**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 Elías Valdivia Pinto

ARICA, 07/12/2018

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
   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. Pruebas

7.1 Descripción de las pruebas realizadas

7.2 Resultados de las pruebas

1. Resultados

8.1 Estado Final del Proyecto

8.2 Problemas Encontrados

8.3 Conclusiones

8.4 Trabajo Futuro

1. Referencias

Anexos

Anexo A: Código de los programas implementados

Anexo B: Robot

1. Panorama General

## 1.1. Introducción

Durante este último tiempo existe un famoso juego llamado “Cubo Rubik”, se trata de un rompecabezas mecánico, el cual a través de un mecanismo de ejes permite a cada cara girar independientemente, mezclando así los colores.

Existen diferentes movimientos, ya sean básicos o complejos, los cuales le permiten poder armar el cubo de diferentes maneras hasta dejarlo en su forma original.

Gracias a este invento, comenzaron a surgir las competencias de armado de este llamado speedcubing. El speedcubing es la actividad de resolver el cubo de Rubik con la mayor rapidez posible. 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.

Por otro lado los ingenieros quisieron llevar su creatividad a otro nivel, esto conlleva a construir un robot que con la ayuda de una persona sea capaz de armar un cubo Rubik en el menor tiempo posible.

Finalmente para aquello se necesita construir un robot programado, lo cual mediante diferentes algoritmos logrará dar total movilidad a las piezas fundamentales del robot y con ello poder realizar el armado del cubo Rubik. Por ende los algoritmos básicos o complejos que realice el robot serán programados a través del lenguaje de programación “Python”.

## 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.
* Operar el robot para que sea capaz de comunicarse vía remota con el computador.

## 1.4. Restricciones

* No se cuenta con los sensores de colores.
* El tiempo asignado para el desarrollo del proyecto es de solo un semestre.
* Lenguaje de Programación Python.

## 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 |
| Informe Final y Presentación Final | Integración de los informes anteriores e inclusión de las pruebas y el estado final del proyecto | 07/12/2018 |
| Manual de Usuario | Instructivo que ayudara al cliente a utilizar el manejo del robot. | 07/12/2018 |
|  |  |  |

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.

