**UNIVERSIDAD DE TARAPACÁ**



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



Área de Ingeniería en Computación e Informática



**WALL•E**

Autor(es): Diego Berrios

Scarlett Oswald

Miguel Rivero

Alisson Visa

Curso: Proyecto I

Profesor: Diego Alberto Aracena Pizarro

Ricardo Elias Valdivia Pinto

ARICA, 11/09/2018

Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 16/08/2018 | 1.0 | Versión preliminar del formato | Scarlett Oswald Alisson Visa |
| 28/08/2018 | 1.1 | Gestión de Riesgos | Scarlett Oswald |
| 02/09/2018 | 2.0 | Corrección Formulación de Proyecto | Scarlett Oswald  Alisson Visa |
| 25/09/2018 | 2.1 | Diseño Arquitectura | Scarlett Oswald |
| 06/11/2018 | 3.0 | Corrección Avance I | Scarlett Oswald  Alisson Visa |
| 08/11/2018 | 3.1 | Diagrama de interacción entre programas | Scarlett Oswald |
| 13/11/2018 | 3.2 | Diseño Interfaz | Scarlett Oswald |
| 13/11/2018 | 3.3 | Código Programas implementados | Alisson Visa |
| 15/11/2018 | 3.4 | Diseño Arquitectura | Alisson Visa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

3.2. Asignación de tiempo

3.3. Personal-rol asignado

3.4. Gestión de Riesgos

1. Planificación de los Recursos

4.1. Recursos Hardware-Software requeridos

4.2. Estimación de Costos

1. Análisis – Diseño

5.1 Especificación de Requerimientos

5.2 Arquitectura Propuesta

5.3 Diseño de la Interfaz Usuario

1. Implementación

6.1 Descripción de los programas implementados

6.2 Diagrama de interacción entre programas

1. Resultados

7.1 Estado actual del proyecto

7.2 Problemas encontrados y soluciones propuestas

7.3 Conclusiones

1. Referencias

Anexos

Anexo A: Código de los programas implementados

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

1. Panorama General

## 1.1. Introducción

Durante este último tiempo existe un famoso juego llamado “cubo Rubik”, el cual consta de armarlo de diferentes maneras, ya que el armado contiene diferentes movimientos para poder dejar cada cara en un solo color. Por otro lado los ingenieros quisieron levar su creatividad a otro nivel, esto conlleva a armar un robot programado para que mediante diferentes algoritmos pueda armar el cubo Rubik.

## 1.2. Objetivo General

Desarrollar un robot funcional que con la ayuda de una persona logre armar un cubo Rubik 3x3.

## 1.3. Objetivos Específicos

* Construir un robot en base a piezas lego, que sea capaz de emplear ciertos algoritmos para armar un cubo Rubik.
* Manipular el robot para que sea capaz de comunicarse vía remota con el computador.

## 1.4. Restricciones

* No se cuenta con los sensores de colores.
* Tiempo asignado para el desarrollo del proyecto

## 1.5. Entregables

|  |  |  |
| --- | --- | --- |
| Identificación Entregable | Descripción Entregable | Fecha de Entrega |
| Formulación del Proyecto  “Informe I” | Planificación y distribución de las tareas de cada miembro del equipo. | 11/09/2018 |
| Presentación Formulación del Proyecto | Presentación sobre la Formulación del Proyecto. | 11/09/2018 |
| Manual de Usuario | Información detallada de cómo se opera el robot adecuadamente. |  |
| Formulación del Proyecto Avance I  “Informe II” | Modificación de la Formulación del Proyecto. | 25/10/2018 |
| Presentación Avance I | Presentación sobre el Avance I. | 25/10/2018 |
| Bitácoras | Breve descripción de lo realizado en la semana y lo que se debe realizar la semana siguiente. | Todos los jueves |
| Informe Avance I  Corregido | Corregir el Informe de Avance I | 23/11/2018 |

2.Organizacion del Personal

## 2.1. Descripción de Roles

* Jefe de Proyecto será el responsable de distribuir las tareas a los miembros del equipo, verificará que se realicen todas las tareas en el tiempo establecido y además estará encargado de actualizar la wiki del Proyecto.
* Programador será responsable de programar y modificar los algoritmos del cubo Rubik.
* Constructor se encargará del armado del robot.
* Documentación se encargará de los informes, bitácoras y la preparación de las presentaciones.

## 2.2. Personal que cumplirá los Roles

|  |  |  |
| --- | --- | --- |
| Actividad | Involucrados | Responsable(s) |
| Jefe de Proyecto | Scarlett Oswald | Scarlett Oswald |
| Programación | Miguel Rivero  Diego Berrios | Miguel Rivero |
| Constructor | Diego Berrios | Diego Berrios |
| Documentación | Scarlett Oswald  Alisson Visa | Scarlett Oswald  Alisson Visa |

