**UNIVERSIDAD DE TARAPACÁ**



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



Área de Ingeniería en Computación e Informática



**Plan de proyecto**

**C.A.M.**

**Autor(es): Nicolas Jorquera**

**José-Ignacio Leblanc**

**Matias Sandoval**

**Luis Soto**

**Asignatura: Proyecto 1**

**Profesor(es): Diego Aracena**

**Ricardo Valdivia**

ARICA, 23 OCTUBRE 2018

# Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 06-09-2018 | 1.0 | Primera formulación de Proyecto | Matias Sandoval |
| 24-10-2018 | 1.1 | Segunda formulación de Proyecto | Nicolás Jorquera  José-Ignacio LeBlanc  Matias Sandoval  Luis Soto |

# 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. Resultados

7.1Estado actual del proyecto

7.2Problemas encontrados y soluciones propuestas

7.3 Conclusiones

1. Referencias(estándar IEEE)

**1. Panorama General**

**1.1 Introducción**

En 1974 se creó un famoso juego interactivo llamado cubo de Rubik y desde entonces ha formado parte de la vida de muchas personas, llegando incluso al punto de haber competencias. Además de aflorar la imaginación de las personas explorando diversas combinaciones de giros, permitiéndoles formar patrones.

Estamos acostumbrados a realizar dichos patrones con algoritmos efectuados con nuestras manos, pero el reto de este proyecto es hacer que un robot sea capaz de hacerlos, recibiendo las instrucciones desde una aplicación.

**1.2 Objetivo General**

Construir un robot con el set Lego EV3, capaz de sostener un cubo de Rubik. Desarrollar programas con el lenguaje Python, en los cuales estén los patrones para que el robot los realice. Además de desarrollar una aplicación para interactuar con dicho robot, enviándole las instrucciones pertinentes para ejecutar los patrones.

**1.3 Objetivos Específicos**

-Estudiar el lenguaje Python.

-Armar el robot Lego “EV3”.

-Realizar prueba de la estructura del robot “EV3”.

-Implementar en la SD el sistema operativo.

-Realizar programa en Python, en el cual definir los movimientos del robot

“EV3”.

-Evaluar si la arquitectura del robot reacciona de buena forma a los

movimientos.

-Evaluar y seleccionar la opción más factible para desarrollar un software mediante el cual poder interactuar con el robot y poder ejecutar algoritmos

sobre el cubo.

-Realizar en conjunto al punto anterior: Adaptar los movimientos usados en dichos algoritmos, debido a las limitaciones de movimiento del robot.

-Mejorar visualmente la aplicación.

-Últimas pruebas de funcionamiento del robot en conjunto de la aplicación.

**1.4 Restricciones**

Para llevar a cabo el proyecto se cuenta con un set de Lego EV3, un cubo Rubik 3x3. La conexión remota debe desarrollarse Python. Posteriormente se buscará implementar la estética de la aplicación.

El tiempo para trabajar en el proyecto durante la semana de manera presencial es de 9 horas, y el tiempo de duración del mismo es hasta que finalice el segundo semestre del presente año.

Una vez puesto el cubo Rubik desarmado en la base del robot “EV3”, el usuario no podrá tocar el cubo Rubik con sus propias manos para realizar movimiento alguno, estos sólo podrán ser realizados mediante la forma de conexión remota.

**1.5 Entregables**

-Bitácoras.

-Robot “EV3”.

-Aplicación u otro método de conexión remota.

-Código Fuente del robot.

-Manual de uso.

-Formulación de proyecto.

-Wiki.

1. **Organización Del Personal**

**2.1 Descripción de Roles**

**Programador:** Escribe, depura y mantiene el código fuente del programa actualizado, libre de errores. Con el cual trabajará el robot “EV3”.

**Ensamblador:** Organizador y montador de la estructura del robot “EV3”,

**Planificador:** Ordenar, guardar y enviar los documentos para llevar un orden en las horas de trabajo grupal.

**Diseñador:** Encargado de la estética de la aplicación.

**Creador Conexión Remota:** Encargado de conectarse al robot “EV3”, por medio de una conexión remota.

**Creador de Algoritmos:** Adapta los algoritmos para ser realizados por el robot “EV3”.

**Desarrollador Aplicación:** Escribe y desarrolla la aplicación para la comunicación con el robot “EV3”.

**2.2 Personal que cumplirá los Roles**

**Programador:** José-Ignacio Leblanc.

**Ensamblador:** Luis Soto, José-Ignacio Leblanc.

**Planificador:** Matias Sandoval.

**Diseñador:** Luis Soto.

**Creador Conexión Remota:** Nicolas Jorquera.

**Creador de Algoritmos:** Matias Sandoval, José-Ignacio Leblanc.

**Desarrollador Aplicación:** Nicolas Jorquera, Luis Soto.

**2.3 Mecanismos de Comunicación**

**Discord:** Programa el cual se puede comunicar mediante videollamada y chat de voz, el cual fue ocupado para organizar las reuniones y algunos trabajos pendientes en clases.

