

**UNIVERSIDAD DE TARAPACÁ**



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

**EUIIIS**

Área de Ingeniería en Computación e Informática



# **Informe Final**

**Autor(es): Pedro Araya**

**Nicolás Colque**

**Gabriel Echeverría**

**Esteban Ovando**

**Asignatura: Proyecto I**

**Profesor(es): Ricardo Valdivia**

**Diego Aracena**

ARICA, 21 de Agosto del 2018

## Historial de Cambios

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
21/07/2018	1.0	Versión inicial del formato	Gabriel Echeverría Nicolas Colque
16/08/2018	1.1	Avance del Plan de proyecto.	Gabriel Echeverría
23/08/2018	1.2	Avance del Plan de proyecto.	Gabriel Echeverría
30/08/2018	1.3	Avance del Plan de proyecto.	Gabriel Echeverría Estaban Ovando
4/08/2018	1.4	Término del Plan de proyecto.	Gabriel Echeverría Estaban Ovando Pedro Araya
25-09-2018	1.8	Corrección de errores del informe	Pedro Araya
06-12-2018	2.0	Informe Final	Gabriel Echeverria Pedro Araya Nicolas Colque Esteban Ovando

# Índice

<b>Historial de Cambios</b>	<b>1</b>
<b>Índice</b>	<b>2</b>
<b>Introducción</b>	<b>3</b>
<b>Panorama General</b>	<b>4</b>
Objetivo General	4
Objetivos Específicos	4
Restricciones	4
Entregables	5
<b>Organización del Personal</b>	<b>6</b>
Descripción y asignación de roles:	6
Mecanismo de Comunicación	7
<b>Planificación del Proyecto</b>	<b>8</b>
Actividades	8
Carta Gantt	9
Gestión de Riesgo	10
<b>Planificación de los Recursos</b>	<b>11</b>
Recursos de Hardware-Software requeridos	11
Estimación de Costos	12
<b>Análisis y Diseño</b>	<b>13</b>
Especificación de requerimientos	13
Arquitectura Propuesta	14
Diseño de Interfaz Usuario	15
<b>Implementación</b>	<b>17</b>
Descripción de los programas implementados	17
Diagrama de interacción entre programas	21
<b>Pruebas</b>	<b>23</b>
<b>Resultados</b>	<b>24</b>
<b>Estado final del Proyecto</b>	<b>24</b>
<b>Problemas encontrados y soluciones propuestas</b>	<b>24</b>
<b>Conclusión</b>	<b>25</b>
<b>Trabajo a Futuro</b>	<b>26</b>

<b>Anexos</b>	<b>27</b>
Anexo A	27
Anexo B	29
<b>Bibliografía y Referencias</b>	<b>31</b>

## Introducción

A mediados de la década de 1970, el escultor y profesor Erno Rubik inventó el rompecabezas mecánico llamado “Cubo Mágico” o también conocido como “Cubo Rubik”.

Gracias a este invento, nació el speedcubing. El Speedcubing es la práctica que intenta resolver un cubo de Rubik en el menor tiempo posible. Existen una serie de competiciones de speedcubing a lo largo del mundo.

Actualmente, el mejor registro humano lo tiene Feliks Zemdegs quien estableció el récord mundial en resolver un cubo de Rubik en 2018 en 4.221 segundos.

Sin embargo, en los robots la cosa es muy distinta, ya que ese registro es demasiado lento comparado al del robot “Sub1 Reloaded”, quien lo resolvió el cubo en 0.637 segundos.

Teniendo estos dos hechos presente, se pensó en hacer un robot que ejecute algoritmos básicos en el cubo rubik, con el fin de que más personas aprendan estos algoritmos y se incluyan a esta actividad.

Si por algo se caracteriza LEGO Mindstorm EV3, es por el infinito números de proyectos que se puede llevar a cabo, gracias a los distintos motores, sensores piezas LEGO y otros elementos incluidos en la caja, por otro lado tenemos a su software, el cual contiene una gran riqueza en programación, yendo desde lo más básico hasta el arrastre de objetos. Una idea de David Gilday

## Panorama General

### Objetivo General

- Implementar un robot que mediante una aplicación permite armar el cubo rubik

### Objetivos Específicos

- Armar un Robot Lego de acuerdo al diseño que se escogió y comprobar si los movimientos que debe realizar estén funcionando correctamente.
- Desarrollar un pseudocódigo con los movimientos básicos del Cubo rubik y posteriormente codificar el pseudocódigo a Python.
- Diseñar un bosquejo de la Interfaz remota.
- Estudiar las plataformas de desarrollo de una interfaz de control remota.
- Desarrollar la Interfaz de control remota que permita al robot moverse mediante control remoto.
- Prueba y Análisis de resultados.

### Restricciones

Se debe respetar un conjunto de limitaciones para el desarrollo del proyecto. A continuación se muestra en la siguiente tabla:

<b>Restricción</b>	<b>Descripción</b>
Tiempo	Fecha límite para entregar los documentos del proyecto.
Partes del Robot (EV3)	El uso completo de las piezas Lego de acuerdo al diseño estudiado.
Lenguaje de Programación (Python)	El único Lenguaje de Programación con el cual se nos permite programar.

## Entregables

El proyecto consta de diferentes resultados en su totalidad, a continuación se muestra una tabla de los entregables:

<b>Entregables</b>	<b>Descripción</b>
Informes de avance y respectivas presentaciones	Esto consiste en: -Formulación de Proyecto. -Informe de Avance I. -Informe Final.
Bitácora semanal	Información sobre el desarrollo de cada semana de trabajo y de lo que se trabajará en su semana siguiente.
Wiki	Consiste en una plataforma para dar a conocer en detalle el proyecto Rubik Peng.
Manual de Usuario	Manual de cómo se opera la aplicación para controlar el robot.
Algoritmos Robot	Código en el cual están las funciones con las que opera el robot para el movimiento del cubo rubik.

