

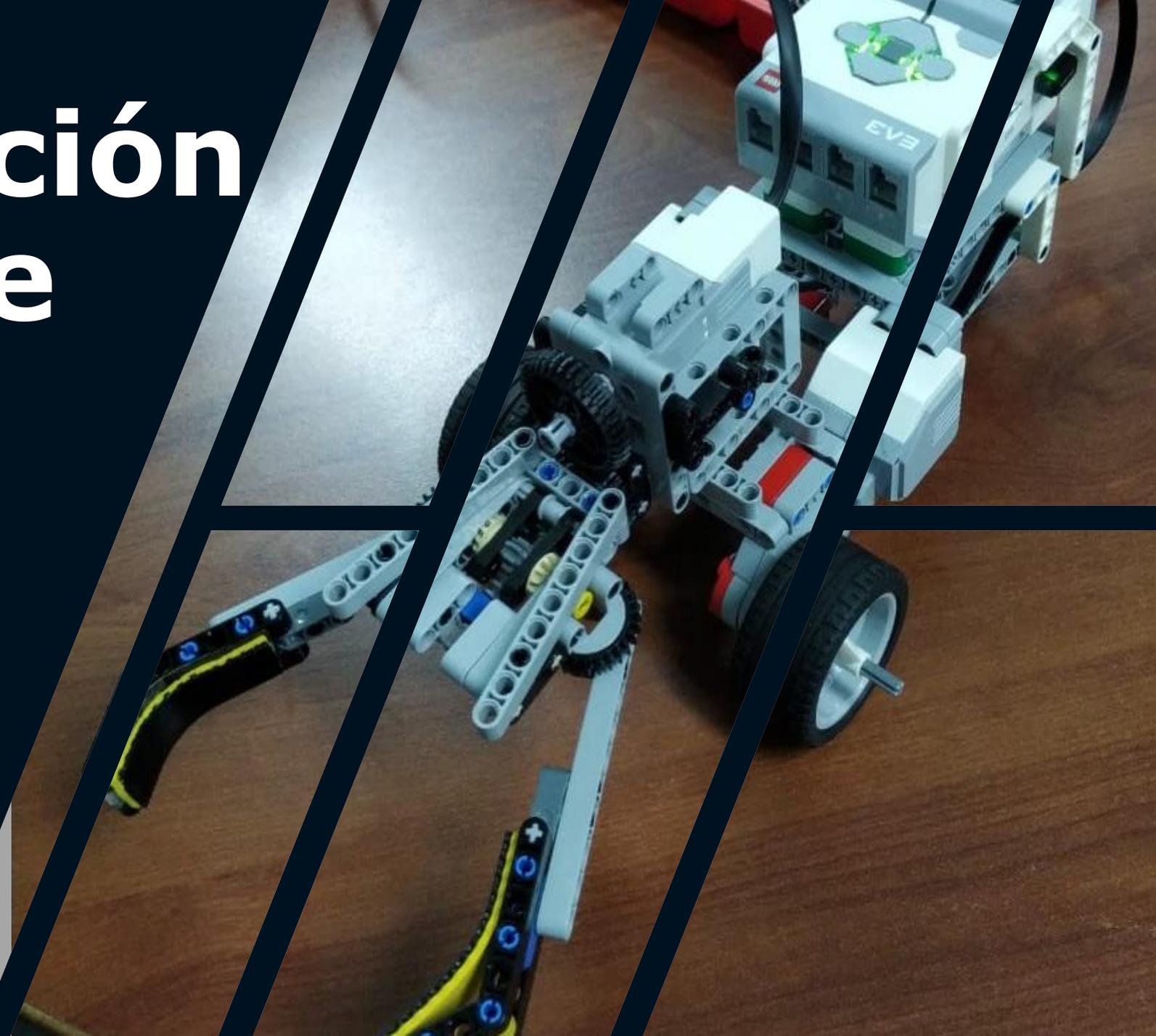
Presentación de avance

Integrantes:

- Ernesto García Ramírez
- Gustavo Olivares Henriquez
- Angelina Orozco Zúñiga
- Benjamín Poblete Araya
- Daniel Ramirez Diaz

Curso: Proyecto1

Profesor: Ricardo Valdivia Pinto



Índice

- Introducción
- Objetivo General
- Objetivos Específicos
- Asignación de tiempo
- Análisis-Diseño
- Implementación
- Resultados
- Conclusión



Introducción

El proyecto “Flip-tac-toe”, consiste en la creación de un robot hecho con piezas del kit lego Mindstorms EV3 education, que deberá participar en el juego llamado “Flip-Tac-Toe”.



