

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA



DEPARTAMENTO DE INGENIERÍA EN
COMPUTACIÓN E INFORMÁTICA



INFORME 2

Heracross

Alumnos: Martin Gonzáles
Hilda Albarracín
Isaac Contreras
Guillermo Pino

Asignatura: Proyecto 1

Profesor: Humberto Urrutia

NOVIEMBRE - 2024

ÍNDICE

Historial de cambios	4
1. Panorama General	5
1.1. Introducción.....	5
1.2. Objetivos	5
1.2.1. Objetivo general.....	5
1.2.2. Objetivos específicos.....	5
1.3. Restricciones	5
1.4. Entregable.....	6
2. Organización del personal	7
2.1. Descripción de los roles	7
2.2. Personal que cumplirá los roles	7
2.3. Métodos de comunicación	7
3. Planificación del proyecto	8
3.1. Actividades (Carta Gantt)	8
3.2. Asignaciones de tiempo.....	9
3.3. Gestión de Riesgo.....	9
4. Planificación de los recursos	10
4.1. Hardware	10
4.2. Software.....	10
4.3. Estimación de costos	11
5. Análisis y diseño	12
5.1. Especificación de requerimiento.....	12
Requerimientos funcionales.....	12
Requerimientos no funcionales.	12
5.2. Arquitectura	12
5.3. Interfaz	13
6. Implementación	14
6.1. Fundamentos de los movimientos.....	14
6.2. Descripción de los programas	15
EV3server.py	15
Interfaz.py.....	16

6.3. Diagramas.....	19
7. Resultados.....	19
7.1. Estado actual del proyecto.....	19
7.2. Problemas encontrados y solucionados	19
8. Pruebas con el robot.....	20
8.1. Pruebas realizadas	20
8.2. Resultados	20
9. Conclusión	21
10. Referencias	21

Historial de cambios

Fecha	Versión	Descripción	Autor(es)
30/08/2024	1.0	Concepción del documento	Guillermo Pino Martin Gonzáles
03/09/2024	1.2	Recopilación y distribución de los datos	Martin Gonzáles
05/09/2024	1.3	Entrega	Guillermo Pino
18/10/2024	1.4	Inicio del desarrollo del avance de para la entrega 2	Guillermo Pino
24/10/2024	1.5	Avance punto 5: Análisis y diseño	Isaac Contreras Hilda Albarracín
07/11/2024	1.6	Avance punto 7	Guillermo Pino Martin Gonzáles
14/11/2024	1.7	Avance punto 6	Guillermo Pino
21/11/2024	1.8	Entrega	Guillermo Pino
10/12/2024	1.9	Avance punto 8	Guillermo Pino Hilda Albarracín Martin Gonzáles
13/12/2024	2.0	Entrega	Guillermo Pino

1. Panorama General

1.1. Introducción

En este semestre de clases, nos embarcaremos en un emocionante proyecto de ingeniería que involucra el desarrollo de un robot utilizando el kit LEGO MINDSTORMS EV3. Este proyecto se llevará a cabo de manera progresiva, con actividades y tareas asignadas semanalmente. A lo largo del semestre, cada integrante del equipo asumirá la responsabilidad de un aspecto específico del proyecto, lo que contribuirá a un mejor y más ordenado desarrollo del mismo. Los roles son: documentador, ensamblador, diseñador, programados y jefe de grupo.

El presente informe detalla los roles asignados a cada miembro del equipo, así como los costos asociados con el desarrollo del robot. Este enfoque no solo nos permitirá aplicar los conocimientos teóricos adquiridos en clase, sino también desarrollar habilidades prácticas en áreas clave como la programación, la ingeniería y el trabajo en equipo.

1.2. Objetivos

1.2.1. Objetivo general

Idear, construir y programar un robot del tipo EV3. Este debe poseer la capacidad de moverse a gusto del usuario y utilizar una garra que pueda sujetar una pelota de ping pong.

1.2.2. Objetivos específicos

- Realizar una planificación de un proyecto.
- Idear y construir un robot funcional a partir del kit LEGO MINDSTORMS EV3.
- Cargar una tarjeta microSD con el sistema operativo necesario para que el robot EV3 sea capaz de correr programas de software programados en Python.
- Confeccionar una aplicación de software que permita el movimiento y transmisión de ordenes al robot EV3 de manera remota.
- Ajustar y optimizar la aplicación anteriormente descrita para que cumpla su propósito lo más eficientemente posible.

1.3. Restricciones

- El robot debe estar armado a base del set LEGO MINDSTORMS EV3 de Lego.
- El lenguaje de programación que se exige para este proyecto es Python.

- El sistema operativo que se debe utilizar a lo largo del proyecto es Linux.
- Este proyecto también posee limitaciones de tiempo que deben ajustarse al formato de un semestre universitario.
- Toda la documentación debe ser publicada en la plataforma Redmine perteneciente a la Universidad de Tarapacá.

