**UNIVERSIDAD DE TARAPACÁ**



**ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL, INFORMÁTICA Y DE SISTEMAS**



Área de Ingeniería en Computación e Informática



**Plan de proyecto  
War Machine Cube Destroyer**

**Autor(es): Cristian Bautista M.**

**Christian Cáceres M.**

**Felipe López C.**

**Alan Ortega G.**

**Asignatura: Proyecto I**

**Profesor(es): Ricardo Valdivia**

**Diego Aracena Pizarro**

ARICA, DÍA MES AÑO

# Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 16/08/2018 | 1.1 | Desarrollo del panorama general | Alan Ortega |
| 28/08/2018 | 1.3 | Organización del Personal | Alan Ortega |
| 06/09/2018 | 1.5 | Planificación del Proyecto | Alan Ortega |
| 06/09/2018 | 1.7 | Planificación de Recursos | Alan Ortega |
| 06/09/2018 | 1.9 | Referencias | Alan Ortega |

# Tabla de Contenidos

1. Panorama General
   1. Introducción (contexto).
   2. Objetivo General.
   3. Objetivos Específicos.
   4. Restricciones.
   5. Entregables.
2. Organización del Personal

2.1. Descripción de Roles.

2.2. Personal que cumplirá los Roles.

2.3. Mecanismos de Comunicación.

1. Planificación del Proyecto

3.1. Actividades (nombre, descripción, responsable, producto).

3.2. Asignación de tiempo (carta Gantt Redmine).

3.3. Personal-rol asignado.

3.3. Gestión de Riesgos (ver plantilla para el Tratamiento de los Riesgos).

1. Planificación de los Recursos

4.1. Recursos Hardware-Software requeridos.

4.2. Estimación de Costos (Hardware, Software, Recursos Humanos).

1. Análisis – Diseño

5.1 Especificación de Requerimientos (incluyendo método/algoritmos considerados para resolver el cubo Rubik).

5.2 Arquitectura Propuesta (incluyendo aspectos de comunicación).

5.3 Diseño de la Interfaz Usuario.

1. Implementación

6.1 Descripción de los programas implementados (entradas, salidas, procesos)

6.2 Diagrama de interacción entre programas

1. Pruebas

7.1 Descripción de las pruebas realizadas

7.2 Resultados de las pruebas

1. Resultados

8.1 Estado Final del Proyecto

8.2 Conclusiones

8.3 Trabajo Futuro

1. Referencias (estándar IEEE)

Anexo A: Código de los programas implementados

Anexo B: Robot (diagrama de construcción, componentes principales)

1. **Panorama General**

# Introducción

Esta ultima década ha sido un gran avance para la tecnología en cuanto a la robótica, creemos que es un punto fuerte para desenvolver nuestras capacidades, por lo tanto, elegimos este tema y lo combinamos con uno de los mejores inventos para desarrollar tu intelecto, el gran cubo Rubik’s.

Se armará un robot a base de piezas legos, el cual tendrá la capacidad de resolver un cubo de Rubik’s mediante algoritmos generados por los integrantes. Dicho robot se desempeñará a través de un computador en el cual mediante una interfaz de usuario ejecutará algoritmos de resolución para el armado del cubo Rubik’s. Además, el producto trae un manual de usuario, el cual contiene indicaciones de instalación y uso, para una utilización adecuada del producto.

# Objetivo General

Armar un robot EV3 capaz de ejecutar algoritmos de resolución para un cubo Rubik’s 3x3x3 a través de un programa diseñado para controlar el hardware del WarMachine: Cube Destroyer.

# Objetivos Específicos

* Diseñar y armar un robot que pueda manipular un cubo Rubik’s.
* Desarrollar un programa en lenguaje Python que logre realizar los algoritmos traducidos para la resolución de un cubo Rubik’s
* Crear una aplicación capacitada para enviar instrucciones remotamente hacia el robot EV3.

# Restricciones

* El tiempo asignado para realizar el proyecto a lo largo del semestre.
* El robot EV3 es el único que se puede utilizar para programar nuestros algoritmos.
* Sólo se dispone de los movimientos que proporcionan los motores para la resolución del cubo Rubik’s.