Objetivo General

Construir un robot lego Mindstorms EV3 education que sea capaz de jugar flip-tac-toe.

Objetivo Especifico:

- Diseñar al robot para que sea capaz de moverse, tomar cosas y girarlas.
- Programar los movimientos necesarios del robot para lograr las acciones requeridas.
- Calibrar los movimientos del robot para verificar un correcto funcionamiento.
- Realizar una interfaz remota para controlar a distancia los movimientos del robot.
- Integrar y entregar al producto final para poder jugar Flip-tac-toe.



✓ Especificación de Requerimientos

Requerimientos funcionales:

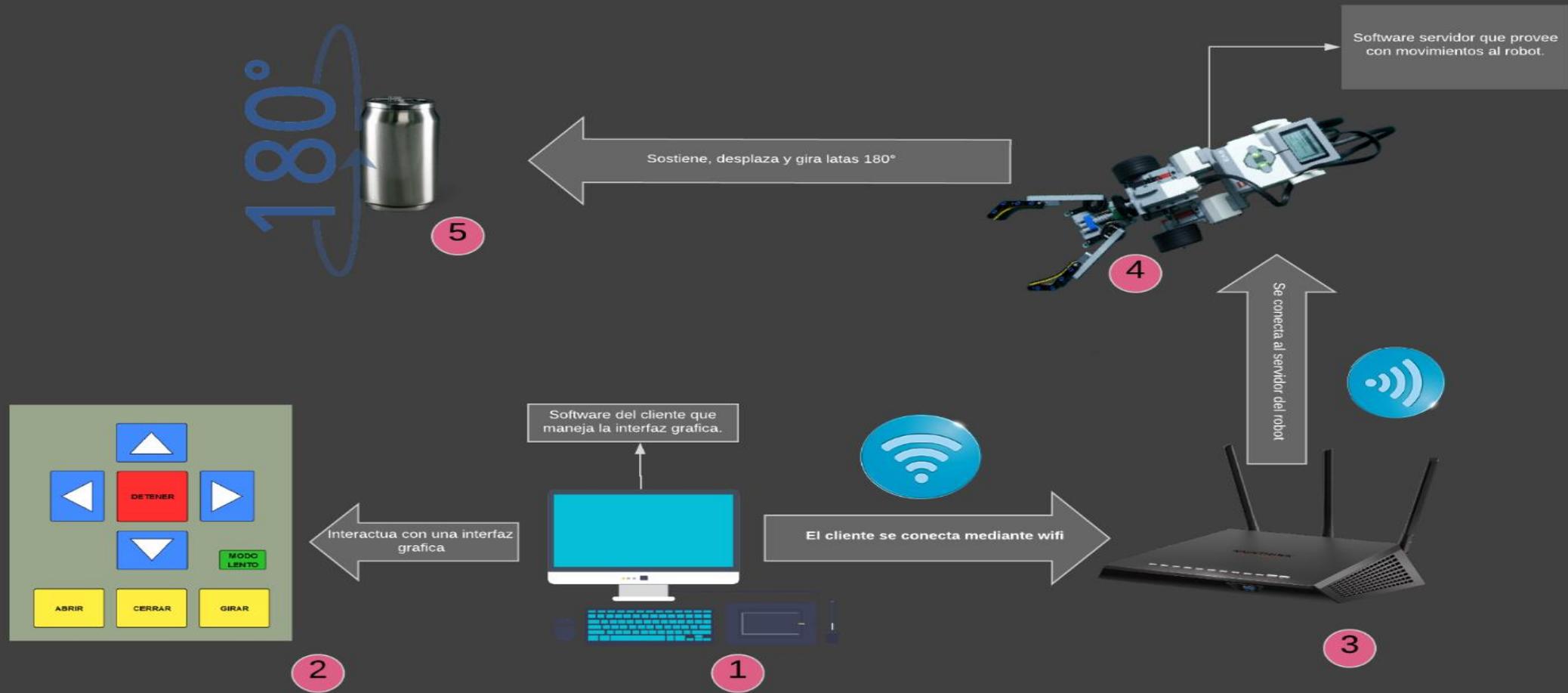
	<i>Descripción</i>	<i>Prioridad</i>	<i>Revisión</i>	<i>Aceptado</i>	<i>Realizado</i>
1	El robot deberá poder moverse a voluntad del usuario mediante una interfaz en pc/smartphone.	1	26/09/2019	✓	✓
2	El robot debe tener la capacidad de tomar una lata con su brazo al presionar un botón específico.	1	26/09/2019	✓	✓
3	El robot debe tener la capacidad de rotar una lata en su brazo al presionar un botón específico.	1	26/09/2019	✓	✓
4	El robot debe tener la capacidad de soltar una lata con su brazo al presionar un botón específico.	1	10/10/2019	✓	✓
5	El robot debe ser capaz de jugar "Flip-Tac-Toe".	1	17/10/2019	✓	✓ ⁶