1.4. Entregable

En medida que avanzamos en la asignatura se debe entregar al profesor una serie de documentos que registren nuestro progreso del proyecto los cuales son:

1. **Bitácoras:** Documento de tipo informe que registra el progreso del proyecto, se requiere redactar uno por semana y debe ser entregado cada domingo a más tardar.
2. **Carta Gantt:** una línea de tiempo en un periodo determinado que contiene una representación gráfica de las tareas a ejecutar y su duración a lo largo del progreso del proyecto.
3. **Informes:** Dentro del plazo que se tiene para desarrollar el proyecto se deben redactar dos informes. El presente conteniendo la formulación del proyecto y el progreso hasta el momento de la entrega. Mientras que el segundo se trata del informe final del proyecto.
4. **Presentaciones:** Cada vez que se entregue un informe, este debe venir acompañado de una presentación relacionada con el contenido del informe en cuestión.
5. **Manual de usuario:** Se debe confeccionar un manual de instrucciones que detallen las funcionalidades del robot construido y una guía de cómo se operan sus funcionalidades.

2. Organización del personal

2.1. Descripción de los roles

1. **Jefe de grupo:** Encargado de la representación del grupo de trabajo ante terceros, además será responsable de la buena organización del mismo.
2. **Diseñador:** Se encargará de todo el apartado de "diseño" referente al proyecto. Ya sea presentaciones, interfaz de usuario, la wiki, etc.
3. **Documentador:** Será el responsable de la, valga la redundancia, documentación del proyecto: toma de fotos y videos del progreso del proyecto, bitácoras semanales, informes, carta Gantt, etc.
4. **Programador:** Encargado de idear e implementar algoritmos computacionales que cumplan con los requerimientos de movimiento del robot y sus funcionalidades.
5. **Ensamblador:** Se compromete con la arquitectura y el armado del robot, lo que engloba una base móvil y una garra que sea capaz de sostener una pelota de ping pong.

2.2. Personal que cumplirá los roles

Rol	Responsable	Implicados
Jefe de grupo	Isaac Contreras	Hilda Albarracin
Diseñador	Hilda Albarracin	Guillermo Pino Pedro Archile
Programador	Martín González	Isaac Contreras
Ensamblador	Pedro Archile	Hilda Albarracin Isaac Contreras
Documentador	Guillermo Pino	Martín González Isaac Contreras

2.3. Métodos de comunicación

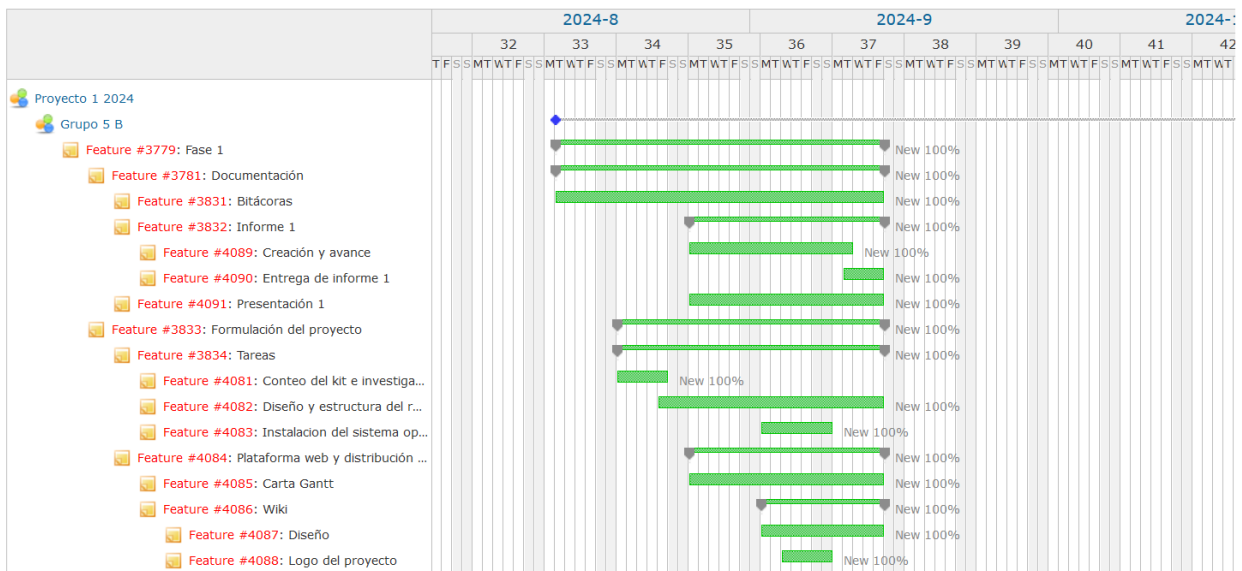
A lo largo del desarrollo del proyecto, se utilizarán principalmente dos medios de comunicación entre los integrantes del grupo. Dichos medios son WhatsApp y Discord ambos medios fiables y gratuitos para el uso público, además de muy convenientes y que en ambos

es posible crear una vía de comunicación tanto verbal como escrita, accesible y de fácil entendimiento.