# Entregables

* Informe Plan de Proyecto
* Informe Final
* Producto Final
* Bitácoras
* Riesgos
* Manual de Usuario

1. **Organización del Personal**

# Descripción de Roles

* Programador: Tiene la responsabilidad de traducir los algoritmos del cubo Rubik’s a lenguaje Python, lograr que el robot EV3 realice las acciones correspondientes y también se encarga del avance de la WIKI.
* Constructor: Tiene la responsabilidad de crear una lista con las piezas faltantes del robot EV3 para el armado completo de éste.
* Secretario: Tiene la responsabilidad de dar avance y desarrollo a los informes, carta Gantt y Bitácora Semanal.
* Líder: Tiene la responsabilidad de organizar y otorgar apoyo a las distintas áreas de trabajo dentro del proyecto.

# Personal que cumplirá los roles

* Programador: Alan Ortega G. y Cristian Cáceres M.
* Constructor: Cristian Bautista M.
* Líder: Felipe López C.
* Secretario: Alan Ortega G.

# Mecanismos de Comunicación

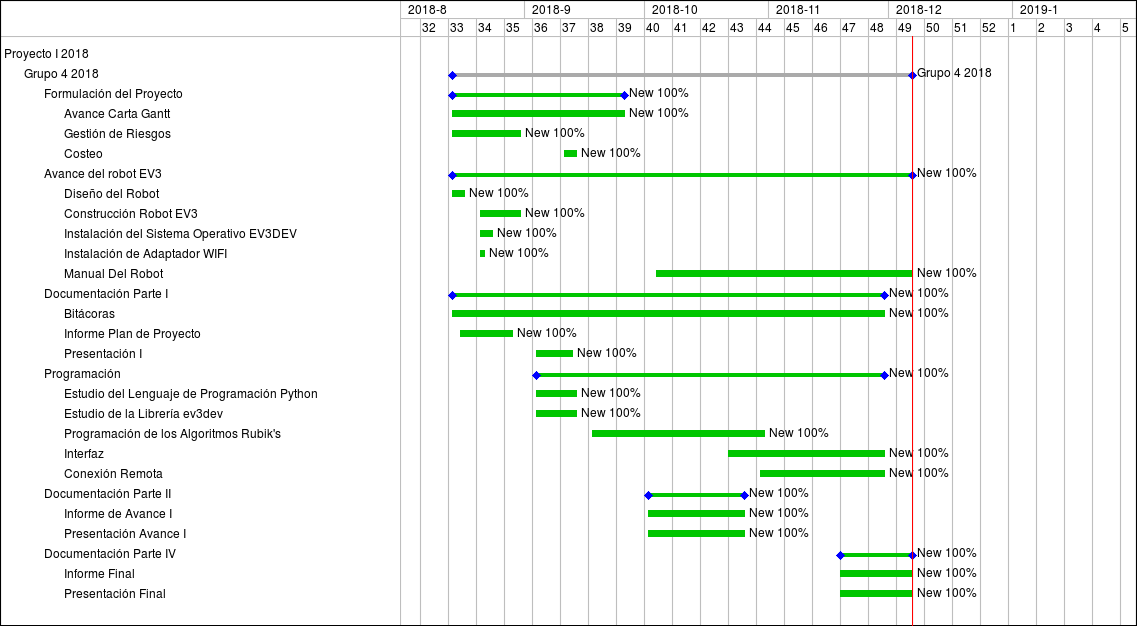
Para poder obtener una comunicación adecuada se decidió crear un grupo de WhatsApp administrado por el líder, para poder resolver problemas encontrados en clases y poder solucionarlos, entonces se llegará a la otra sesión con el material adecuado.

1. **Planificación del Proyecto**

# Actividades

* Armar robot EV3.
* Instalar sistema operativo en la tarjeta SD para trabajar con el robot EV3.
* Configurar conexión WIFI mediante un adaptador USB Tenda.
* Instalar programas necesarios para la programación y comunicación con el robot EV3.
* Diseñar e implementar algoritmos que resuelvan el cubo de Rubik’s.
* Estudiar e implementar un sistema de comunicación para controlar el robot remotamente.

