**UNIVERSIDAD DE TARAPACÁ**



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



Área de Ingeniería en Computación e Informática



**Plan de proyecto
Mindcuber G3**

 **Autor(es): Huber Ticona**

 **Byron Yavi**

 **Gabriel Martínez**

 **Rodrigo Gonzales**

 **Jorge Fernández**

 **Asignatura: Proyecto I**

 **Profesor(es):** DIEGO ARACENA PIZARRO

 RICARDO VALDIVIA PINTO

ARICA, 9 De septiembre 2017

 **Historial de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 09/09/2017 | 0.5 | Primera versión del formato. |  Byron Yavi  Hubert Ticona  |
|  10/09/2017 | 0.6 | Revisión y corrección de loredactado en el informe. | Gabriel Martínez |
|   21/10/2017 |  0.8 | Se agregaron nuevos entregables además de corregir lo anteriormente implementado. | Gabriel MartínezByron Yavi |
|  |  |  |  |

**1. Introducción**

El propósito es elaborar un proyecto tecnológico trabajando en grupo, el cual está orientado a desarrollar un mecanismo automático para la creación de patrones sobre un cubo Rubik. Durante el semestre el equipo de trabajo tendrá que organizarse de forma que cumpla los tiempos establecidos.

En el presente informe se especifica aspectos como: especificación del problema, definición de las actividades, estructura organizacional etc.

**1.1 Alcance de Proyecto**

El proyecto consiste en diseñar y programar un robot capaz de armar patrones sobre un cubo rubik, basándose en el proyecto Mindcuber.

Propósito: El producto final será un robot programado con la finalidad de crear patrones sobre un cubo rubik.

Alcance: El robot será capaz de realizar patrones en un cubo rubik, se espera que lo haga de una manera eficiente.

Objetivo: diseñar un robot y programarlo para realizar patrones en un cubo rubik en el menor tiempo posible, en base a sus algoritmos.

Suposiciones y restricciones: El proyecto debe seguir estrictamente los pasos indicados por el proyecto Mindcuber.

**1.2 Entregables del proyecto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificación Entregable** | **Descripción Entregable** | **Fecha de entrega** | **Lugar de entrega** | **Condiciones satisfacción** |
| Terminar el robot | El grupo debe haber terminado la construcción del robot. | 09-09-2017 | La sesión correspondiente a la fecha. Y notificación en la plataforma de administración. | El robot debe estar terminado de acuerdo al manual de armado. |
| Programación del robot 0.1 | La programación del debe cumplir con los estándares de avance según la fecha de entrega. | 01-10-2017 | La sesión correspondiente y las notificaciones serán mediante la plataforma de administración. | La programación debe estar avanzada y debe hacer algo útil. |
| Programación del robot 0.2 | El robot debe estar programado en su totalidad y debe ser capaz de armar un cubo rubik | 20-10-2017 | La entrega es de forma presencial y además debe ser notificada en la plataforma de administración | El robot debe ser capaz de armar un cubo rubik |
| Informes y presentaciones | Informe sobre todo lo que trata el proyecto | 19-11-2017 | Entrega en forma presencial  | Aprobación del proyecto. |

**2. REFERENCIAS**

*Mindcuber:* Se ocupará el manual de instrucciones del proyecto Mindcuber

*Programación avanzada*: Se hará uso de los conocimientos adquiridos para la programación del robot.

*YouTube:* El grupo ocupará la plataforma de YouTube para buscar tutoriales.

*Lego Manual*: En primera instancia se ocupa el manual y proporcionado por Lego que ayuda a la orientación y la funcionalidad de cada parte del robot.

**3. Organización del proyecto**

En una descripción general, el rol a seguir por cada miembro del grupo ya está descrita mediante la implementación de un plan de proyecto el cual se establece de la siguiente forma: En caso de existir una tarea esta se dividirá su trabajo dividiendo el grupo en 2 subgrupos y repartiendo las tareas de forma equitativa. El esquema de comunicación y la organización del grupo se plantean de la siguiente forma.