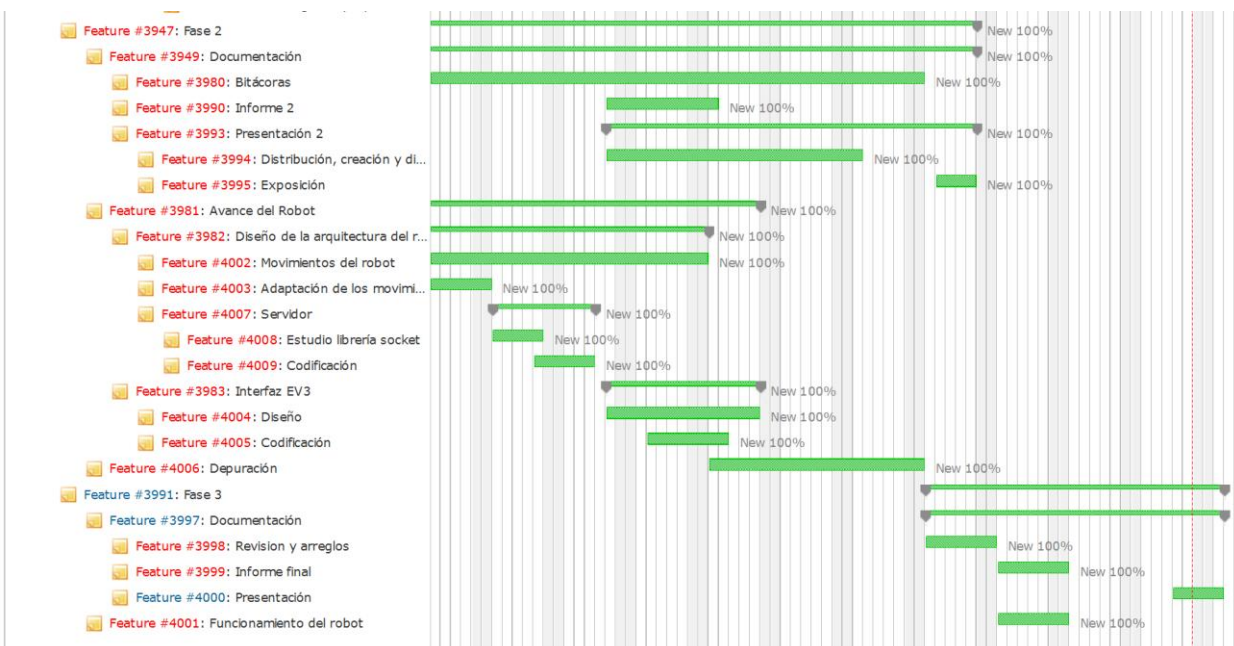
3. Planificación del proyecto

3.1. Actividades (Carta Gantt)

Agosto 2024



Noviembre - Diciembre 2024



3.2. Asignaciones de tiempo

- Planificación del proyecto: semana 4 semanas.
- Ejecución del proyecto: 6 semanas.
- Toques finales: 3 semanas.
- Tiempo total de proyecto: 13 semanas.

3.3. Gestión de Riesgo

Niveles de impacto:

1. **Catastrófico (1):** Daño que debe ser reparado inmediatamente, pues pondrá al proyecto en pausa de manera indefinida.
2. **Crítico (2):** En caso de ocurrencia, debe ser resuelto a la brevedad, debido a que tener uno de estos problemas retrasará el proyecto en varias etapas.
3. **Circunstancial (3):** Problema que es necesario resolver en el momento, de lo contrario podría retrasar el proyecto en una etapa en específico.
4. **Irrelevante (4):** El daño causado por este tipo de riesgo es mínimo por lo tanto es un problema menor que puede ser resuelto en cualquier momento.

Tipo de Riesgo	Probabilidad de Ocurrencia	Nivel de Impacto	Acciones Por Tomar
Error en la codificación	90%	2	Revisar y analizar el código escrito para identificar el problema encontrado y resolverlo.
Desarme del robot por contusión o caída	10%	2	Se le informa a los implicados en la arquitectura del robot y se efectúa un rearmado del mismo.
Carencia de una pieza específica requerida	30%	4	Se consulta al personal a cargo de las piezas.
Falta de un miembro por enfermedad o problemas personales	90%	3	Se reorganizan las tareas que el miembro faltante debía cumplir entre

			los miembros restantes.
Incompatibilidad de la arquitectura del robot con los objetivos que debe cumplir	60%	2	Rediseñar y armar un nuevo robot que sea capaz de cumplir con los requerimientos.
Perdida parcial o total de archivos	20%	2	Recrear lo más exacto posible todo el material perdido.

4. Planificación de los recursos

4.1. Hardware

Todos los recursos físicos que se utilizarán para llevar a cabo el proyecto serán los siguientes:

- Kit LEGO MINDSTORMS EV3.
- Tarjeta microSD con su correspondiente adaptador para PC.
- Adaptador Wi-Fi USB.
- Notebook con sistema operativo Linux que permita codificar en el robot.
- Galaxy Tab S8.

4.2. Software

- Linux como sistema operativo.
- Discord y WhatsApp como medios de comunicación.
- Visual Studio Code como editor de código y conexión al robot EV3.
- Canva para crear las presentaciones.