## Organización del Personal

### Descripción y asignación de roles:

Para un desarrollo eficiente del proyecto se definen distintos roles a los integrantes del equipo, con el fin de tener un responsable en cada área del proyecto. A continuación se muestra una tabla con la información detallada.

<b>Rol</b>	<b>Descripción</b>	<b>Responsable</b>
Programador	Diseñar los algoritmos del robot y la interfaz para la comunicación remota con el robot.	Nicolás Colque
Secretario	Encargado de realizar los informes de avances, bitácoras y publicación de los avances.	Gabriel Echeverría
Ensamblador	Encargado de ensamblar el robot.	Esteban Ovando
Líder	Organiza el equipo e inspecciona el avance del proyecto.	Pedro Araya

## Mecanismo de Comunicación

Para poder tener una buena comunicación durante las horas de trabajo en casa o en caso de que un integrante del proyecto falte, se utilizan diferentes redes sociales y herramientas de trabajo:

**Discord:** Software gratuito de comunicación online, este cuenta con un sistema de salas, el proyecto cuenta con una sala de trabajo con un chat y llamada de voz grupal. Este funciona tanto en Windows como en Android.

**Facebook Messenger:** En caso de tener de una emergencia y/o un inconveniente se creó un grupo en “Facebook Messenger”. Es una aplicación de celular que permite una comunicación a través de mensajes en cualquier momento.

**Google Drive:** Es una Plataforma, donde se almacenarán los informes, bitácoras, tratamientos de riesgos, etc. También se usa como herramienta para el desarrollo de informes y presentaciones, ya que, cuenta con un sistema de trabajo online en el cual se puede avanzar paralelamente en ello.

Por último, en horas libres se realizarán juntas de trabajo para avanzar o perfeccionar el proyecto.



## Planificación del Proyecto

### Actividades

Con el fin de tener una mayor organización en el proyecto se definieron diversas actividades a realizar en el proyecto junto con su responsable en cada área.

Actividad	Descripción	Responsable	Producto
Administración del Sitio de RedMine (ICCI)	-Desarrollo de la Wiki. -Bitácora (semanal). -Diseño de la carta Gantt. -Administrar documentos (entregables).	-Gabriel Echeverria	-Dar a conocer el avance del proyecto.
Armado del Robot EV3	-Designar el modelo del robot. -Verificación y búsqueda de piezas faltantes. -Ensamblaje del robot.	-Esteban Ovando	-Hacer un Robot que cumpla con las expectativas del equipo.
Programación del brick Ev3	-Instalación del sistema operativo. -Investigación de Python y librería ev3dev-lang-python. -Conexión remota con cliente SSH. -Implementar y calibrar los movimientos básicos del robot. -Estudiar los patrones de movimientos de Cubo Rubik. -Desarrollar un pseudocódigo con los movimientos básicos del Cubo rubik. -Programar el pseudocódigo en Python. -Investigar e implementar métodos de armado para el cubo de Rubik.	-Nicolás Colque	-Programar robot para que luego ejecute una línea de comandos.
Programación de la interfaz de usuario.	-Realizar un boceto de la interfaz de usuario. -Investigar comunicación remota con el brick EV3. -Investigar plataforma para el	-Pedro Araya	-Hacer una aplicación para que remotamente el robot ejecute métodos del armado del cubo de Rubik.

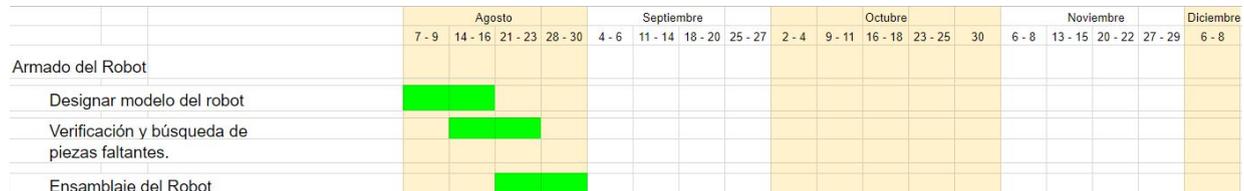
	envió de comandos al brick EV3. -Programación de la interfaz de usuario.		
Documentación del Proyecto	-Realización del manual de usuario.	-Pedro Araya	-Dar conocimiento básicos al usuario acerca del manejo del robot.

## Carta Gantt

Para tener una estimación del tiempo que se utilizará en cada actividad del proyecto se ha confeccionado una Carta Gantt, con el propósito de organizar el tiempo entre actividades para una mayor eficiencia.

### Armado del Robot:

Responsable: Esteban Ovando



### Administración del sitio Redmine:

Responsable: Gabriel echeverria



### Programación del Brick:

Responsable Nicolas Colque



Programación de la interfaz de usuario:      Responsable: Pedro Araya

	Agosto				Septiembre				Octubre				Noviembre				December	
	7 - 9	14 - 16	21 - 23	28 - 30	4 - 6	11 - 14	18 - 20	25 - 27	2 - 4	9 - 11	16 - 18	23 - 25	30	6 - 8	13 - 15	20 - 22	27 - 29	6 - 8
Programación de la interfaz de usuario.																		
Realizar un boceto de la interfaz de usuario.																		
Investigar comunicación remota con el brick EV3.																		
Investigar plataforma para el envío de comandos al brick EV3.																		
Programación de la interfaz de usuario.																		

## Gestión de Riesgo

El nivel de impacto de los riesgos pronosticados se clasificaron desde el nivel Catastrófico(1) - Crítico(2) - Marginal(3) - Despreciable(4), tomando el nivel Catastrófico como el más grave y el Despreciable el más leve.