**WhatsApp:** Aplicación para comunicarse mediante mensajes o audios de voz. Se utilizó para comunicarse de maneras más rápida y compartir detalles breves.

**Dropbox:** Software para almacenar datos en la nube. Se utilizó para subir la documentación de los avances del proyecto.

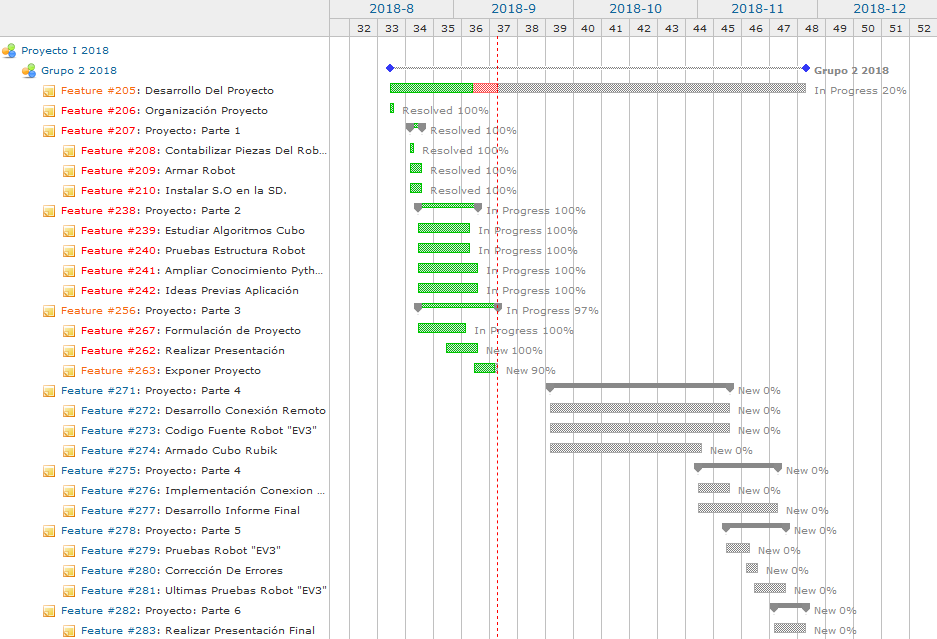
**Correo Electrónico:** Software para enviar archivos. Se utilizó de apoyo al momento que Dropbox no nos permitía subir algunos archivos.

1. **Planificación del Proyecto**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE** | **DESCRIPCION** | **RESPONSABLE** | **PRODUCTO** |
| Organización Proyecto | Distribución de tareas a los integrantes del grupo, donde cada uno dedicara su mayor tiempo. | Equipo de Trabajo | Realizar las actividades asignadas de forma organizada. |
| Contabilizar Piezas del Robot | Revisión de piezas contenidas dentro de la caja del Robot EV3, para lograr el objetivo de armado. | Luis Soto  Jose Le-Blanc | Disponer de las piezas correctas de forma ordenada. |
| Armar Robot | Armado del Robot, donde se realizará la estructura dependiendo de las dimensiones del cubo. | Luis Soto  Jose Le-Blanc | Robot Ev3. |
| Instalar S.O en la SD | Ingresar el sistema operativo al “brick” para lograr controlar el robot. | Nicolas Jorquera | Contar con el S.O en brick. |
| Estudiar Algoritmos del Cubo | Aprendizaje de los algoritmos del cubo Rubik, para lograr llegar a los patrones deseados. | Matias Sandoval  Jose Le-Blanc | Patrones del Cubo Rubik. |
| Pruebas Estructura Robot | Realizar movimientos con algoritmos básicos en Python, para comprobar que la estructura del robot responde de buena manera a dichos movimientos. | Jose Le-Blanc | Estructura optima del Robot, sin ningún detalle. |
| Estudiar Python | Aprendizaje de Python en la realización del programa fuente. | Jose Le-Blanc  Nicola Jorquera | Conocimiento de las librerías de Python. |
| Desarrollo conexión Remota | Establecer conexión entre el robot y la computadora, desde la cual se emitirán los patrones de armado. | Nicolas Jorquera | Conexión entre robot y la computadora. |
| Aplicación EV3 | Donde se realizará una aplicación, usando la conexión remota. | Luis Soto  Nicolas  Jorquera | Aplicación de la conexión remota. |
| Armado del cubo de Rubik | Armar el cubo Rubik, adaptándose los movimientos del robot. | Matias Sandoval | Traducción de los patrones “normales” a forma del robot. |
| Realizar Código Fuente EV3 | Programar los patrones del cubo Rubik previamente adaptados a los movimientos del robot. “Código fuente”. | José Le-Blanc  Matias Sandoval | Programa que contiene los algoritmos de patrones. |
| Pruebas Robot EV3 | Realizar las pruebas del robot, pero ahora con todos los patrones ya programados. | Jose Le-Blanc  Nicolas Jorquera  Matias Sandoval | Respuesta óptima del robot al ejecutarse los patrones en éste. |
| Realización de Documentos. | Se realizarán las bitácoras semanales, informes, etc. | Matias Sandoval | Un orden al realizar el proyecto. |