# Asignación de Tiempo



|  |  |
| --- | --- |
| Responsable a cargo | Actividad |
| Alan Ortega | Instalación de programas, configuración Wifi, Desarrollo del Plan de Proyecto y programación en Python |
| Felipe López | Presentación y lista de riesgos |
| Cristian Cáceres | Traducción de algoritmos del cubo Rubik’s, avance en Redmine (WIKI) e instalar Sistema Operativo EV3DEV Stretch Beta |
| Christian Bautista | Construcción robot EV3 y diseño de la interfaz |

# Personal-rol Asignado

# Gestión de Riesgos

|  |  |
| --- | --- |
| CATEGORIA DE RIESGOS | |
| 1 | CATASTROFICO |
| 2 | CRÍTICO |
| 3 | MARGINAL |
| 4 | DESPRECIABLE |

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgos | Probabilidad de ocurrencia | Nivel de impacto | Acción remedial |
| Perdida de información | 60% | 1 | Hacer un respaldo de la información correspondiente continuamente en los dispositivos del equipo. |
| Enfermedades/Accidentes | 40% | 3 | Repartir tareas del integrante en cuestión al equipo y el integrante cuando se reintegre necesitara adelantar trabajo. |
| Falta de herramientas para desarrollar el proyecto | 30% | 3 | Buscar mediante otros medios estas herramientas. |
| Falta de piezas para el robot ev3 | 70% | 2 | Comprar las piezas faltantes para el robot en el mercado. |
| Daño a tarjeta de SD del robot | 60% | 2 | Hacer el respaldo debido en algún dispositivo del equipo. |
| Falla del software necesario | 30% | 3 | Buscar mediante internet otras opciones similares al software necesario. |
| Errores de programación | 50% | 3 | Buscar errores en internet para su debida resolución. |
| Abandono de integrantes | 30% | 2 | Tomar las tareas del integrante y repartir deberes a los integrantes del equipo. |

1. **Planificación de Recursos**

# Recursos Hardware-Software Requeridos

Hardware:

* Robot Lego EV3
* Adaptador WIFI Tenda
* Cubo Rubik’s
* Memoria SD

Software:

* Sistema Operativo EV3DEV – STRETCH BETA
* Visual Studio Code
* Putty (SSH)
* Python 3.7.0

# Estimación de Costos

* Robot Lego EV3 - $490.209 CLP
* Cubo Rubik’s - $15.000 CLP
* Micro/Adapater SD ADATA 8GB - $3.550 CLP
* Adaptador WIFI Tenda - $6.000 CLP

# Recursos humanos

Para llevar a cabo el proyecto, se debió gestionar un aproximado de sueldo para el equipo, el cual llego al monto de $2.592.000 CLP por el periodo de desarrollo del proyecto, debido a que la hora de trabajo por cada integrante esta evaluada en $9.000 CLP, son 4,5 horas de trabajo a la semana y 16 semanas en las cuales se debe realizar el proyecto, el sueldo para cada integrante es de $648.000 CLP.