Riesgo	Probabilidad de Ocurrencia	Nivel de Impacto	Acciones Remediales
Ausencia de un(os) integrante a las juntas de trabajo.	10%	4	Distribuir las tareas equitativamente del miembro faltante dependiendo de cada una de sus capacidades.
Falta de piezas del robot	25%	3	Buscar un reemplazo para las piezas improvisando con otras piezas.
Atraso de actividad(es)	20%	3	Enfocarse en la actividad(es) atrasada en horarios extras.
Pérdida de documento(s)	5%	3	Tener un respaldo de los documentos y/o diseños del proyecto.
Retirada de miembro(s) del proyecto.	5%	3	Reasignar las actividades del proyecto con los miembros restantes.
Cambios de actividades	5%	3	Planificar un horario para la reunión que no se pudo congregar de trabajo

## Planificación de los Recursos

### Recursos de Hardware-Software requeridos

Los recursos que se utilizaran durante el periodo del proyecto cumplen un rol importante en este, por lo que daremos a conocer todos los recursos que necesitaremos junto con su finalidad.

<b>Hardware</b>	<b>Descripción de uso</b>
Computador	Para la programación del robot y la realización de los entregables, también para la operación remota del robot.
Brick (EV3)	Parte del Kit de Lego Mindstorms(EV3), este realiza los algoritmos para el cubo mediante una conexión remota.

<b>Software</b>	<b>Descripción de uso</b>
Microsoft Visual Studio Code	IDE para el desarrollo del programa para el robot.
Redmine	Publicación de avance del proyecto y respectivas informaciones.
Lenguaje de Programación (Python)	Lenguaje utilizado para la programación del robot(EV3).
Putty	Ciente de SSH para la conexión remota con el robot.
Glade	Herramienta para el desarrollo de la interfaz del cliente.

## Estimación de Costos

Los productos son estimados mediante un precio promedio, es decir que se toman varias referencias del producto y se calculó su promedio.

<b>Productos y Horas</b>	<b>Cantidad</b>	<b>Costo</b>
Cubo Rubik	1 unidad	\$10.000
Red Wireless Usb TP-Link	1 unidad	\$9.000
Kit de Lego Mindstorms(EV3)	1 unidad	\$500.000
Micro SD 8GB (kingston)	1 unidad	\$7.000
Hora trabajo	1 hora	\$9.000
Hora trabajo x persona	9.000 * 4 personas	\$36.000
Horas Totales	72 hrs * 36.000	\$2.592.000
Hora Extra	1 hora	\$13.000
Hora Extra x persona	13.000 * 4 personas	\$52.000
Horas Extras	32 hrs * 52.000	\$3.744.000
		Total: \$6.972.000

