**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 |
|  |  |  |  |
|  |  |  |  |

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

3.3. 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. | 25/10/18 |
| Bitácoras | Breve descripción de lo realizado en la semana y lo que se debe realizar la semana siguiente. | Todos los jueves |

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

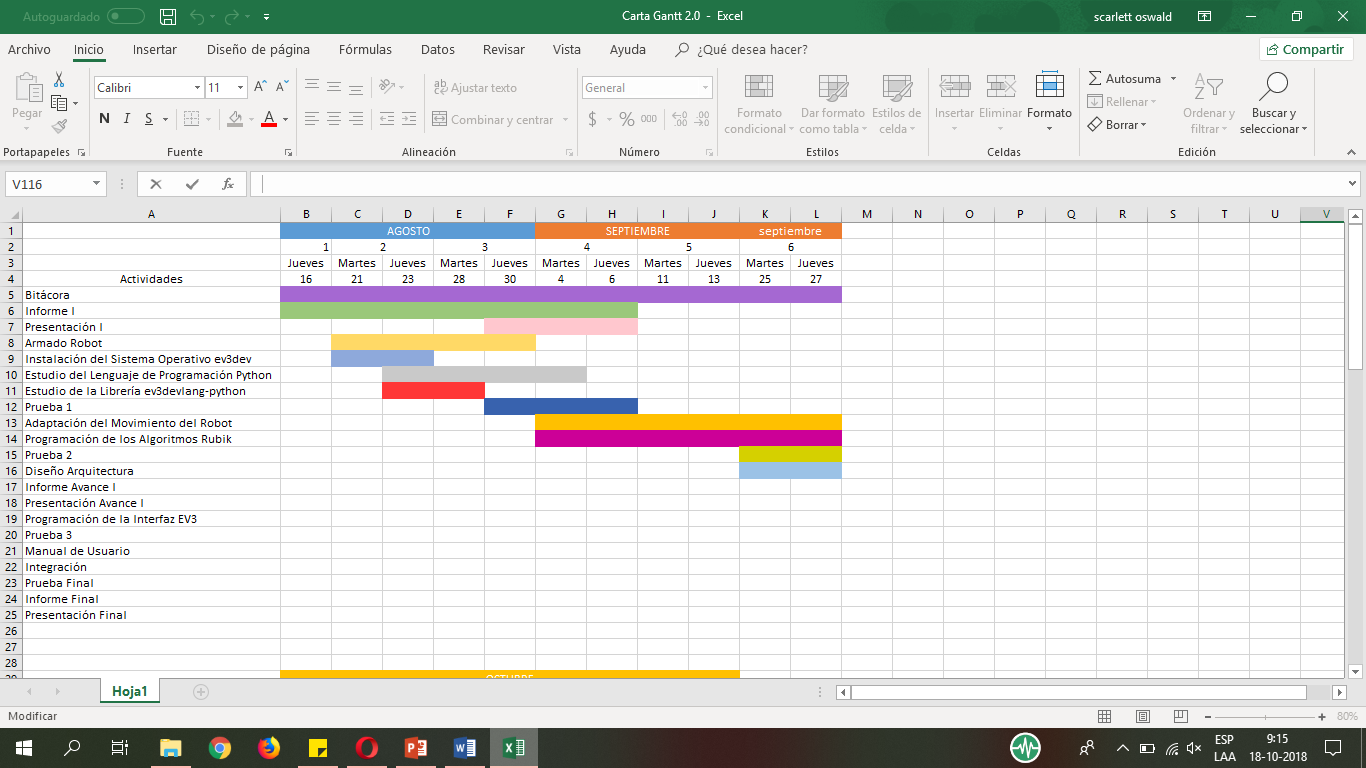
Toda comunicación de los miembros del equipo con sus respectivos docentes será por medio del sistema intranet de la Universidad, al mismo tiempo la comunicación entre los integrantes del equipo será mediante WhatsApp y Google Drive donde se compartirán los informes, presentaciones o cualquier otro material 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.