**3.2 Asignación de tiempo (carta Gantt Redmine)**



**3.3 Gestión de Riesgos (Ver plantilla para el tratamiento de los riesgos)**

|  |  |  |  |
| --- | --- | --- | --- |
| **RIESGOS** | **PROBABILIDAD DE OCURRENCIA** | **NIVEL DE IMPACTO** | **ACCIÓN REMEDIAL** |
| Problemas estructurales del robot en general. | 80% | 1 | Solucionar dichos problemas de la mejor manera. |
| El personal no cumple con las tareas en el tiempo estimado. | 40% | 2 | Intercambiar roles entre los mismos integrantes del grupo. |
| Un miembro del personal no asiste por problemas de salud o motivos personales. | 50% | 2 | Que otro miembro cubra la actividad asignada al personal ausente. |

**4. Planificación de los Recursos**

**4.1 Recursos Hardware-Software requeridos**

* **Hardware:**

-Robot “EV3” compañía de lego.

-Cubo Rubik.

-Notebook.

-Smartphone.

-Tarjeta SD.

-Dongle Wifi.

* **Software:**

-Visual Studio Code.

-Discord.

-Dropbox.

-Correo Electrónico.

-WhatsApp.

-App Inventor 2.

-Vi.

-Python.

-So ev3dev.

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

|  |  |
| --- | --- |
| **Hardware/Software/RH** | **Costos ($ CLP)** |
| Robot “EV3” | $490.209 |
| Cubo Rubik | $5.000 |
| Notebook’s | $600.000 |
| Celular | $150.000 |
| Tarjeta SD | $5.000 |
| Dongle wi-fi | $8.000 |
| HH Desarrollador Aplicación | $8.000 ($936.000) |
| HH Programador | $7.000 ($819.000) |
| HH Ensamblador | $4.000 ($468.000) |
| HH Planificador | $8.000 ($936.000) |
| HH Creador de Algoritmo | $4.000 ($468.000) |
| HH Diseñador | $2.000 ($234.000) |
| HH Creador Conexión Remota | $7.000 ($819.000) |
| Software de Apoyo | Libre De Costo |
| **TOTAL** | **$5.938.209** |

|  |  |
| --- | --- |
| **Fecha de inicio/Fecha de término** | **Fecha vacaciones Septiembre** |
| 16-08🡪 27-11 | 11-09 🡪 25-09 |

**\*SE CONSIDERARON 9 HORAS SEMANALES DE TRABAJO Y TRABAJAMOS 13 SEMANAS DANDO UN TOTAL DE 117 HORAS\*.**

**5. Análisis – Diseño**

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

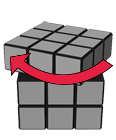
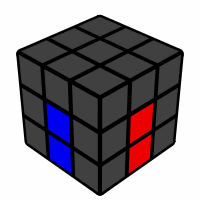
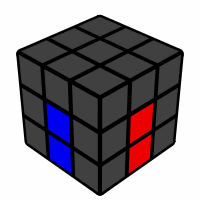
Requerimientos del Proyecto:

* Piezas de LEGO Mindstorms.
* Notebooks’s.
* Cubo Rubik.
* Guía de patrones Cubo Rubik (Armado y figuras).