# Análisis y Diseño

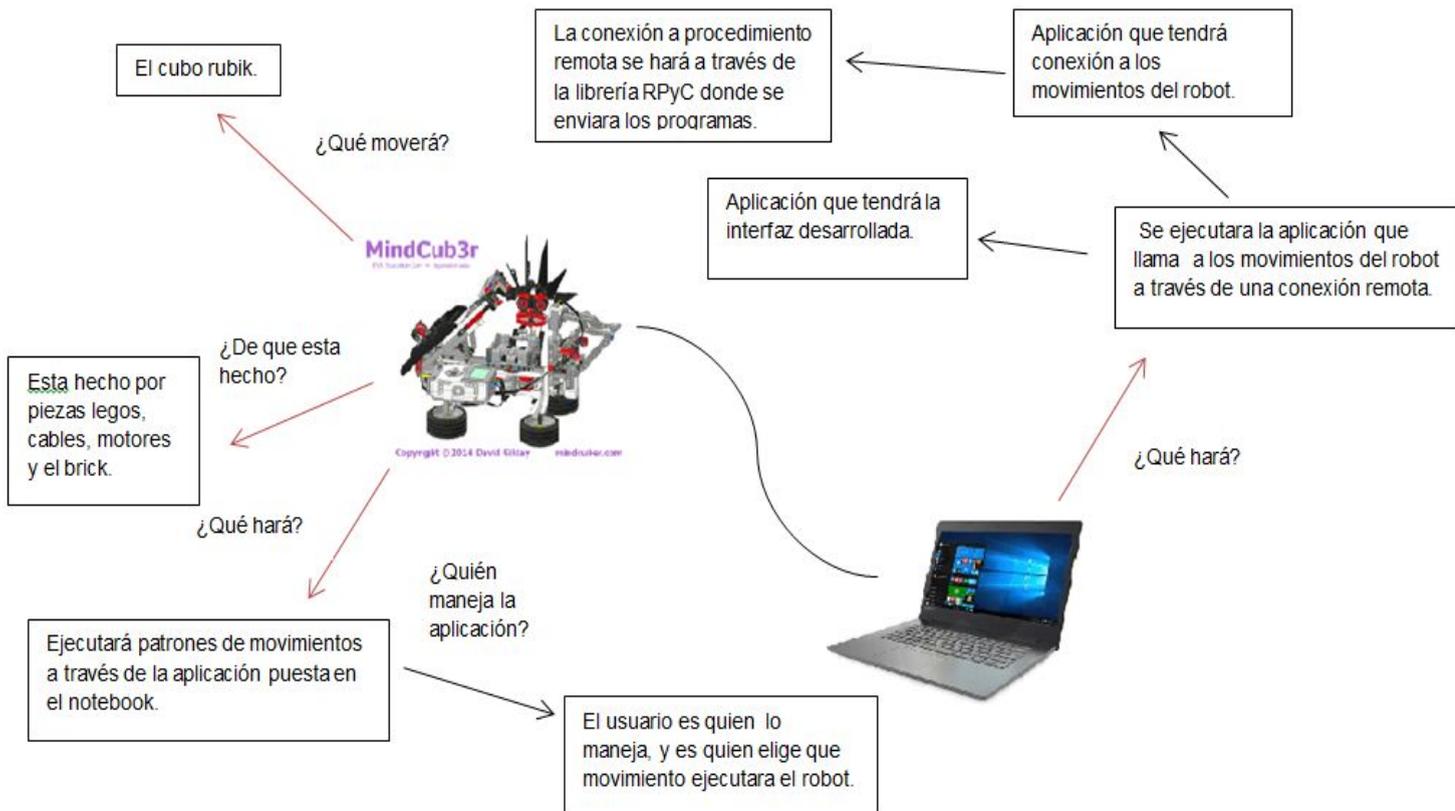
## Especificación de requerimientos

Para este proyecto se requerirá aplicar todos conocimiento obtenidos hasta ahora, tiempo de dedicación y responsabilidad. Los requerimientos necesarios para el desarrollo de este son los siguientes:

Funcional	No funcional
<ul style="list-style-type: none"> <li>- Un robot lego (MindStorms EV3) que armará mediante comandos un cubo rubik con movimientos predeterminados.</li> <li>- Los movimientos costarán desde movimientos básicos a diferentes métodos de resolución para el cubo de rubik.</li> <li>- Una aplicación de computador para la programación que se le hará al robot lego(EV3), está realizará una conexión remota para la ejecución de los comandos hacia el robot.</li> </ul>	<ul style="list-style-type: none"> <li>- Desarrollar el proyecto del robot EV3 con un plazo determinado 4 meses.</li> <li>- Programar en el lenguaje de programación Python los movimientos que se le dará al robot.</li> <li>- Se ocupará la plataforma de redmine para el seguimiento del proyecto.</li> </ul>

## Arquitectura Propuesta

En el siguiente diagrama se puede observar lo planteado en el proyecto.



## Diseño de Interfaz Usuario

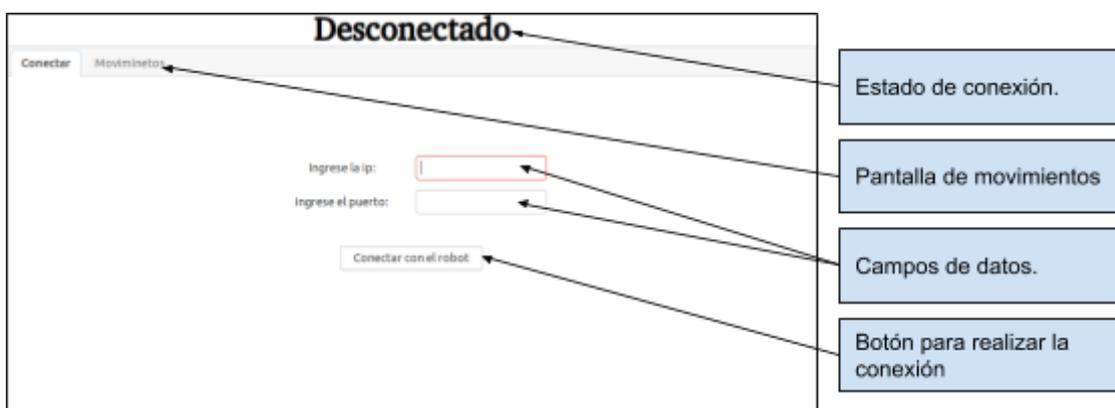
El diseño de la interfaz fue desarrollado en Glade que es un RAD(Rapid Application Development) y programada en python 3. Esta contiene el cliente rpyc el cual envía comandos al brick EV3 para realizar los movimientos básicos del robot.

Análisis de especificación de la interfaz:



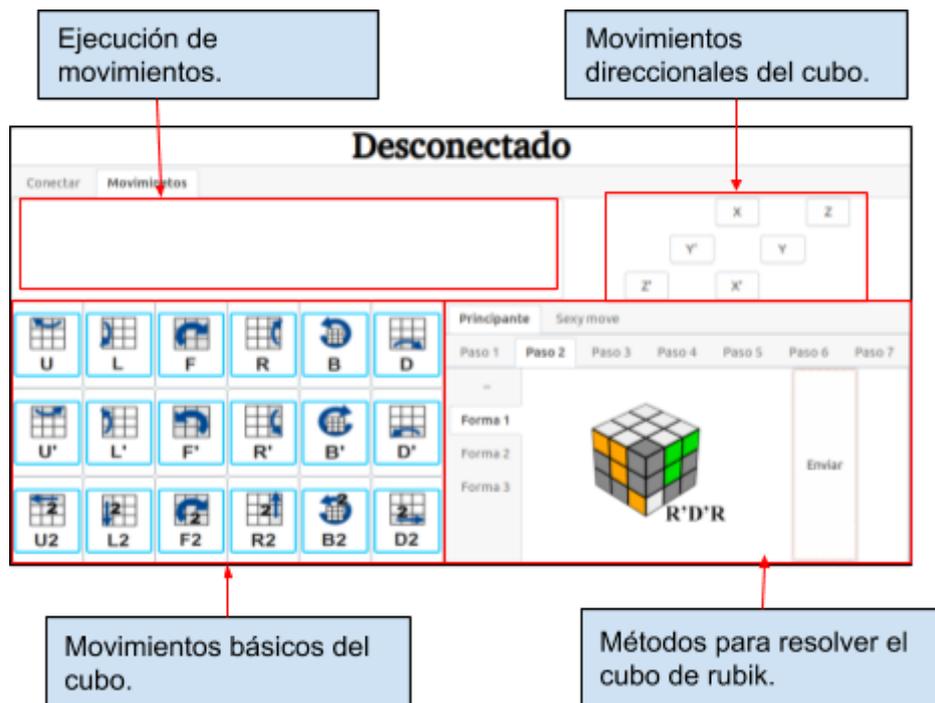
Descripción de las pantallas:

Pantalla de conexión: Es la cual se establecerá conexión con el brick EV3 entregando la ip y el puerto de red.



