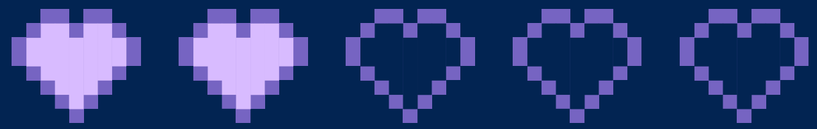
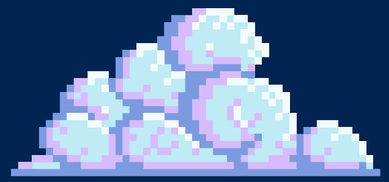
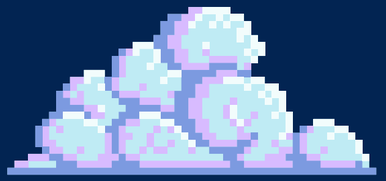
UNIVERSIDAD DE TARAPACÁ
Universidad del Estado

Ingeniería
Computación e Informática





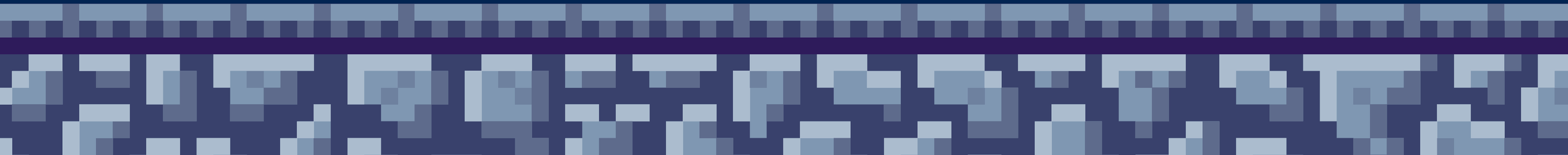


ALLIGATOR 3000

PROYECTO I

PROFESOR: HUMBERTO URRUTIA

START

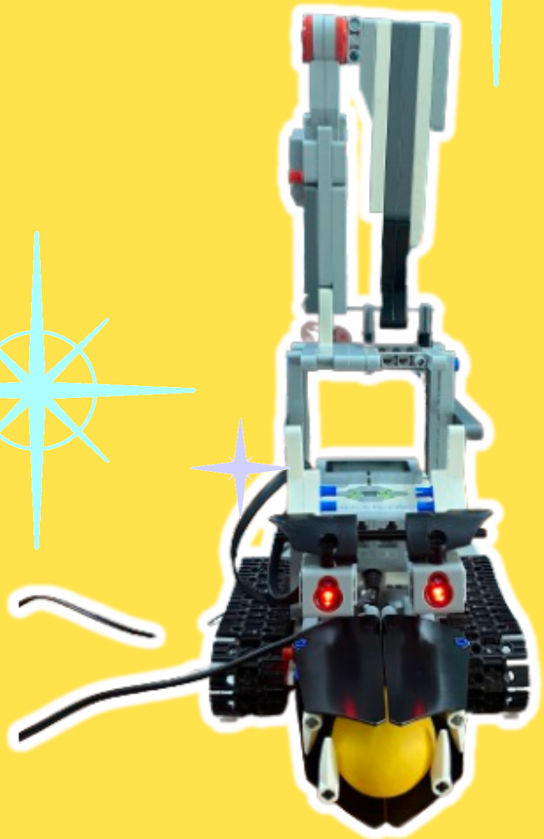


01

INTRODUCCIÓN

OBJETIVOS

La misión primordial consiste en la concepción y manufactura de un robot EV3, dotado con la capacidad de golpear una pelota mediante una interfaz gráfica meticulosamente confeccionada a través del lenguaje de programación Python.



02. ENTREGABLES

- Manual de usuario
- Documentación subida a Redmine:
  - Wiki
  - Bitácoras
  - Carta Gantt
  - Informes
  - Presentaciones
- Robot “Alligator 3000”: El robot armado y funcional.

03. RESTRICCIONES

- Para realizar la conexión del robot con el ordenador es necesaria una red wifi estable.
- El producto del robot no será más que un prototipo requerimientos del proyecto.
- El robot debe moverse y golpear una pelota.

04. ROLES

BRIAN LÓPEZ

- Jefe de grupo

Involucrado:

- Programación y diseño.

KAREN MAMANI

- Ensamblador

ANGIE MARTINEZ

- Documentador

Involucrado:

- Ensamblador.

POLETTE MONTT

- Programador

Involucrado:

- Documentador.