## 2.3. Mecanismos de Comunicación

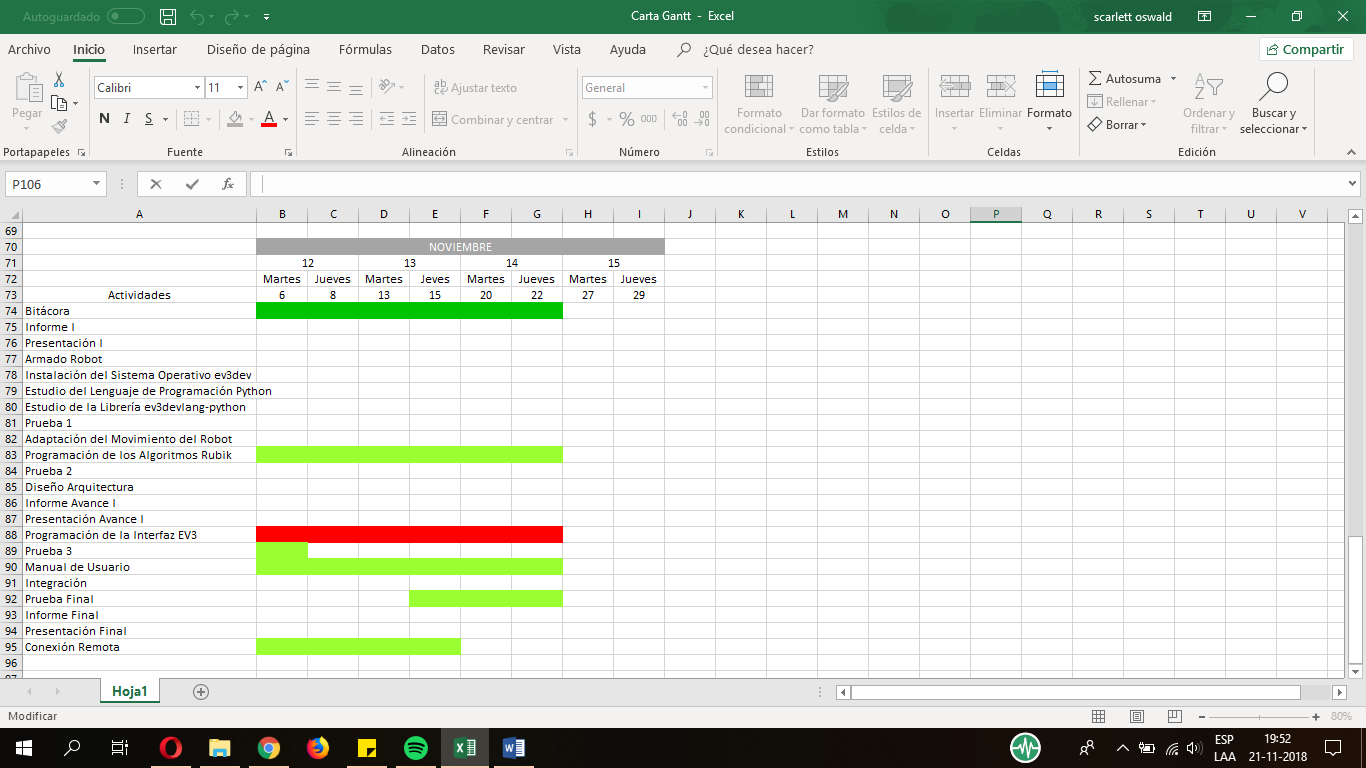
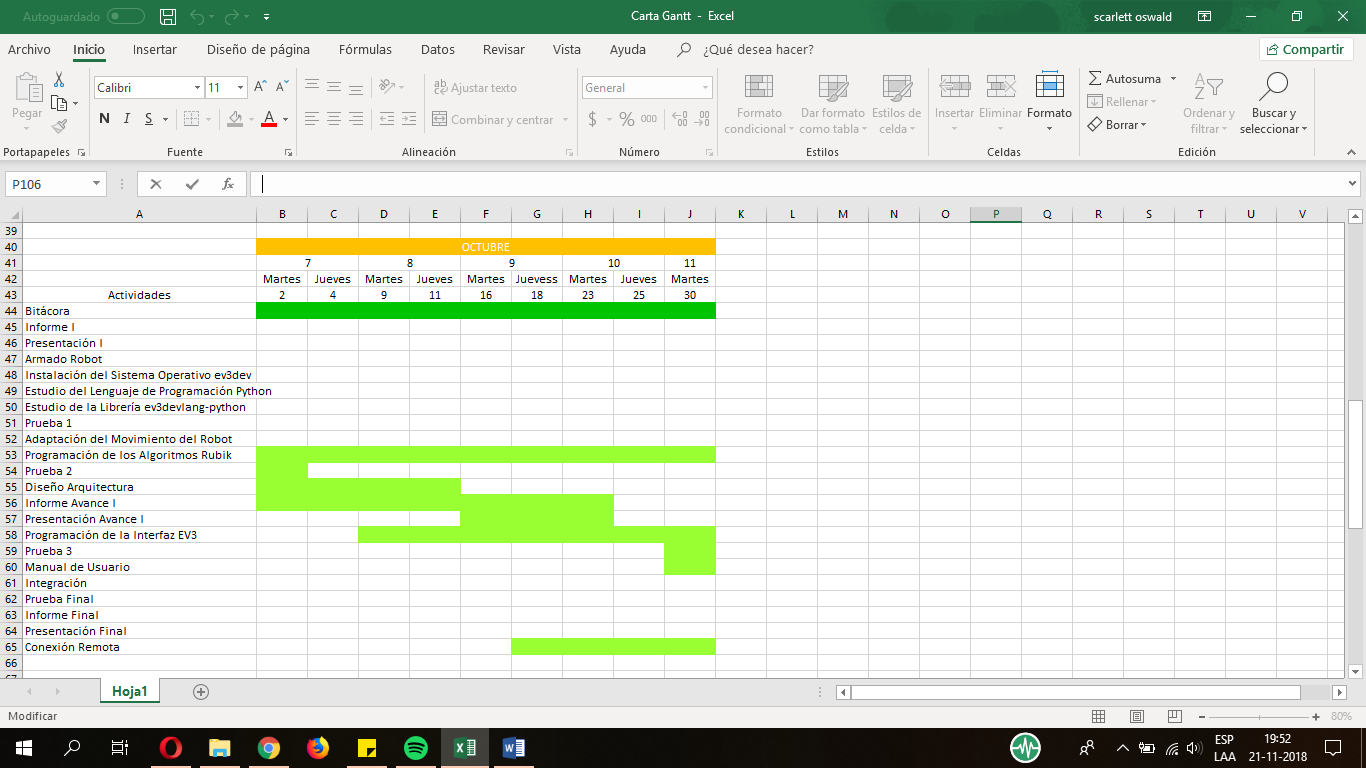
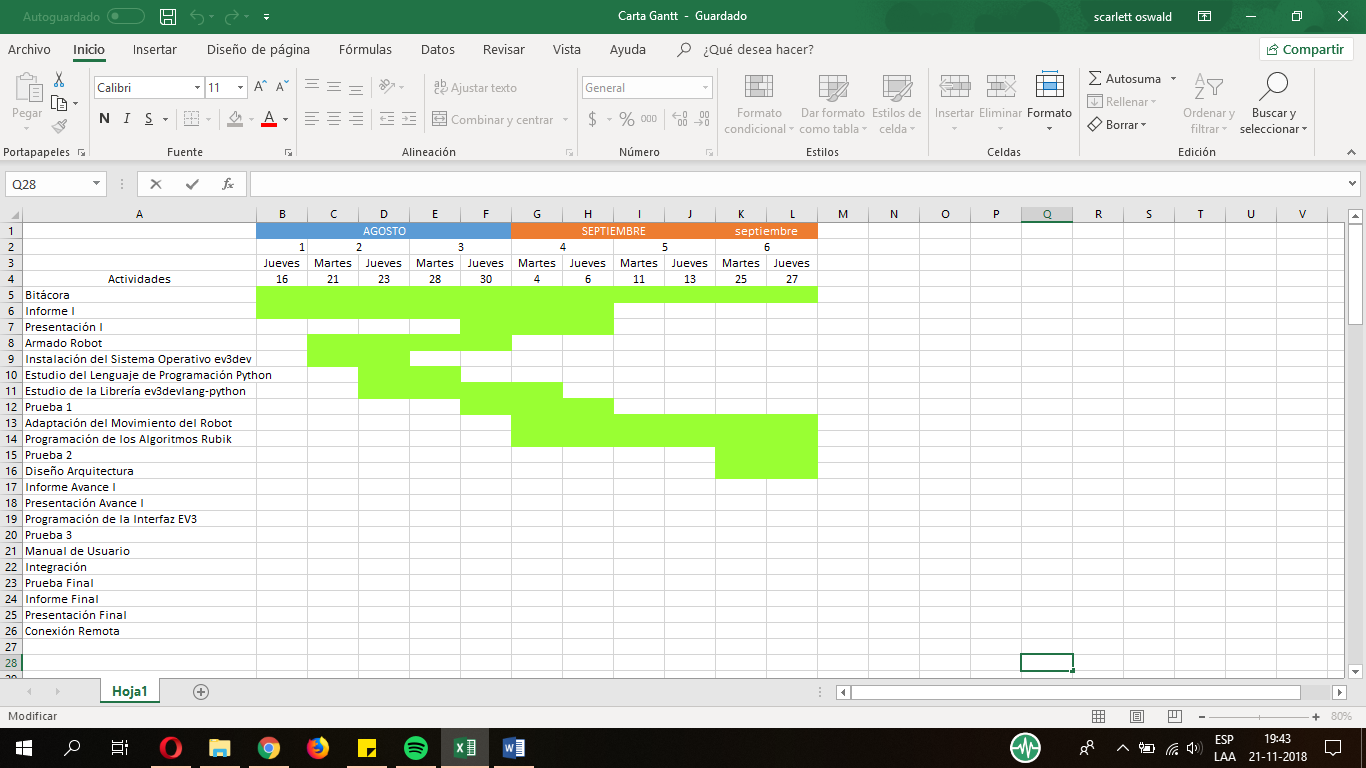
Toda comunicación de los miembros del equipo con sus respectivos docentes será por medio del sistema intranet de la Universidad, al mismo tiempo la comunicación entre los integrantes del equipo será mediante WhatsApp y Google Drive donde se compartirán los informes, presentaciones o cualquier otro material que sea necesario para el desarrollo del Proyecto. Por otra parte para el acceso a cualquier tipo de archivo publicado por los docentes será por medio del sistema redmine.