Líder de grupo

(Coordinador y administrador)

Integrante Encargado

50% (Tarea x)

Integrante Encargado

50% (Tarea x)

Integrante de soporte

Integrante de soporte

**3.1-Detalles y acuerdos de organización:**

Cabe destacar que la organización del grupo estará en función de la complejidad de la tarea, y en base a esto se toma la decisión de distribuir el trabajo en subgrupos, el máximo de subgrupos es de 2, en los cuales se asignará un encargado para cada sub tarea siendo el Líder del grupo mediador y en cargado de administrar y mantener la coordinación del avance entre los sub grupos.

**3.2- Roles y responsabilidades**

Programador: será el encargado de implementar los algoritmos dados para hacer que el robot cumpla con lo previsto.

Coordinador: deberá velar por el cumplimiento de los tiempos establecidos dentro de los entregables del proyecto.

Secretario: será el encargado de gestionar las diligencias establecidas por las sesiones bitácoras, avances etc.

Ayudante: estará a cargo de una tarea “x” la cual estará encabezada por un jefe de subgrupo el cual dirá las directrices para que el ayudante cumpla con su tarea.

**3.3- Mecanismos de Comunicación**

Para mantener la comunicación entre los miembros del grupo, se ha optado por crear un grupo en Facebook y WhatsApp y el intercambio de archivos se hará mediante la plataforma de gestión de grupos.

**4-Planificacion de los procesos de gestión y costeo**

Planificación de estimaciones:

|  |  |
| --- | --- |
|  PROCESO  |  COSTE ESTIMADO |
| Compra de cubo rubik |  10000$ |
| Compra de materiales faltantes para la construcción del robot | 20000$ |
| Compra de materiales complementarios para el proyecto | 8000$ |

**4.1-Planificación de recursos humanos**

Se acuerdan los siguientes roles: 1 programador, 1 Jefe de proyecto,

 2 ayudante y 1 diseñador.

**4.2-Actividades del proyecto**

El proyecto consta de seguir los pasos indicados en la página del proyecto Mindcuber, siendo el reto establecer una organización entre los miembros del grupo, de forma tal, que se logre cumplir las tareas estipuladas dentro de los tiempos establecidos.

La ejecución del proyecto comienza con el plan de proyecto, una vez puesto en marcha es necesario gestionar de forma adecuada las tareas indicadas previamente, además de cumplir las fechas de entrega acordada.

**Asignación de tiempo:**

|  |  |
| --- | --- |
|  **trabajo** |  **Tiempo(semanas)** |
|  Planificación de proyecto |  **2-4** |
|  Ejecución del proyecto |  **11-12**  |
| Cierre del proyecto |  **1** |
|  Presentación final |  **1**  |

**4.3-Planificación de Riesgos**

**Nivel de impacto:**

**1=Catastrófico, 2=critico, 3=Marginal, 4=insignificante.**

|  |  |  |  |
| --- | --- | --- | --- |
| Riegos | Probabilidad Ocurrencia | Nivel de impacto | Acción remedial |
| Se enferma un integrante |  **60%** |  **2** | Se reasignan las tareas entre los integrantes que no estén enfermos |
| Integrante falta a una sesión |  **40%** |  **3** | Debe justificar la falta, además deberá ponerse al día con lo avanzado. |
| Descoordinación entre sub Grupos  |  **40%** |  **1** | El subgrupo que este avanzado deberá ayudar al subgrupo atrasado. |
| Falta presupuesto |  **30%** |  **1** | Se pedirá una cuota entre los miembros del grupo |
| Un integrante abandona el grupo |  **20%** |  **2** | Se reasignan tareas y cargos para solventar la perdida |
| Un integrante Llega tarde a la sesión  |  **20%** |  **4** | Siguen las actividades con normalidad, se espera 10 min si no llega otro toma su rol hasta el fin de sesión. |
| Se muere un integrante |  **3%** |  **1** | Se reasignan las tareas y los cargos |