- Estado de conexión: Esta etiqueta mostrará el estado actual de conexión con el brick.
- Pantalla de movimientos: Acceso hacia la pantalla respectiva.
- Campos de datos: En estos campos se ingresa la ip y puerto de red del brick EV3.

Pantalla de comandos: Esta pantalla es el mecanismo principal de la comunicación con el brick EV3.



- Ejecución de movimientos: Este es un panel donde se mostrará en cola de movimientos que se van ingresando y ejecutando en el robot.
- Métodos para resolver el cubo de rubik: Selector de tipos de métodos y diferentes pasos a de resolución para el cubo de Rubik con sus respectivas combinaciones de movimientos.
- Movimientos direccionales del cubo: estos movimientos sirven para rotar el cubo sin alterar su estado.
- Movimientos básicos del cubo: Estos movimientos hacen cambiar de estado el cubo. Estos movimientos están basados en las notaciones del cubo de rubik.



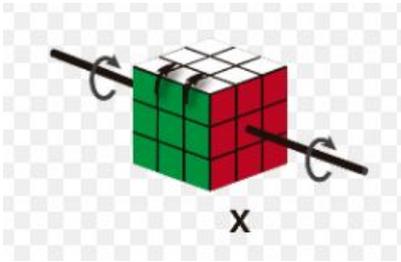
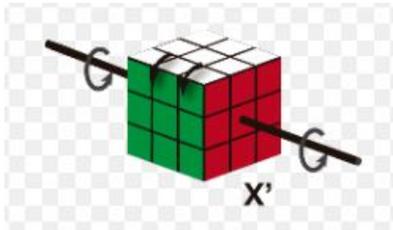
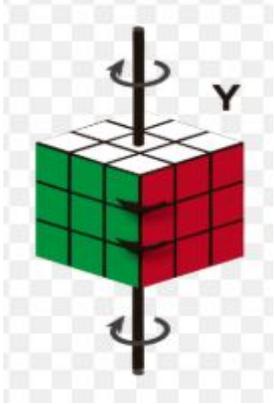
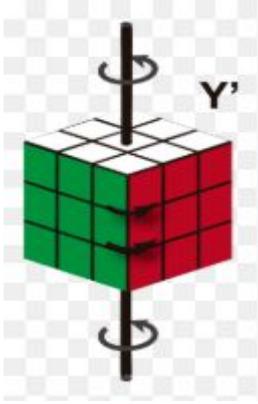
cubo rote a la derecha.

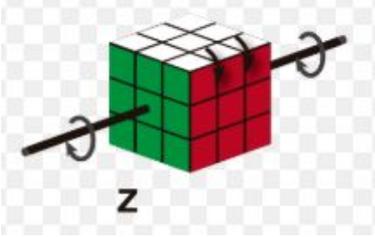
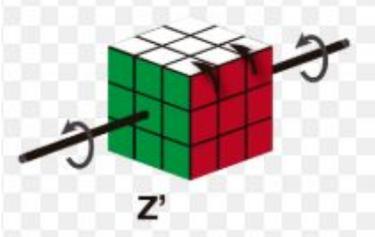
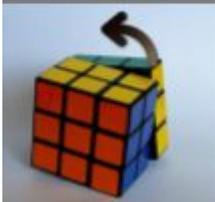
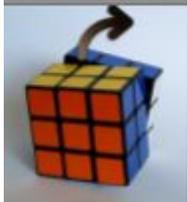
Entrada: Motor del brazo, Motor de la base, n° de vueltas.

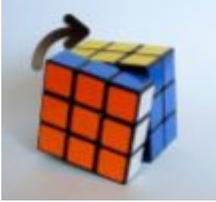
**def rotate\_left(brazo, base, vueltas):** Girar la base en 90° a la izquierda y a su vez el brazo del robot sujeta el cubo que estará en la base para que la parte inferior del cubo rote a la izquierda.

Entrada: Motor del brazo, Motor de la base, n° de vueltas.

En la siguiente tabla se mostrarán los programas desarrollados actualmente con los cinco movimientos bases del robot a los que denominamos “Movimiento Básicos” .

<b>Movimientos Básicos</b>	
<p>Def X(brazo, vueltas)</p> 	<p>Def Xprima(brazo)</p> 
<p>Def Y(base, vueltas)</p> 	<p>Def Yprima(base, vueltas)</p> 

<p>Def Z(brazo,base, vueltas)</p> 	<p>Def Zprima(brazo, base, vueltas)</p> 
<p>def U(brazo, base, vueltas):</p> 	<p>def Uprima(brazo, base, vueltas):</p> 
<p>def D(brazo, base, vueltas):</p> 	<p>def Dprima(brazo, base, vueltas):</p> 
<p>def M(brazo, base, vueltas):</p> 	<p>def Mprima(brazo, base, vueltas):</p> 
<p>def B(brazo, base, vueltas):</p> 	<p>def Bprima(brazo, base, vueltas):</p> 
<p>def R(brazo, base, vueltas):</p>	<p>def Rprima(brazo, base, vueltas):</p>

