una batería, la cual es recargable para su posterior uso.
El proyecto debe contar con manual de usuario, el cual indicará el buen uso del robot de manera clara y entendible.

Arquitectura

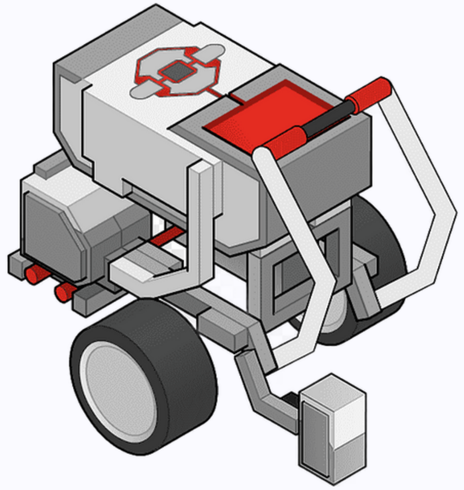
cliente

internet

servidor

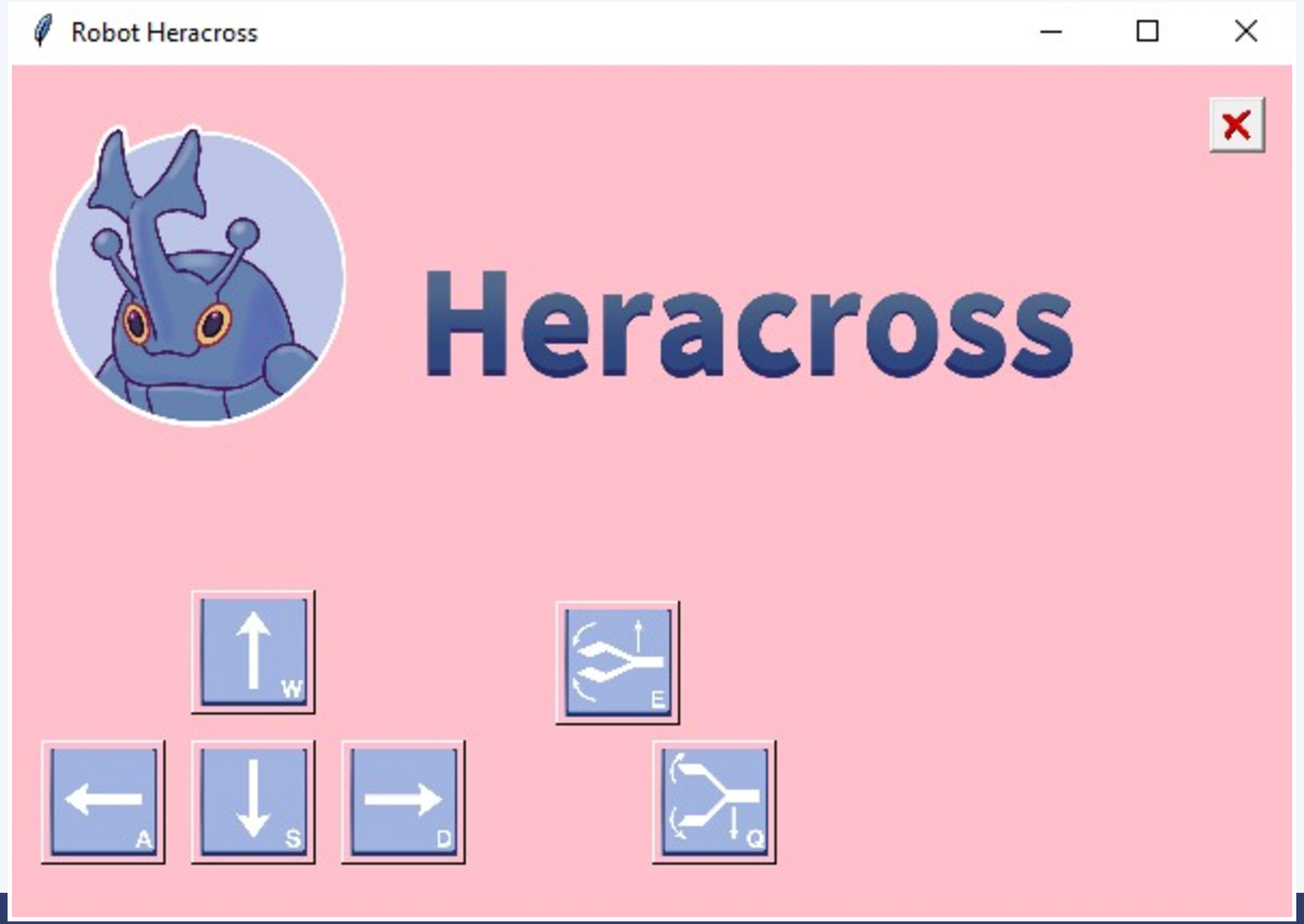


interfaz grafica

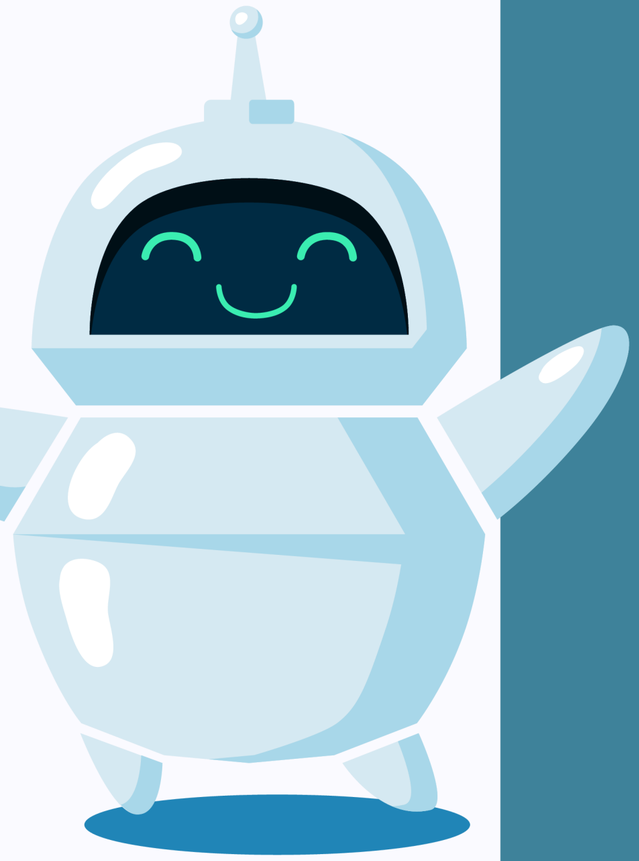


robot

Interfaz gráfica



Implementación



01 Fundamentos de los movimientos

- La fuerza mínima que debe ejercer el robot para levantar la pelota:

$$F = (2,7 * 10^{-3})(kg) * 9,8\,(m/s^2)$$

$$F = 0,2646\,(N)$$

- El trabajo mecánico mínimo que realiza el robot al levantar la pelota:

$$W = 0,2646\,(N) * 0,1\,(m)$$

$$W = 2646 * 10^{-5}\,(J)$$

02 Descripción de los programas

El código es simple y ordenado, dividido en dos archivos, el primero encargado de enviar información al servidor usando Sockets mediante una interfaz de Tkinter, y el segundo recibe información del servidor usando Sockets, y le comanda al robot usando Pybricks