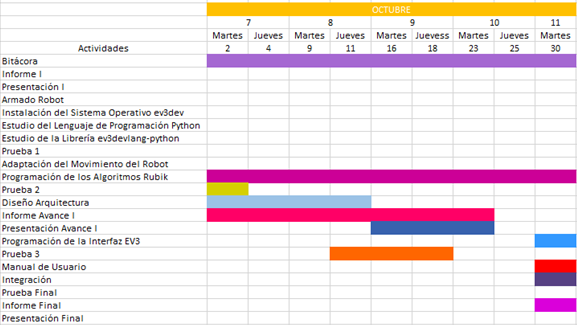
3.Planificacion del Proyecto

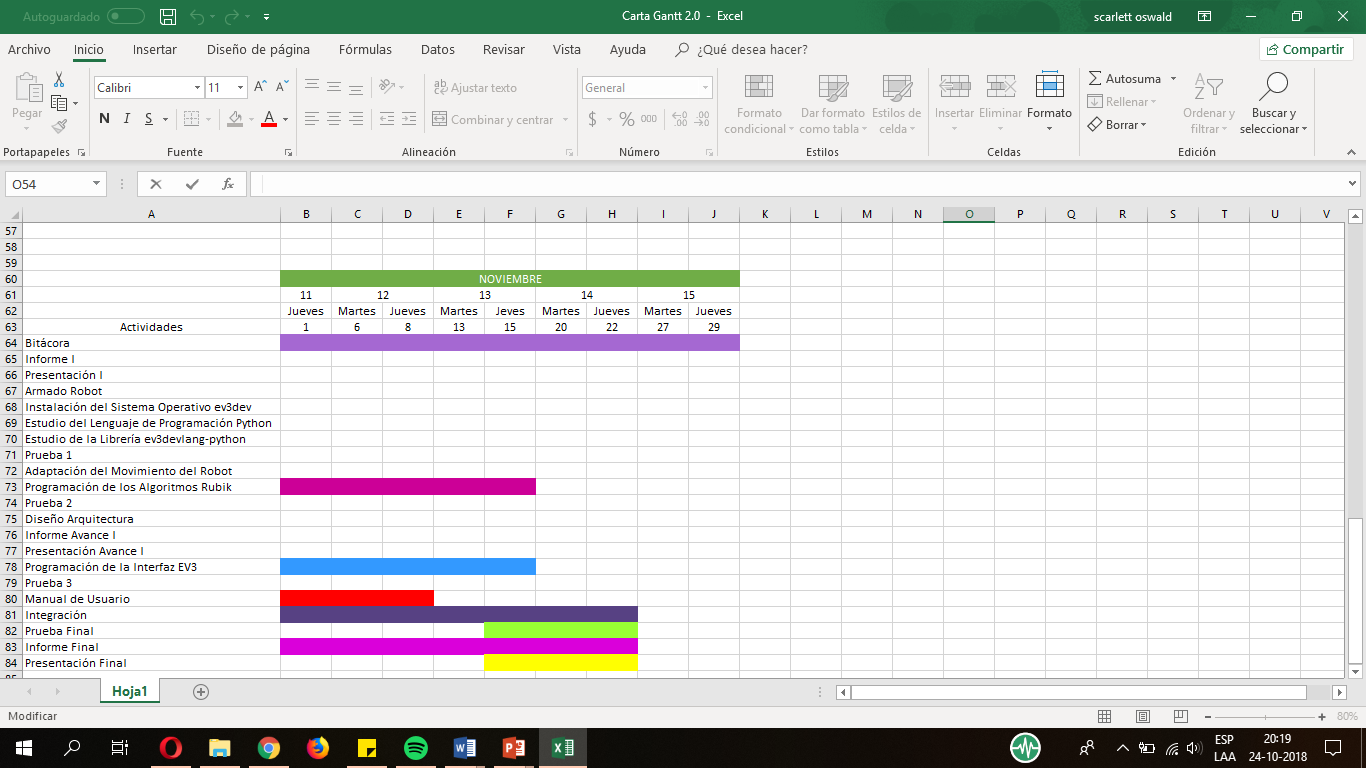
## 3.1. Actividades

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | No 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 | No 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