1. **Análisis – Diseño**

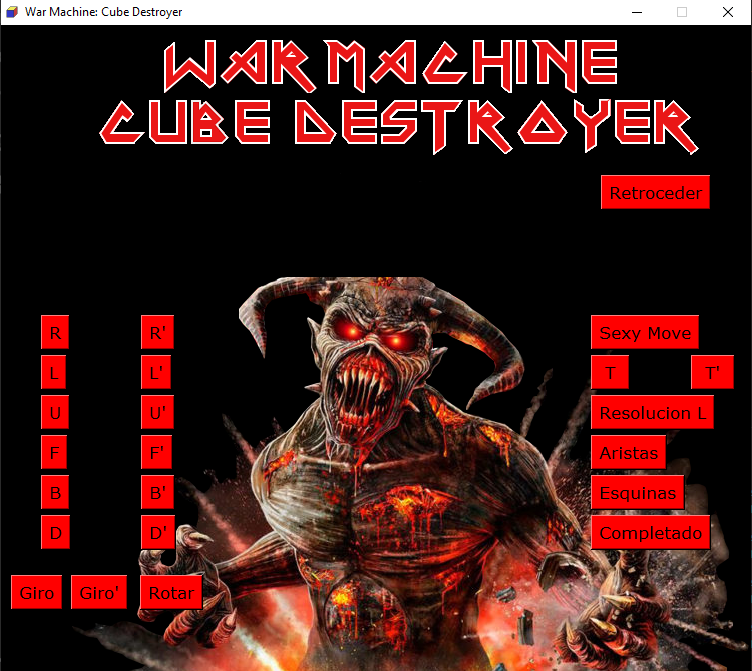
# Especificación de Requerimiento

* El usuario dispondrá de una interfaz en un computador, a través de esta podrá controlar el robot para que ejecute los algoritmos de resolución del cubo. Esta interfaz contiene imágenes descriptivas de cada movimiento en específico con su debido algoritmo de resolución y estará conectado con el Brick del robot vía wifi para que se puedan comunicar como es debido.
* Se implementará un robot que estará capacitado con motores para poder realizar los movimientos correspondientes al efectuar el armado del cubo mediante un brazo y una base.
* Se dispondrá de un manual de usuario con el cual se podrá proceder a utilizar la interfaz correctamente, señalando los algoritmos para armar el cubo.

# Arquitectura Propuesta

Imagen 1, muestra la arquitectura propuesta.

# Diseño de la Interfaz de Usuario



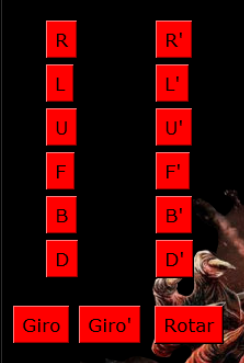
****

Imagen 2, Interfaz de Usuario

-En esta parte de la interfaz se encuentran los botones que ejecutan los movimientos básicos para armar el cubo.

-Los últimos tres botones son movimientos de los motores del robot (base y brazo).