**5.1-Planificación y procesos Técnicos**

En primera instancia, se pretende formular el proyecto, luego lo llevaremos a cabo aplicando lo solicitado según el proyecto Mindcuber

**5.2-Herramientas técnicas**

En el transcurso del proyecto contaremos con plataformas de desarrollo tales como: Lenguaje de programación NXC, Lenguaje de programación Python, Microsoft office, PowerPoint, entre otros.

Algunas de las técnicas que se usarán serán: Divide para conquistar (Organización de tareas), Modularidad, etc.

**5.3-Planificación y aceptación del producto**

Para que una tarea “x” sea considerada como terminada, esta deberá de pasar una evaluación por cada uno de los integrantes del equipo de trabajo.

Se considerarán los siguientes puntos:

1- La tarea cumple con lo estipulado.

2- El producto final es de calidad.

3- La tarea entregada cumple con los estándares del grupo.

Cada miembro dará una nota del 1 al 7 y si el promedio obtenido a partir de los 5 miembros del grupo es >=5.0 se considerará como tarea finalizada, a continuación, una plantilla.

Encuesta de satisfacción.

|  |  |
| --- | --- |
|  PREGUNTA |  Nota |
| ¿La tarea cumple con lo estipulado? |  5 |
| ¿El producto final es de calidad? |  5  |
| ¿Cumple con los estándares? |  6 |

**6-Modelo pseudocódigo general**

Movimiento que hará girar el cubo de manera frontal.

Movimiento de brazo.

La base hará rotar el cubo hacia la izquierda o derecha según se le indique

Movimiento de base.

Agregar patrones.

Se agregaran los movimientos del patrón y el número de veces que se debe realizar para luego ser llamado.

Al entregarle los datos del patrón,

la función realizara los movimientos necesarios para la creación del patrón.

Función de movimiento

según patrón.

**6.1-Diseño de interface de Usuario**

En el proceso del proyecto surge la necesidad de generar una comunicación entre el usuario y el producto, de esta forma se establecieron métodos de vinculación entre usuario y máquina, es decir, el vínculo entre quien hará uso del producto y el producto, de esta manera se establecen la Comunicación directa y comunicación remota.

*-*Método de comunicación directa:

El usuario puede interactuar con el robot de forma directa, es decir podrá establecer los patrones que este desee a través del “ladrillo”, seleccionando el patrón que desee.

*-*Método de comunicación remota:

El usuario posee una aplicación que permite establecer una conexión blurtooth con el robot para poder enviar datos recibidos por el mismo para que este haga un patrón u otro.

**6.2- Especificación de Requerimientos:**

***-N°.1 Propósito de requerimientos:***

El proyecto necesita establecer algunas normas, y objetivos para determinar de forma concisa la culminación y/o la necesidad de postergar la finalización de una etapa de desarrollo del proyecto.

***-N°.2 Alcance y personal involucrado:***

A continuación se especificaran las tareas, los procesos y sus respectivos encargados con los respectivos requerimientos para la finalización de esta tarea, en caso de no cumplirse se regresa al punto de partida para analizar el posible problema, véase diagrama 1.1

No cumple requisitos, se detectan problemas

Sub tarea

Cumple requisitos

Siguiente tarea

|  |  |
| --- | --- |
|  Análisis y planificación Del proyecto  |   |
|  Análisis de ejecutables | Se determinan cuáles serán las tareas prioritarias y sub tareas a ejecutar a futuro, encargado Jorge Fernández. |
| Requerimiento: Aprobación del grupo. |
| Organización de los  trabajos | Establecer cuáles serán las tareas de cada uno de los integrantes, Byron yavi. |
| Requerimiento: Predisposición de los integrantes |
|  Proceso de avance y  mantención | La gestión y organización del proyecto se hace mediante la plataforma facilitada en la asignatura proyecto I, Encargado Jorge Fernández |
| Requerimiento: Mantención constante hasta la finalización del proyecto. |