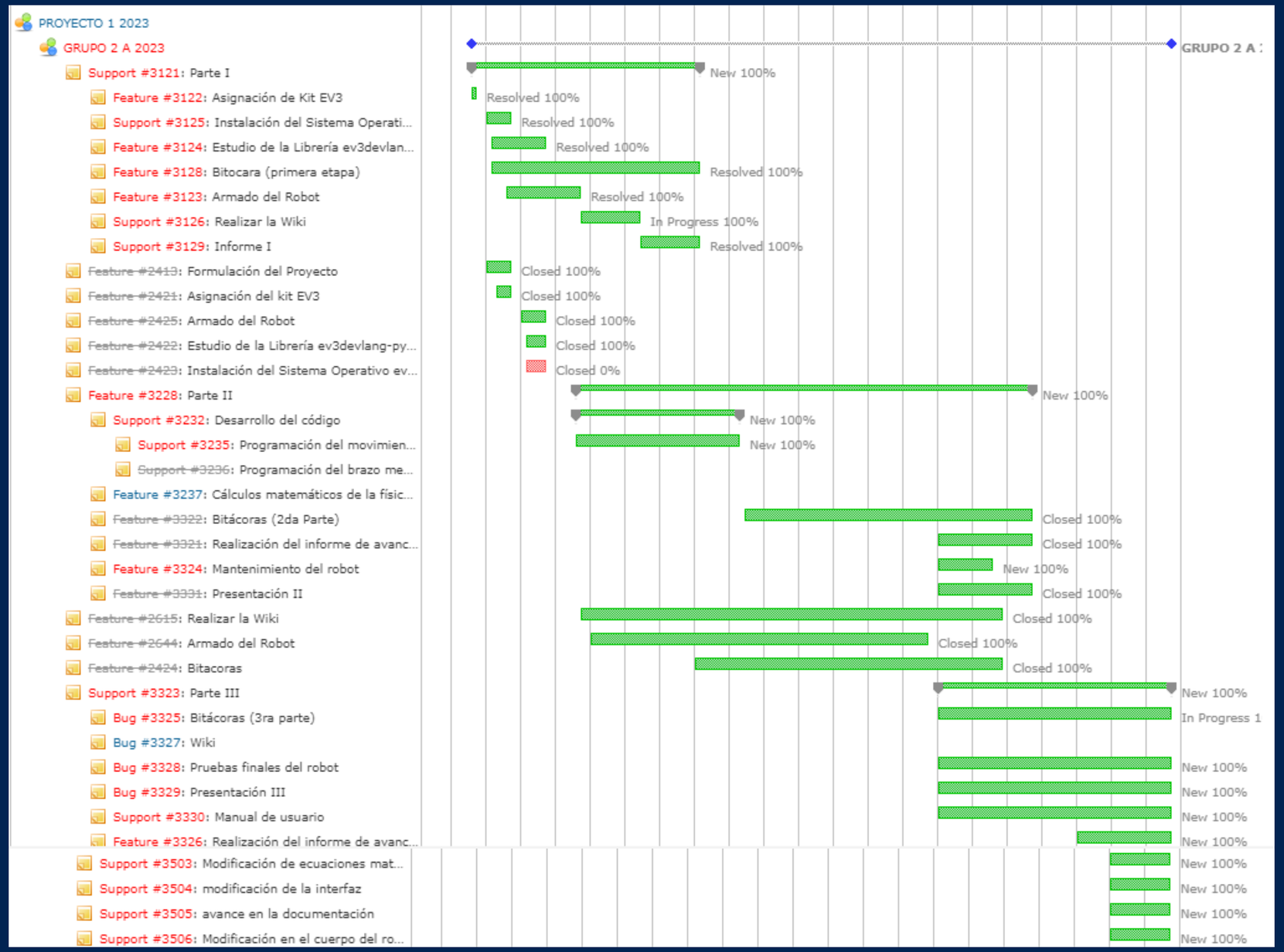
BASTIAN SUCSO

- Diseñador

Involucrado:

- Programación y documentación.

05. CARTA GANTT



06. ESTIMACIÓN DE COSTOS

Nombre del producto	Precio	Cantidad	Total
EV3 LEGO MINDSTORMS	$1,299,000	1	$1,299,000
Lego Mindstorms Ev3 L-servo Motor	$112,500	3	$337,500
Toshiba tecra z40-c	$349,000	2	$698,000
Tarjeta MicroSD 64GB	$3,990	1	$3,990
Tp-link Ac600	$13,495	1	$13,495
Asus VivoBook X409FB	$491,720	1	$491,720

Nombre del producto	Precio	Cantidad	Total
Tablet Samsung Galaxy Tab S4	$499,990	1	$499,990
Canva (Versión Gratuita)	Gratuito	1	$0
Visual code Studio	Gratuito	1	$0
Librería EV3	Gratuito	1	$0
Total			$3,343,695

Costos de hardware y software

Costos de gestión:

Integrante	Valor por hora	Horas Trabajadas	Valor total
Angie Martinez	$10,000	53	$530,000
Bastian Sucso	$11,000	52,75	$580,250
Brian Lopez	$12,000	60,3	$723,600
Polette Montt	$10,000	50,75	$507,500
Karel Mamani	$10,000	50,75	$507,500

Costos totales::

Nombre	Costo total
Costos de Software y Hardware	$3.343.695
Costos de Gestión	$2,848,850
Costo total proyecto	$6,192,545

07. ANÁLISIS Y DISEÑO

REQUERIMIENTOS FUNCIONALES:

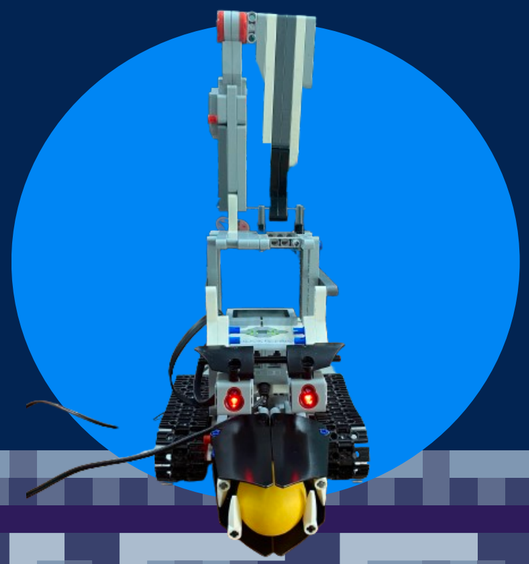
- El robot debe tener la capacidad de moverse.
- Los usuarios controlan el robot a través de una interfaz gráfica.
- El robot golpea una pelota.
- Se espera que el robot exhiba la habilidad de impactar una pelota.