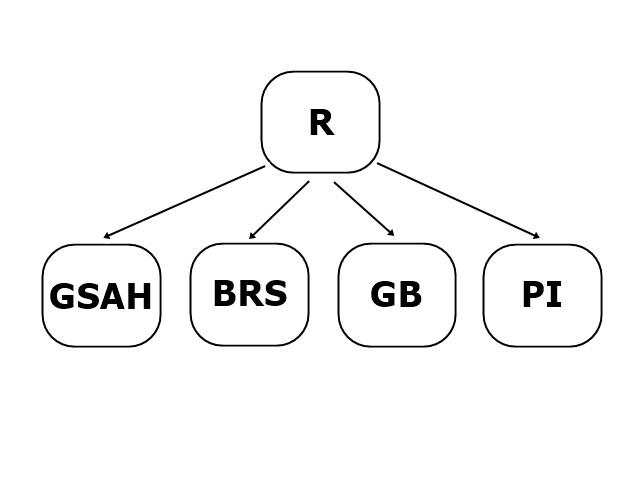
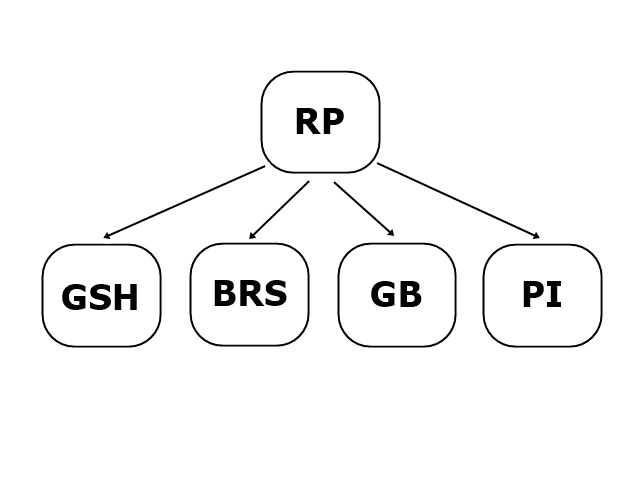
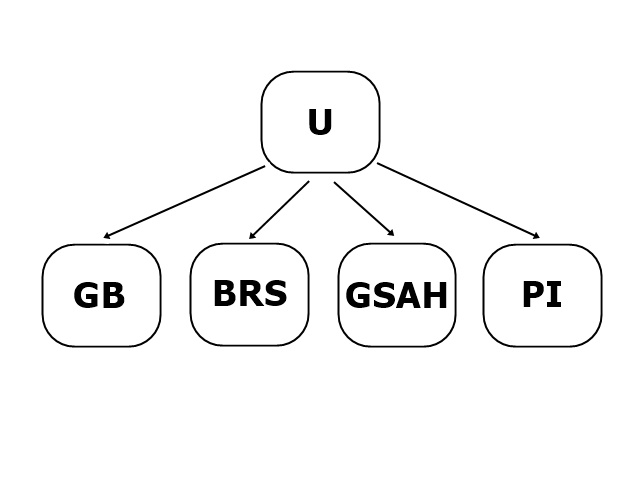
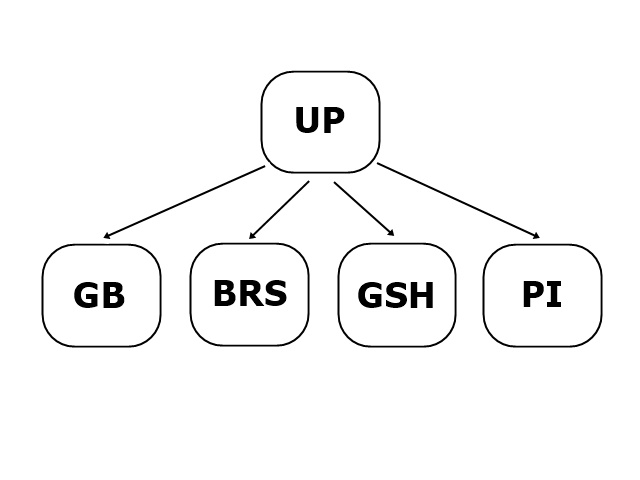
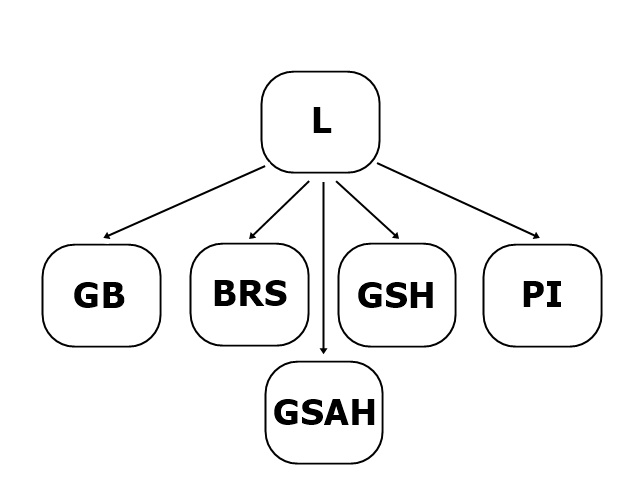
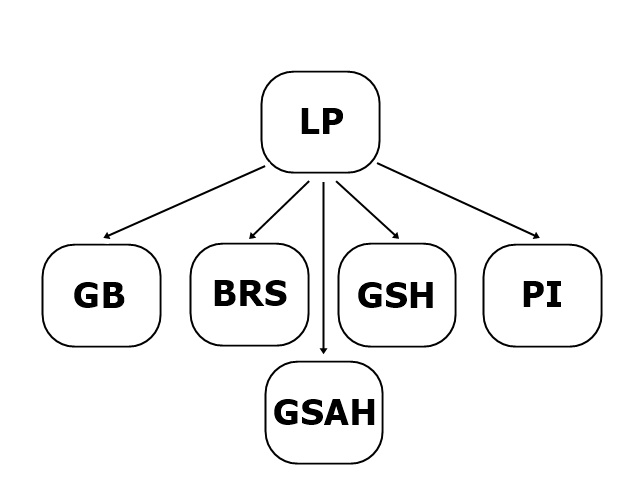
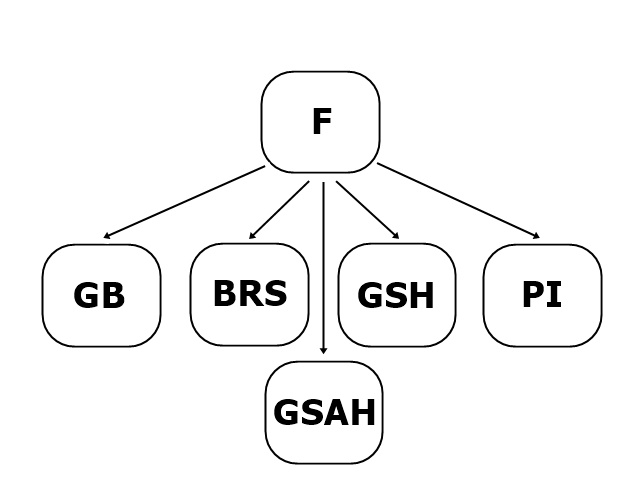