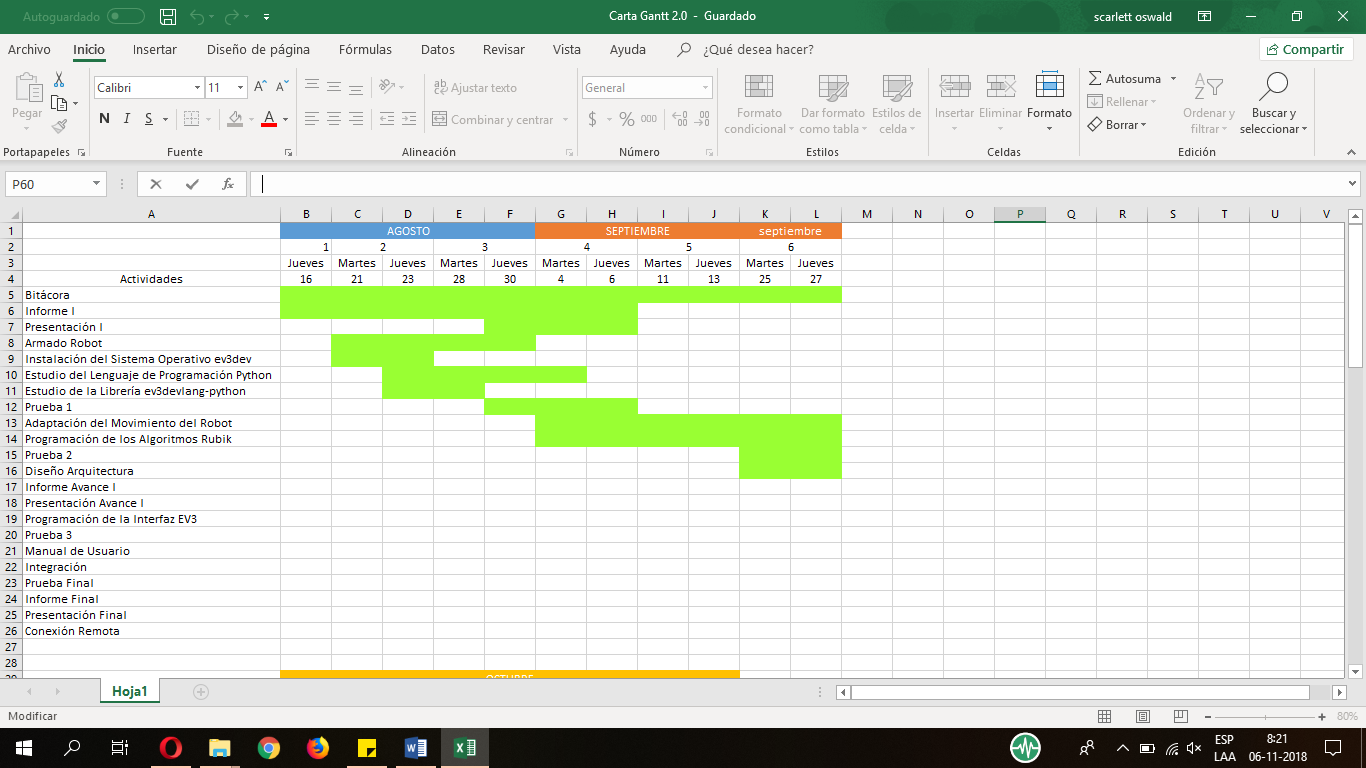


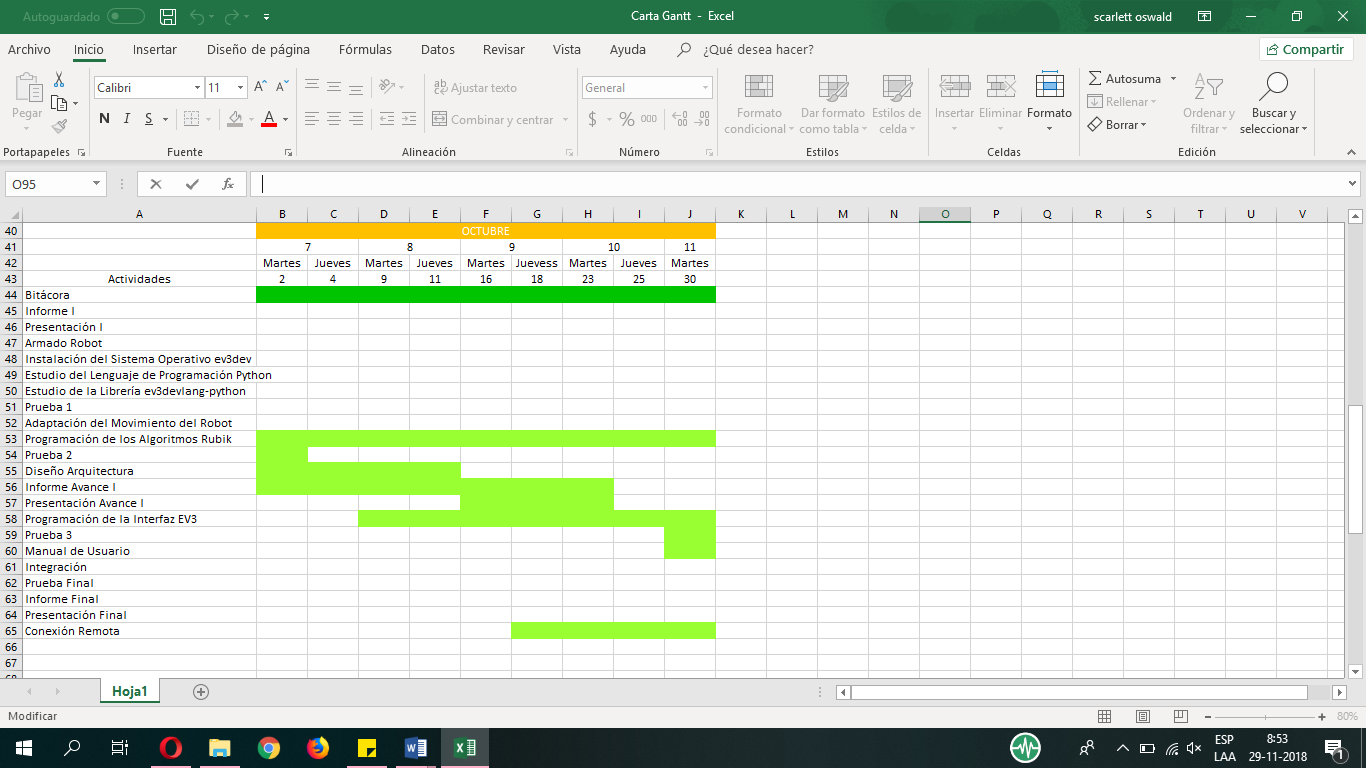
3.Planificacion del Proyecto

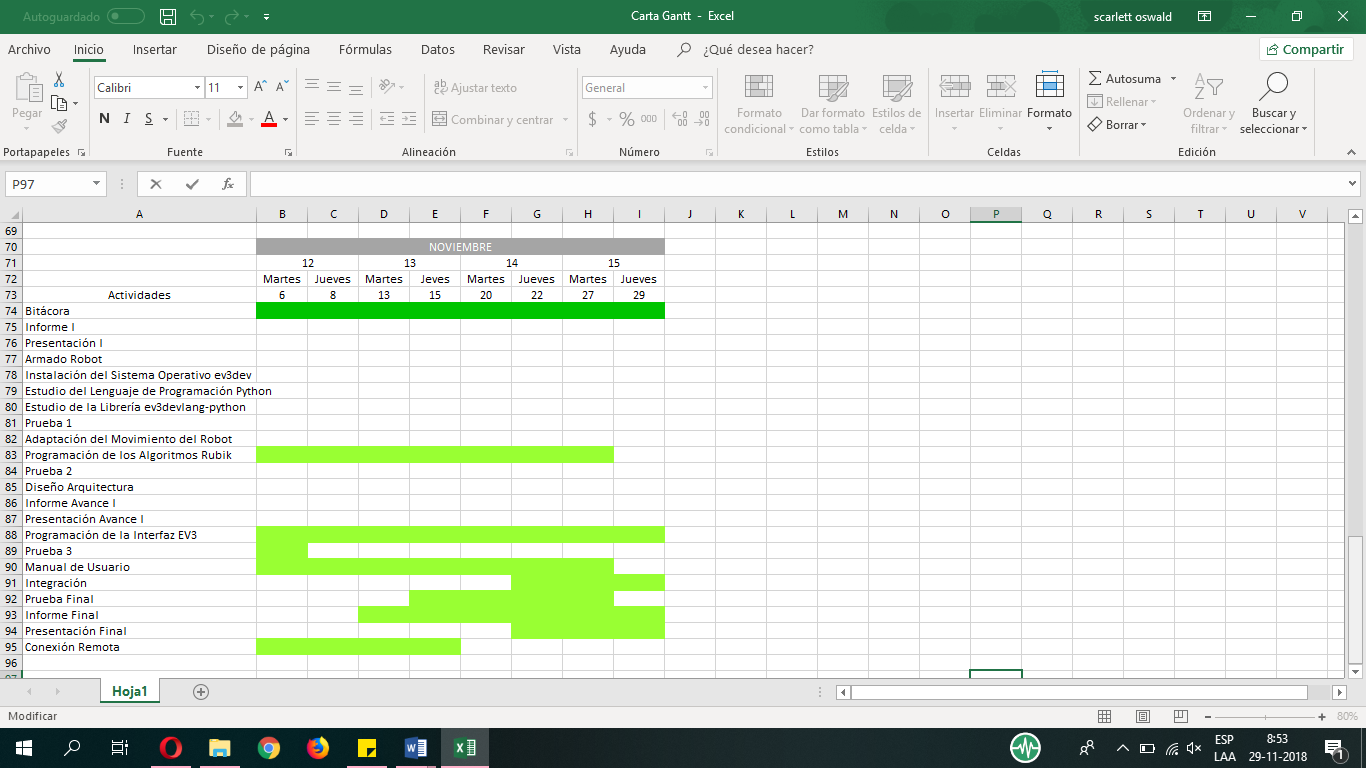
## 3.1. Actividades

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Concretado |
| Programación de la Interfaz EV3 | Programación de los movimientos del robot mediante una aplicación. | Diego Berrios | Concretado |
| Pruebas | Pruebas del funcionamiento del robot con el cubo Rubik. | Diego Berrios  Miguel Rivero | Concretado |
| 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 | Concretado |
| 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 | Concretado |
| Bitácoras | Breve descripción de lo realizado en la semana y lo que se debe realizar la semana siguiente. | Alisson Visa | Concretado |

## 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

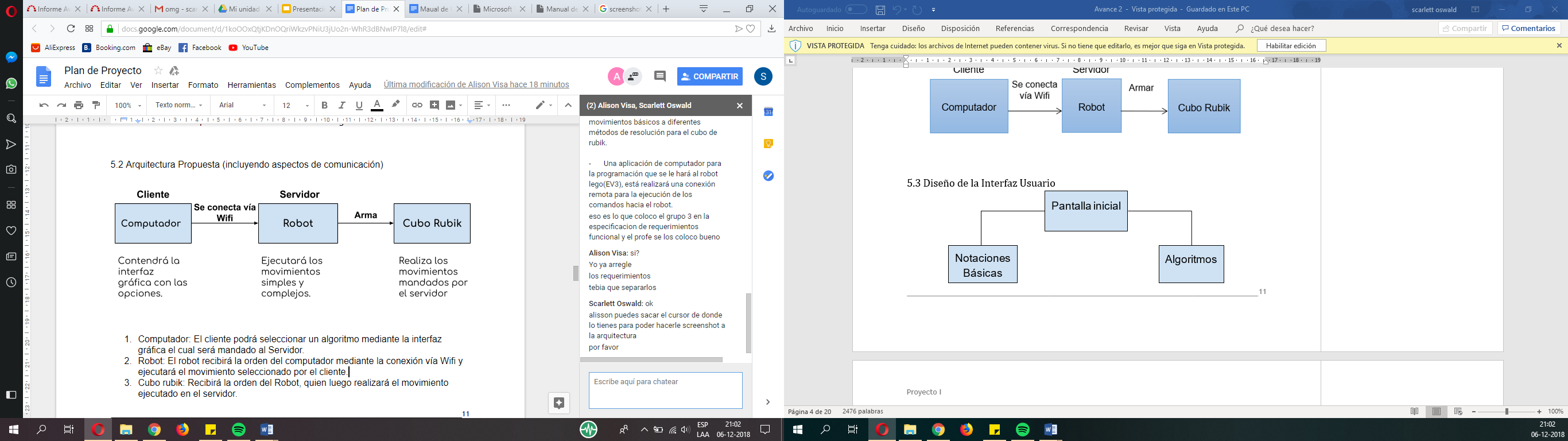
Requerimientos Funcional:

* Se implementará un robot el cual se va a comunicar vía Wifi con el computador, además mediante la conexión el robot podrá realizar el armado del cubo Rubik.
* Se desarrollará una interfaz gráfica adecuada para el entendimiento del cliente.
* La interfaz gráfica contendrá los algoritmos necesarios para resolver el cubo.
* El robot podrá ejecutar más de cinco algoritmos con la ayuda de la persona.

Requerimientos No Funcionales:

* Cada movimiento que ejecutará el robot deberá realizarse en un tiempo 20 segundos aproximado.
* El sistema debe contar con manuales de usuario estructurados adecuadamente.
* El sistema debe poseer un buen diseño de interfaces gráficas.

## 5.2 Arquitectura Propuesta



1. Computador: El cliente podrá seleccionar un algoritmo mediante la interfaz gráfica el cual será mandado al Servidor.
2. Robot: El robot recibirá la orden del computador vía Wifi y ejecutará el movimiento seleccionado por el cliente.
3. Cubo Rubik: Recibirá la orden del Robot, quien luego realizará el movimiento ejecutado en el servidor.