|  |  |  |
| --- | --- | --- |
|  Ejecución del proyecto  |  Descripción y Encargados |  Requerimiento |
| Gestión de Tareas  | Supervisión de las tareas a nivel general, programación, diseño etc. encargado: Byron yavi | Es necesario que las tareas estén ejecutándose dentro de los plazos establecidos. |
| Actualización de planificación  | En caso de un error o problema se harán cambios de planes mininos, encargado: Byron yavi | Es necesario el avance constante dentro de las tareas en el tiempo. |
| Refinamiento y optimización | Cada encargado es responsable de realizar en su labor el mejor resultado posible, encargado: Encargados de sub tareas. | Es necesario tener el producto final terminado para realizar mejoras en el mismo. |

**7-Modelo de diseño detalles pseudocódigo**

**7.1-Diagrama de flujo**

Encender Robot

No

Si

Restaurar cubo

Terminar programa

¿Seguir operando?

Realizar movimientos

Escoger patrón

Asignar movimientos

Conectarse vía

bluetooth

**1-Coneccion bluetooth:** Mediante una aplicación móvil creada previamente, se enlazara al brick del robot para iniciar el programa.

**2-Escoger patrón:** Ya estando conectado al brick, la aplicación móvil ofrecerá la opción de escoger entre distintos patrones que se podrán formar en el cubo rubik.

**3-Asignar movimientos:** Se escaneara la opción escogida en la App y se asignaran los movimientos designados para realizar el patrón escogido.

**4-Realizar movimientos:** El robot se encargara de realizar lo movimientos necesarios para armar el patrón escogido previamente.

**5- Restaurar cubo:** El cubo rubik volverá a su forma inicial para poder realizar nuevos patrones.

**6- Terminar programa:** Se desconectara el enlace bluetooth y se apagara el robot.

**8-**

**8.1-Plan de integración**

A continuación, se expondrán los procesos a seguir que competen al plan de integración del presente proyecto.

1) Planificación del proyecto:

Tiene como finalidad establecer las directrices y además esclarecer el objetivo principal que tiene cada proceso dentro del proyecto.

Dentro del ámbito técnico e implementador, cabe señalar las actividades respectivas en este proceso queda expuesto en el siguiente diagrama.

 Análisis del objetivo

Análisis de implementación

Planificación de entregables

A continuación, una descripción de los procesos mencionados.

|  |  |
| --- | --- |
| Análisis del objetivo | El equipo en general debate cual debe ser el objetivo final del proyecto. |
| Análisis de implementación | El equipo discute los métodos para llevar a cabo las tareas de prioridad, ejemplo: armado de robot. |
| Planificación de Entregables | Una vez dirimida la idea, se oficializa el objetivo del proyecto.  |

2) Organización Estructural:

El proyecto ve necesario el uso de un modelo organizacional que se adapta a las necesidades del proyecto en función del factor, tiempo y complejidad de una tarea específica, es por ello que se decide estudiar la forma de organizar y repartir bien el trabajo conforme la situación del equipo de trabajo.

Dentro de los aspectos fundamentales en el ámbito técnico e implementador, se puede destacar el siguiente esquema que sintetiza esto.

 Organización Estructural

Tareas y desarrollo

 Análisis y aprobación

Siguiente tarea

Equipo de trabajo

(Recursos humanos)

3) Ejecución e implementación de tareas

Las ejecuciones de las tareas agenda das se llevan a cabo en función del orden de prioridad según estas estén establecidas (véase entregables y encargados).

Aa continuación, se expondrán el orden de implementación de las tareas más importantes y agenda das, ya llevadas a cabo y pendientes, cabe destacar que el trabajo entre grupo técnico y programador y el grupo de Analistas es realizado de forma Paralela.