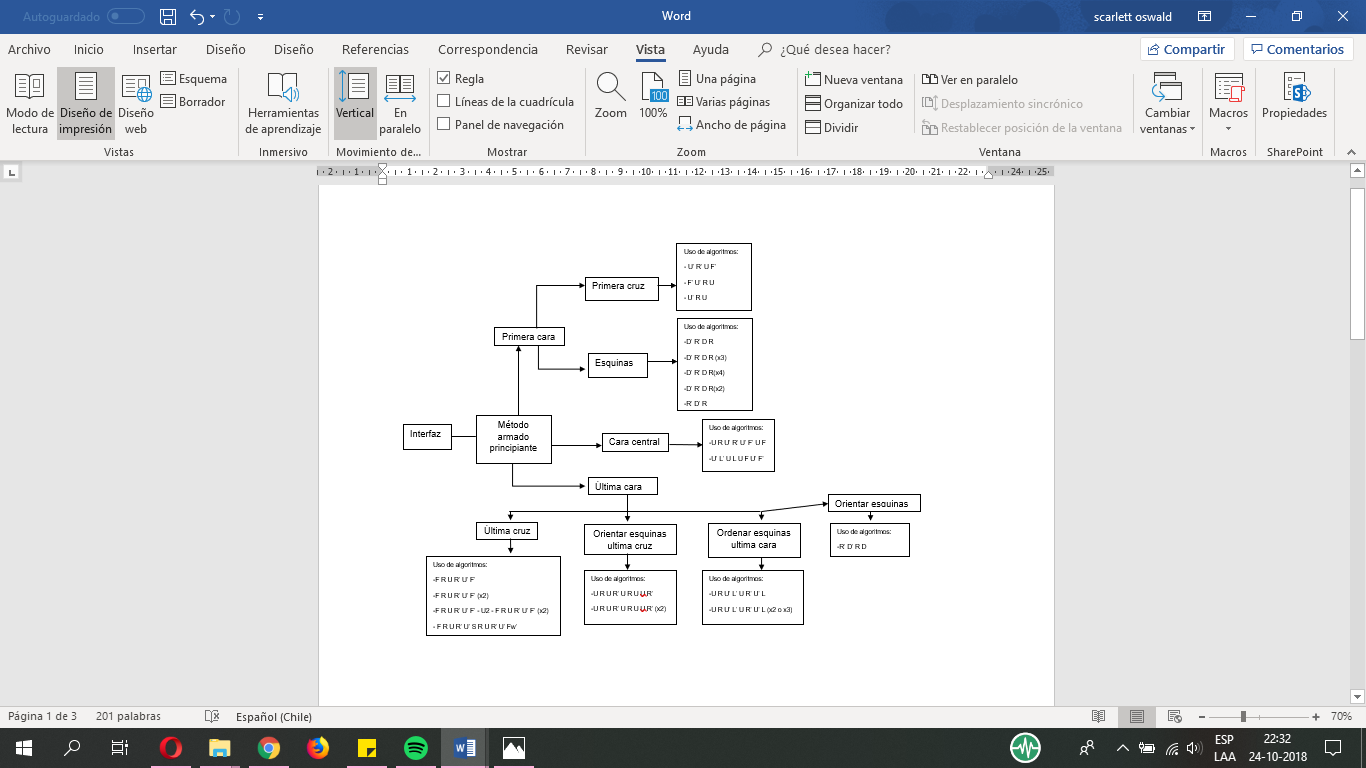
## 5.1 Especificación de Requerimientos

Requerimientos Funcional:

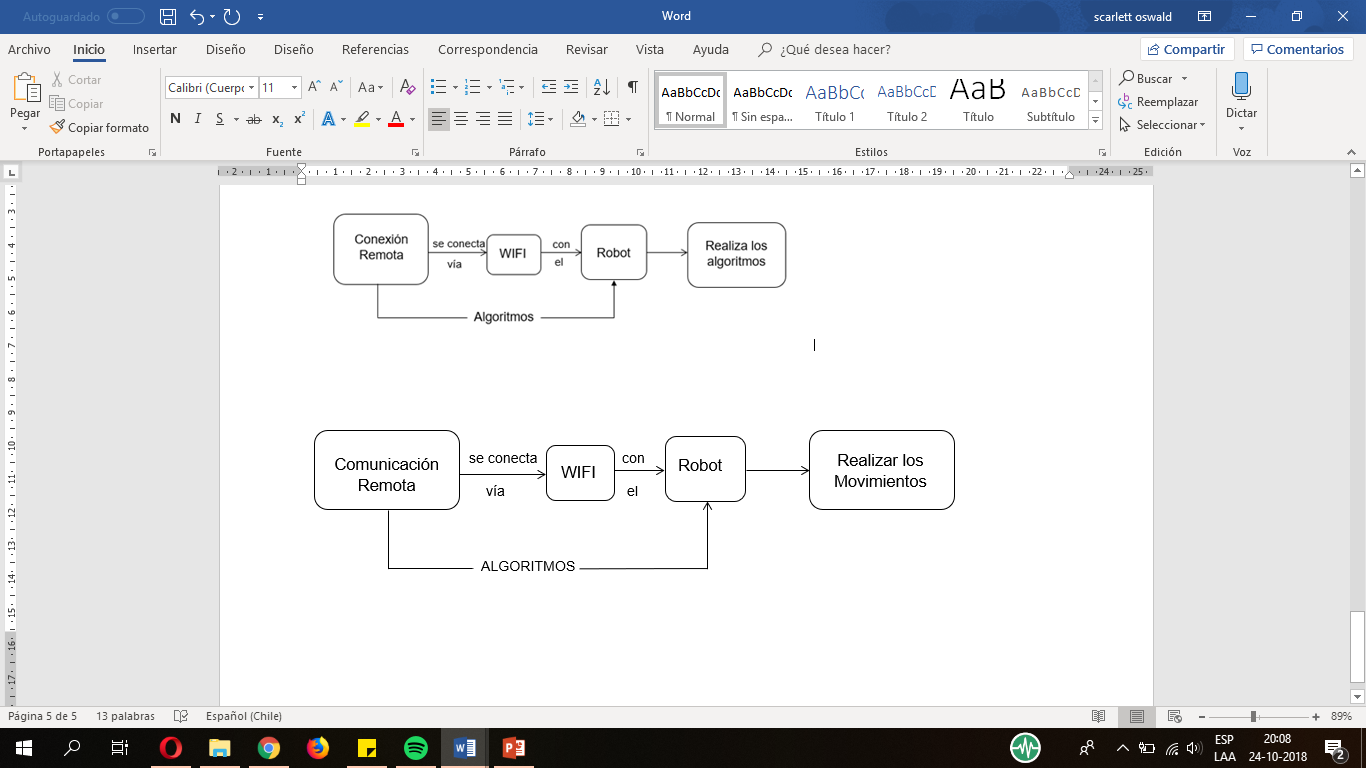
* Se implementará un robot el cual se va a comunicar vía Wifi con el computador, por lo tanto le permitirá al usuario poder elegir algoritmos mediante una **interfaz gráfica desarrollada en él.**
* La interfaz gráfica contendrá los algoritmos necesarios para resolver el cubo.

Requerimientos No Funcionales:

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



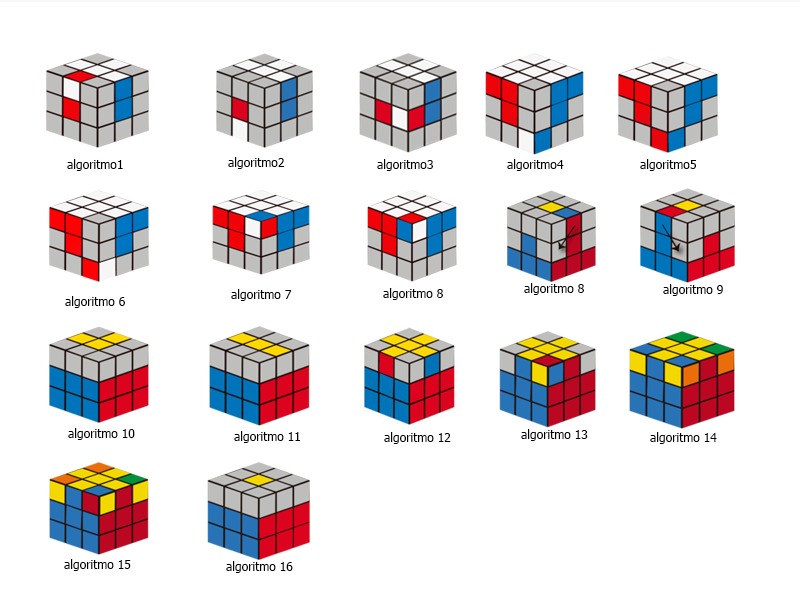
## 5.2 Arquitectura Propuesta



**Descripción de la Arquitectura:**

1. Comunicación remota: El computador se conecta vía wifi con el robot, el cual manda los algoritmos que se deben realizar al robot.
2. Robot: Recibe los algoritmos que son mandados por el computador.
3. Realizar movimientos: Ejecuta el movimiento del algoritmo que fue elegido.