REQUERIMIENTOS NO FUNCIONALES:

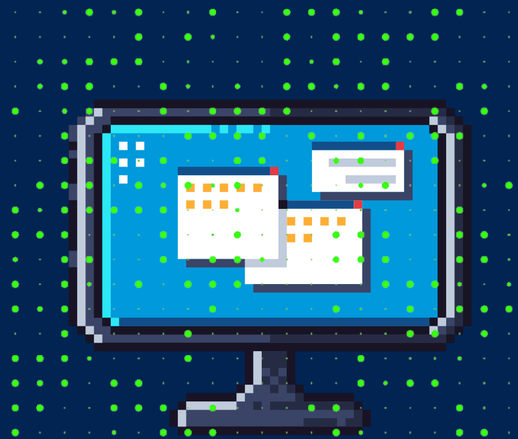
- Los movimientos del robot se logran mediante la interfaz gráfica.
- Los movimientos y la interfaz gráfica es programada en Python.
- La simulación del Swing de golf se realiza mediante el uso de las piezas del LEGO EV3.

ARQUITECTURA

Cuerpo del robot.



Programación

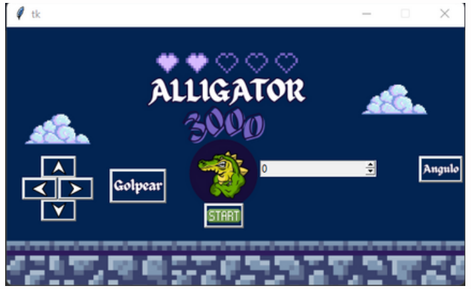


CONEXIÓN DEL ROBOT

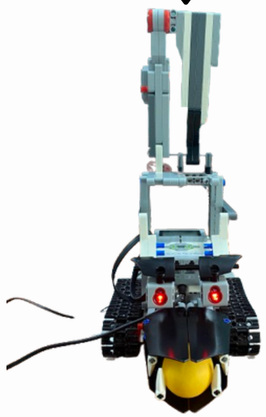
4 Usuario

1 Wifi

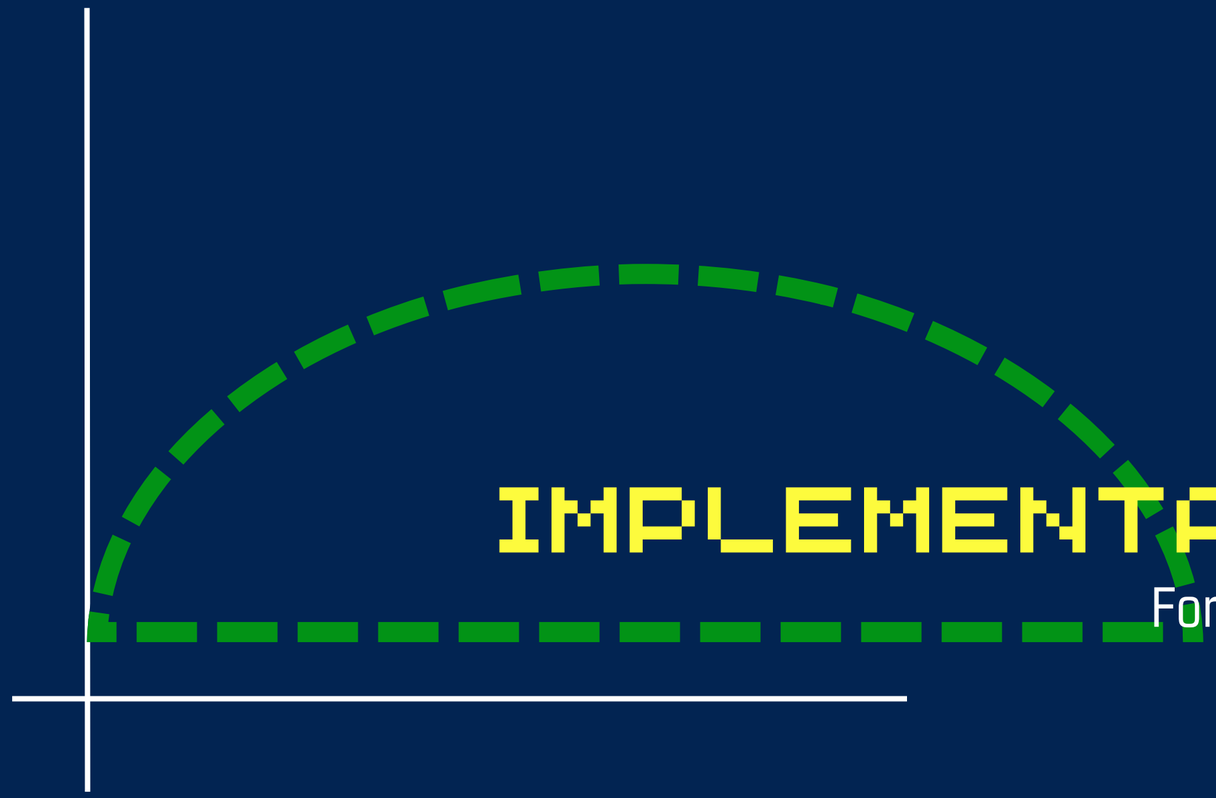
2 Sevidor



3 Interfaz



5 Robot



03

IMPLEMENTACIÓN

Formulas físicas

FUNDAMENTOS DEL MOVIMIENTO PARABÓLICO Y DE VELOCIDAD TANGENCIAL.