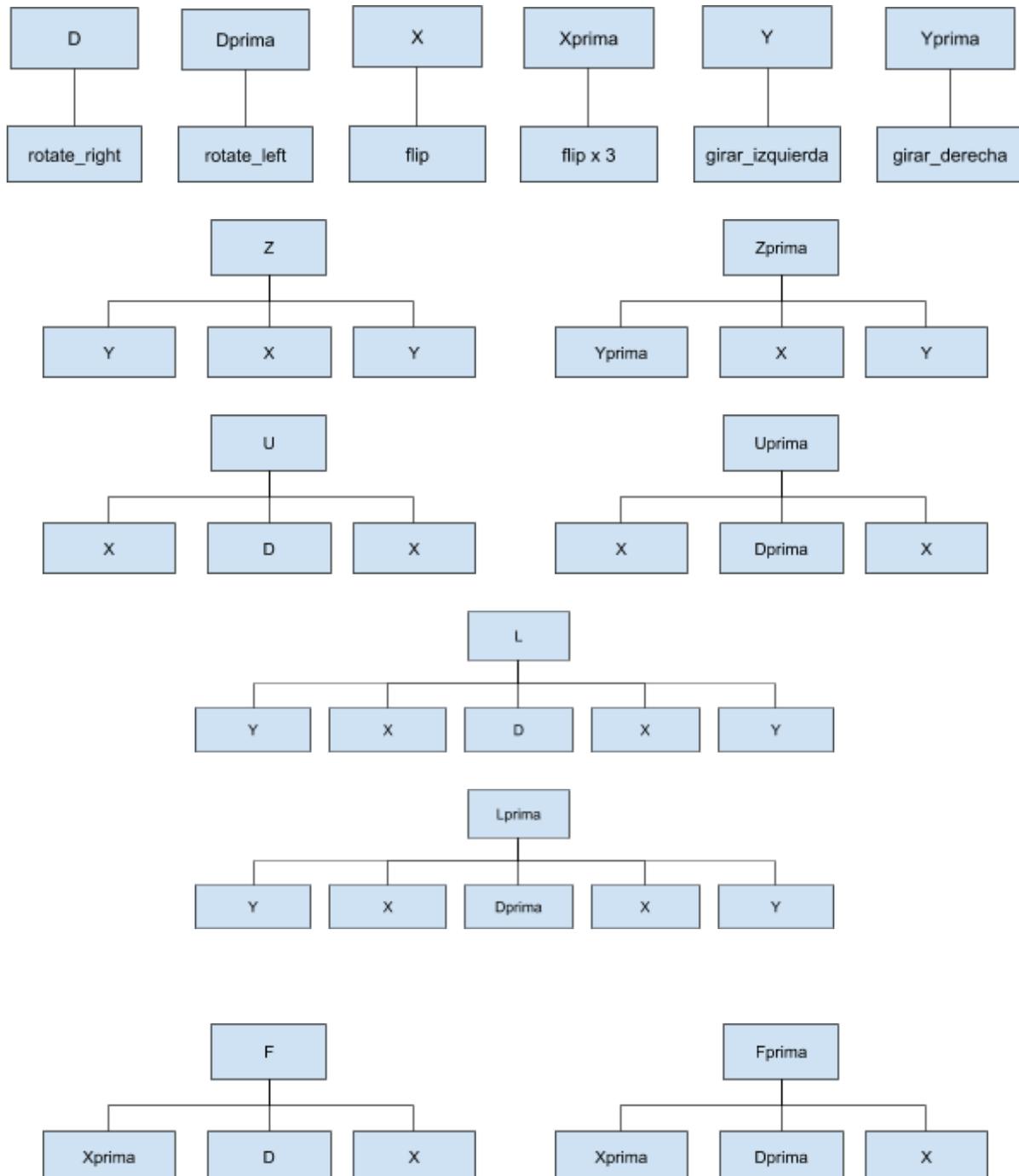
	
<p>def F(brazo, base, vueltas):</p> 	<p>def Fprima(brazo, base, vueltas):</p> 
<p>def L(brazo, base, vueltas):</p> 	<p>def Lprima(brazo, base, vueltas):</p> 

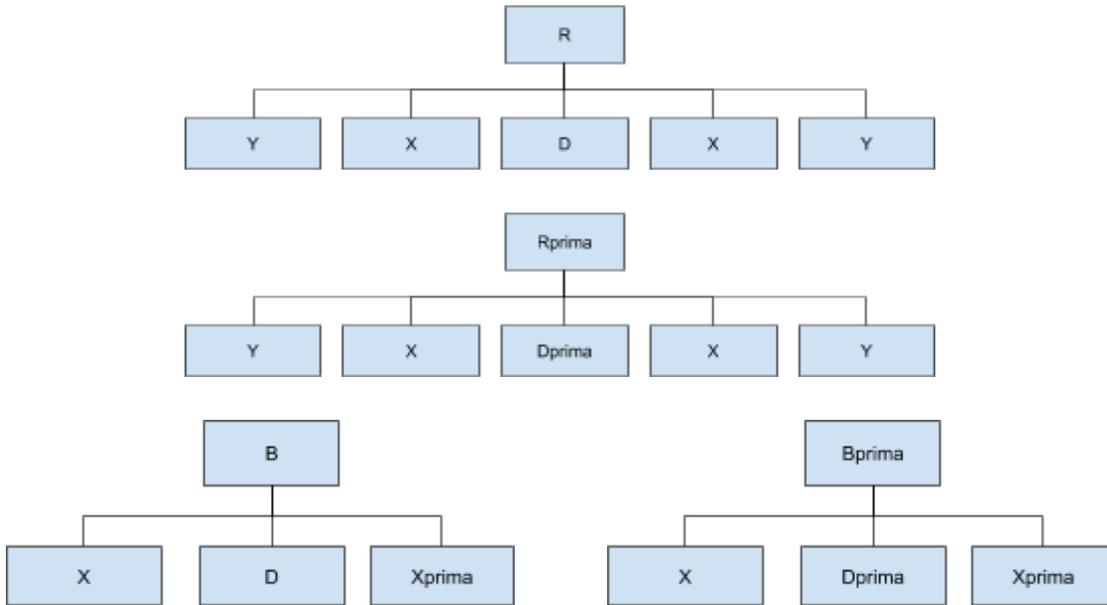
En el próximo informe se tendrán los programas que se formarán a partir de los "Movimientos Básicos", como por ejemplo para realizar el Algoritmo "Sexy Move" se necesitan los Movimientos Básicos  $R - U - R' - U'$ .