 Grupo Técnico y programador Grupo de Analistas

Investigación de patrones para el proyecto

 Armado del robot

 Análisis del código para el armado de

 patrones

Refinamiento y optimización del código

Diseño gráfico e interface de usuario de la aplicación de conexión remota

Corrección de errores

Pruebas de conexión remota

Administración y gestión de informes finales

Optimización

Corrección de Errores

Pruebas de armado de patrón

Prueba de conexión

Producto Final

4) Procesos de Optimización y refinamiento

Los procesos de optimización y refinamiento del proyecto comenzarán después de que se den por cumplidas las siguientes condiciones.

1) El producto debe darse por terminado en primera instancia, es decir el producto final debe funcionar y hacer lo que se le pide.

2) La Optimización se hace si y solo si es posible realizar una mejora perceptible.

3) Debe existir un tiempo prudente para la realización de la optimización

A continuación, una representación gráfica del modelo de creación de Optimización y refinamiento.

Producto Terminado

Estudio de posibilidad de Mejora

Cumple las condiciones

No cumple las condiciones

Es optimizarle

No es necesario o no es posible la optimización

**8.2-Descripción de arquitectura**

Se encarga de verificar si el la base que sostiene al cubo debe rotar hacia la izquierda o derecha según se le indique.

void rotarcubo(){

 rotar(motor A)

 esperar(200)

 apagar(motor A)}

Desde la posición inicial del brazo este debe avanzar y agarrar el cubo para evitar que este se mueva.

void agarrarcubo(){

 atras(motor C)

 esperar(400)

 apagar(motor C)}

Luego de realizar las acciones se debe soltar el cubo y regresar el brazo a su posición inicial.

void soltarcubo(){

 adelante(motor C)

 esperar(550)

 apagar(motor C)}

Recibe los datos del patrón escogido para luego realizar los movimientos asignados al patrón.

void movimientos(patron,movimiento,pasos){

 while(pasos){

 escogercara

 realizarmovimientos

 devolverainicio }}

Guardar los datos del patrón: cara a mover, cantidad de movimientos y el número de pasos para realizar el patron.

void patron(){

string patron

int movimiento

int pasos}

Main principal, el switch en él debe recibir el número del patrón que debe realizar el programa.

task main(){

 switch(){

 escogerpatron() }}

**8.3-Modelo de implementación**

-Visitar anexo n°1 para ver el código del programa.

-Diseño de la App de conexión remota

 

**8.4-Módulos implementados**

1-Giro rotatorio de la base que sostiene el cubo

void giro\_izquierda(){

 RotateMotor(OUT\_A, 80, -208);

 Wait(200);

 Off(OUT\_C);

}

void giro\_derecha(){

 RotateMotor(OUT\_A, 80 , 209);

 Wait(200);

 Off(OUT\_C);

}

2-Movimiento de brazo

void voltear\_cubo(){

 OnRev(OUT\_C,70);

 Wait(270);

 Off(OUT\_C);

 Wait(200);

 OnFwd(OUT\_C,70);

 Wait(280);

 Off(OUT\_C);

 Wait(200);

}

void soltar\_cubo(){

 OnFwd(OUT\_C,50);

 Wait(550);

 Off(OUT\_C);

 Wait(200);

}

void agarrar\_cubo(){

 OnRev(OUT\_C,60);

 Wait(400);

 Off(OUT\_C);

 Wait(100);

}

**8.5 Historial de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
|  **Fecha** |  **Versión** |  **Descripción** |  **Autor(es)** |
| 26/09/2017 | 0.5 | Primer código aceptado. | Gabriel Martínez |
| 5/10/2017 | 0.6 | Se agregaron dos patrones y se optimizo el código. | Rodrigo Gonzales |
| 12/10/2017 | 0.8 | Modificación de código respecto a la potencia de motores. | Rodrigo GonzalesGabriel Martínez |
|  |  |  |  |
|  |  |  |  |

**9-Avances de acuerdo a la Gantt:**

Las tareas asignadas en la Gantt están yendo según lo planeado, teniendo en cuenta que hay pequeños retrasos causados por fallos en la estructura del robot y organización del grupo pero estos no influyen en la fecha de entrega de las tareas propuesta en la Gantt.