3 posibles situaciones a suceder, de acuerdo a 3 tipos de ángulos de inclinación del brazo mecánico (0º, 45º y 90º)

Datos ángulo 0º:

X0=0 m

Y0=0,22 m

Xf= 0,56 m

Yf=0 m

t = 0,5 s

Datos ángulo 45º:

X0=0 m

Y0=0,22 m

Xf= 0,71 m

Yf=0 m

t = 0,6 s

Datos ángulo 90º:

X0=0 m

Y0=0,22 m

Xf= 0,76 m

Yf=0 m

t = 0,65 s

Velocidad Tangencial para el ángulo de inclinación del brazo en 0º:

Fórmula:

$vt = r \cdot \omega$

Datos:

r = 0,134 m

ω = 160 RPM = 16,8 rad/s

Resultado:

vt = 2,25 (m/s)

Calcular el ángulo de inclinación con el brazo en inclinación en 0º

Fórmula:

$Xf = X0 + V0 \cos\theta\, t$

Datos:

Xf = 0,56 m

X0 = 0 m

V0 = vt = 2,25 m/s

t = 0,5 s

Resultado:

θ = 60,26º

Sacar la velocidad inicial del brazo con inclinación

Fórmula:

$Xf = X0 + V0 \cos\theta\, t$

Con 45º:

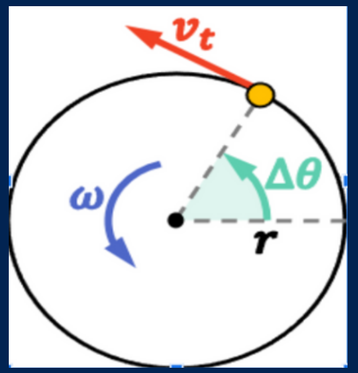
Resultado:

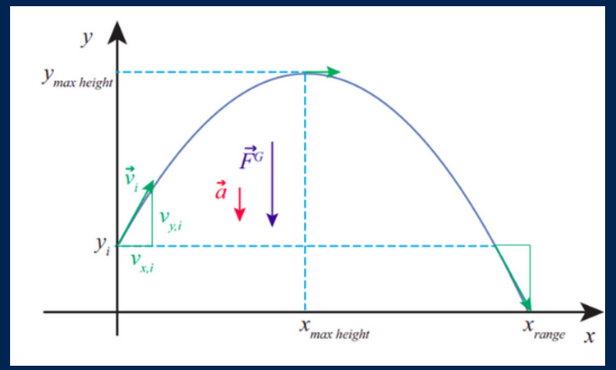
V0 = 2,37 m/s

Con 90º:

Resultado:

V0 = 2,39 m/s





09

CÓDIGO

Server

```python
import socket
from Function import *
s = socket.socket()
print("Socket creado")
port = 8080
s.bind(('', port))
print("El socket se creo con puerto: {}".format(port))
s.listen(5)
print("EL socket is listening....")
connect, addr = s.accept()
print("Se conecto a  {}".format(addr))
while True:
    rawByte = connect.recv(1)
    char = rawByte.decode('utf-8')
    if (char == 'w'):
        Adelante()
    if (char == 's'):
        Atras()
    if (char == 'd'):
        Derecha()
    if (char == 'a'):
        Izquierda()
    if (char == 'p'):
        Brazo()
    if (char =='i'):
        Inclinacion0()
    if (char == 'o'):
        Inclinacion45()
    if (char == 'u'):
        Inclinacion90()
    if (char == ' '):
        Parar()
    if (char == 'q'):
        print("Terminada la sesion...")
        break
```

Funciones

```
from ev3dev2.motor import LargeMotor, MediumMotor, OUTPUT_A, OUTPUT_B,
OUTPUT_C, OUTPUT_D, SpeedPercent, MoveTank
import time
from ev3dev2.sound import Sound

sound = Sound()

ruedas = MoveTank(OUTPUT_B,OUTPUT_C)
brazo= LargeMotor(OUTPUT_D)

def Adelante():
    ruedas.on(SpeedPercent(100),SpeedPercent(100))

def Atras():
    ruedas.on(SpeedPercent(-70),SpeedPercent(-70))

def Izquierda():
    ruedas.on(SpeedPercent(70),SpeedPercent(-70))

def Derecha():
    ruedas.on(SpeedPercent(-70),SpeedPercent(70))

def Brazo():
    brazo.on_for_rotations(100,-1)
def Inclinacion0():
    brazo.on_for_degrees(10, -0,brake=True)
def Inclinacion45():
    brazo.on_for_degrees(10, 45, brake=True)
```

```
def Inclinacion45():
    brazo.on_for_degrees(10, 45, brake=True)
def Inclinacion90():
    brazo.on_for_degrees(10, 90, brake=True)
def Parar():
    ruedas.stop()
```

Interfaz

```
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
import tkinter as tk
import socket
import time
import sys

#Funciones de los botones

def getAddres():
    w_ip = Toplevel()
    w_ip.geometry("300x169")
    imagen1 = PhotoImage(file = "imagen2.png")
    labelimagen1 = Label(w_ip, image = imagen1)
    labelimagen1.pack()

    dato = StringVar(w_ip)
    w_ip.title("Configurar Ip")
    ip = ttk.Entry(w_ip,textvariable=dato).place(x=8,y=55)
    boton_aplicar = tk.PhotoImage(file="aplicar.png")
                            button        =        Button(w_ip,image
boton_aplicar,command=lambda:[conectar(dato.get(),port),
w_ip.destroy()]).place(x=195,y=50)
    w_ip.mainloop()

def conectar(addres,port):
    try:
        clientSocket.connect((addres, port))
        messagebox.showinfo("Mensaje Servido", "Cliente conectado al robot:
{0} : {1}".format(ipAddress, port))
    except (socket.error, socket.timeout) as e:
         messagebox.showerror("Error", f"No se pudo conectar al robot en la
dirección {addres} : {port}")

def forward():
    clientSocket.send(bytes([ord('w')]))
```

```
def backward():
    clientSocket.send(bytes([ord('s')]))

def right():
    clientSocket.send(bytes([ord('d')]))

def left():
    clientSocket.send(bytes([ord('a')]))

def hit():
    clientSocket.send(bytes([ord('p')]))

def end_session():
    clientSocket.send(bytes([ord('q')]))

def soltar(event):
    clientSocket.send(bytes([ord(' ')]))

def Angulo():
    if(datos.get()=="0"):
        clientSocket.send(bytes([ord('i')]))
    elif(datos.get() == "45"):
        clientSocket.send(bytes([ord('o')]))
    elif(datos.get() == "90"):
        clientSocket.send(bytes([ord('u')]))

window = Tk()
window.geometry("533x300")
```