Para más información revisar la sección de anexos donde se podrá ver de forma más detallada lo que hace cada programa desarrollado.

## Diagrama de interacción entre programas

Los Movimientos Básicos del robot se forman a partir de una combinación entre los Movimientos Bases. A continuación se muestra el diagrama de cada uno de los movimientos implementados en el brick:





Cabe destacar que los métodos están implementados en la interfaz de usuario a partir de los movimientos ya descritos y no se implementaron nuevas funciones.

## Pruebas

En esta tabla se describirán las pruebas a las que se sometió el robot junto con su resultado.

Pruebas	Descripción	Resultado
Prueba I	Como primera prueba, al robot se le ordenó realizar los movimientos básicos implementados, sin utilizar la interfaz remota. Para comprobar si cada patrón de movimiento el robot lo realiza adecuadamente.	El robot realizó cada movimiento correctamente, sin embargo, después de realizar una gran cantidad de patrones este se descalibra e interfiere con los siguientes patrones.
Prueba II	Como segunda prueba, el robot se le ordenó nuevamente realizar cada movimiento básico pero esta vez utilizando la interfaz remota. Para ver si los botones del robot funcionan correctamente.	La prueba se realizó exitosamente, ya que cada botón ejecuta el programa que debía realizar.

# Resultados

## Estado final del Proyecto

Actualmente el proyecto ya se termino la programación de los movimientos del cubo Rubik, con respecto a la operación remota con el robot se usó RPyC el cual nos permitirá hacer una conexión remota vía wifi entre el servidor (Robot) y el cliente (notebook), con lo relacionado a la interfaz ya se termino el diseño como la programación de la interfaz, para realizarlo se uso la libreria de python llamada Glade la cual nos facilitó mucho el diseño de la interfaz así como su programación.

## Problemas encontrados y soluciones propuestas

En el desarrollo del proyecto nos encontramos con diferentes problemas los cuales nos provocaron retrasos en el desarrollo de algunas actividades.

Un problema que se encontró fue que el visual studio code no ejecutaba el código en el que estaban las funciones del robot. Para resolverlo solo se envió el código al robot sin necesidad de que lo revise el visual studio, debido a que el que tiene que ejecutar esas funciones son el robot y no el computador.

El diseño que se eligió de la base no estaba centrado con el motor lo cual provocaba que en algunas partes de la base el cubo no hiciera sus funciones de forma correcta y en otras sí. Para resolver este problema se hizo un nuevo diseño para la base la cual fue creación original del equipo la que actualmente no provoca problemas como el diseño anterior.

Con lo que respecta al diseño y la interfaz se encontró problemas con que libreria usar para el diseño y programación de la interfaz, inicialmente se usó tkinter el cual no cumplía con nuestras necesidades al diseño de la interfaz, también se vio la librería kivy la cual nos complicó mucho en el sentido de programación de la interfaz, para solucionarlo se investigó sobre librerías en las cuales se encontró Glade el cual nos facilitó mucho el diseño de la interfaz como su programación además de cumplir con todas nuestras necesidades.

## Conclusión

Al terminar el Proyecto, se pudo entender que cada punto trabajado es esencial para el desarrollo del proyecto y que éste va a ser mucho más complejo de lo que se había pensado en un principio.

Durante el desarrollo del Informe final se pueden destacar varios puntos del proyecto, uno de ellos es la seria corrección de los objetivos, ya que no estaban bien explicados y la faltas de otras. Una observación destacada es la improvisación que tuvo el equipo, en el excelente re armado que se le dio a la base del robot(EV3), ya que fue la causa de los errores que se presentaban en la ejecución de pruebas que se realizaban en este.

Unas de las experiencias compartidas que se tiene con todos los integrantes, es la comprensión con cada compañero, la motivación que se tiene por finalizar con éxito este proyecto y el querer sacar el mejor aprendizaje.

## Trabajo a Futuro

Este proyecto puede continuar si se realizan implementaciones de más secciones como por ejemplo:

- Armado del cubo sólo a partir de los colores entregados.
- Desarmado del cubo en forma aleatoria.
- Un minijuego donde el robot deja en un estado desarmado el cubo y el usuario tendrá que armarlo en la menor cantidad de movimientos posibles.

Otra sugerencia de trabajo futuro es implementar sensores para solucionar el problema de calibración en la base de el robot.

Un estudio en profundidad de python y la librería ev3dev permitira realizar estas implementaciones de manera eficiente sin problemas.