03 Diagramas

Motores Grandes → Movimiento del robot

Motor Pequeño → Movimiento de Garra

Descripción de los programas

```python
#!/usr/bin/env pybricks-micropython

import tkinter as tk
import socket

def send_command(command):
    ev3_ip = '192.168.74.196' # Check each time
    port = 63383 # Get-NetTCPConnection | Where-Object { $_.State -eq 'Listen' }
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.sendto(command.encode(), (ev3_ip, port))

def key_press_w(event):
    send_command("W")
    WButton.config(bg="red")

def key_press_a(event):
    send_command("A")
    AButton.config(bg="red")

def key_press_s(event):
    send_command("S")
    SButton.config(bg="red")

def key_press_d(event):
    send_command("D")
    DButton.config(bg="red")

def key_press_e(event):
    send_command("E")
    EButton.config(bg="red")

def key_press_q(event):
    send_command("Q")
    QButton.config(bg="red")

def key_release_w(event):
    send_command("stop")
    WButton.config(bg="white")

def key_release_a(event):
    send_command("stop")
    AButton.config(bg="white")

def key_release_s(event):
    send_command("stop")
    SButton.config(bg="white")

def key_release_d(event):
    send_command("stop")
    DButton.config(bg="white")
```

```
    DButton.config(bg="white")

def key_release_e(event):
    send_command("stopClaw")
    EButton.config(bg="white")

def key_release_q(event):
    send_command("stopClaw")
    QButton.config(bg="white")

root = tk.Tk()
root.title("EV3 Control")
root.geometry("400x400")

root.bind("<KeyPress-w>", key_press_w)
root.bind("<KeyRelease-w>", key_release_w)

root.bind("<KeyPress-a>", key_press_a)
root.bind("<KeyRelease-a>", key_release_a)

root.bind("<KeyPress-s>", key_press_s)
root.bind("<KeyRelease-s>", key_release_s)

root.bind("<KeyPress-d>", key_press_d)
root.bind("<KeyRelease-d>", key_release_d)

root.bind("<KeyPress-e>", key_press_e)
root.bind("<KeyRelease-e>", key_release_e)

root.bind("<KeyPress-q>", key_press_q)
root.bind("<KeyRelease-q>", key_release_q)

frame = tk.Frame(root)
frame.pack()

WButton = tk.Button(frame, text="W", width= 5, height= 2)
WButton.grid(row=0, column=1)

AButton = tk.Button(frame, text="A", width= 5, height= 2)
AButton.grid(row=1, column=0)

SButton = tk.Button(frame, text="S", width= 5, height= 2)
SButton.grid(row=1, column=1)

DButton = tk.Button(frame, text="D", width= 5, height= 2)
DButton.grid(row=1, column=2)

EButton = tk.Button(frame, text="E", width= 5, height= 2)
EButton.grid(row=0, column=2)

QButton = tk.Button(frame, text="Q", width = 5, height=2)
QButton.grid(row=0, column=0)

root.mainloop()
```

Servidor

```python
#!/usr/bin/env pybricks-micropython

import socket

from pybricks.ev3devices import Motor
from pybricks.robotics import DriveBase
from pybricks.parameters import Port

leftMotor = Motor(Port.D)
rightMotor = Motor(Port.B)
clawMotor = Motor(Port.C)

flagGrabbed = False

robot = DriveBase(leftMotor, rightMotor, wheel_diameter=55.5, axle_track=104)

ip_address = '0.0.0.0'
port = 63383
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((ip_address, port))
```