Los elementos que están en su breve término es todo lo relacionado con la programación de movimiento del robot además de la implementación de patrones que este puede realizar.

Las tareas relacionadas con los informes y presentaciones futuras están yendo según lo asignado y no ha habido problemas que afecten a su fecha de término.

**Problemas encontrados:**

1- Inestabilidad del giro del cubo: Fallo en el comportamiento de la base giratoria que sostiene el cubo, esta no siempre entrega un resultado estable teniendo un pequeño margen de error que influye en el correcto desempeño del robot, además de influir la fecha de término.

-Solución: Se modificara la potencia con la que funciona la base y se realizara alguna modificación a la misma, para reducir el error.

2- Traspaso de información: Se percató que integrantes del equipo no tiene conocimiento de algunas áreas trabajadas por los demás, es decir, no se está trasmitiendo toda la correctamente información entre los miembros del equipo.

-Solución: Se acordó una reunión para trasmitir y archivar lo realizado por todos los integrantes del equipo y así poder mantenerse informado.

**Conclusión:**

Trabajar con Mindcuber no es complicado, su fácil manejo de estructura y la sencilla comprensión de su lenguaje de programación invitan a cualquiera a elaborar algún proyecto con él.

Hacer que Mindcuber realice patrones en vez de armar el cubo es tarea sencilla no requiere de un vasto conocimiento, solo comprender su funcionamiento y darle las acciones necesarias, siendo lo más laborioso crear una aplicación para conectarse al robot y enviar ordenes mediante conexión remota.

**Trabajo a futuro:**

-Se tiene planeado modificar el código del programa para lograr obtener una versión mejorada, optimiza y ligera del mismo.

-Se mejorara la aplicación de conexión remota con el robot, haciendo que esta sea aún más intuitiva y pueda ser usada fácilmente por cualquier usuario sin importar su condición. Además de mejorar el diseño de la misma logrando ser más agradable para todo usuarios.

-Se realizara una pequeña modificación con los soportes del robot para mantenerlo más estable y evitar posibles fallos relacionados con la inclinación del robot.