3.Planificacion del Proyecto

## 3.1. Actividades

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Descripción | Responsables | Producto |
| Formulación del Proyecto “Informe I” | Planificación y distribución de las tareas a cada miembro del equipo. | Scarlett Oswald  Alisson Visa | Concretado |
| Armado del Robot | Armado del robot EV3. | Diego Berrios | Concretado |
| Instalación Sistema Operativo | Instalación del Software en el robot. | Miguel Rivero | Concretado |
| Programación de los Algoritmos | Programación de los algoritmos del cubo Rubik con el lenguaje Python. | Miguel Rivero | Iniciado |
| Programación de la Interfaz EV3 | Programación de los movimientos del robot mediante una aplicación. | Diego Berrios | Iniciado |
| Pruebas | Pruebas del funcionamiento del robot con el cubo Rubik. | Diego Berrios  Miguel Rivero | Iniciado |
| Diseño de la Arquitectura | Proceso de comunicación entre el Robot ev3dev y vía remota. | Scarlett Oswald | Concretado |
| Integración | Proceso en el cual se junta toda información desarrollada durante el tiempo transcurrido para obtener el proyecto final. | Diego Berrios  Miguel Rivero  Scarlett Oswald | iniciado |
| Informe II | Información detallada de todo lo que se trabajó en el proyecto. | Alisson Visa | Concretado |
| Manual de Usuario | Información detallada de cómo se opera el robot adecuadamente. | Scarlett Oswald | iniciado |
| Bitácoras | Breve descripción de lo realizado en la semana y lo que se debe realizar la semana siguiente. | Alisson Visa | Iniciado |

## 3.2. Asignación de tiempo



## 3.3. Personal rol-asignado

* Jefe de Proyecto asignado a Scarlett Oswald.
* Programación asignada a Miguel Rivero y Diego Berrios.
* Constructor asignado a Diego Berrios.
* Documentación asignada a Alisson Visa y Scarlett Oswald.