# Anexos

## Anexo A

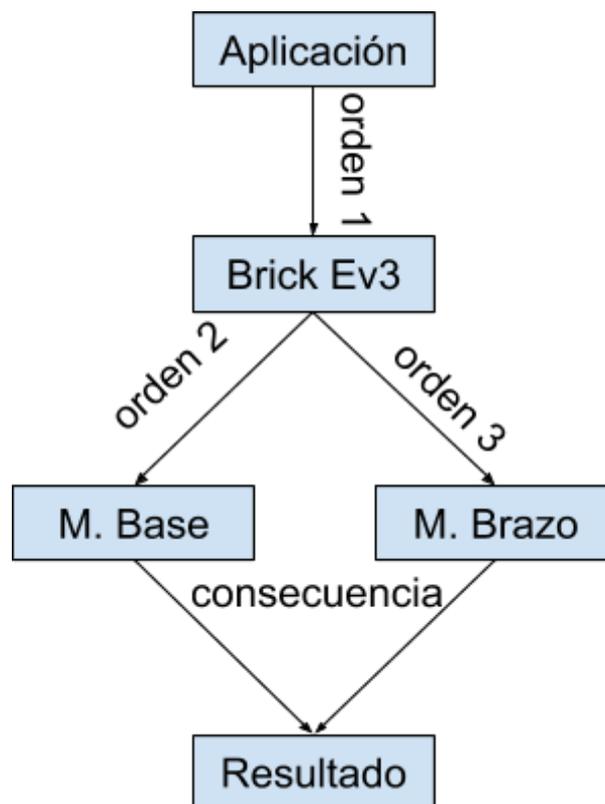
<b>Movimiento Bases</b>
<pre>def girar_derecha(base, vueltas): for x in range(0, vueltas):     base.on_for_degrees(SpeedPercent(-40), 261.5) base.stop()</pre>
<pre>def girar_izquierda(base, vueltas): for x in range(0, vueltas):     base.on_for_degrees(SpeedPercent(40), 261.5) base.stop()</pre>
<pre>def rotate_right(brazo, base, vueltas): brazo.on_for_seconds(35, 0.28) brazo.stop()</pre>
<pre>def rotate_left(brazo, base, vueltas): brazo.on_for_seconds(35, 0.28) brazo.stop()</pre>
<pre>def flip(brazo, vueltas): brazo.on_for_seconds(35, 0.28) brazo.stop() for x in range(0, vueltas):     brazo.on_for_seconds(35, 0.305)     brazo.on_for_seconds(-35, 0.305)     brazo.on_for_seconds(-35, 0.3) brazo.stop()</pre>

<b>Movimientos Básicos</b>	
def X(brazo, n): flip(brazo, n)flip(brazo, 3)	def Xprima(brazo): flip(brazo, n)
def Y(base, n): girar_izquierda(base, n)	def Yprima(base, n): girar_derecha(base, n)
def Z(brazo, base, n): Yprima(base,1) Xprima(brazo, n) Y(base,1)	def Zprima(brazo, base, n): Y(base, 1) Xprima(brazo, n) Yprima(base,1)
def U(brazo, base, vueltas): Xprima(brazo, 2) D(brazo, base, n) Xprima(brazo, 2)	def Uprima(brazo, base, vueltas): Xprima(brazo, 2) Dprima(brazo, base, n) Xprima(brazo, 2)
def L(brazo, base, vueltas): Y(base, 1) Xprima(brazo, 1) D(brazo, base, n) X(brazo) Yprima(brazo, 1)	def Lprima(brazo, base, vueltas): Y(base, 1) Xprima(brazo, 1) Dprima(brazo, base, n) X(brazo) Yprima(brazo, 1)
def F(brazo, base, vueltas): X(brazo) D(brazo, base, n) Xprima(brazo, 1)	def Fprima(brazo, base, vueltas): X(brazo) Dprima(brazo, base, n) Xprima(brazo, 1)
def R(brazo, base, vueltas): Yprima(base, 1) Xprima(brazo, 1) D(brazo, base, n) X(brazo) Y(base, 1)	def Rprima(brazo, base, vueltas): Yprima(base, 1) Xprima(brazo, 1) Dprima(brazo, base, n) X(brazo) Y(base, 1)
def B(brazo, base, vueltas):	def Bprima(brazo, base, vueltas):

Xprima(brazo, 1) D(brazo, base, n) X(brazo)	Xprima(brazo, 1) Dprima(brazo, base, n) X(brazo)
def D(brazo, base, vueltas): rotate_right(brazo, base, n)	def Dprima(brazo, base, vueltas): rotate_left(brazo, base, n)

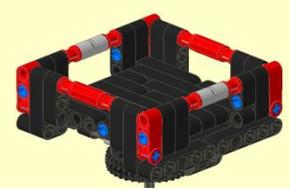
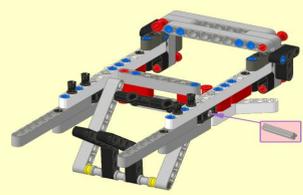
## Anexo B

Este diagrama representa como opera el robot al enviar una orden por la aplicación.



Desde la Aplicación se envía una Orden 1 al Brick Ev3 y dependiendo de la orden que reciba nacerán dos nuevas órdenes “orden 2” y “orden 3” (estas varían según la orden 1 que reciba), estas trabajaran con los Motores de la Base y del Brazo respectivamente, lo que provocará que el robot realice un conjunto de orden a lo que llamaremos resultado.

En esta tabla se podrán los componentes principales del robot:

<p>Cuerpo del Robot</p>		<p>El Cuerpo del Robot es la parte principal del robot ya que es donde está la conexión con los otras piezas.</p>
<p>Base</p>		<p>La Base es para dejar el cubo rubik, gracias a esto y al brazo, se podrá hacer distintos movimientos en el cubo rubik.</p>
<p>Brazo</p>		<p>El Brazo hará "flip" en el cubo rubik, y eso ayudará para poder hacer los movimientos</p>
<p>Brick</p>		<p>El Brick es la pieza más importante ya que esta todas las funciones en código de los movimientos.</p>

<p>Imagen de referencia</p>	
-----------------------------	--

## Bibliografía y Referencias

[1] D. Gilday. (2013-2016). Networks (versión 1.1) [Online]. Available: [mindcuber.com/mindcub3r/mindcub3r.html](http://mindcuber.com/mindcub3r/mindcub3r.html)

[2] R. Hempel (2015). Pure python bindings for ev3dev [Online]. Available: <https://github.com/ev3dev/ev3dev-lang-python>

[3] Ruwix Community (2018) Rubik's Cube Wiki[Online]. Available: <https://ruwix.com/>

[4] Ev3Dev community (2018). Ev3Dev is you EV3 re-Imagined[Online]. Available: <https://www.ev3dev.org/>