Requerimientos del Usuario:

* Manual de Usuario.
* Notebook.
* Aplicación C.A.

Se consideraron 3 métodos para la formación de patrones y la formación del cubo Rubik mediante el robot, estos 3 se consideraron en un movimiento de 90°. Más que se consideraran 3 métodos, fueron las restricciones que propusimos en este proyecto.

1.  B) C)

**Patrones de figuras dentro del cubo Rubik.**

**Seis Lunares:**

A + 2C – A + C + A + 2C – A + B + C + A + 2C – A + B + C + A + 2C - A.

****

**Cubo en X:**

2A + 2C + 2A + B + C + 2A + 2C + 2A + B + C + 2A + 2C + 2A.

****

**5.2 Arquitectura**

Se dispone una pantalla con

***Robot***

Se obtiene

***Patrón Seleccionado***

Se establece conexión

***Aplicación***n

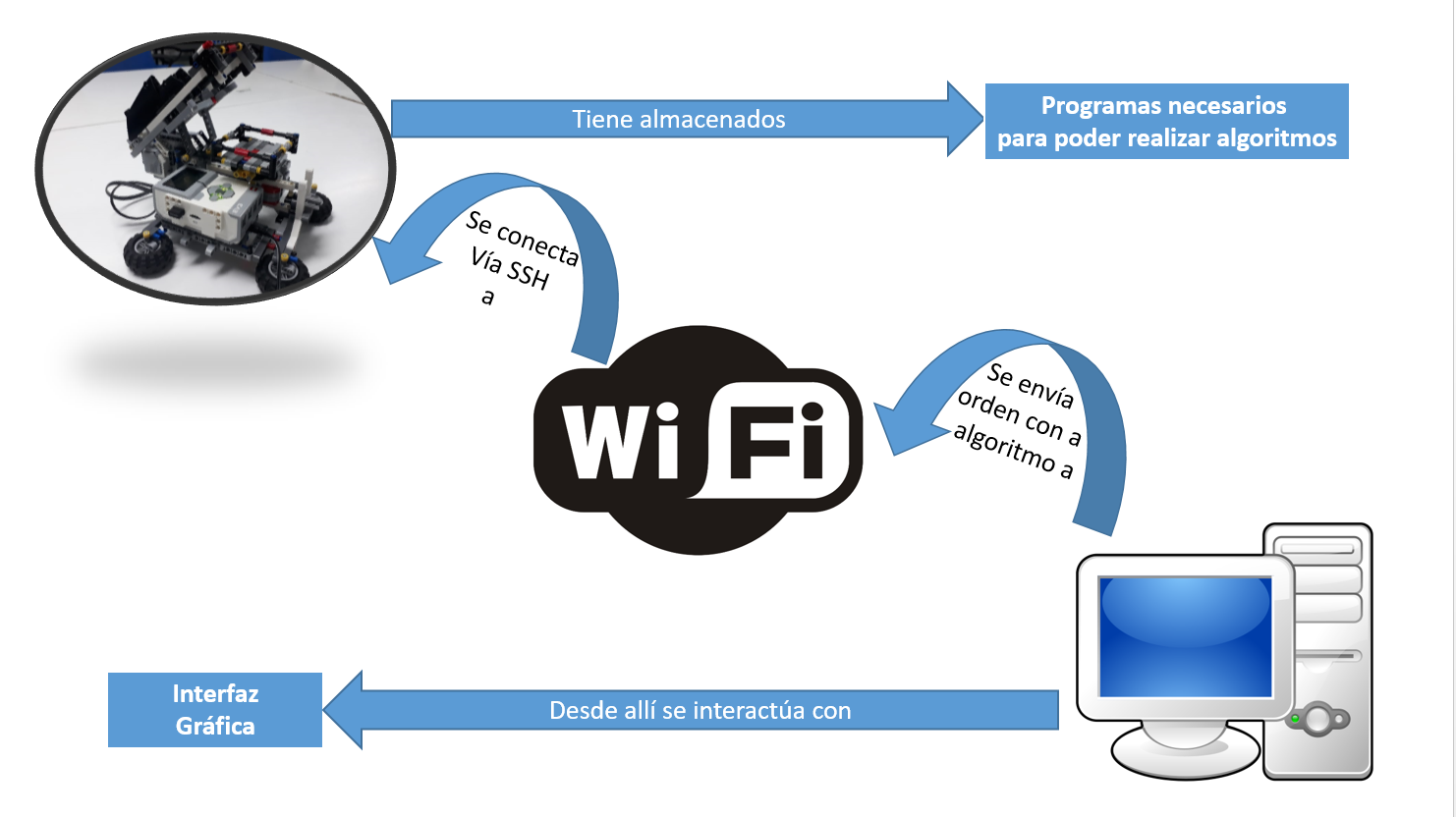
***Cubo con dicho patrón***

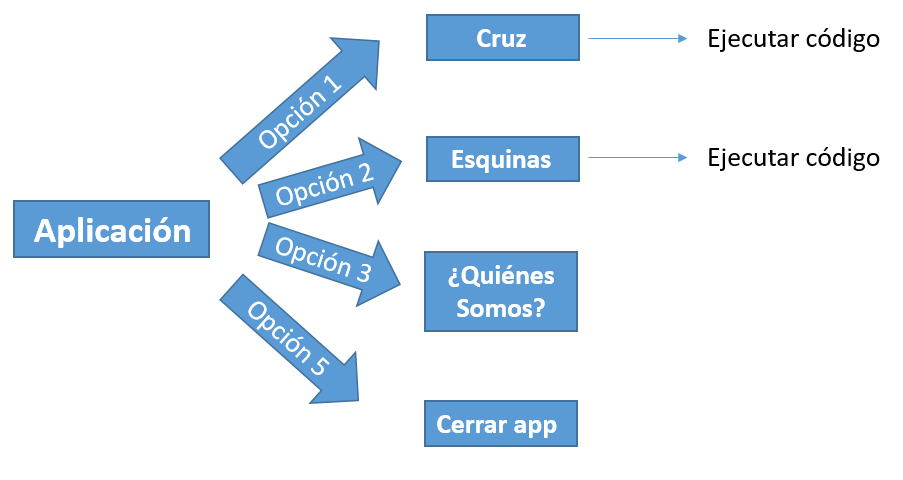
***Lista de patrones***

**5.3 Diseño de la interfaz Usuario**

|  |  |
| --- | --- |
| **Panel** | **Función** |
| Panel Principal | Menú principal, se encontrarán las opciones de patrones y salir del programa. |
| Patrones | Donde se ubicarán los algoritmos, y se le dará la opción al usuario de una vez realizado el patrón, se pueda devolver a su estado original(resuelto). |
| ¿Quiénes somos? | Se encontrará toda la información del proyecto, y del equipo de trabajo. |

**6. Implementación**

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

**6.2 Diagrama de interacción entre programas**

**7. Resultados**

**7.1 Estado actual del proyecto**

A la fecha el proyecto cuanta con:

-El robot Ev3 en su estructura definitiva, sin ningún problema al realizar movimientos.

-El código fuente, donde hay una cantidad de 2 patrones ya programados, con un faltante de 4 más los cuales ya están realizados con la restricción de los 3 movimientos del robot.

-La aplicación realizada con todas sus opciones. Nos referimos, a que cuenta con un menú que dispone de patrones y el usuario escoge el que desea ejecutar.

**7.2 Problemas encontrados y soluciones propuestas.**

|  |  |
| --- | --- |
| **PROBLEMAS ENCONTRADOS** | **SOLUCIONES PROPUESTAS** |
| Base del Robot: Al momento de posicionar el cubo Rubik en la base, esté quedaba muy endeble. Ocasionando un mal movimiento de giro. | Reconstruir la base del Robot con otro tipo de piezas, de tal manera que quedara con dimensiones muy parecidas al del cubo Rubik. |
| Conexión remota con el App Inventor, ocasionando el problema de cómo establecer la conexión. | Desarrollar la aplicación del cubo Rubik, con otro método en el que ya se pueda realizar la conexión remota con el Robot. |
| Brazos del Robot: Una vez realizando los movimientos del robot, no proporcionaba un agarre adecuado sobre el cubo Rubik. | Reforzar los laterales de los brazos, agregándole piezas entrecruzadas. |

**7.3 Conclusiones**

Con lo visto en el tiempo que ha transcurrido durante el proyecto, se han logrado todos los objetivos que teníamos propuestos, tales como entablar conexión remota con el brick del robot vía SSH, dejándonos así un control de los motores que nos permite poder mover el brazo y la base para llegar a los patrones deseados.