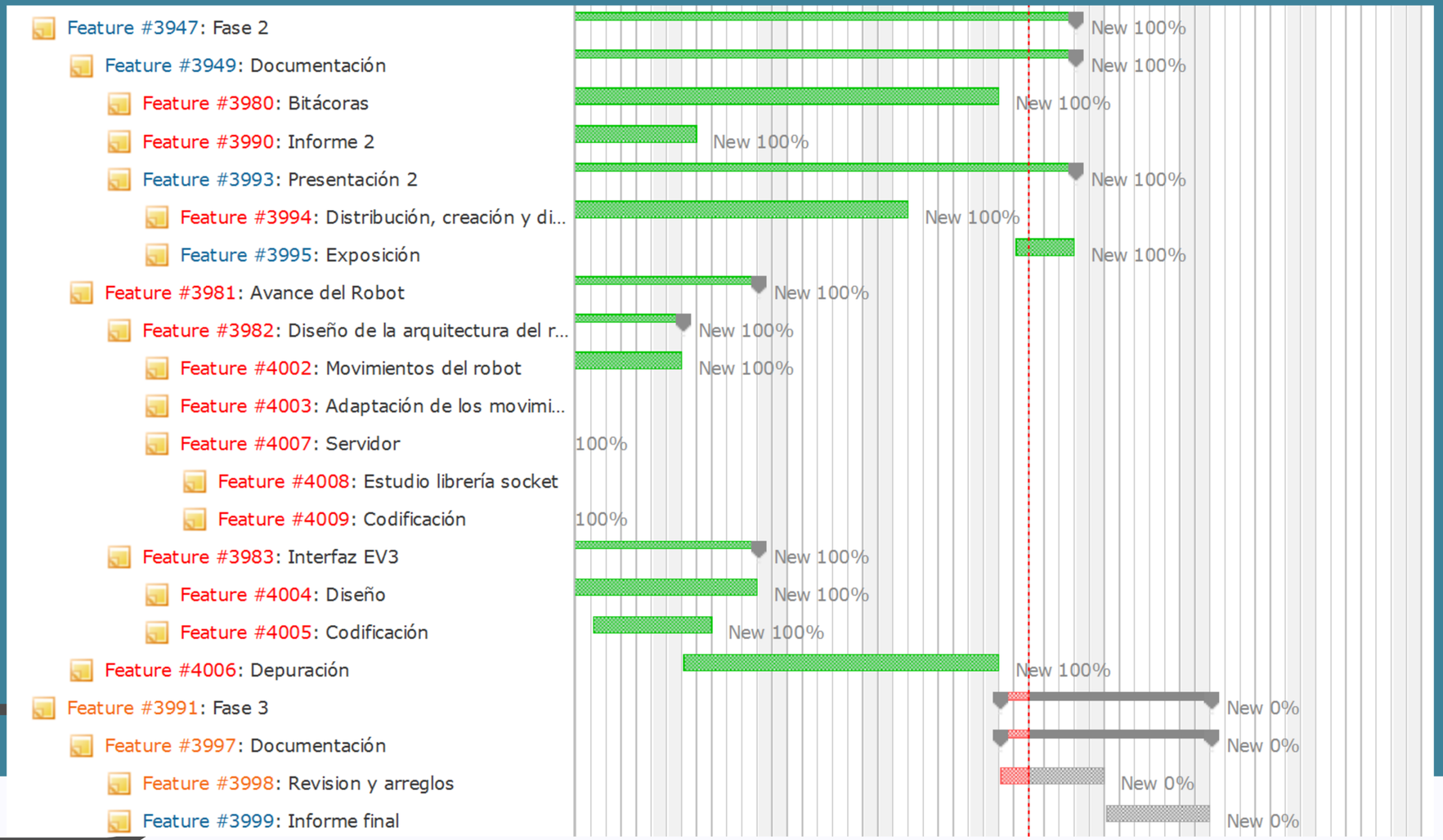
```python

print("Listening...")

while True:
    data, addr = sock.recvfrom(1024)
    command = data.decode()

    if command == 'W':
        robot.drive(1000,0)
    elif command == 'A':
        robot.drive(0,-120)
    elif command == 'S':
        robot.drive(-1000,0)
    elif command == 'D':
        robot.drive(0,120)
    elif command == 'E':
        clawMotor.run(-500)
    elif command == 'Q':
        clawMotor.run(500)
    elif command == 'stop':
        robot.stop()
    elif command == 'stopClaw':
        clawMotor.stop()
```