## 5.3 Diseño de la Interfaz Usuario



6. Implementación

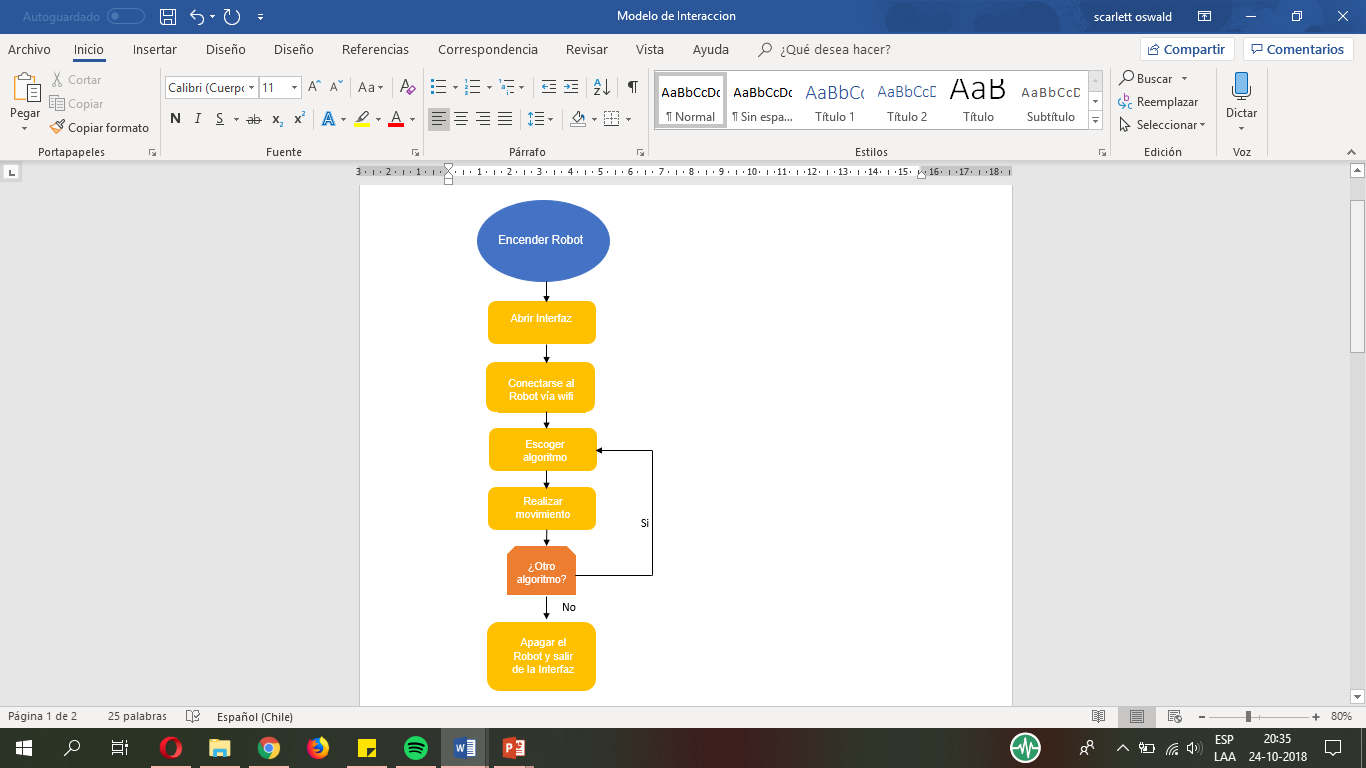
6.1 Descripción de los programas implementados

Programas de entrada: Se coloca el cubo en la base del robot.

Proceso: Se realiza la ejecución de la notación del cubo Rubik que fue implementado en nuestro programa.

Programas de Salida: Se realiza la ejecución del movimiento elegido.