A la vez de haber creado un menú provisorio que permite escoger de entre la lista de algoritmos disponibles a la fecha, los cuales fueron programados en Python.

**8. Referencias (estándar IEEE)**

[1]M. Martínez. (2018, Jun 08). Te enseñamos a hacer el cubo de Rubik [Online]. Available: <https://www.nobbot.com/off-topic/como-hacer-cubo-de-rubik/>

[2]C. Angosto. (2015, May 05). El cubo Rubik de la A a la Z [Online]. Available: <http://www.rubikaz.com/resoluciones.php>

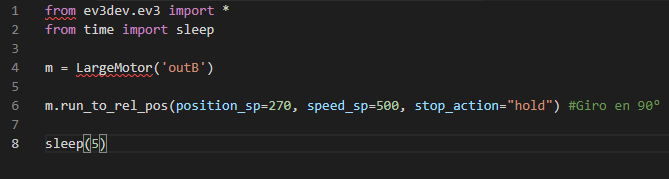
[3]D. Gilday. (2013). MindCuber for lego Mindstorms nxt [Online]. Available: <http://mindcuber.com>

[4]A. Hinojosa. *“Introducción al lenguaje de programación Python”*. 3ª Edición. Oficina de Software Libre de la Universidad de Granada.

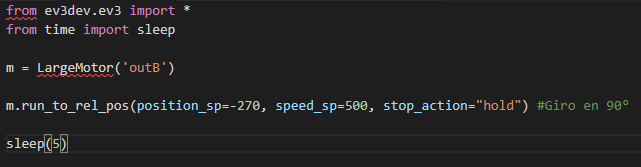
**Anexos**

Anexo A: Código de los programas implementados.

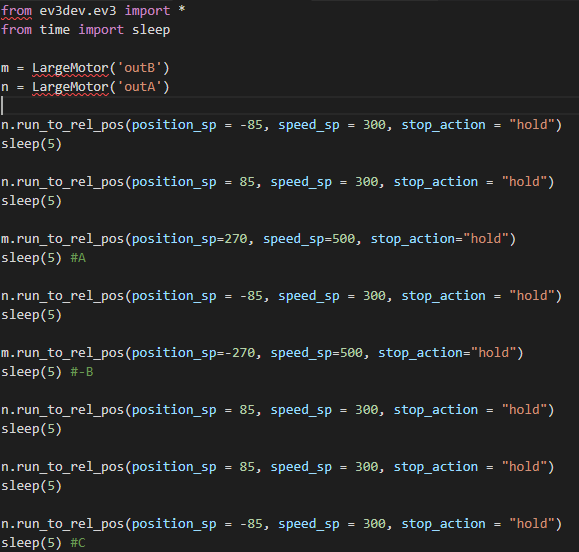
Girar a la izquierda.

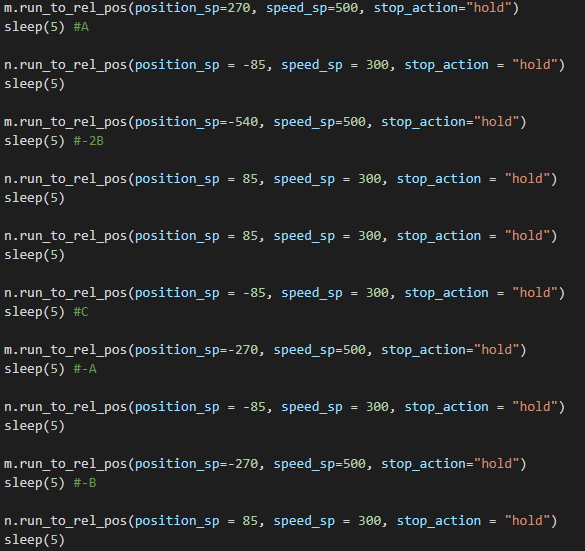


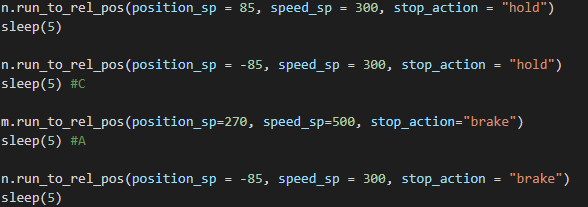
Girar a la derecha.