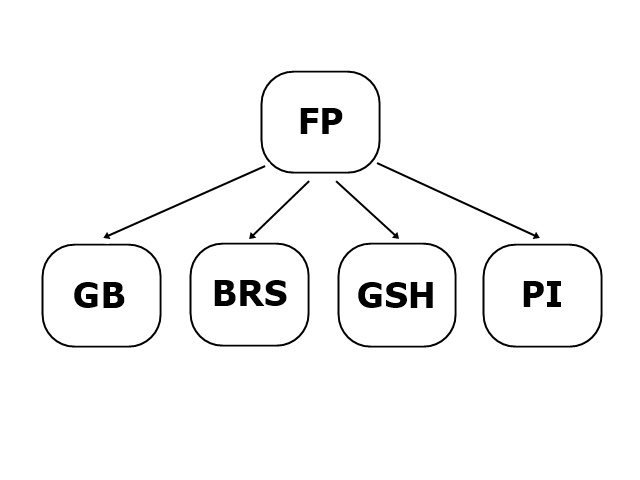
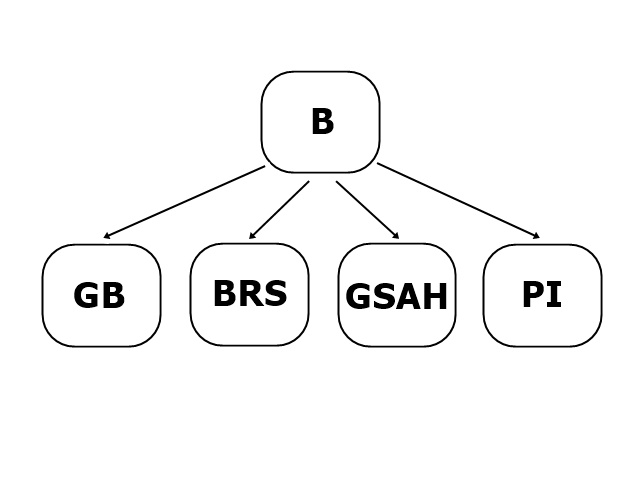
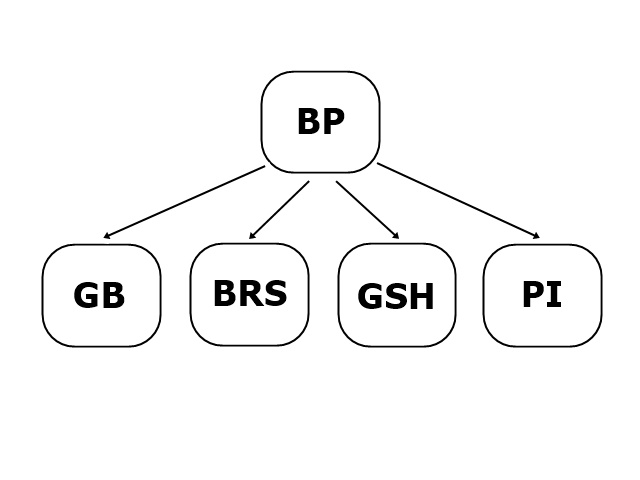
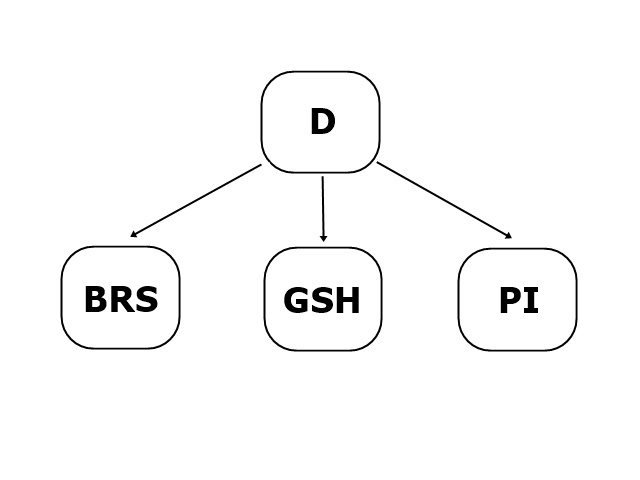
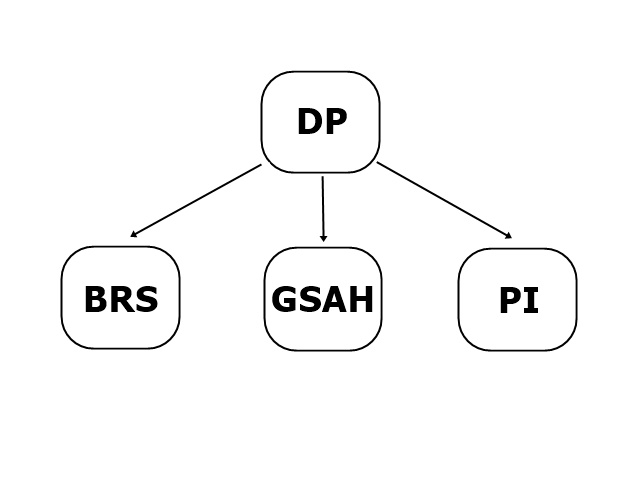
-El botón retroceder deshace el movimiento anterior.