Interfaz

```
imagen = PhotoImage(file = "imagen1.png")
labelimagen = Label(window, image = imagen)
labelimagen.pack()
window.resizable(False, False)

#Botones en la ventana

boton_arri = tk.PhotoImage(file="arriba.png")
button_1 = Button(image= boton_arri,repeatdelay= 50, repeatinterval=
50,command=forward)
button_1.place(x=40,y=150)
```

```
boton_abajo = tk.PhotoImage(file="abajo.png")
button_2 = Button(image= boton_abajo,repeatdelay= 50, repeatinterval= 50,
command=backward)
button_2.place(x=40,y=200)

boton_derech = tk.PhotoImage(file="derecha.png")
button_3 = Button(image= boton_derech,repeatdelay= 50, repeatinterval= 50,
command=right)
button_3.place(x=60,y=175)

boton_izq = tk.PhotoImage(file="izquierda.png")
button_4 = Button(image= boton_izq,repeatdelay= 50, repeatinterval= 50,
command=left)
button_4.place(x=18,y=175)

boton_golpear = tk.PhotoImage(file="golpear.png")
button_5 = Button(window, image = boton_golpear,
command=hit).place(x=120,y=165)
```

```
datos = StringVar(window)
angulo = Spinbox(window, state ="readonly", values = ("0", "45", "90"),
textvariable=datos).place(x=295,y=155)
boton_angulo = tk.PhotoImage(file="angulo.png")
button_6 = Button(window, text = "ingresar", command=Angulo, image =
boton_angulo).place(x=480,y=150)
```

```
button_1.bind('<ButtonRelease-1>',soltar)
button_2.bind('<ButtonRelease-1>',soltar)
button_3.bind('<ButtonRelease-1>',soltar)
button_4.bind('<ButtonRelease-1>',soltar)

boton_conec = tk.PhotoImage(file="conectar.png")
button_connect = Button(window, image = boton_conec, command=lambda:
getAddres()).place(x=230,y=205)
ipAddress = "192.168.84.197"


# Read the command line argument for the IP address of the server
if len(sys.argv) > 2:
```

Interfaz

```
    print("Usage: client-laptop.py [IP-addr-of-robot]")
    sys.exit(1)
elif len(sys.argv) == 2:
    ipAddress = sys.argv[1]
    print("Using specified IP address: {}".format(ipAddress))
else:
    print("Using default IP address: {}".format(ipAddress))


clientSocket = socket.socket()
port = 8080
window.mainloop()
```