## 3.4. Gestión de Riesgos

Niveles de impacto:

1. Catastrófico
2. Crítico
3. Circunstancial
4. Irrelevante

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgos | Probabilidad de Ocurrencia | Nivel de Impacto | Acciones Remediables |
| Salida de un integrante del equipo | 10% | 2 | Repartir todas las tareas de dicha persona con el resto de los miembros del equipo. |
| Enfermedad o accidente de algún integrante | 10% | 3 | Reorganizar el equipo de tal forma que se pueda cubrir en su totalidad la labor asignada a dicho miembro. |
| Falta de piezas | 15% | 4 | Reemplazar la pieza faltante con alguna pieza parecida. |
| Error en construcción del robot | 25% | 2 | Identificar la parte donde esta el error y se procederá a arreglar dicha parte. |
| Daño de la tarjeta SD | 50% | 1 | Cambiar la tarjeta SD por una nueva y volver a realizar todo lo ya realizado en la tarjeta dañada |
| Falla en los motores | 45% | 2 | Cambiar el motor averiado del robot por uno en buen funcionamiento. |

4. Planificación de Recursos

## 4.1. Recursos Hardware-Software requeridos

|  |  |
| --- | --- |
|  | Producto |
| Hardware | Robot EV3dev |
| Software | Software NXC |

## 4.2. Estimación de Costos

|  |  |
| --- | --- |
| **Producto** | **Valor CLP** |
| **Software** | **$0 (Software libre)** |
| **Set lego Mindstorms** | **$490.209** |
| **Cubo Rubik** | **$20.000** |
| **Tarjeta SD** | **$5.000** |
| **Adaptador nano** | **$7.000** |

* 1 Hora de trabajo: $9.000
* Total de horas: 81 horas
* Costo por persona: $729.000
* Costo total: 2.916.000

|  |  |
| --- | --- |
|  | **Costo** |
| **Recursos** | **$522.209** |
| **Recursos Humanos** | **$2.916.000** |
| **Total** | **$3.438.209** |

5. Análisis – Diseño

## 5.1 Especificación de Requerimientos

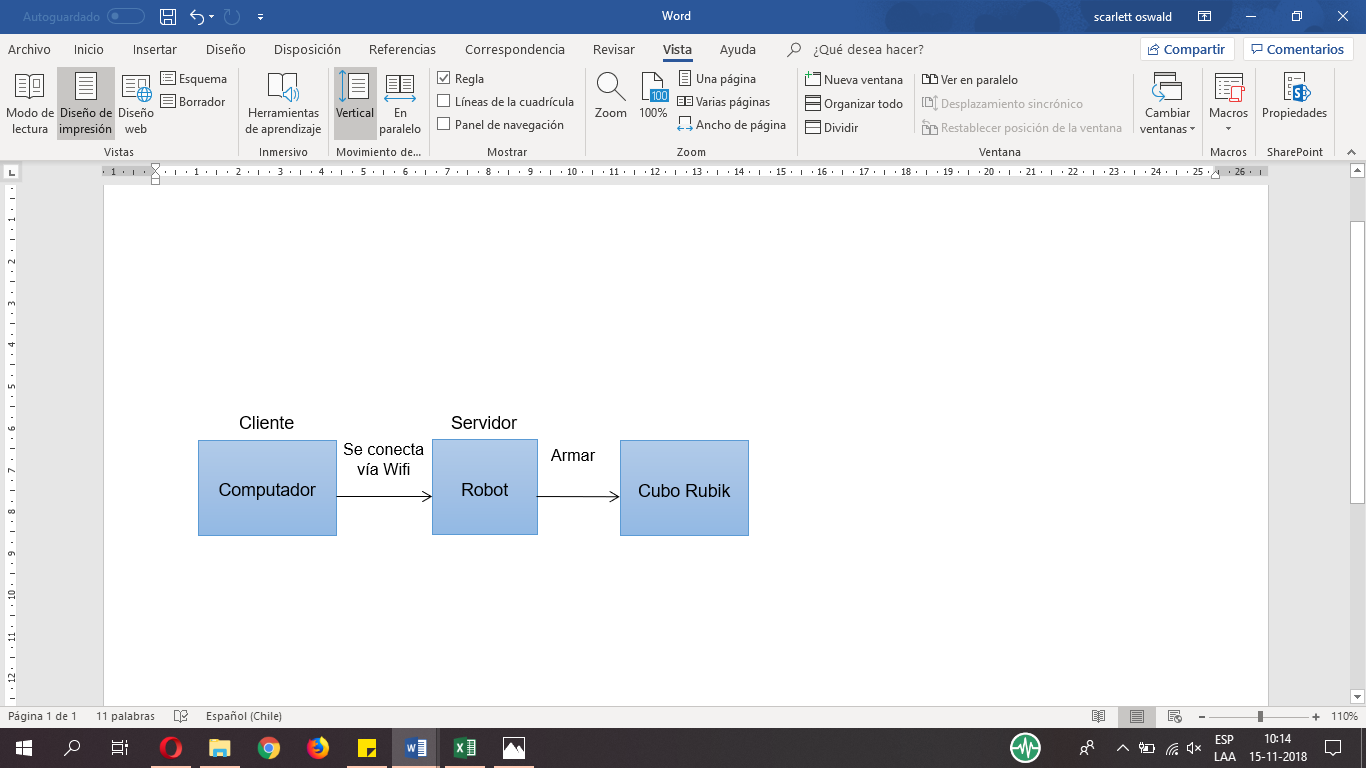
Requerimientos Funcional:

* Se implementará un robot el cual se va a comunicar vía Wifi con el computador, por lo tanto le permitirá al usuario poder elegir algoritmos mediante unainterfaz gráfica desarrollada en él.
* La interfaz gráfica contendrá los algoritmos necesarios para resolver el cubo.

Requerimientos No Funcionales:

* Cada movimiento que ejecutará el robot deberá realizarse en un tiempo de 20 segundos.
* El sistema debe contar con un manual de usuario estructurado adecuadamente.
* El sistema debe poseer interfaces gráficas bien formadas.