6.2 Diagrama de interacción entre programas



Primero se debe encender el robot, luego abrir la interfaz. Cuando ya este el robot encendido y la interfaz este abierta se debe proceder a conectar la interfaz con el robot vía wifi, ya que esté listo la conexión se debe elegir el algoritmo que se desea realizar, con el algoritmo ya elegido el robot comenzara a realizar los movimientos de dicho algoritmo. Este proceso se realiza hasta que el cubo este completamente armado, cuando el cubo ya este armado se debe proceder a apagar el robot y la interfaz.

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).

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 | No se ha encontrado hasta ahora una solución factible. |

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

**1 #!/usr/bin/env python3**

**2 from time import sleep**

**3 from ev3dev.ev3 import \***

**4 def D():**

**5 ma = LargeMotor(‘outA’)**

**6 mb = LargeMotor(‘outB’)**

**7 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=200, stop\_action="hold")**

**8 sleep(3)**

**9 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**10 sleep(3)**

**11 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

12 **def R():**

**13 ma = LargeMotor(‘outA’)**

**14 mb = LargeMotor(‘outB’)**

**15 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**16 sleep(3)**

**17 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=200, stop\_action="hold")**

**18 sleep(3)**

**19 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**20 sleep(3)**

**21 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=200, stop\_action="hold")**

**22 sleep(3)**

**23 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**24 sleep(3)**

**25 ma.run\_to\_rel\_pos(position\_sp=-150, speed\_sp=300, stop\_action="hold")**

**26   def L():**

**27 ma = LargeMotor(‘outA’)**

**28 mb = LargeMotor(‘outB’)**

**29 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**30 sleep(3)**

**31 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**32 sleep(3)**

**33 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**34 sleep(3)**

**35 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**36 sleep(3)**

**37 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**38 sleep(3)**

**39 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**40   def B():**

**41 ma = LargeMotor(‘outA’)**

**42 mb = LargeMotor(‘outB’)**

**43 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**44 sleep(3)**

**45 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**46 sleep(3)**

**47 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**48 sleep(3)**

**49 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**50 sleep(3)**

**51 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**52   def U():**

**52 ma = LargeMotor(‘outA’)**

**53 mb = LargeMotor(‘outB’)**

**54 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**55 sleep(3)**

**56 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**57 sleep(3)**

**58 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**59 sleep(3)**

**60 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**61 sleep(3)**

**62 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**63 sleep(3)**

**64 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**65 sleep(3)**

**66 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**67   def F():**

**68 ma = LargeMotor(‘outA’)**

**69 mb = LargeMotor(‘outB’)**

**70 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**71 sleep(3)**

**72 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**73 sleep(3)**

**74 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**75 sleep(3)**

**76 