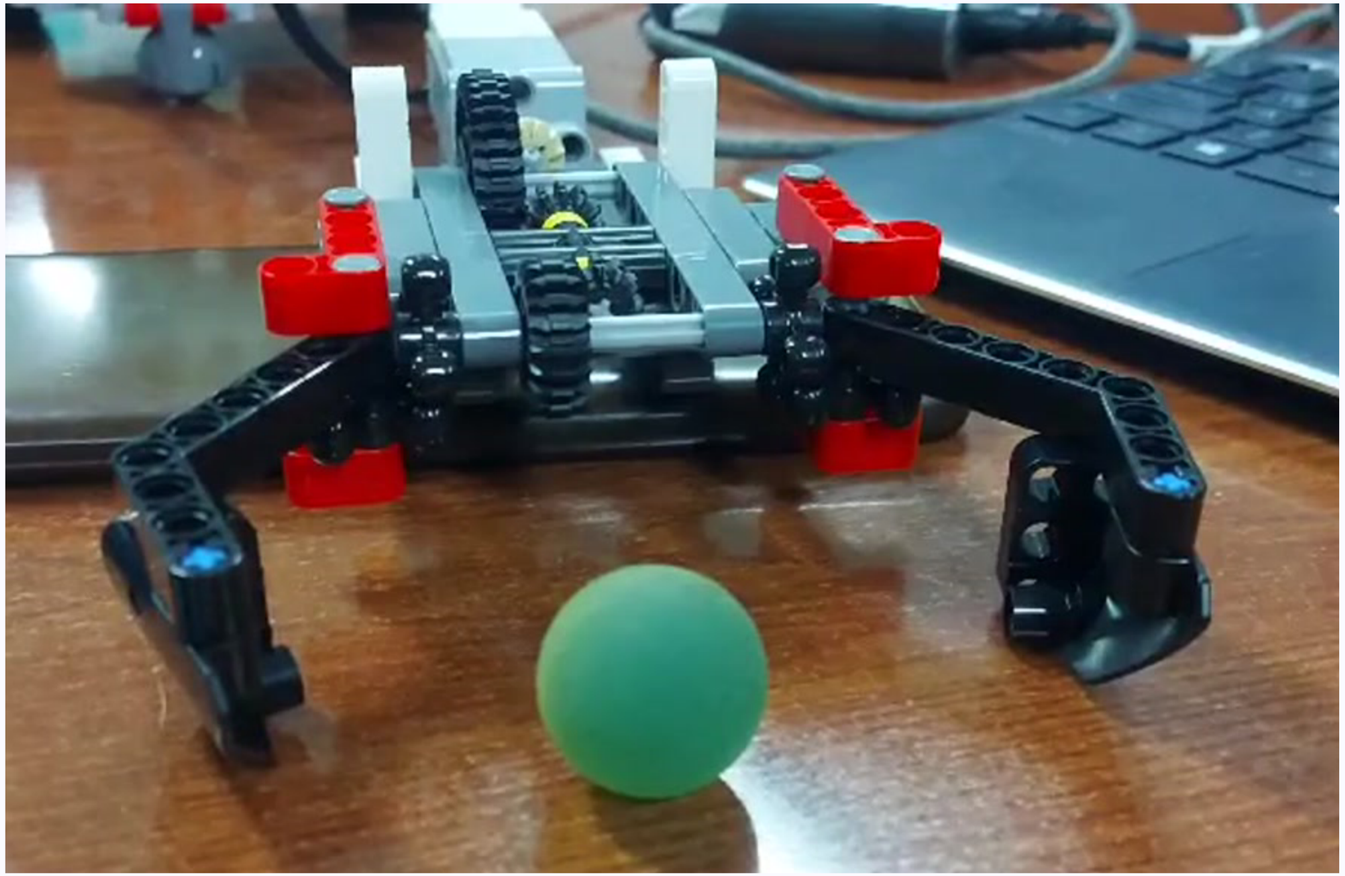
Carta Gantt


Feature #3947: Fase 2
Feature #3949: Documentación
Feature #3980: Bitácoras
Feature #3990: Informe 2
Feature #3993: Presentación 2
Feature #3994: Distribución, creación y di...
Feature #3995: Exposición
Feature #3981: Avance del Robot
Feature #3982: Diseño de la arquitectura del r...
Feature #4002: Movimientos del robot
Feature #4003: Adaptación de los movimi...
Feature #4007: Servidor
Feature #4008: Estudio librería socket
Feature #4009: Codificación
Feature #3983: Interfaz EV3
Feature #4004: Diseño
Feature #4005: Codificación
Feature #4006: Depuración
Feature #3991: Fase 3
Feature #3997: Documentación
Feature #3998: Revision y arreglos
Feature #3999: Informe final
New 100%
New 100%
New 100%
New 100%
New 100%
New 100%
New 100%
New 100%
New 100%
New 100%
100%
100%
New 100%
New 100%
New 100%
New 100%
New 0%
New 0%
New 0%
New 0%

Problemas y soluciones

Diseño de los iconos de los botones de la interfaz gráfica.	Rediseño de los iconos por un acuerdo de gusto en común por los integrantes del equipo. ✅
El uso de Tkinter y sockets combinados.	Comprensión y mayor trabajo en el uso de estos mismo. ✅
Codificación en una función, para la activación del motor.	Se mejoro cambiando la opción a un cierto límite de tiempo. ✅

Resultados



Conclusiones

Durante la segunda mitad del avance del proyecto, desarrollamos el diseño final del robot, su interfaz gráfica y pruebas con código, durante este proceso hemos notado una mejor confianza para trabajar con nuestro equipo como también una buena distribución del tiempo que se nos fue dando cada semana que nos sirvió bastante, también hay que recalcar que hasta ahora hemos logrado el proyecto con éxito.