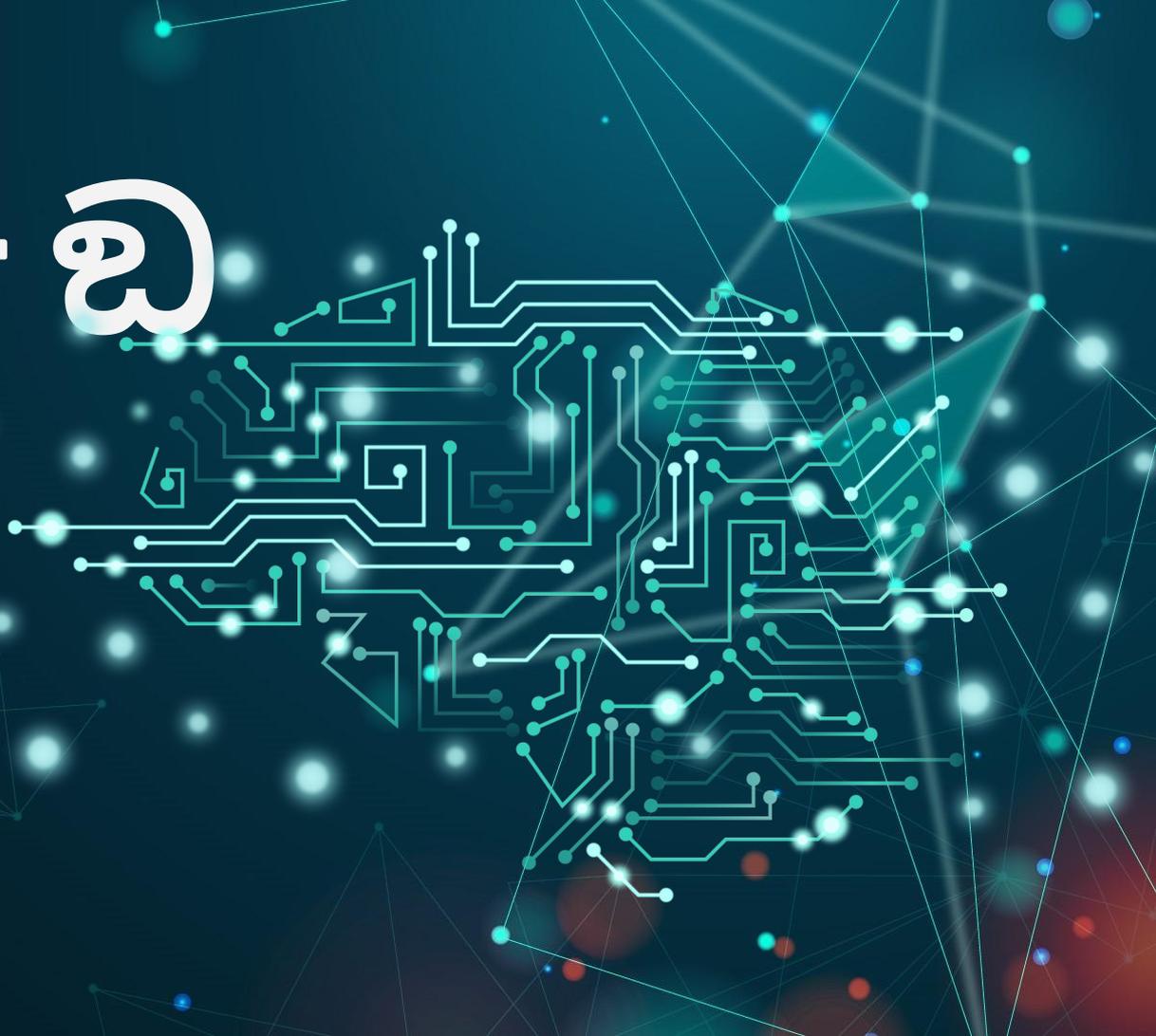


# Panzer 2



Alumno(os): Miguel Fernández  
Edson Galdames  
Lukas Torres  
Bastian Vega

Asignatura: Proyecto I

Profesor: Humberto Urrutia

# ÍNDICE

01 | Introducción

03 | Diseño de la  
Interfaz

05 | Funcionalidades

02 | Objetivos

04 | Fundamentos  
de proyectil

06 | Conclusiones

El proyecto Panzer 2) consiste en el desarrollo de un robot utilizando el kit de lego EV3 mindstorm que se pueda desplazar y hacer un disparo haciendo uso de la mecánica clásica, además de ser controlado remotamente desde una interfaz.

# Introduccion

# Objetivos

## Objetivo General

El objetivo general de este proyecto consta en utilizar el kit de lego MindStorms Ev3 para armar un robot con un cañón que se pueda desplazar, apuntar a un objetivo y disparar.