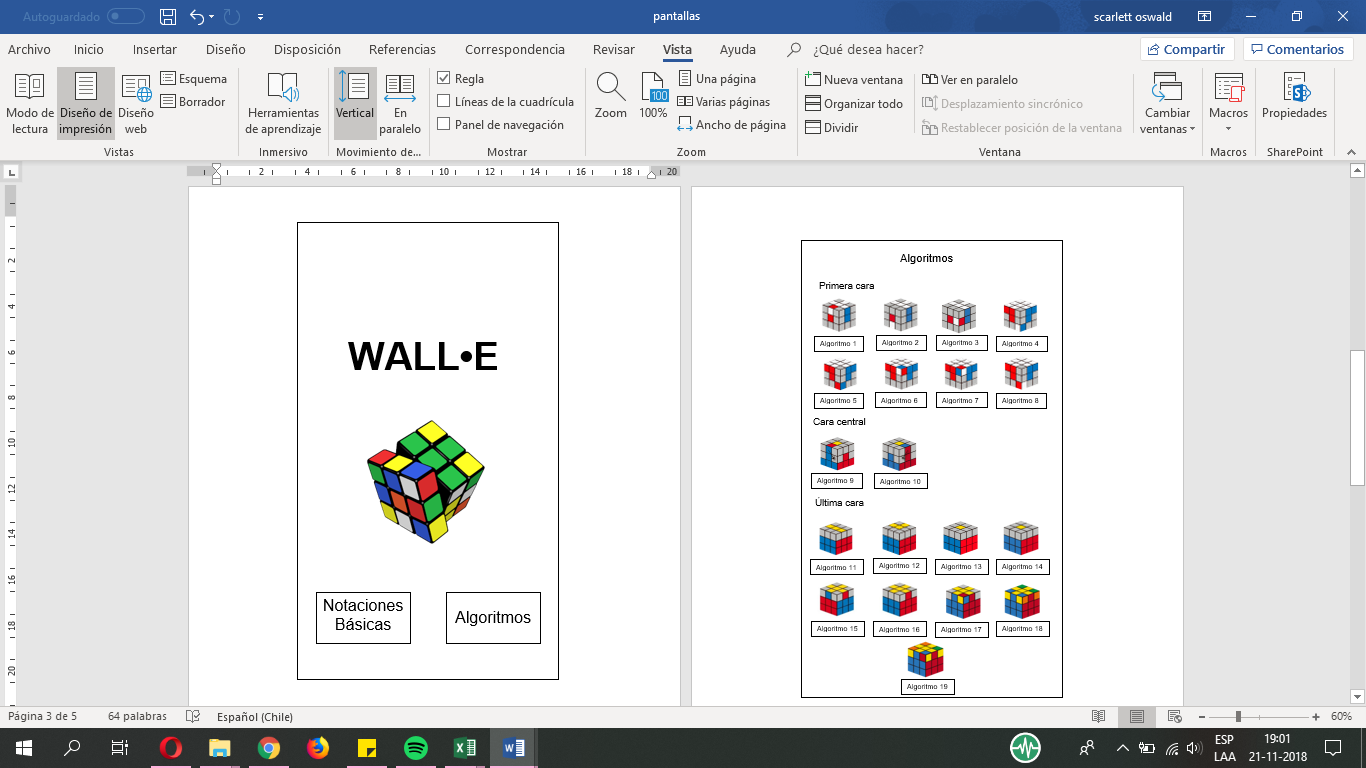
## 5.3 Diseño de la Interfaz Usuario

Pantalla inicial

Notaciones Básicas

Algoritmos

Descripción de las pantallas:



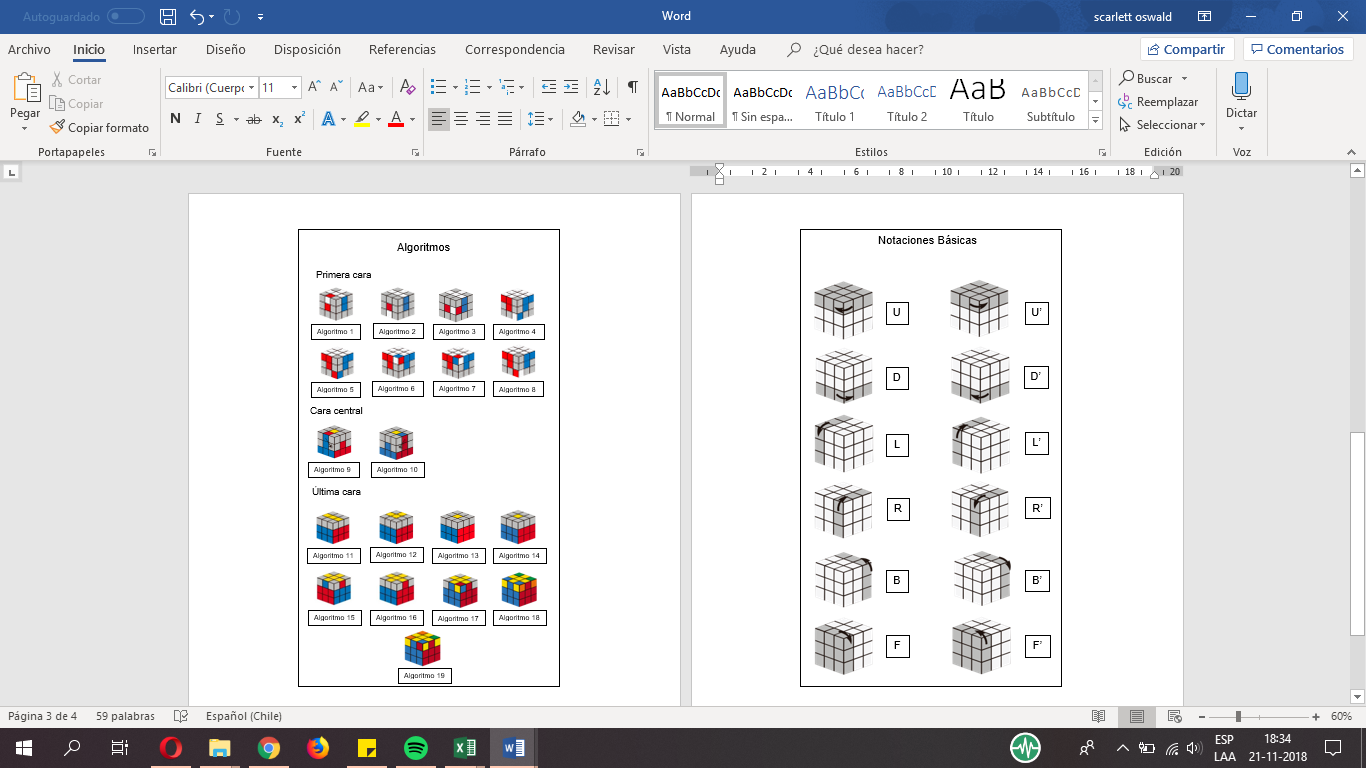
Pantalla inicial

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

* Notaciones Básicas
* Algoritmos

Notaciones Básicas

Algoritmos

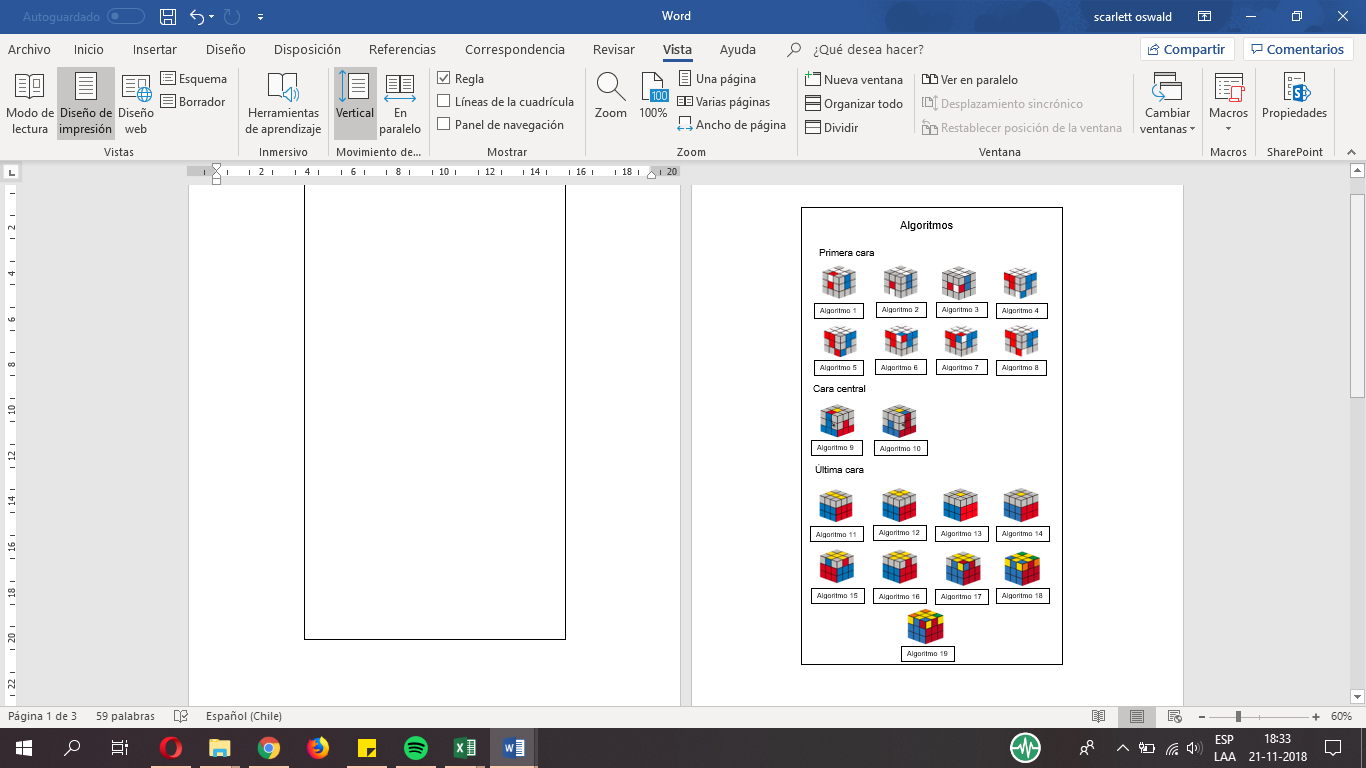


Pantalla Notaciones Básicas

Esta pantalla consta de los movimientos básicos del robot, el cual funcionará 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 la implementación de los programas fue realizada 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” está 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” está 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. |