✓ Especificación de Requerimientos

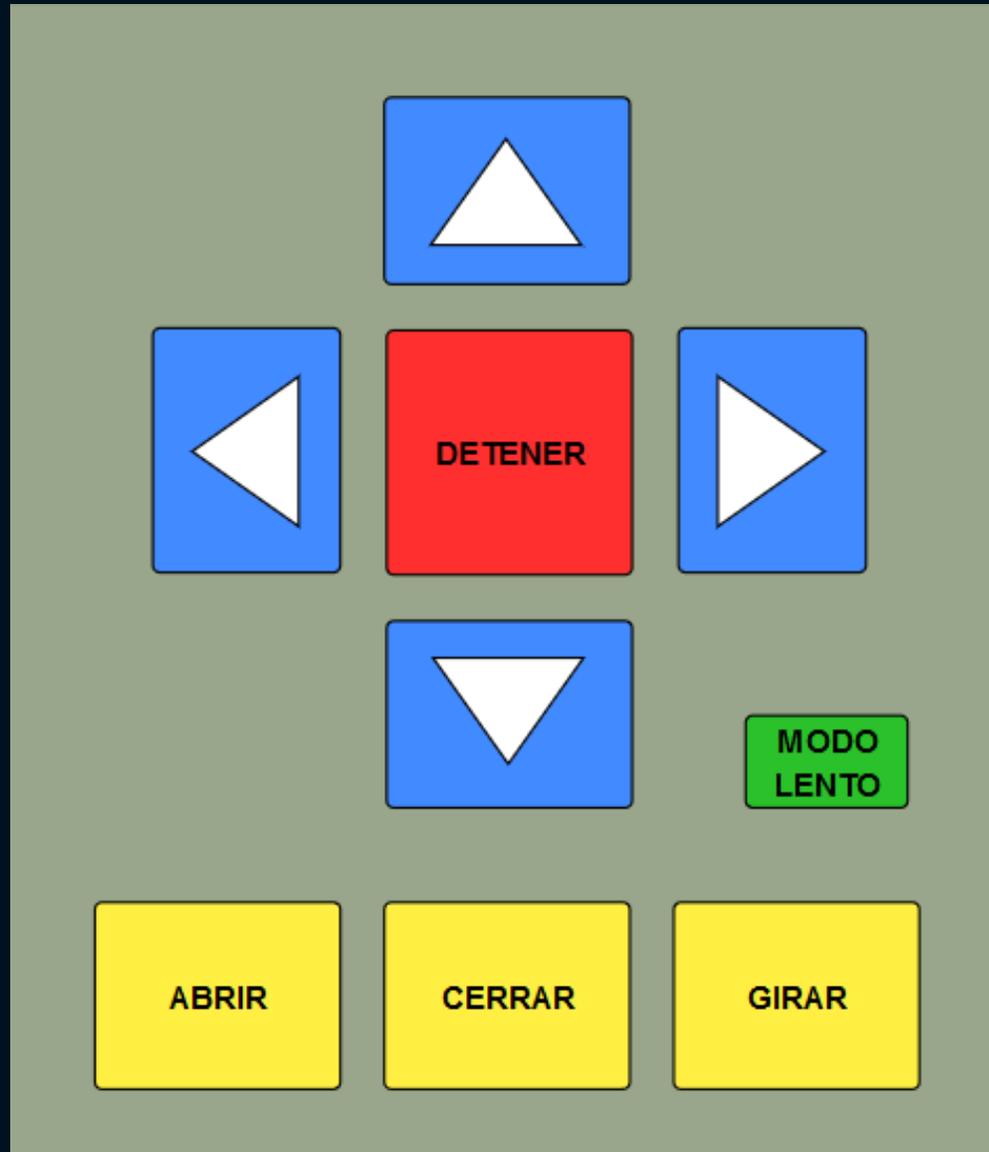
Requerimientos no funcionales:

	<i>Descripción</i>	<i>Prioridad</i>	<i>Revisión</i>	<i>Aceptado</i>	<i>Realizado</i>
6	La programación debe estar hecha en Python.	1	26/09/2019	✓	✓
7	El robot debe poder tomar una lata en un tiempo, como máximo, de cinco segundos.	2	26/09/2019	✓	✓
8	El robot debe poder girar una lata mientras la toma en un tiempo, como máximo, de cinco segundos.	2	26/09/2019	✓	✓
9	El robot debe estar hecho con piezas del kit lego Mindstorm EV3 education.	1	01/10/2019	✓	✓

Arquitectura propuesta



Diseño de la interfaz de usuario



Esta es la interfaz de movimiento del robot, en la parte superior hay cuatro botones que al ser presionados generarán el movimiento del robot.

El botón "Detener" en medio de estos sirve para interrumpir cualquier actividad.

Los tres botones inferiores sirven para utilizar la garra.

Por último, el botón "MODO LENTO" sirve para que los movimientos del robot sean más lentos.

Implementación

```
class robservice(Service):
    def exposed_add(self, x, y):
        return x + y
```

* All other names (not prefixed by ``exposed``) are local (not accessible to the other party)

.. note::
You can override ``_rpyc_getattr``, ``_rpyc_setattr`` and ``_rpyc_delattr`` to change attribute lookup -- but beware of possible ****security implications!****

Method resolution order:

[MyService](#)
[rpyc.core.service.Service](#)
[builtins.object](#)

Methods defined here:

exposed_Abrir(self, outga_b)
[exposed_Abrir](#)(self, outga_b)

Hace abrir la garra del robot.

Parametros

self : [MyService](#)
es una referencia a la instancia de [MyService](#) creada a partir de la clase.

outga_b : string
es el nombre del motor que se ocupa para cerrar/abrir la garra del robot.

exposed_Avanzar(self, outa, outd)
[exposed_Avanzar](#)(self, outa, outd)

Hace avanzar el robot.

Parametros

self: [MyService](#)
es una referencia a la instancia de [MyService](#) creada a partir de la clase.

outa: string
es el nombre del motor de la rueda derecha que se va a ocupar.

outd: string

Funcions

Abrir(event)

[Abrir](#)(event)

Activa la funcion Abrir del robot a traves de un evento del teclado.

event: evento
recibe un evento del teclado.

Avanzar(event)

[Avanzar](#)(event)

Activa la funcion Avanzar del robot a traves de un evento del teclado.

event: evento
recibe un evento del teclado.

Cerrar(event)

[Cerrar](#)(event)

Activa la funcion Cerrar del robot a traves de un evento del teclado.

event: evento
recibe un evento del teclado.

Derecha(event)

[Derecha](#)(event)

Activa la funcion Derecha del robot a traves de un evento del teclado.

event: evento
recibe un evento del teclado.

Detener(event)

[Detenet](#)(event)

Activa la funcion Detener del robot a traves de un evento del teclado.

Resultados

ESTADO ACTUAL DEL PROYECTO

Actualmente el proyecto cuenta con:

- La versión final del robot “R.S.B.A.”.
- Las funciones de movimientos ya están implementadas.
- La wiki del proyecto
- La conexión de una interfaz del robot con la ayuda de la librería RPyC en ev3dev/Python.
- La programación del apartado visual

PROBLEMAS ENCONTRADOS Y SOLUCIONES PROPUESTAS

Problemas Encontrados

Soluciones

La garra del robot estaba muy alta por lo que no le era posible tomar una lata pequeña.

Se cambia el diseño del robot de manera que la garra quede a la altura de la lata.

Es un principio, el sistema de rotación no era capaz de girar la garra .

Se cambió el diseño del Sistema de rotación de manera que la garra gire en torno a su propio eje.

En un principio se intento desarrollar los algoritmo en Visual Studio, sin embargo, no se pudo.

Se trabajó en un editor de texto directamente en el robot.

Al momento de rotar la garra, por su diseño, se abría, por lo que la lata se soltaba automáticamente.

Se compensó su apertura cerrándolo al momento de la rotación.



CONCLUSIÓN