## Objetivos Específicos

- Diseñar la estructura del robot de manera que se pueda desplazar y disparar
- Programar algoritmos para que el robot cumpla lo anterior
- Desarrollar una interfaz para controlar de manera remota al robot

# Diseño de la Interfaz

## Movimiento

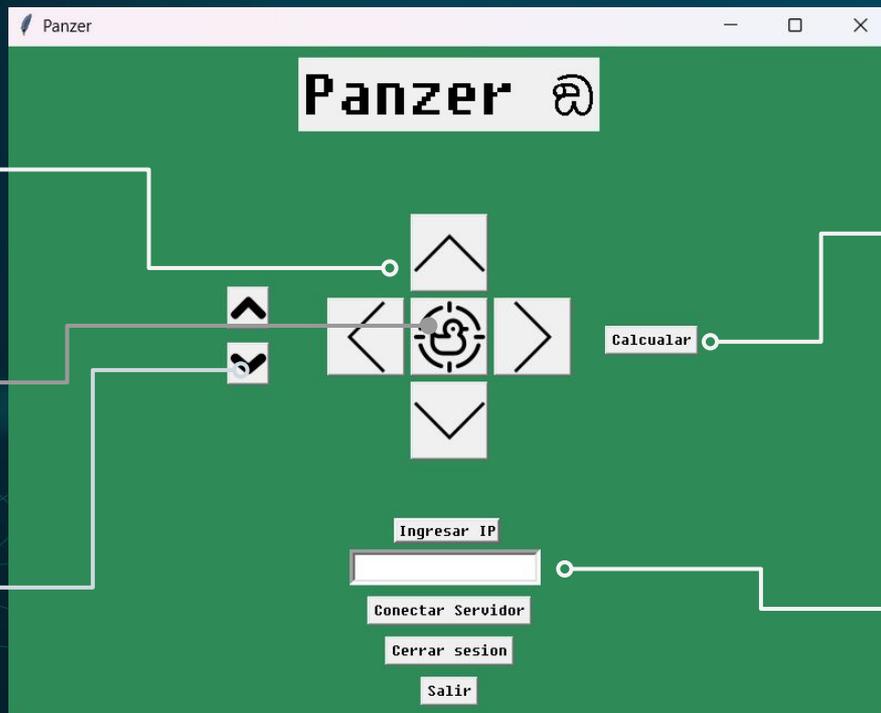
Botones que controlan el movimiento del robot

## Disparo

Botones que ejecuta el comando de disparo

## Elevación del cañón

Botones que controlan manualmente la elevación del cañón.



## Calcular Angulo

Botón el cual permite al robot cambiar a un ángulo donde el tiro sea más preciso

## Dirección IP

Campo donde el usuario puede ingresar la ip del servidor

# Fundamentos de proyectiles

$$x = 100\text{cm} \text{ (distancia del proyectil)}$$

$$v = 121\left(\frac{\text{cm}}{\text{s}}\right)$$

$$1) x = x_0 + v_0 * t \rightarrow x = 50 \text{ (punto medio)}$$

$$50\text{cm} = 121\left(\frac{\text{cm}}{\text{s}}\right) * \cos(\theta) * t$$

$$2) v_f = v_{0y} - g * t$$

$$0\left(\frac{\text{cm}}{\text{s}}\right) = 121\left(\frac{\text{cm}}{\text{s}}\right) * \text{sen}(\theta) - 98\left(\frac{\text{cm}}{\text{s}}\right) * t$$

Despejando t en 2:

$$0 = 121\left(\frac{\text{cm}}{\text{s}}\right) * \text{sen}(\theta) - 98 * t$$

$$t = \frac{121}{98}\left(\frac{\text{cm}}{\text{s}}\right) * \text{sen}(\theta)$$

Reemplazando t en 1:

$$50 = 121 * \cos(\theta) * \frac{121}{98} * \text{sen}(\theta)$$

$$50 = 149,3979 * \cos(\theta) * \text{sen}(\theta)$$

$$\frac{50}{149,3979} = \cos(\theta) * \text{sen}(\theta)$$

$$0,3346 = \frac{\text{sen}(2\theta)}{2}$$

$$0,6693 = \text{sen}(2\theta) \rightarrow \sin^{-1}(0,6693) = 2\theta$$

$$19,1755 = \theta$$

# Funcionalidades

## Movimiento

```
def moveUp():
    print("Moving up...")
    tankmoves.on(-100,-100)

def moveDown():
    print("Moving down...")
    tankmoves.on(100,100)

def moveRight():
    print("Moving right...")
    tankmoves.on(100,-100)

def moveLeft():
    print("Moving Left...")
    tankmoves.on(-100,100)
```

## Elevación del cañón

```
def elevar():
    global val
    if (val < 25):
        print("eLevando...")
        print(val)
        val+=5
        md.on_to_position(10,val)

def bajar():
    global val
    if (val > -35):
        print("bajando...")
        print(val)
        val= val - 5
        md.on_to_position(10, val)
```

## Disparo

```
def disparar():
    print("fuego")
    mc.on_for_rotations(100, 5)
```

## Detener motores

```
def stop():
    tankmoves.stop()
    print("parar")
    md.off(brake=True)
```

## Apuntar

```
def getDistancia():
    distance = us.value()/10
    return distance
```

```
def calcularAngulo():
    distance = getDistancia()
    while (distance < 95 or distance > 105):
        if distance > 90:
            tankmoves.on(-100,-100)
            distance = getDistancia()
        else:
            tankmoves.on(100,100)
            distance = getDistancia()
            tankmoves.stop()
            print(distance)
    md.on_to_position(10,25)
    disparar()
```

# Conclusión