## 5.2 Arquitectura Propuesta



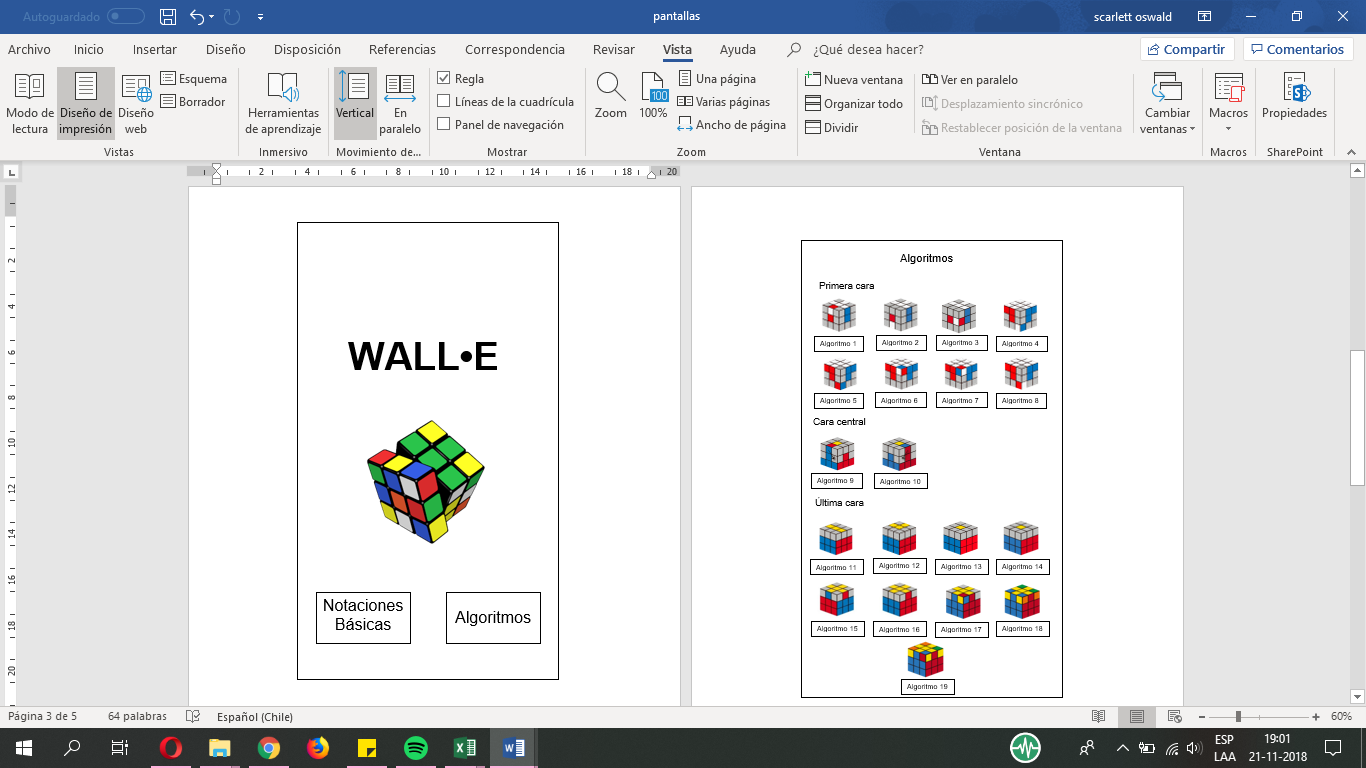
## 5.3 Diseño de la Interfaz Usuario

Pantalla inicial

Notaciones Básicas

Algoritmos

Descripción de las pantallas:



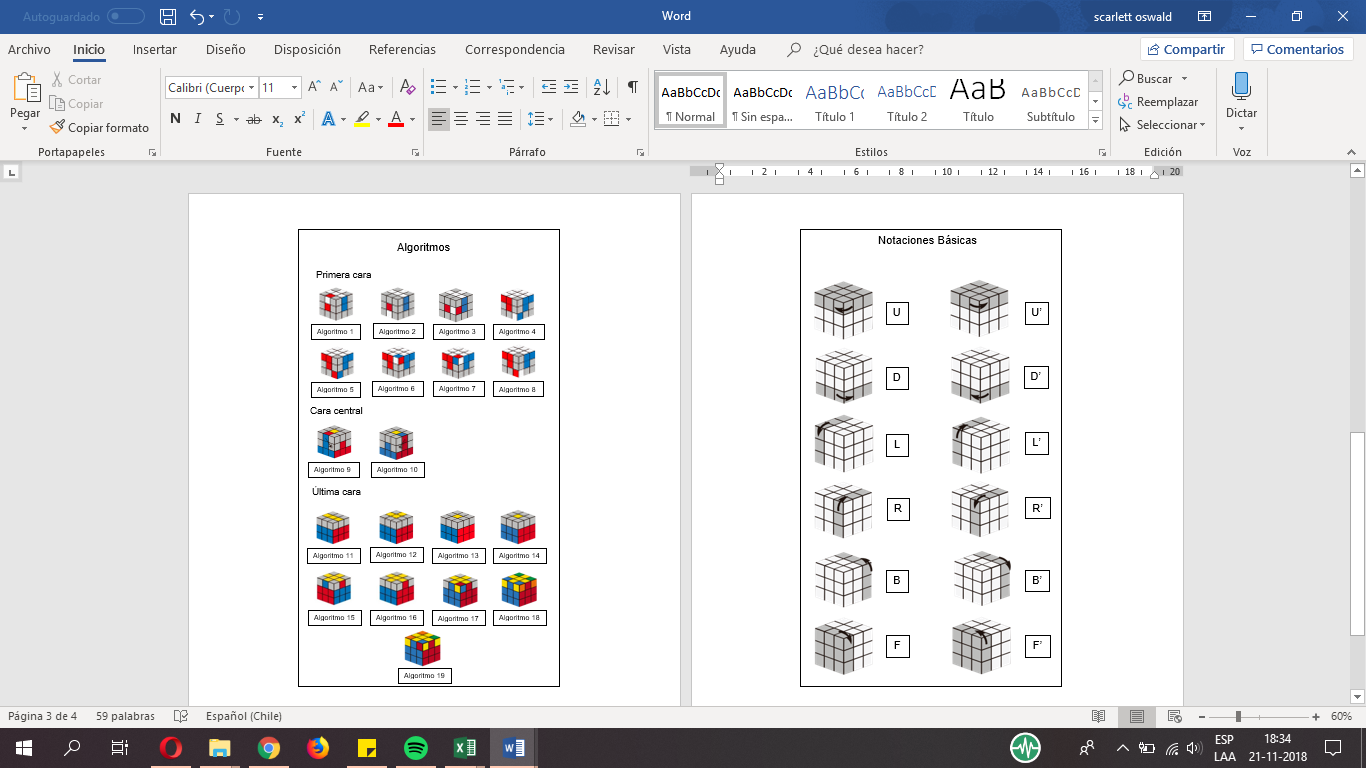
Pantalla inicial

Esta pantalla consta de dos opciones que llevaran a dos ventanas distintas que son:

* Notaciones Básicas
* Algoritmos

Notaciones Básicas

Algoritmos

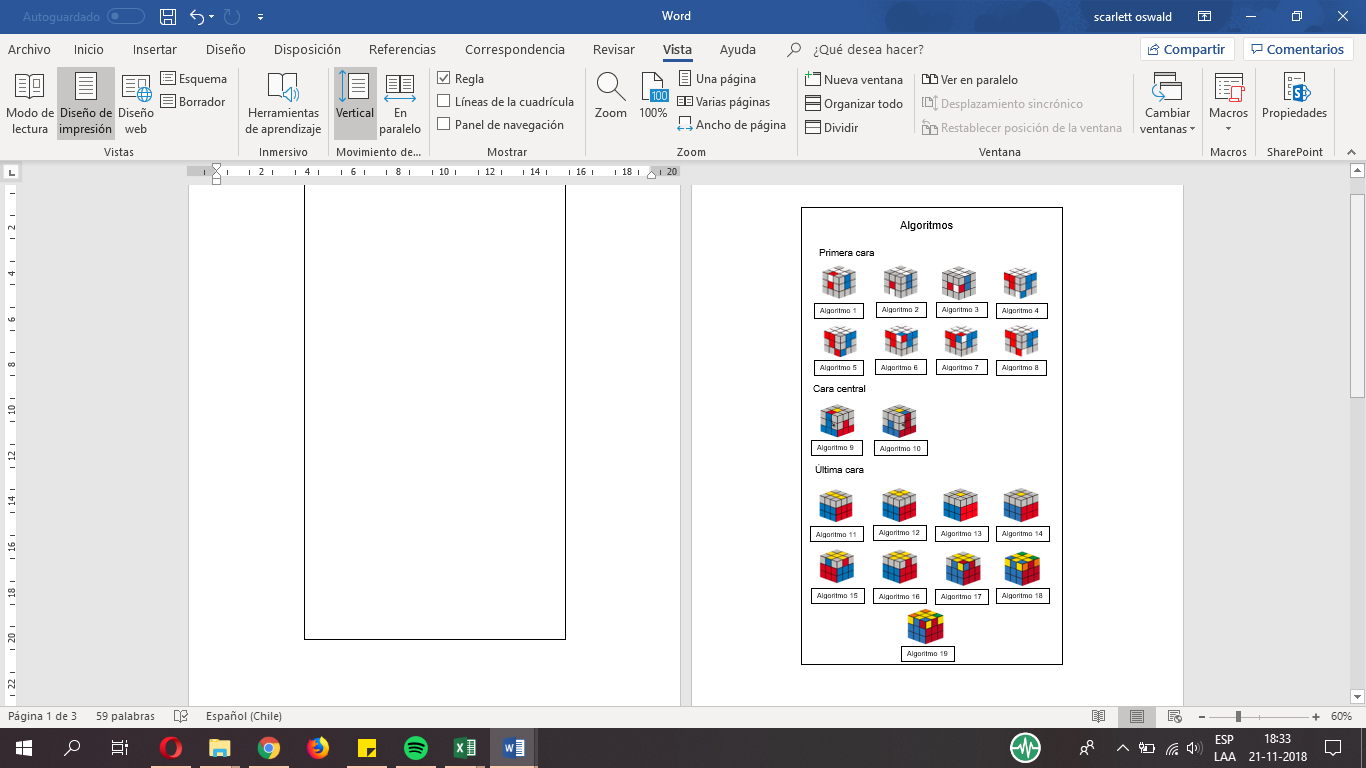


Pantalla Notaciones Básicas

Esta pantalla consta de los movimientos básicos del robot, el cual funcionara al apretar el movimiento que se desea realizar por ejemplo, si se quiere realizar el movimiento B se debe apretar en donde salga B y si se quiere realizar el movimiento opuesto se debe apretar en donde salga B´.

B

B´

Pantalla Algoritmos

Esta pantalla consta de todos los algoritmos básicos que se requieren para armar el cubo, el cual funciona presionando en la parte baja de las imágenes. Se debe elegir el algoritmo adecuado para la parte que se desea mover.

Algoritmo 7

Algoritmo 18

6. Implementación

## 6.1 Descripción de los programas implementados

Cabe mencionar que en la implementación de los programas fueron realizados con la clase importada Large Motor, por ende estos no devuelven nada ya que solo realizan acciones.