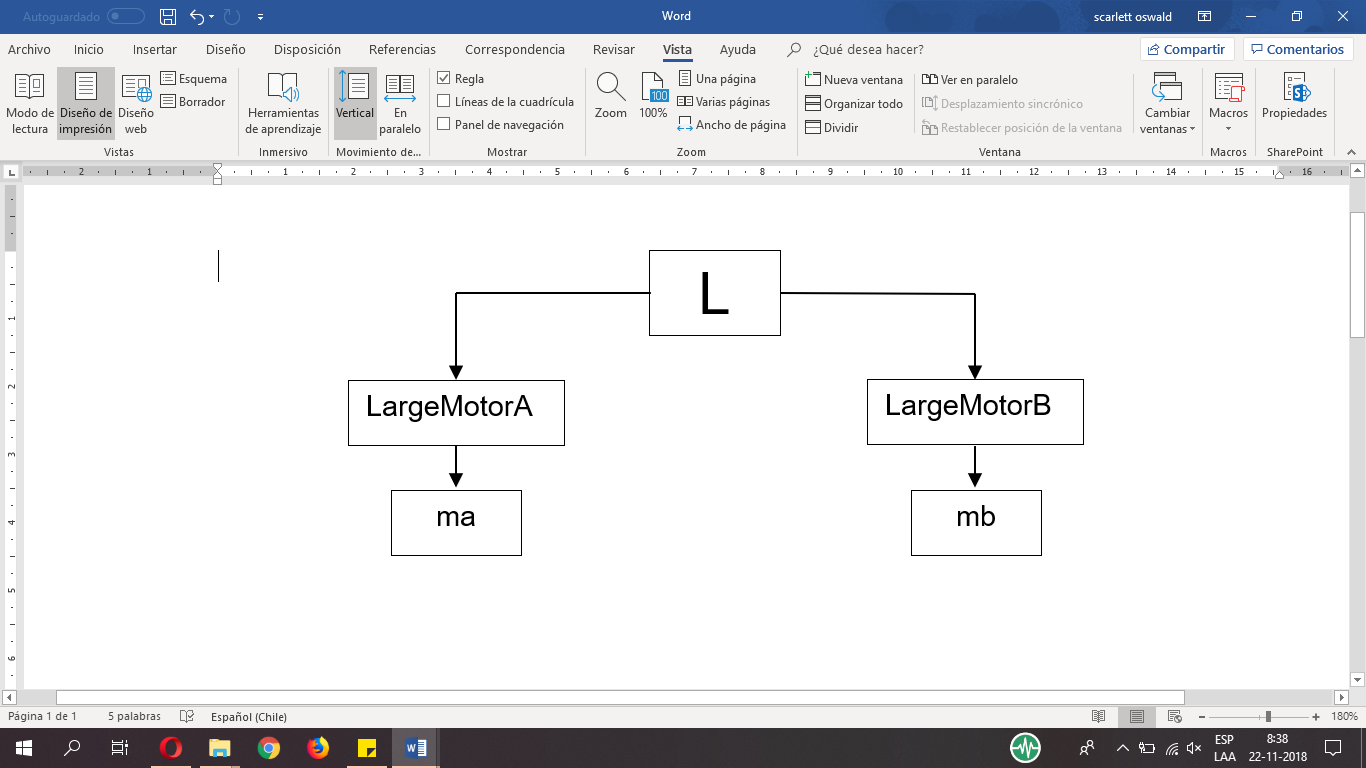
|  |  |
| --- | --- |
| Primera Cara | |
| Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |
| Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. | Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. |
| Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. | Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |
| Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. | Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. |

|  |  |
| --- | --- |
| Cara Central | |
| Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. | Imagen que contiene shoji, edificio  Descripción generada automáticamente  Entrada: ma, mb. |

|  |  |
| --- | --- |
| Ultima Cara | |
| Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. | Imagen que contiene edificio, shoji  Descripción generada automáticamente  Entrada: ma, mb. |
| Entrada: ma, mb. | Entrada: ma, mb. |
| Entrada: ma, mb. | Imagen que contiene artículos de aseo personal  Descripción generada automáticamente  Entrada: ma, mb. |
| Imagen que contiene shoji  Descripción generada automáticamente  Entrada: ma, mb. | 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 se quiere realizar el movimiento básico “L” se realizaría de la siguiente forma:



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

Sexy Move

R

R´

U´

U

7. Pruebas

## 7.1 Descripción de las pruebas realizadas

Al finalizar el armado del robot, la programación de la base, del brazo y los algoritmos se realizaron cinco pruebas para ver el funcionamiento del robot. Las cuales fueron:

* Prueba 1: Se realizo la prueba del funcionamiento del brazo y de la base del robot.
* Prueba 2: Se realizo la prueba de los movimientos básicos del cubo D, L, R, B, U y F.
* Prueba 3: Se realizo la prueba de los movimientos básicos y de las notaciones primas del cubo D, D’, L, L’, R, R´, B, B’, U, U´, F Y F’.
* Prueba 4: Se realizo la combinación de los movimientos de la notación básica y la notación prima del cubo.
* Prueba 5: Se realizo la prueba del armado del robot con las notaciones básicas y los algoritmos implementados.

## 7.2 Resultados de las pruebas

Los resultados que se obtuvieron en la realización de las cinco pruebas fueron las siguientes:

* En la primera prueba se observó que con una velocidad de 1000 las piezas salieron eyectadas.
* En la segunda prueba se observó que al realizar ciertos movimientos básicos la base se descalibra, ya que se realizaron mal los cálculos de los grados y del tiempo de espera para realizar el siguiente movimiento.
* En la tercera prueba se observó que al no tener claro cual era el frente nos complico a la hora de ejecutar los movimientos.
* En la cuarta prueba y en la quinta prueba se observó que al realizar varios algoritmos seguidos el robot se traba por lo que hay que dejar un tiempo entre la realización de cada algoritmo. Además necesita la intervención humana para las veces que el cubo no queda en su posición.