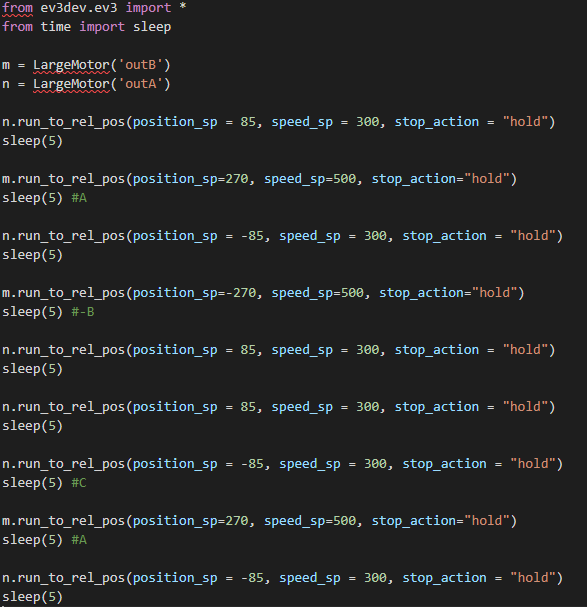
Cruz Blanca.

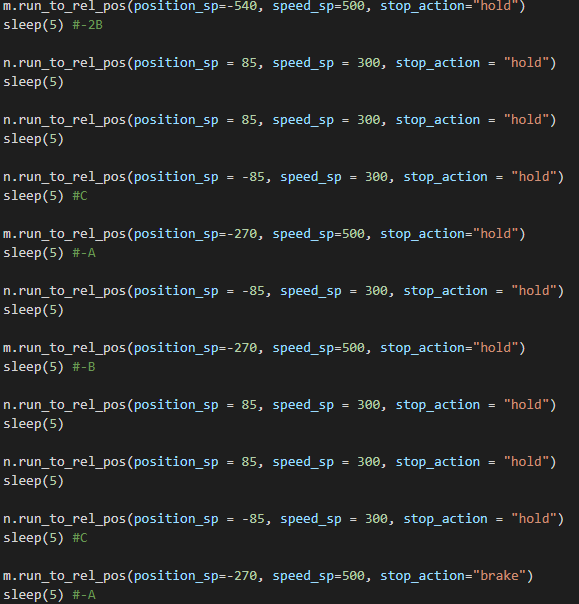






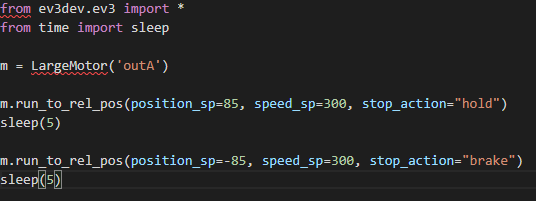
Completar esquinas.





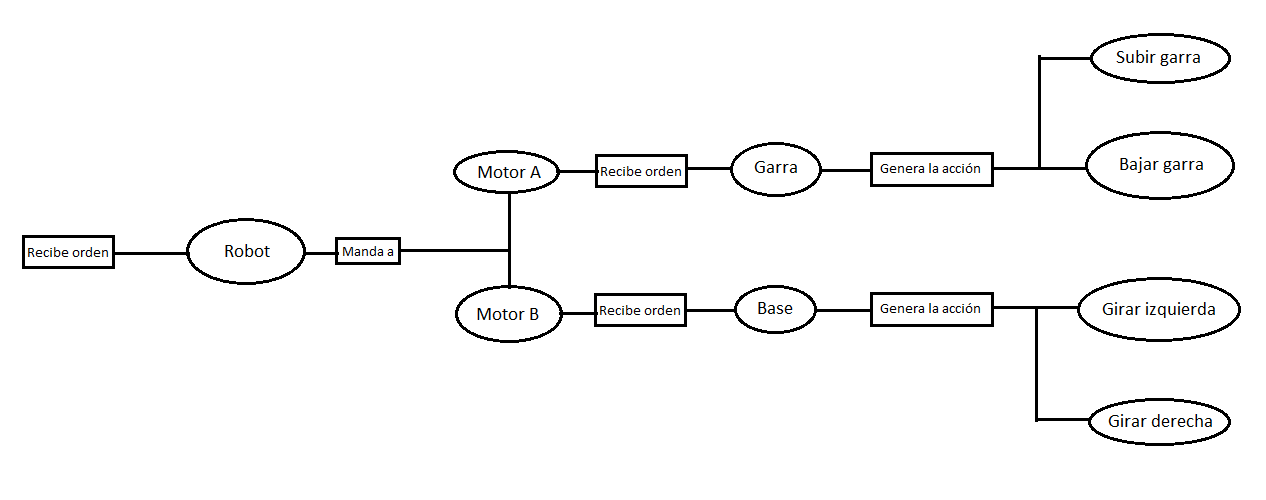


Acción de la garra.