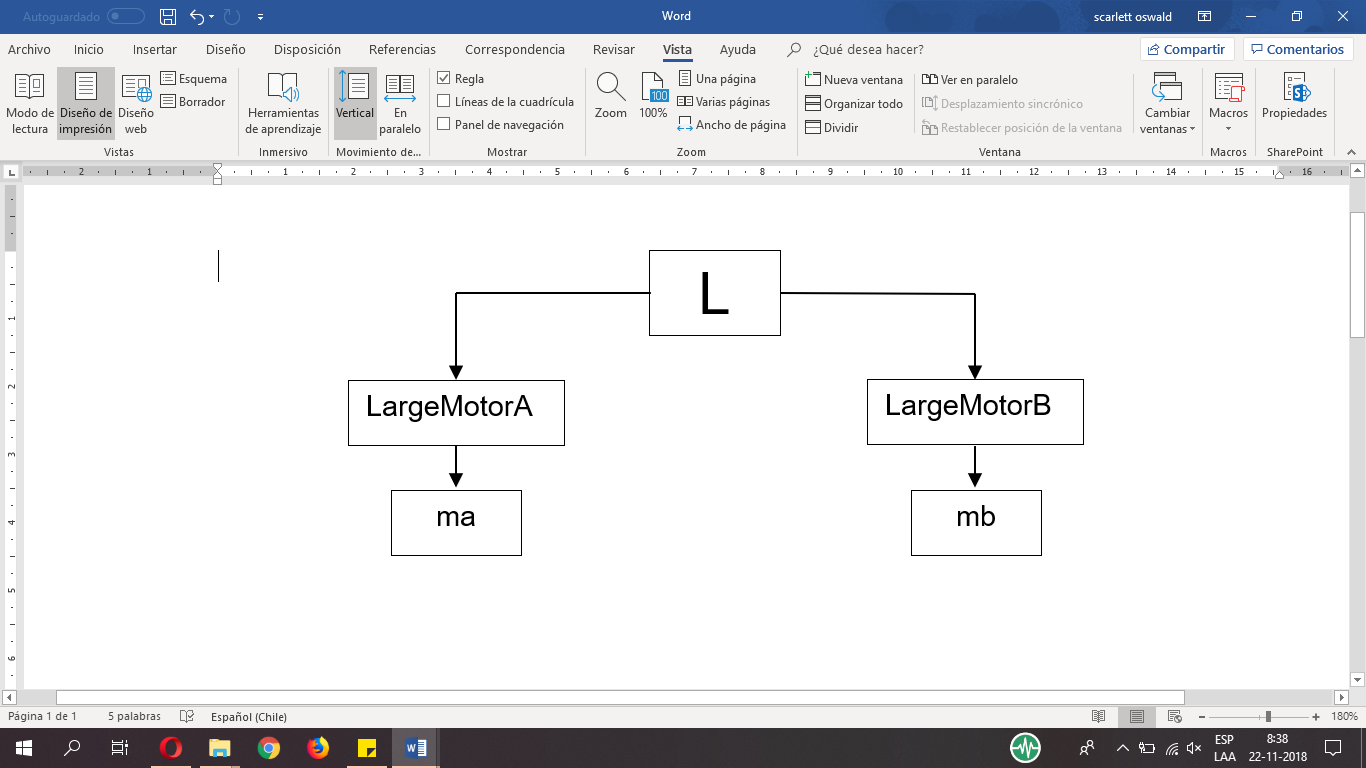
* + - LargeMotorA: Esta clase tiene el deber de realizar movimientos en el brazo.
    - LargeMotorB: Esta clase tiene el deber de realizar el movimiento de la base.
    - mb.run\_to\_rel\_pos: La variable “mb” esta asignada a LargeMotorB, por ende “mb” será quien haga actuar la base, la cual realiza movimientos dados ciertos grados y cierta velocidad.
    - ma.run\_to\_rel\_pos: La variable “ma” esta asignada a LargeMotorA, por ende “ma” será quien haga actuar el brazo, la cual el brazo realizará movimientos hacia adelante y atrás, además de sostener el cubo.

La siguiente tabla incluye las notaciones básicas junto con sus respectivas notaciones primas.

|  |  |
| --- | --- |
| **Movimientos Básicos** | **Notaciones Primas** |
| **def D:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | **def DPrima:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |
| **def L:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | **def L Prima:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |
| **def R:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | **def R Prima:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |
| **def B:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | **def BPrima:**  Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. |
| **def U:**  **Imagen que contiene shoji, edificio  Descripción generada automáticamente**  Entrada: ma, mb. | **def U Prima:**  Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. |
| **def F:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | **def F Prima:**  Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |

## 6.2 Diagrama de interacción entre programas

Los movimientos básicos del robot se forman a partir de las notaciones básicas del cubo Rubik, por ejemplo si queremos realizar el movimiento básico “L” se realizaría de la siguiente forma:



Posteriormente, con las notaciones básicas podemos formar algoritmos, por ejemplo si queremos formar el algoritmo “Sexy Move” seria de la siguiente forma:

Sexy Move

R

R´

U´

U

7. Resultados

## 7.1 Estado actual del proyecto

El proyecto actualmente se encuentra en la implementación de los algoritmos del armado del robot, el desarrollo de la interfaz gráfica y en el inicio del manual de usuario. El proyecto ya cuenta con los códigos de los movimientos básicos del cubo que son:  RIGHT(R), FRONT(F), UP(U), BACK(B), LEFT(L), DOWN(D). Dicho lo anterior también tenemos implementado las notaciones primas de lo mismo, además logramos realizar la conexión entre el computador y el robot, lo cual no fue tan sencillo como lo esperábamos.

Imagen que contiene interior, suelo, mesa, pared

Descripción generada con confianza muy alta

Imagen que contiene suelo, interior, pequeño, pared

Descripción generada con confianza muy alta

## 7.2 Problemas encontrados y soluciones propuestas

|  |  |
| --- | --- |
| Problemas encontrados | Soluciones propuestas |
| Fallo en el sistema operativo | Formatear la tarjeta SD y volver a instalar el sistema operativo. |
| Conexión del robot con el computador | Cambiar la versión de Python, de la versión 3.7.0 a la versión 3.4.0 |

## 7.3 Conclusiones

A lo largo de la realización del proyecto, nos encontramos con varios problemas, como la construcción del robot, falta de piezas, problemas con la tarjeta de SD, conexión del robot con el computador, lo cual como equipo se lograron solucionar todos estos problemas. Por otro lado la documentación del proyecto cabe destacar que no fue un trabajo fácil.

8. Referencias

Programación NXC - Robots Lego NXT Mindstorms.