4.3. Estimación de costos

Categoría	Descripción	Cantidad	Costo (CLP)
Hardware	Kit básico LEGO MINDSTORMS EV3	1	\$ 1.006.350
	MicroSD c/ Adaptador	1	\$ 13.000
	Adaptador wifi	1	\$ 9.450
	Notebook	3	\$ 1.920.000
	Samsung Galaxy Tab S8	1	\$ 520.000
Software	Licencia Canva Pro Equipo / 4 meses	1	\$ 120.000
	Plan Discord Nitro / 5 personas / 4 meses	1	\$ 107.200

La cantidad de horas trabajadas y horas extra se considerarán utilizando la planificación de la asignatura Proyecto I. Horas trabajadas siendo horas presenciales (72) y horas extra siendo horas autónomas (82).

Cargo	Valor hora (CLP)	Horas presenciales	Horas autónomas	Total horas	Sueldo mensual (CLP)
Jefe de grupo	\$ 9.231	72	82	154	\$ 1.421.574
Programador	\$ 6.701	72	82	154	\$ 1.031.954
Ensamblador	\$ 7.385	72	82	154	\$ 1.137.290
Diseñador	\$ 3.692	72	82	154	\$ 568.568
Documentador	\$ 8.615	72	82	154	\$ 1.326.710
Total					\$ 5.486.096

5. Análisis y diseño

5.1. Especificación de requerimiento

Requerimientos funcionales

El robot tiene una conexión estable con computadora y se comunica mediante la interfaz gráfica a través del bluetooth y Wi-Fi.

El robot es capaz de moverse hacia adelante, hacia atrás y también girar gracias al sensor de giro, además de una pinza que puede tomar una pelota de ping pong.

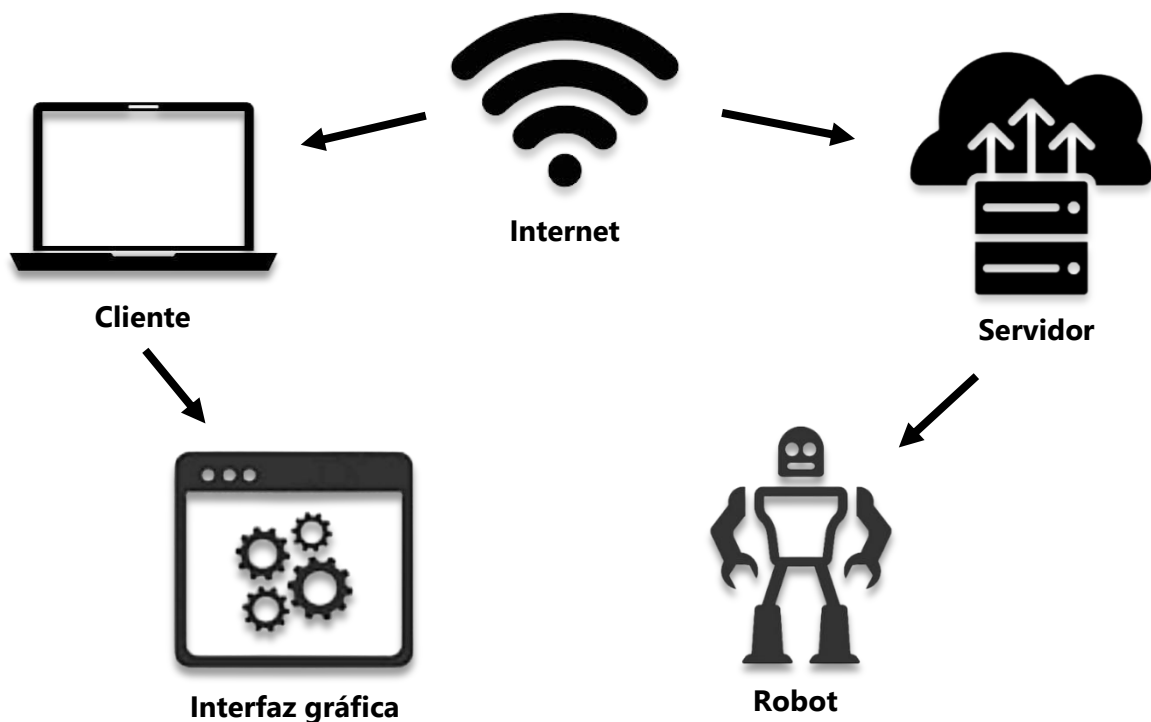
Requerimientos no funcionales.

Interfaz gráfica de usuario fácil de entender y usar, además de los botones de movimiento también cuenta con botones de soporte y 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.