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

Diagrama construcción



**Componentes Principales:**

* MindCube.



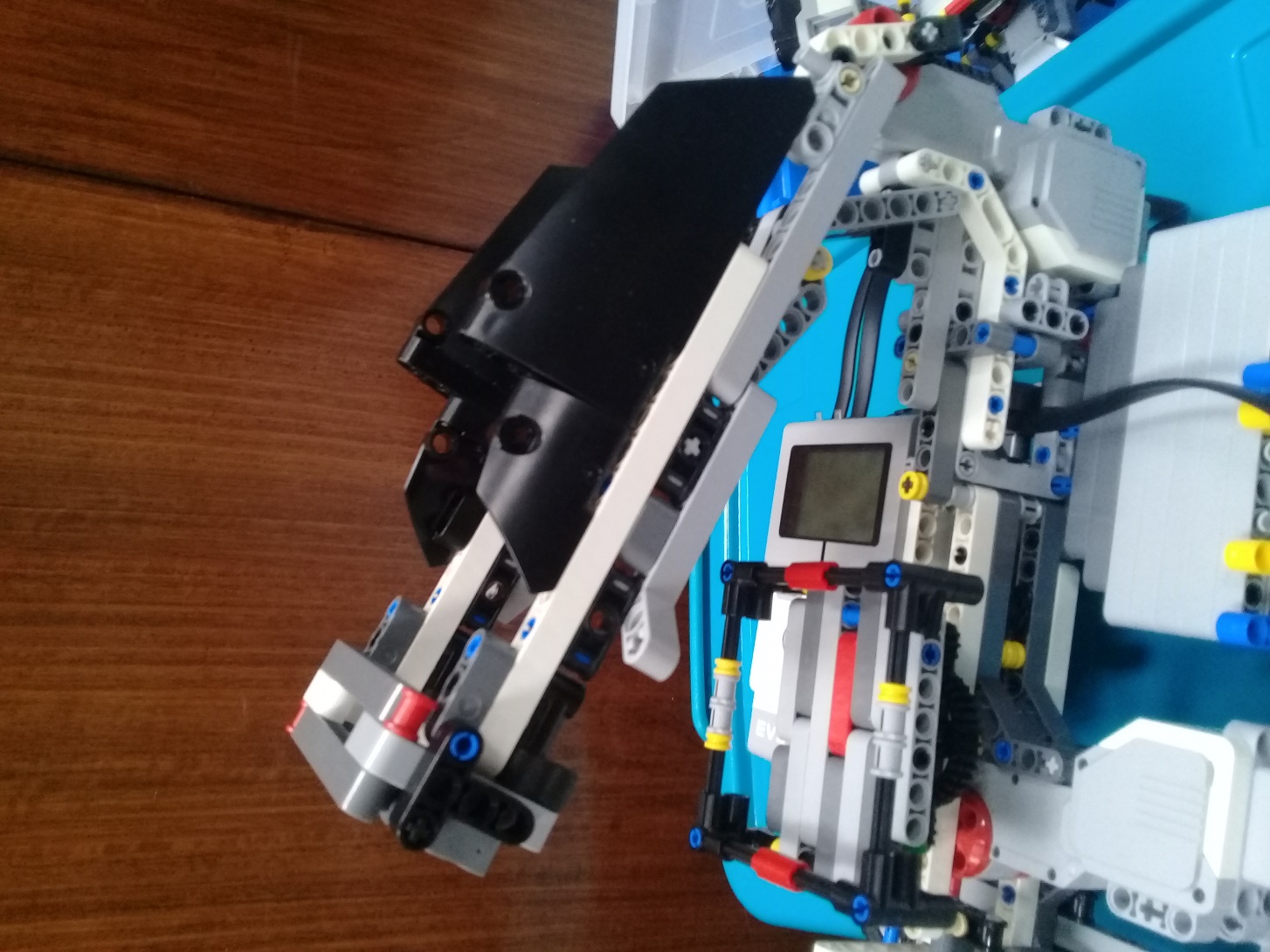
* Motor (A, B).



* Base (contiene cubo Rubik).



* Garra.



* Cables de conexión (de motor a MindCube).