**10-Anexo**

**N°1 Código actual del programa**

|  |  |
| --- | --- |
| **int i = 0;****void giro\_izquierda(){** **//Giro de 90° izquierda** **RotateMotor(OUT\_A, 80, -208);** **Wait(200);** **Off(OUT\_C);****}****void giro\_derecha(){** **//Giro de 90° derecha** **RotateMotor(OUT\_A, 80 , 209);** **Wait(200);** **Off(OUT\_C);****}****void agarrar\_cubo(){** **OnRev(OUT\_C,60);** **Wait(400);** **Off(OUT\_C);** **Wait(100);****}****void soltar\_cubo(){** **OnFwd(OUT\_C,50);** **Wait(550);** **Off(OUT\_C);** **Wait(200);****}****void voltear\_cubo(){** **//atras** **OnRev(OUT\_C,70);** **Wait(270);** **Off(OUT\_C);** **Wait(200);** **//adelante** **OnFwd(OUT\_C,70);** **Wait(280);** **Off(OUT\_C);** **Wait(200);****}****void agarrar\_voltear\_cubo(){** **agarrar\_cubo();** **voltear\_cubo();****}****void girar\_de(int mv[]){** **repeat(mv[i]){** **giro\_derecha();** **}** **soltar\_cubo();****}****void girar\_iz(int mv[]){** **repeat(mv[i]\*-1){** **giro\_izquierda();** **}** **soltar\_cubo();****}****void movimientos(string c[], int m[], int p){** **while(i <= p){** **//-----Girar cara ARRIBA-----** **if(c[i] == "Ar"){//Arriba** **agarrar\_voltear\_cubo();** **voltear\_cubo();** **if(m[i] > 0){//m+ giro derecha** **girar\_de(m);** **}** **else{//m- giro izquierda** **girar\_iz(m);** **}** **agarrar\_voltear\_cubo();** **voltear\_cubo();** **soltar\_cubo();** **}//Fin Arriba** **//--------------------------** **//-----Girar cara ABAJO-----** **else if(c[i] == "Ab"){** **agarrar\_cubo();** **if(m[i] > 0){//m+ giro derecha** **girar\_de(m);** **}** **else{//m- giro izquierda** **girar\_iz(m);** **}** **}** | **//--------------------------** **//---Girar cara IZQUIERDA---** **else if(c[i] == "Iz"){** **giro\_derecha();** **agarrar\_voltear\_cubo();** **if(m[i] > 0){//m+ giro derecha** **girar\_de(m);** **}** **else{//m- giro izquierda** **girar\_iz(m);** **}** **agarrar\_cubo();** **repeat(3){** **voltear\_cubo();** **}** **soltar\_cubo();** **giro\_izquierda();** **}** **//--------------------------** **//----Girar cara DERECHA----** **else if(c[i] == "De"){//Derecha** **giro\_izquierda();** **agarrar\_voltear\_cubo();** **if(m[i] > 0){//m+ giro derecha** **girar\_de(m);** **}** **else{//m- giro izquierda** **girar\_iz(m);** **}** **agarrar\_cubo();** **repeat(3){** **voltear\_cubo();** **}** **soltar\_cubo();** **giro\_derecha();** **}** **//--------------------------** **//----Girar cara FRENTE-----** **else if(c[i] == "Fr"){//Frente** **agarrar\_voltear\_cubo();** **if(m[i] > 0){//m+ giro derecha** **girar\_de(m);** **}** **else{//m- giro izquierda** **girar\_iz(m);** **}** **agarrar\_cubo();** **repeat(3){** **voltear\_cubo();** **}** **soltar\_cubo();** **}//Fin Frente** **//--------------------------** **//-----Girar cara ATRAS-----** **else{//(c[i] == "At") Atras** **agarrar\_cubo();** **repeat(3){** **voltear\_cubo();** **}** **if(m[i] > 0){//m+ giro derecha** **girar\_de(m);** **}** **else{//m- giro izquierda** **girar\_iz(m);** **}** **agarrar\_voltear\_cubo();** **soltar\_cubo();** **}//Fin Atras** **//--------------------------** **i++;** **}//Fin while** **i = 0;****}****void patron\_6T(){** **string cara\_6T[] = {"Iz","Fr","Ar","Iz","De","Ab","At","Iz","De"};** **int mov\_6T[] = {2,2,2,-1,1,2,2,1,1};** **int pasos = 9;** **movimientos(cara\_6T,mov\_6T,pasos);****}****void patron\_ojos(){** **string cara\_ojos[] = {"Iz","At","De","Ar","At","Iz","De","Fr","Iz","De","Ab","De","At","Iz"};** **int mov\_ojos[] = {1,1,1,1,1,2,2,-1,2,2,-1,-1,-1,-1};** **int pasos = 14;** **movimientos(cara\_ojos,mov\_ojos,pasos);****}****void patron\_cuboxcubo(){** **string cara\_cuboxcubo[] = {"Iz","At","Iz","Ar","Fr","Ar","Iz","At","Ar","At","De","Ab","De","At","Ar"};** **int mov\_cuboxcubo[] = {1,1,1,-1,1,1,2,2,-1,-1,1,-1,-1,2,1};** **int pasos = 15;** **movimientos(cara\_cuboxcubo,mov\_cuboxcubo,pasos);****}****task main(){** **patron\_ojos();****}** |