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**77 sleep(3)**

**78 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**79 sleep(3)**

**80 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**81 sleep(3)**

**82 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**83   def DPrima():**

**84 ma = LargeMotor(‘outA’)**

**85 mb = LargeMotor(‘outB’)**

**86 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=200, stop\_action="hold")**

**87 sleep(3)**

**88 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**89 sleep(3)**

**90 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**91   def LPrima():**

**92 ma = LargeMotor(‘outA’)**

**93 mb = LargeMotor(‘outB’)**

**94 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**95 sleep(3)**

**96 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=200, stop\_action="hold")**

**97 sleep(3)**

**98 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=300, stop\_action="hold")**

**99 sleep(3)**

**100 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=200, stop\_action="hold")**

**101 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**102 sleep(3)**

**103 ma.run\_to\_rel\_pos(position\_sp=150, speed\_sp=300, stop\_action="hold")**

**104   def RPrima():**

**105 ma = LargeMotor(‘outA’)**

**106 mb = LargeMotor(‘outB’)**

**107 mb.run\_to\_rel\_pos(position\_sp=-278.8, speed\_sp=500, stop\_action="hold")**

**108 sleep(3)**

**109 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**110 sleep(3)**

**111 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**112 sleep(3)**

**113 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**114 sleep(3)**

**115 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**116 sleep(3)**

**117 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**118   def BPrima():**

**119 ma = LargeMotor(‘outA’)**

**120 mb = LargeMotor(‘outB’)**

**121 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**122 sleep(3)**

**123 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**124 sleep(3)**

**125 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**126 sleep(3)**

**127 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**128 sleep(3)**

**129 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**130   def UPrima():**

**131 ma = LargeMotor(‘outA’)**

**132 mb = LargeMotor(‘outB’)**

**133 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**134 sleep(3)**

**135 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**136 sleep(3)**

**137 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**138 sleep(3)**

**139 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**140 sleep(3)**

**141 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**142 sleep(3)**

**143 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**144 sleep(3)**

**145 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**146   def FPrima():**

**147 ma = LargeMotor(‘outA’)**

**148 mb = LargeMotor(‘outB’)**

**149 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**150 sleep(3)**

**151 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**152 sleep(3)**

**153 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

**154 sleep(3)**

**155 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

**156 sleep(3)**

**157 ma.run\_to\_rel\_pos(position\_sp=-160, speed\_sp=250, stop\_action="hold")**

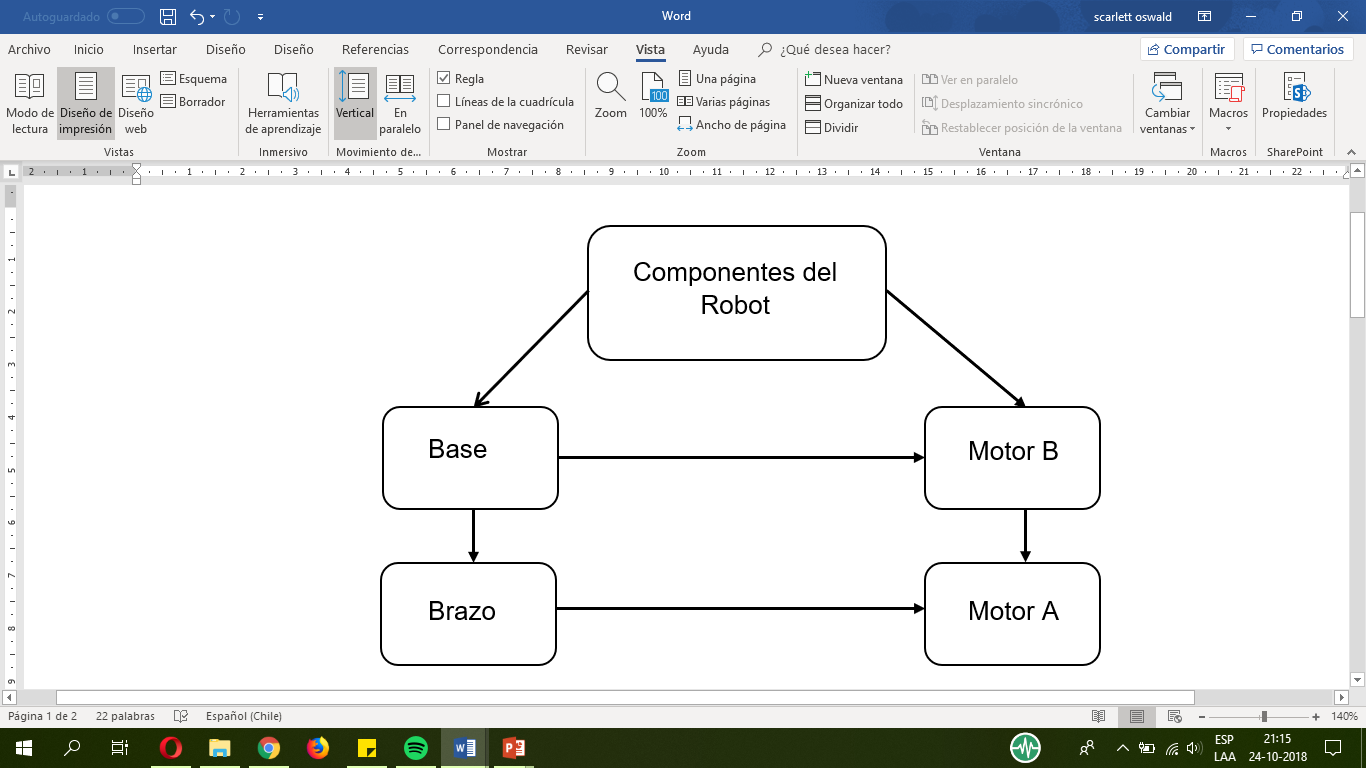
**158 sleep(3)**

**159 mb.run\_to\_rel\_pos(position\_sp=278.8, speed\_sp=500, stop\_action="hold")**

**160 sleep(3)**

**161 ma.run\_to\_rel\_pos(position\_sp=160, speed\_sp=300, stop\_action="hold")**

## 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.