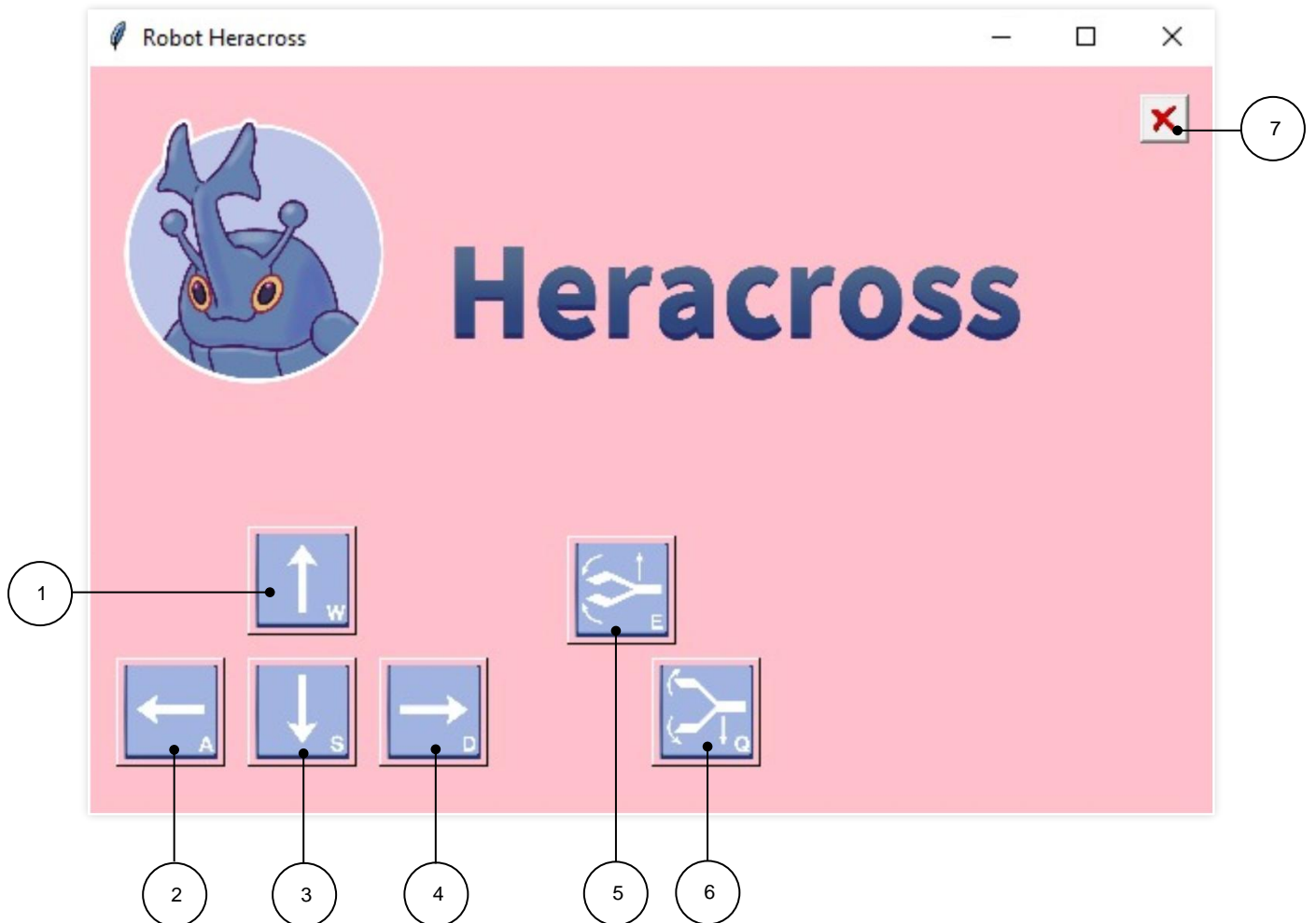
El proyecto debe contar con manual de usuario, el cual indicará el buen uso del robot de manera clara y entendible.

5.2. Arquitectura



5.3. Interfaz

Imagen de la interfaz gráfica con la descripción de las funciones que hace cada icono de ella.



1. Botón de movimiento hacia adelante.
2. Botón de giro hacia la izquierda.
3. Botón de movimiento hacia atrás.
4. Botón de giro hacia la derecha.
5. Botón para cerrar la garra y luego subir la garra.
6. Botón para bajar y luego abrir la garra.
7. Botón para cerrar el programa.

6. Implementación

6.1. Fundamentos de los movimientos

Como equipos determinamos que en el robot actúa la fuerza de gravedad y su trabajo mecánico realizado, esto en el levantamiento de la pelota de ping pong.

Para determinar la fuerza que debe realizar el robot para levantar tenemos que:

$$\vec{F} = m \cdot \vec{a}, \quad \text{con } \vec{a} = \vec{g} = 9,8 \frac{m}{s^2}$$

por lo tanto:

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

$$\vec{F} = 0,2646 N$$

La fuerza necesaria para levantar la pelota de ping pong debe ser mayor a 0,2646 N.