8. Resultados

## 8.1 Estado Final del Proyecto

Imagen que contiene suelo, interior, pequeño, pared

Descripción generada con confianza muy alta

Imagen que contiene interior, suelo, mesa, pared

Descripción generada con confianza muy alta

Imagen que contiene interior, electrónica

Descripción generada automáticamente

## 8.2 Problemas Encontrados

|  |  |
| --- | --- |
| 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 |

## 8.3 Conclusiones

A lo largo de la realización del proyecto, nos hemos encontrado con variados problemas de todos los ámbitos, ya sean prácticos, como la construcción del robot, falta de piezas, problemas con el sistema operativo, lo cual como equipo logramos solucionar el problema satisfactoriamente sin ningún otro percance. Por otro lado con la documentación del proyecto, cabe destacar que no fue un trabajo fácil, ya que hubo altos y bajos como en los otros procesos.

Cabe señalar que adquirimos mucha experiencia a lo largo del Proyecto, ya sea en el ámbito de programación en sí de conocer un lenguaje nuevo el cual trabajamos de forma paralela con una herramienta externa, en este caso el robot.

También tuvimos muchas complicaciones con la conexión del computador hacia el robot, lo cual gracias a los profesores y ayudantes logramos resolver aquel problema.

## 8.4 Trabajo Futuro

Como equipo hemos realizado la etapa final del proyecto, se tiene planeado incluir más algoritmos complejos para resolver el cubo de diferentes maneras. Por otro lado se realizará el mejoramiento del diseño de la interfaz gráfica y además se realizará un breve cambio en el Manual de Usuario.

9. Referencias

Programación NXC - Robots Lego NXT Mindstorms.

<https://blog.kubekings.com/como-hacer-el-cubo-de-rubik/> , como hacer el cubo de rubik

<https://www.lego.com/es-es/mindstorms/support> , Mindstorms ev3

<http://repositori.uvic.cat/bitstream/handle/10854/3799/trealu_a2014_lopez_albert_manipulador_manual_usuari.pdf?sequence=1&isAllowed=y> , Manual de Usuario Manipulador robótico con visión artificial, Alberto López Germán.

Anexos

## Anexo A: Código de los programas implementados

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

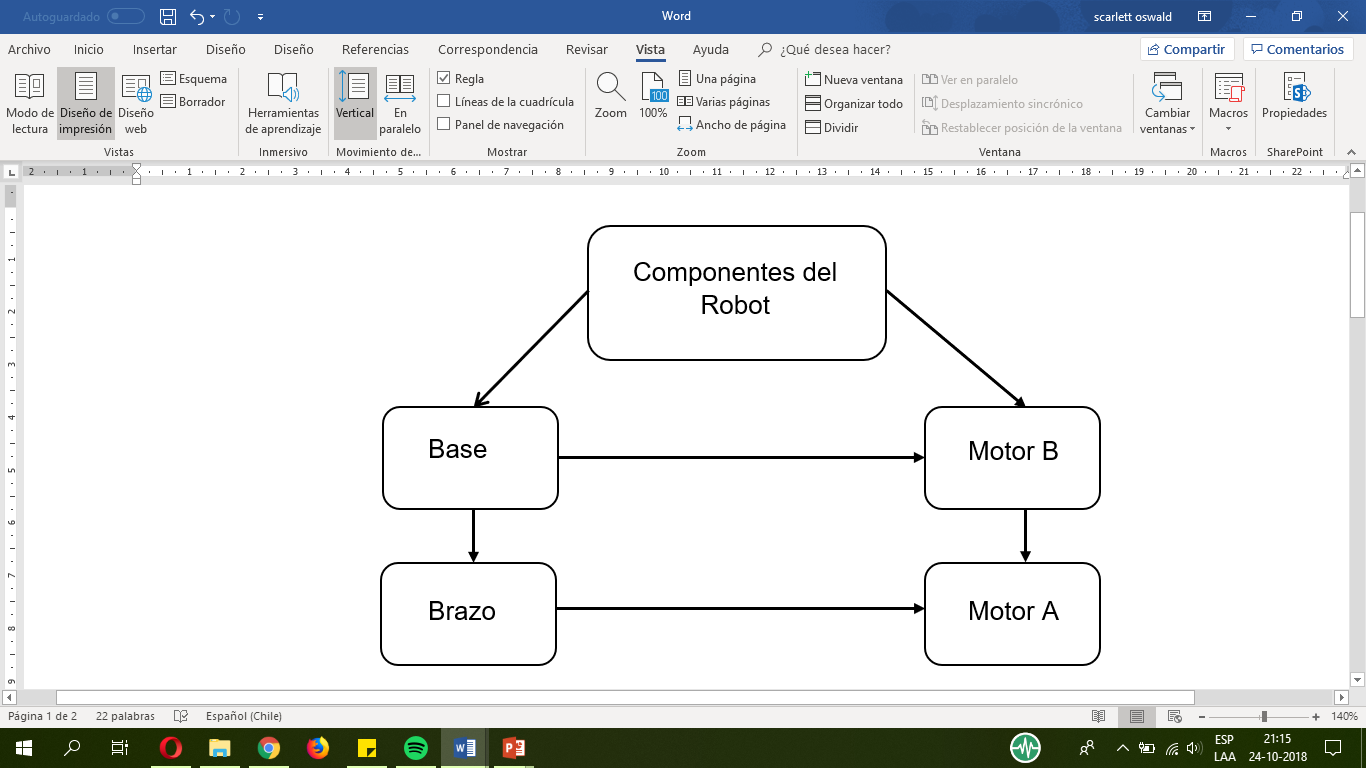
|  |
| --- |
| **Movimientos Básicos del Cubo Rubik** |
| **def    D():**  **En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior del mismo para ejecutar este movimiento.** |
| **def    L():**  **En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, luego el brazo gira el cubo tres veces y 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 la base gira 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 la base gira, 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.** |