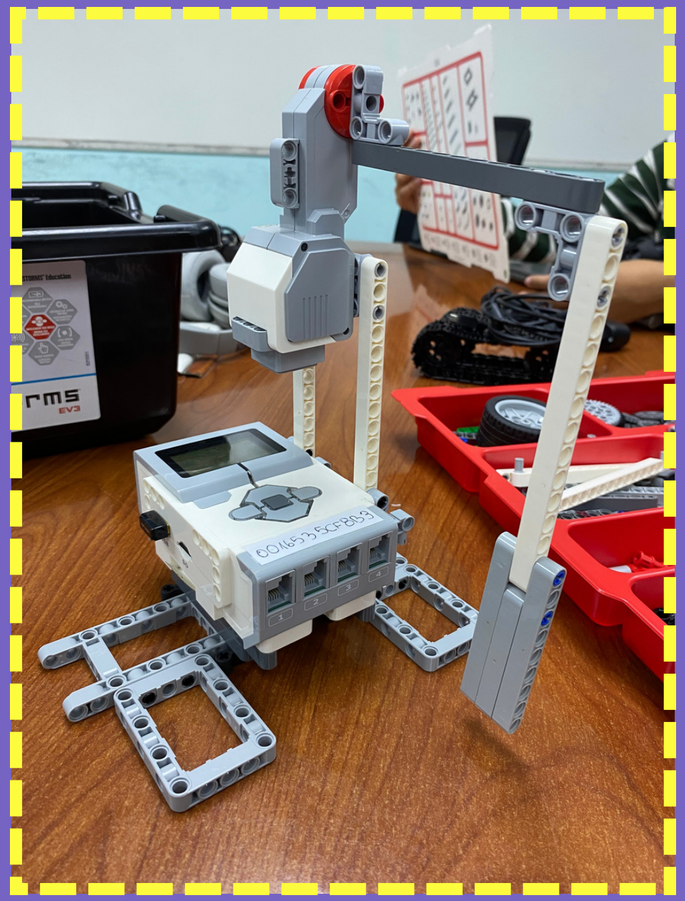
10

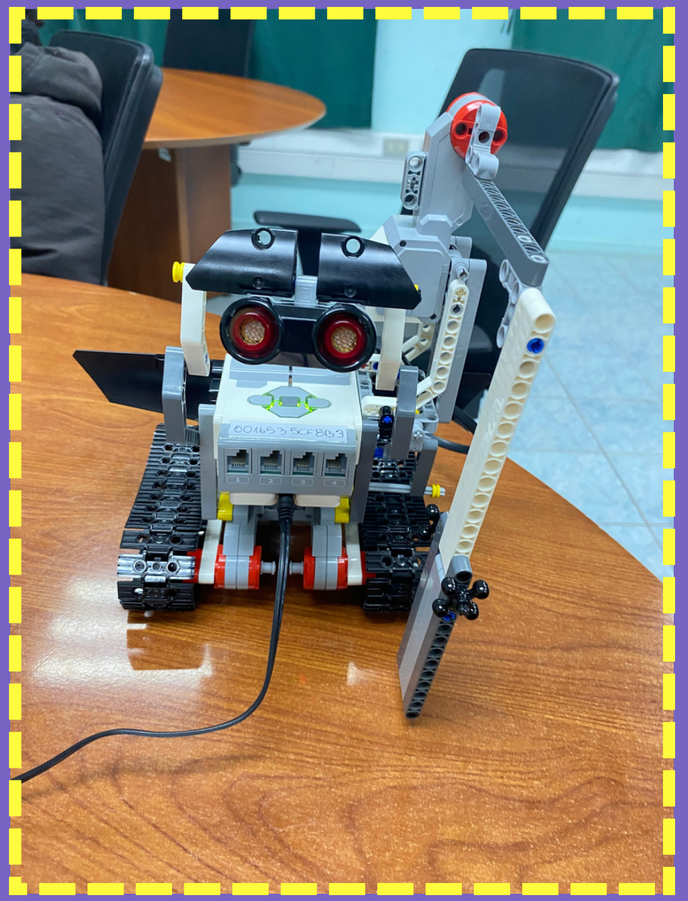
ROBOT FINALIZADO

Modificaciones del prototipo

PROTOTIPO I

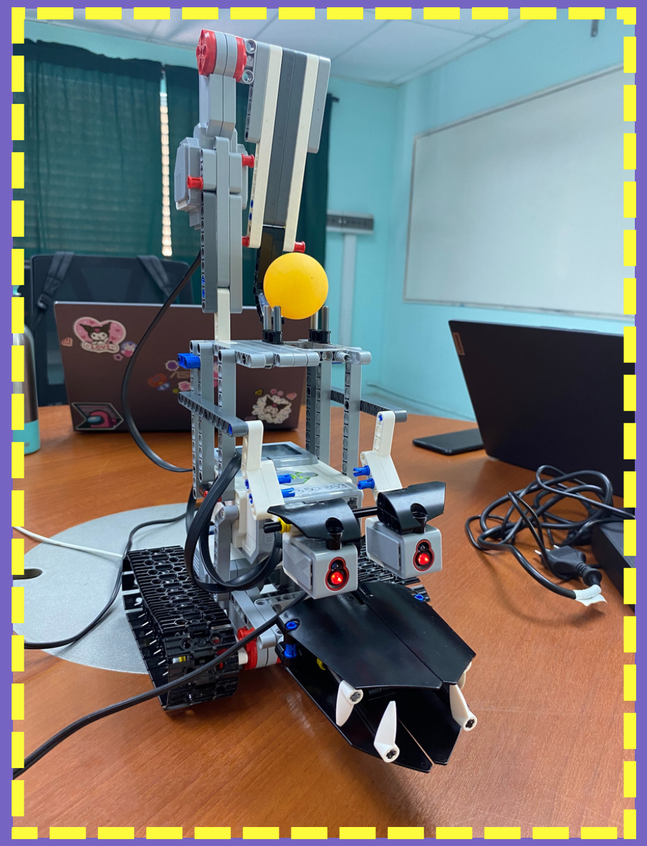
Inestabilidad en la estructura, además se dificultaba la conexión de los cables.

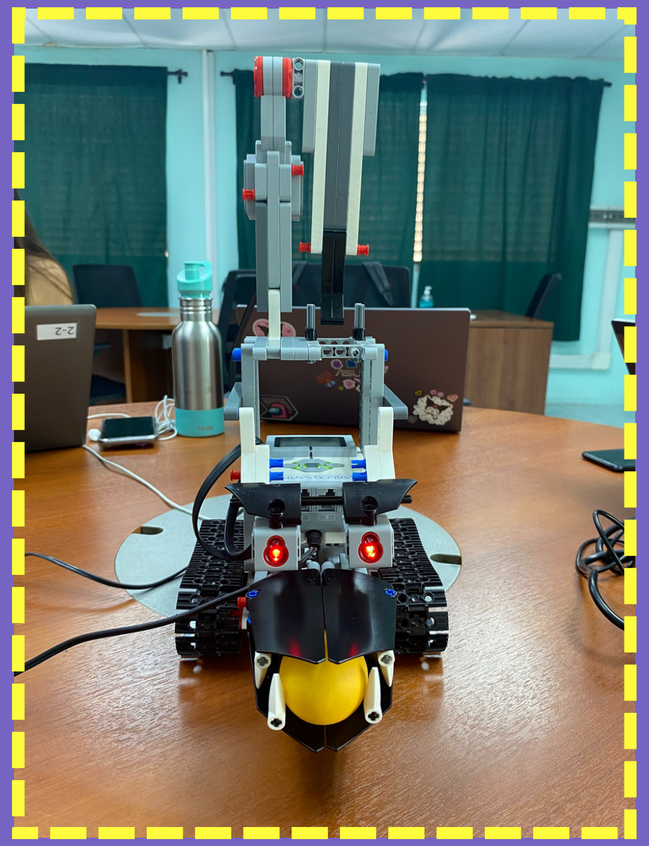




PROTOTIPO FINAL

Modificación en el brazo para mejor movimiento, se aplicó el refuerzo en la estructura. En lo estético se trato de reflejar a un cocodrilo.