****

-En esta otra parte se encuentran los movimientos complejos para el armado del cubo.

1. **Implementación**

# Descripción de los Programas Implementados y Diagrama de Interacción entre Programas

* GSAH: Función que gira el cubo en sentido anti-horario.
* GSH: Función que gira el cubo en sentido horario.
* BRS: Función que ordena al motor del brazo trabajar como un soporte.
* GB: Función que ordena al motor del brazo girar el cubo Rubik’s.
* PI: Función que ordena al motor del brazo regresar a la posición inicial.
* R(): Esta función realiza el movimiento R en sentido horario de los algoritmos básicos del cubo Rubik’s.
* RP(): Esta función realiza el movimiento R en sentido anti-horario de los algoritmos básicos del cubo Rubik’s.
* U(): Esta función realiza el movimiento U en sentido horario de los algoritmos básicos del cubo Rubik’s.
* UP(): Esta función realiza el movimiento U en sentido anti-horario de los algoritmos básicos del cubo Rubik’s.
* L(): Esta función realiza el movimiento L en sentido horario de los algoritmos básicos del cubo Rubik’s.
* LP(): Esta función realiza el movimiento L en sentido anti-horario de los algoritmos básicos del cubo Rubik’s
* F(): Esta función realiza el movimiento F en sentido horario de los algoritmos básicos del cubo Rubik’s.
* FP(): Esta función realiza el movimiento F en sentido anti-horario de los algoritmos básicos del cubo Rubik’s.
* B(): Esta función realiza el movimiento B en sentido horario de los algoritmos básicos del cubo Rubik’s.
* BP(): Esta función realiza el movimiento B en sentido anti-horario de los algoritmos básicos del cubo Rubik’s.
* D(): Esta función realiza el movimiento D en sentido horario de los algoritmos básicos del cubo Rubik’s.
* DP(): Esta función realiza el movimiento D en sentido anti-horario de los algoritmos básicos del cubo Rubik’s.

1. **Pruebas**

# Descripción de pruebas realizadas

**-Prueba de movimiento de los motores del robot:**

Se configuraron los motores de tal manera que se tuvo que calibrar sus ángulos y velocidades con el software pertinente, para que ejecutara movimientos concretos lo cual más adelante nos ayudara a realizar movimientos simples y complejos para el armado del cubo.

**-Prueba de movimientos simples:**

Se programaron los movimientos simples los cuales son fundamentales para el armado del cubo y se configuraron de tal manera que el robot ejecute los algoritmos de resolución. Las pruebas realizas a estos movimientos simples fueron exitosas, ya que el giro de las caras del cubo se concretó de manera correcta se prosiguió a la realización de movimientos más complejos.