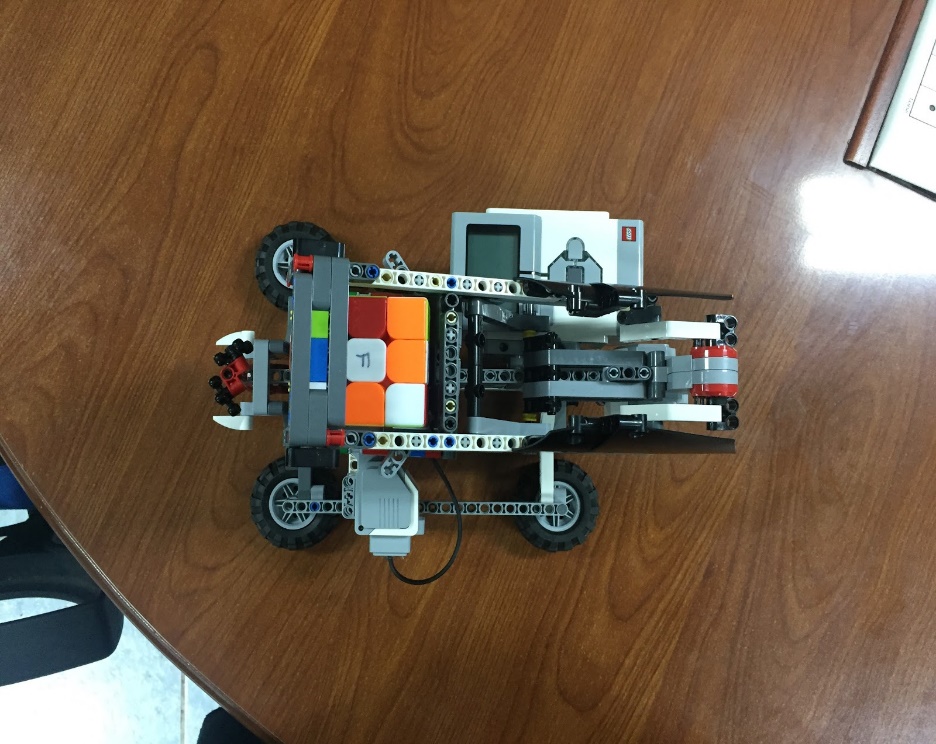
|  |
| --- |
| **Movimientos (Notaciones Primas)** |
| **def    DPrima():**  **En esta función el brazo sostendrá el cubo mientras la base gira la capa inferior, por último el brazo gira siete veces.** |
| **def    LPrima():**  **En esta función el brazo sostendrá el cubo mientras la base gira la capa 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 tres veces, a continuación la base gira y 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 la base gira, 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    F()Prima:**  **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.** |

|  |  |
| --- | --- |
| Algoritmos | |
| def exposed\_line\_Algoritmo1(self):  UPrima()  RPrima()  U()  FPrima() | def exposed\_line\_Algoritmo2(self):  FPrima()  UPrima()  R()  U() |
| def exposed\_line\_Algoritmo3(self):  UPrima()  R()  U() | def exposed\_line\_Algoritmo4(self):  DPrima()  RPrima()  D()  R() |
| def exposed\_line\_Algoritmo5(self):  i = 0  while i < 3:  DPrima()  RPrima()  D()  R()  i += 1 | def exposed\_line\_Algoritmo6(self):  i = 0  while i < 4:  DPrima()  RPrima()  D()  R()  i += 1 |
| def exposed\_line\_Algoritmo6(self):  i = 0  while i < 4:  DPrima()  RPrima()  D()  R()  i += 1 | def exposed\_line\_Algoritmo7(self):  i = 0  while i < 2:  DPrima()  RPrima()  D()  R()  i += 1 |
| def exposed\_line\_Algoritmo8(self):  RPrima()  DPrima()  R() | def exposed\_line\_Algoritmo9(self):  U()  R()  UPrima()  RPrima()  UPrima()  FPrima()  U()  F() |

|  |  |
| --- | --- |
| def exposed\_line\_Algoritmo10(self):  UPrima()  LPrima()  U()  L()  U()  F()  UPrima()  FPrima() | def exposed\_line\_Algoritmo11(self):  F()  R()  U()  RPrima()  UPrima()  FPrima() |
| def exposed\_line\_Algoritmo12(self):  RPrima()  UPrima()  FPrima()  U()  F()  R() | def exposed\_line\_Algoritmo13(self):  RPrima()  U()  U()  R()  U()  RPrima()  U()  R() |
| def exposed\_line\_Algoritmo14(self):  R()  U()  U()  RPrima()  UPrima()  R()  UPrima()  RPrima() | def exposed\_line\_Algoritmo15(self):  LPrima()  U()  U()  L()  B()  UPrima()  LPrima()  UPrima()  L()  U()  B() |
| def exposed\_line\_Algoritmo16(self):  LPrima()  B()  LPrima()  FPrima()  FPrima()  L()  BPrima()  LPrima()  FPrima()  FPrima()  LPrima()  LPrima() | def exposed\_line\_Algoritmo17(self):  RPrima()  U()  RPrima()  UPrima()  RPrima()  UPrima()  RPrima()  U()  R()  U()  R()  R() |
| def exposed\_line\_Algoritmo18(self):  R()  R()  UPrima()  RPrima()  UPrima()    R()  U()  R()  U()  R()  UPrima()  R() |  |

## Anexo B: Robot





Motor B



Motor A

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.