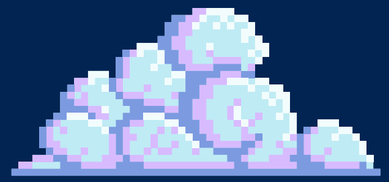


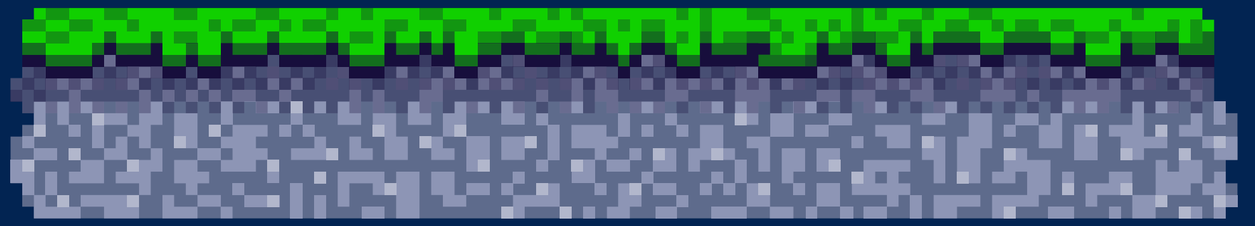
11

PROBLEMAS Y SOLUCIONES

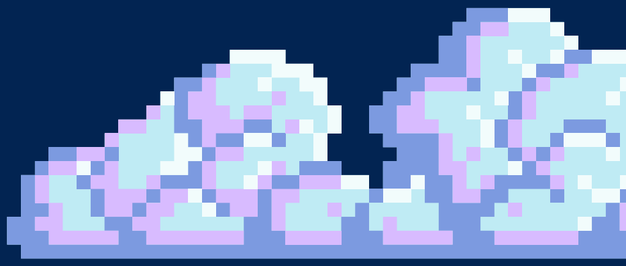
Problemas encontrados	Soluciones
El robot, se encontraba descargado muchas veces.	Cargarlo al principio de la sesión de clases
La inestabilidad del robot.	Se modificaron las fuerzas del motor y se agregaron piezas para reforzar.
Incongruencias en los tiempos de la carta gantt, con la realidad del proyecto.	Se reajustó el tiempo teniendo en cuenta las horas trabajadas en clases, autónomas y los días feriados.
El uso de una librería errónea (Pybrics) en la implementación de los movimientos del robot “Pybrics”	Se cambió de libreria, a la librería: Ev3dev2
Falta de conocimientos en el manejo de Visual Studio Code y conocimiento en redes y conexiones.	Preguntar a personas que manejan más el tema y aprender de forma autónoma en base a prueba y error.

12. RESULTADOS



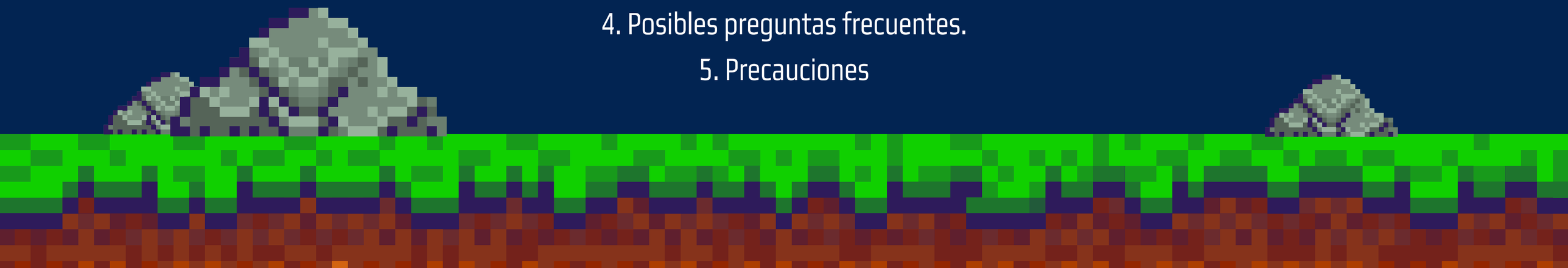


- La versión finalizada del robot “Alligator 3000”
- Las funciones de movimiento.
- Interfaz gráfica, se realizó mediante el uso de la librería “Tkinter”.
- La instauración del servidor involucró el empleo de la biblioteca de Python llamada "Socket". Este servidor está implementado en el entorno EV3DEV bajo el nombre de Server.py. Actualmente, se ha concluido su desarrollo y se encuentra plenamente operativo.
- La conexión vía remota.
- La wiki del proyecto.
- Carta Gantt actualizada.
- Bitácoras, informes y presentaciones concretadas.



MANUAL DE USUARIO

https://docs.google.com/document/d/1qzmGJ9ksB4Eqg6leiUnEjzP_9wN5p84X0KzQwG1FMZU/edit