**-Prueba de movimientos complejos:**

Cerciorarse que los movimientos más complejos sean los que ocupan los dos motores y que estos funcionen correctamente.

El termino de los movimientos simples nos abrió camino hacia el desarrollo de algoritmos más complejos para resolver el cubo, que consisten en una secuencia de movimientos simples, con esto logramos que el robot este habilitado para poder ayudar a armar el cubo Rubik’s.

# Resultado de las pruebas

Los resultados de las pruebas realizadas se mostrarán en video con los respectivos links.

**-Prueba de movimientos simples:**

[**https://mega.nz/#!4EY01AbD!q7DXd-xwq91oOI7CZgV7ObvgtR8V9FmeOYMfa3DrPcM**](https://mega.nz/#!4EY01AbD!q7DXd-xwq91oOI7CZgV7ObvgtR8V9FmeOYMfa3DrPcM)

[**https://mega.nz/#!0BAQnSIR!AzQ4L9QF43aMnXWvta7naJmjv4A4LcY\_tpD9EL10910**](https://mega.nz/#!0BAQnSIR!AzQ4L9QF43aMnXWvta7naJmjv4A4LcY_tpD9EL10910)

**8. Resultados**

# Estado Actual del Proyecto

El robot se encuentra completamente terminado, realiza de manera efectiva los algoritmos de resolución para el armado del cubo Rubik’s. Estos algoritmos se ejecutan a través de la interfaz gráfica, la cual está finalizada y funcional. Se terminó el manual de usuario en el cual se indica paso por paso como instalar y usar nuestro producto.  
 Se realizaron las tareas pendientes de la carta Gantt y se actualizo esta misma, se concretaron los informes, presentaciones, bitácoras, etc.

# Conclusiones

Se finalizó de manera correcta el proyecto, solucionamos los problemas que se presentaron a lo largo del semestre. Desarrollamos los conocimientos necesarios para la construcción, programación y el desarrollo de la interfaz gráfica de un proyecto básico de ingeniero.  
 Se finalizó el producto, el cual consta de todas las condiciones necesarias para llegar al mercado del cubo Rubik.

# Trabajo a Futuro

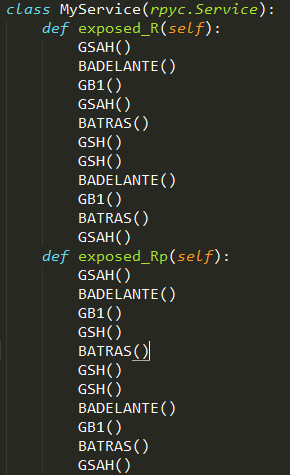
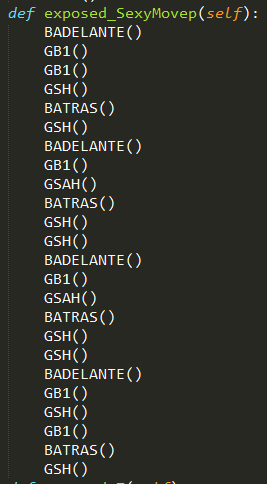
Se pensó en perfeccionar el robot para que el producto sea una herramienta esencial para el jugador promedio y que además este sea capaz de estar al nivel de la competencia del mercado. Una vez dentro de este se quiere desarrollar distintos tipos de robots, en el cual siga siendo una herramienta para el jugador, pero con mejores características tanto en la construcción y programación del robot. Por otra parte, se pretende desarrollar un robot que pueda armar de manera autónoma el cubo y en el menor tiempo posible.

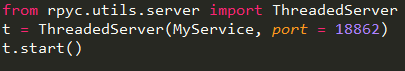
1. **Referencias**

<https://python-ev3dev.readthedocs.io/en/ev3dev-stretch/motors.html#ev3dev.motor.Motor.on_for_degrees>

Anexo A:







Anexo B:

El robot dispondrá de las siguientes piezas, las cuales permiten que se arme el cubo rubik’s:

* Base o plataforma: Permite girar completamente (o lo solicitado) de forma horizonal el cubo rubik’s.
* Brazo o gancho: Permite girar verticalmente las veces que sean necesarias el cubo rubik’s.