Para determinar el trabajo mecánico que realiza la fuerza el robot para levantar la pelota:

$$\vec{W} = \vec{F} \cdot \vec{d}$$

$$\vec{W} = 0,2646 N \cdot 0,1 m$$

$$\vec{W} = 2646 \cdot 10^{-5} J$$

6.2. Descripción de los programas

EV3server.py

```
1  #!/usr/bin/env pybricks-micropython
2
3  import socket
4
5  from pybricks.ev3devices import Motor
6  from pybricks.robotics import DriveBase
7  from pybricks.parameters import Port
8
9  leftMotor = Motor(Port.D)
10 rightMotor = Motor(Port.B)
11 clawMotor = Motor(Port.C)
12
13 flagGrabbed = False
14
15 robot = DriveBase(leftMotor, rightMotor, wheel_diameter=55.5, axle_track=104)
16
17 ip_address = '0.0.0.0'
18 port = 63383
19 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
20 sock.bind((ip_address, port))
21
22 print("Listening...")
23
24 while True:
25     data, addr = sock.recvfrom(1024)
26     command = data.decode()
27
28     if command == 'W':
29         robot.drive(1000,0)
30     elif command == 'A':
31         robot.drive(0,-120)
32     elif command == 'S':
33         robot.drive(-1000,0)
34     elif command == 'D':
35         robot.drive(0,120)
36     elif command == 'E':
37         clawMotor.run(-500)
38     elif command == 'Q':
39         clawMotor.run(500)
40     elif command == 'stop':
41         robot.stop()
42     elif command == 'stopClaw':
43         clawMotor.stop()
```

Interfaz.py

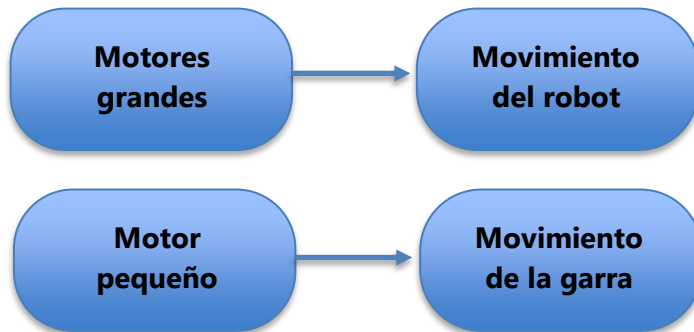
```
1 import os
2 import tkinter as tk
3 import socket
4 #Constantes
5 ventana= tk.Tk()
6 IMAGENWoR= tk.PhotoImage(file= "Ola/bt_w.png").subsampling(8,8)
7 IMAGENWoP= tk.PhotoImage(file= "Ola/bt_w.png").subsampling(5,5)
8 IMAGENAoR=tk.PhotoImage(file= "Ola/bt_a.png").subsampling(8,8)
9 IMAGENAoP=tk.PhotoImage(file= "Ola/bt_a.png").subsampling(5,5)
10 IMAGENSor=tk.PhotoImage(file= "Ola/bt_s.png").subsampling(8,8)
11 IMAGENSop=tk.PhotoImage(file= "Ola/bt_s.png").subsampling(5,5)
12 IMAGENDoR=tk.PhotoImage(file= "Ola/bt_d.png").subsampling(8,8)
13 IMAGENDoP=tk.PhotoImage(file= "Ola/bt_d.png").subsampling(5,5)
14 IMAGENEoR=tk.PhotoImage(file= "Ola/bt_cerrarGarra.png").subsampling(8,8)
15 IMAGENEoP=tk.PhotoImage(file= "Ola/bt_cerrarGarra.png").subsampling(5,5)
16 IMAGENQoR=tk.PhotoImage(file= "Ola/bt_abrirGarra.png").subsampling(8,8)
17 IMAGENQoP=tk.PhotoImage(file= "Ola/bt_abrirGarra.png").subsampling(5,5)
18 IMAGENSALIR=tk.PhotoImage(file= "Ola/Salir Img.png").subsampling(10,10)
19 IMAGENLOGO=tk.PhotoImage(file= "Ola/logo_borde.png").subsampling(3,3)
20 IMAGENTITULO=tk.PhotoImage(file= "Ola/tituloHeracross.png").subsampling(3,3)
21 #Funciones
22 def send_command(command): #Envia comandos al servidor
23     ev3_ip = '192.168.74.196' # Check each time
24     port = 63383 # Get-NetTCPConnection | Where-Object { $_.State -eq 'Listen' }
25     with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
26         s.sendto(command.encode(), (ev3_ip, port))
27     #Comandos Boton W
28 def W_onPress(event):
29     send_command("W")
30     boton_w.config(image=IMAGENWoP)
31 def W_onRelease(event):
32     send_command("stop")
33     boton_w.config(image=IMAGENWoR)
34 #Comandos Boton A
35 def A_onPress(event):
36     send_command("A")
37     boton_a.config(image= IMAGENAoP)
38 def A_onRelease(event):
39     send_command("stop")
40     boton_a.config(image= IMAGENAoR)
41 #Comandos Boton S
42 def S_onPress(event):
```



```
43     send_command("S")
44     boton_s.config(image= IMAGENSOP)
45 def S_onRelease(event):
46     send_command("stop")
47     boton_s.config(image=IMAGENSOR)
48     #Comandos Boton D
49 def D_onPress(event):
50     send_command("D")
51     boton_d.config(image= IMAGENDOOP)
52 def D_onRelease(event):
53     send_command("stop")
54     boton_d.config(image=IMAGENDOR)
55     #Comandos Boton E
56 def E_onPress(event):
57     send_command("E")
58     boton_e.config(image= IMAGENEOOP)
59 def E_onRelease(event):
60     send_command("stopClaw")
61     boton_e.config(image=IMAGENEOOR)
62     #Comandos Boton Q
63 def Q_onPress(event):
64     send_command("Q")
65     boton_q.config(image= IMAGENQOOP)
66 def Q_onRelease(event):
67     send_command("stopClaw")
68     boton_q.config(image=IMAGENQOOR)
69
70 #Configuracion TKinter
71 ventana.title("Robot Heracross")
72 ventana.geometry("600x400")
73 ventana.resizable(0,0)
74 frame = tk.Frame(ventana, bg= 'pink')
75 frame.pack(fill="both", expand= True)
76 #logo y titulo
77 logo_label=tk.Label(frame,image=IMAGENLOGO, bg='pink')
78 logo_label.place(x=15,y=25)
79 titulo_label=tk.Label(frame,image=IMAGENTITULO,bg='pink')
80 titulo_label.place(x=180,y=85)
81 #Boton W
82 boton_w= tk.Button(frame, image=IMAGENWoR, bg='pink',activebackground='pink', bd=1)
83 boton_w.place (x=85, y=245)
84 boton_w.bind("<ButtonPress-1>", W_onPress)
```

```
85 boton_w.bind("<ButtonRelease-1>",W_onRelease)
86 ventana.bind("<KeyPress-w>", W_onPress)
87 ventana.bind("<KeyRelease-w>", W_onRelease)
88 #Boton A
89 boton_a= tk.Button(frame, image= IMAGENAoR, bg='pink',activebackground='pink', bd=1)
90 boton_a.place(x= 15, y= 315)
91 boton_a.bind("<ButtonPress-1>", A_onPress)
92 boton_a.bind("<ButtonRelease-1>",A_onRelease)
93 ventana.bind("<KeyPress-a>", A_onPress)
94 ventana.bind("<KeyRelease-a>", A_onRelease)
95 #Boton S
96 boton_s= tk.Button(frame, image=IMAGENSor, bg='pink',activebackground='pink', bd=1)
97 boton_s.place(x=85 ,y=315)
98 boton_s.bind("<ButtonPress-1>", S_onPress)
99 boton_s.bind("<ButtonRelease-1>", S_onRelease)
100 ventana.bind("<KeyPress-s>", S_onPress)
101 ventana.bind("<KeyRelease-s>", S_onRelease)
102 #Boton D
103 boton_d= tk.Button(frame, image=IMAGENDoR, bg='pink',activebackground='pink', bd=1)
104 boton_d.place(x=155,y=315)
105 boton_d.bind("<ButtonPress-1>", D_onPress)
106 boton_d.bind("<ButtonRelease-1>", D_onRelease)
107 ventana.bind("<KeyPress-d>", D_onPress)
108 ventana.bind("<KeyRelease-d>", D_onRelease)
109 #Boton E
110 boton_e=tk.Button(frame, image=IMAGENEoR, bg='pink',activebackground='pink', bd=1)
111 boton_e.place(x=255,y= 250)
112 boton_e.bind("<ButtonPress-1>", E_onPress)
113 boton_e.bind("<ButtonRelease-1>", E_onRelease)
114 ventana.bind("<KeyPress-e>", E_onPress)
115 ventana.bind("<KeyRelease-e>", E_onRelease)
116 #Boton Q
117 boton_q=tk.Button(frame, image=IMAGENQoR, bg='pink',activebackground='pink', bd=1)
118 boton_q.place(x=300,y= 315)
119 boton_q.bind("<ButtonPress-1>", Q_onPress)
120 boton_q.bind("<ButtonRelease-1>", Q_onRelease)
121 ventana.bind("<KeyPress-q>", Q_onPress)
122 ventana.bind("<KeyRelease-q>", Q_onRelease)
123 #Boton Salir
124 boton_salir= tk.Button(frame, image= IMAGENSALIR, command= ventana.destroy)
125 boton_salir.place (x=560,y=15)
126 ventana.mainloop()
```