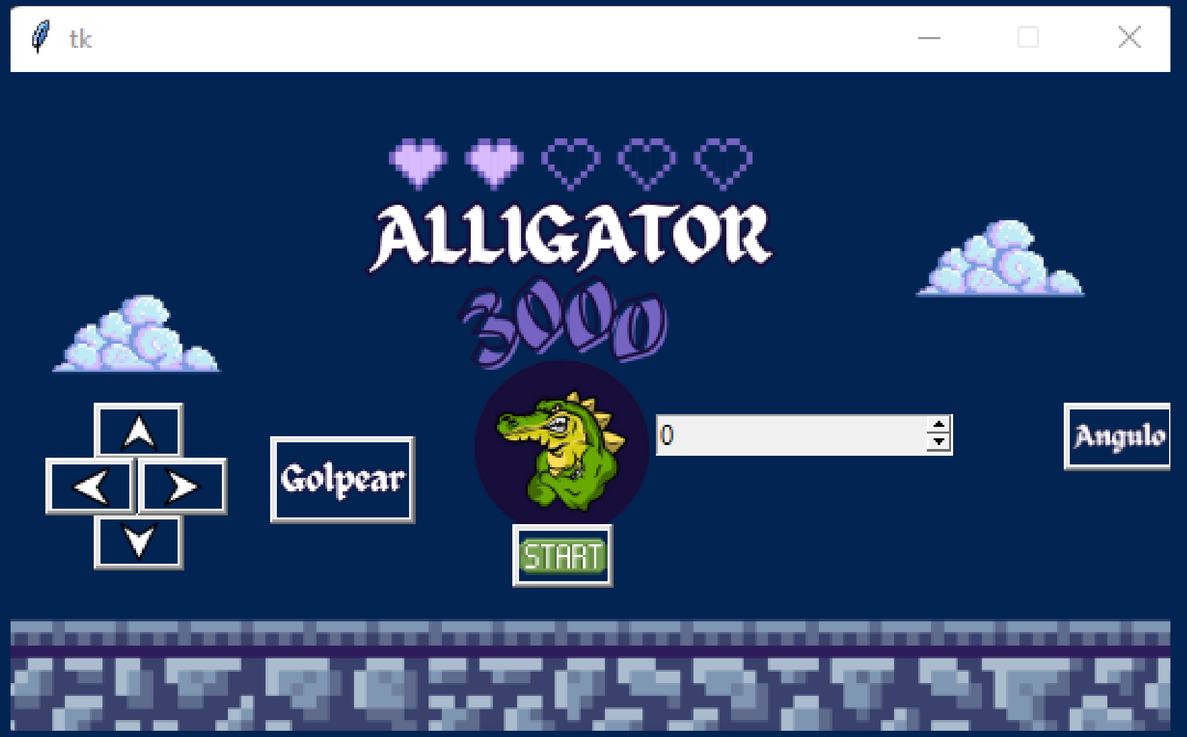


INTERFAZ GRÁFICA

1. Flechas direccionales para indicar las distintas orientaciones del robot.

2. "Golpear", el cual activará la función de golpear la pelota.

3. En la barra desplegable, se modifican los valores del ángulo que se utilizaran para posicionar el brazo al golpear.

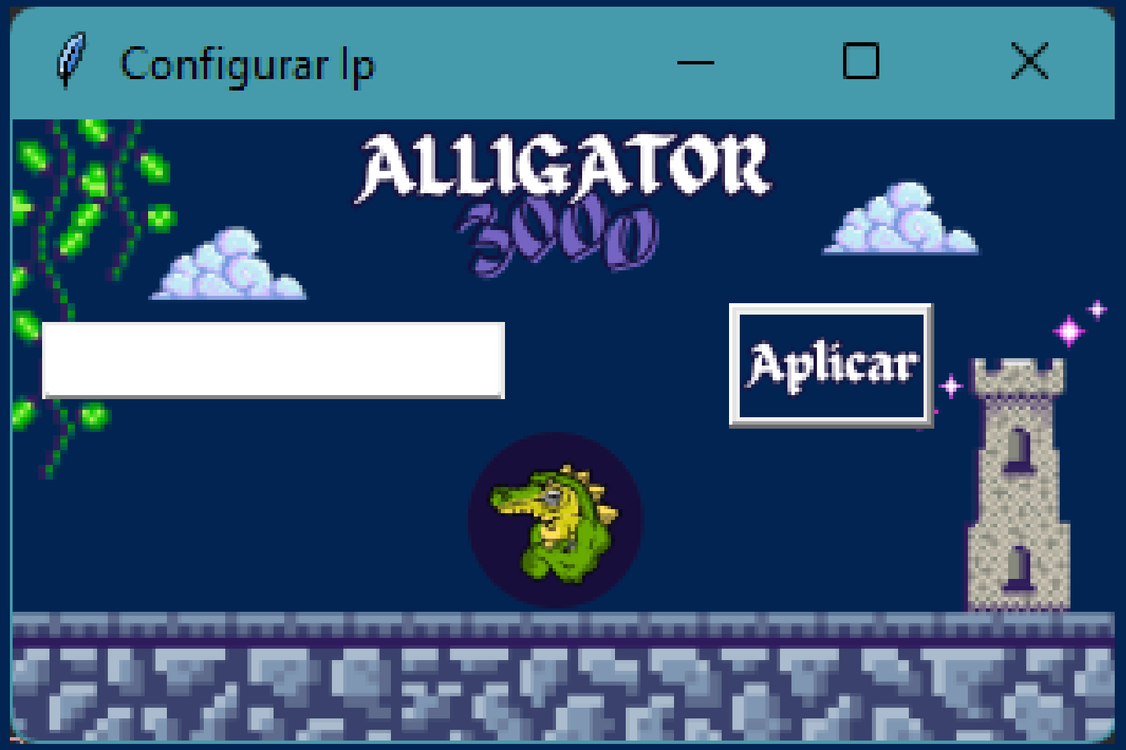


4. El botón “Ángulo” se ajusta con alguno de los 2 valores para seleccionar (45° o 90°)

5. El botón “Start”, se usa para establecer la conexión entre el cliente y el servidor en el EV3

INTERFAZ GRÁFICA

6. Al presionar el botón “Start” se abre una pestaña que contiene opciones para poder permitir al usuario vincularse mediante la dirección IP del servidor.



7. Ingresa la dirección IP en el campo correspondiente.

8. Haz clic en el botón de "Aplicar" para establecer la conexión.

CONCLUSIÓN

Gracias por su atención

ALLIGATOR 3000