Anexos

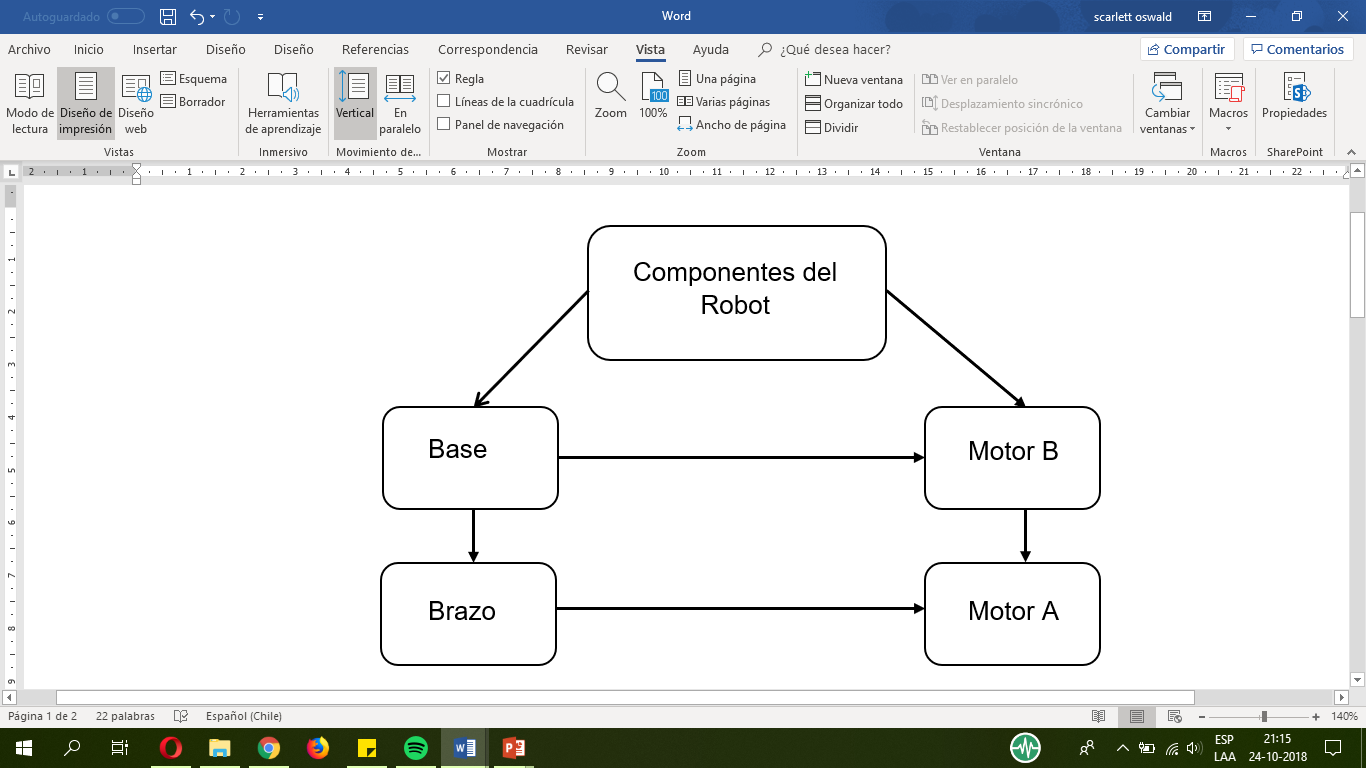
## Anexo A: Código de los programas implementados

Debido a que los códigos son bastante extensos, daremos una breve explicación de lo que realiza cada función:

|  |
| --- |
| Movimientos Básicos del Cubo Rubik |
| **def    D():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, por último el brazo gira siete veces el cubo. |
| **def    L():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, luego el brazo gira el cubo tres veces y luego vuelve a girar la base, después el brazo gira el cubo y gira la base, por último el brazo gira cuatro veces el cubo. |
| **def    R():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, luego el brazo gira tres veces el cubo y gira la base, después el brazo gira el cubo y gira la base dos veces, por último el brazo gira cuatro veces el cubo y gira la base. |
| **def    B():**  En esta función el brazo gira tres veces el cubo y gira la base, por último gira cinco veces el cubo. |
| **def    U():**  En esta función el brazo gira cinco veces el cubo y gira la base, por último gira tres veces el cubo |
| **def    F():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior dos veces, luego el brazo gira tres veces el cubo, por último la base gira y el brazo gira el cubo. |

|  |
| --- |
| Movimientos (Notaciones Primas) |
| **def    DPrima():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, por último el brazo gira siete veces. |
| **def    LPrima ():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, luego el brazo gira tres veces, después la base gira y luego el brazo gira el cubo, la base gira y el brazo gira cuatro veces, por último la base gira dos veces. |
| **def    RPrima ():**  En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, luego el brazo gira el brazo tres veces, luego la base gira , luego el brazo gira el cubo, después la base gira dos veces, por último el brazo gira cuatro veces el cubo y gira la base. |
| **def    BPrima ():**  En esta función el brazo gira tres veces y gira la base, por último el brazo gira cinco veces el cubo. |
| **def    UPrima ():**  En esta función el brazo gira cinco veces y gira la base, por último el brazo gira tres veces el cubo. |
| **def    FPrima ():**  En esta función el brazo sostendrá el cubo mientras la base gira dos veces la capa inferior, luego el brazo gira tres veces el cubo, por último gira la base y el brazo gira el cubo. |

## Anexo B: Robot



Descripción:

* Base:  Plataforma del cubo encargada de girar el cubo en 360° en su propio eje.
* Brazo: La funcionalidad del motor A es mover las caras de cubo.

1. **Levantar y Bajar**: Para ejecutar estas acciones se requiere el uso del Motor A, el cual le permitirá la movilidad de los brazos del robot, de esta forma al accionar el motor correspondiente con una velocidad del 50% y con un tiempo de x segundos.
2. **Giro**: Para llevar a cabo el giro se deberá tener los brazos del robot abajo y además se usará el motor B, ya que este poseerá el control de la plataforma donde se encuentra el cubo, de esta manera al girar la plataforma al lado derecho o izquierdo, según se requiera, los brazos detendrán al cubo mientras la plataforma girará la base del cubo y así realizar el movimiento de girar las caras.
3. **Vertical**: Esta acción consiste en que la funcionalidad del motor A es mover las caras del cubo.