6.3. Diagramas



7. Resultados

7.1. Estado actual del proyecto

La construcción del robot se encuentra totalmente completa y con su código funcionando en perfecto estado, con una interfaz gráfica intuitiva y fácil de usar que permiten a través de ella la actividad de los movimientos que realiza el robot. Los movimientos capaces de realizar son el de desplazarse hacia adelante y atrás, girar a la derecha e izquierda, la elevación y descenso de los brazos y por último abrir y cerrar las pinzas para tomar la pelota de ping pong.

7.2. Problemas encontrados y solucionados

Problemas en el diseño de los iconos de los botones de la interfaz gráfica y su solución fue rediseñando los iconos por un acuerdo de gusto en común por los integrantes del equipo.

Problemas en el uso de Tkinter y sockets combinados, se solucionó entendiendo más el uso de estos mismo.

Errores de codificación en una función, en el uso del motor en que se activara sin límite de tiempo. Se mejoro cambiando la opción a un cierto límite de tiempo.

8. Pruebas con el robot

8.1. Pruebas realizadas

Establecimos la conexión entre el computador y el sistema EV3 permitiendo programar y codificar algoritmos, también como subir los archivos necesarios para la interfaz gráfica y permitir la funcionalidad del robot.

Realizamos pruebas de movimientos del robot (adelante, hacia atrás y giros) y también pruebas de los movimientos de la garra para determinar que funciona correctamente al levantar y dejar la pelota de ping pong.

Determinar la funcionalidad del robot mediante una interfaz gráfica.

8.2. Resultados

La conexión entre el computador y el equipo EV3 lo logramos mediante Wi-Fi local entre el robot y el computador.

Los movimientos del robot lo lograron mediante la conexión establecida anteriormente y la programación realizada en clases con el IDE Visual Studio Code y el lenguaje de programación Python.

Mediante la programación logramos encajar los movimientos del robot con la interfaz gráfica usando Tkinter en Python.

9. Conclusión

Durante el desarrollo de este proyecto se cumplieron los objetivos planteados. Como conclusión para nuestro equipo, este proyecto fue una experiencia en la cual pudimos aplicar nuestros conocimientos aprendidos en los primeros semestres. El desarrollo y aplicación de nuestros conocimientos en este proyecto nos ayudará en un futuro entorno profesional de para trabajar de manera formal y eficiente.

10. Referencias

LEGO El juego de construcción MINDSTORMS EV3 incluye 3. (s. f.). Amazon. <https://www.amazon.com/-/es/construcci%C3%B3n-MINDSTORMS-servomotores-interactivos->

redise%C3%B1ado/dp/B016SF8GWY/ref=sr_1_4?_mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=2WBMO7SQXAJLG&dib=eyJ2IjoiMSJ9.p7tS_F0gBAKeTwGBJCbB4UkO5GleFi-vz-uPW2z2UBkvmKRBB-Q59De-WcnqoP_CgLOCrW_2nTzP2TxFPBbhKj8WVFO9KM3FrVUYtGI9iHDjYKTJAswBN_EmqelSeXupG3xGPhMTseQt3A31XMeihvq7vOoNNCv5G3MMVDbCxPg39fIGqrZ15z-hgyg7i6ND-MBOHhOUUnu_6xFrkZ1iXq08UTiN_6n8n4HcP6xgnL-2JuCqkg1bRuQwGEymJsYbRa9pHnHYzmcnQ1rH9SIX6dberxnizuYKM_nKzpmQ2XKM.ZmM0iz9cRtDIE_tIT9uCN3PR62ls5rI5-juebMjVK8Q&dib_tag=se&keywords=ev3%2Blego%2Brobotics%2Bkit&qid=1725937954&sprefix=ev3%2Blego%2Brobotics%2Bkit%2Caps%2C415&sr=8-4

Memoria Micro Sd Hc 8gb Sandisk Clase 4. (s. f.). MercadoLibre. https://articulo.mercadolibre.cl/MLC-1889109532-memoria-micro-sd-hc-8gb-sandisk-clase-4-_JM#position%3D4%26search_layout%3Dstack%26type%3Ditem%26tracking_id%3D081e5023-6eda-4abe-b26d-9f6e10d7e644

Adaptador de Wifi Nano Inalambrico 150 mbps. (s. f.). MercadoLibre. https://articulo.mercadolibre.cl/MLC-511410157-adaptador-wifi-usb-nano-inalambrico-150mbps-wn725n-tp-link-_JM#position%3D22%26search_layout%3Dstack%26type%3Ditem%26tracking_id%3Dda65db98-0232-4dd5-a29c-2bb94ca8bd95

Sueldo Graphic Designer en Santiago, Chile 2024. (2024). Glassdoor. https://www.glassdoor.com.ar/Sueldos/santiago-graphic-designer-sueldo-SRCH_IL0,8_IM1004_KO9,25.htm

Salario para Ingeniero Eléctrico en Chile. (2024). Talent.
<https://cl.talent.com/salary?job=ingeniero+electrico#:~:text=%C2%BFCu%C3%A1nto%20gana%20un%20Ingeniero%20electrico%20en%20Chile%3F&text=El%20salario%20ingeniero%20electrico%20promedio,con%20un%20ingreso%20de%20%2410.800>.

Sueldos de ingenieros software. (2024, septiembre) Chiletrabajos.
<https://www.chiletrabajos.cl/sueldos/ingeniero/software#:~:text=Salario%20o%20sueldo%20promedio%20de,840%20en%20Septiembre%202024>

Salario para Jefe De Proyecto en Chile- Salario Medio. (2024). Talent.
<https://cl.talent.com/salary?job=jefe+de+proyecto#:~:text=%C2%BFCu%C3%A1nto%20gana%20un%20Jefe%20de%20proyecto%20en%20Chile%3F&text=El%20salario%20jefe%20de%20proyecto,a%3%B1o%20o%20%249.231%20por%20hora>.

Salario para Administrador de Proyecto en Chile. (2024). Talent.
<https://cl.talent.com/salary?job=administrador+de+proyecto#:~:text=%C2%BFCu%C3%A1nto%20gana%20un%20Administrador%20de%20proyecto%20en%20Chile%3F&text=El%20salario%20administrador%20de%20proyecto%20promedio%20en%20Chile%20es%20de%20%2416.800>.

Documents - Grupo 5 B – Redmine. (2024. 12 septiembre). Redmine.
<http://pomerape.uta.cl/redmine/projects/grupo-5-b/documents>