**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 lo  redactado en el informe. | Gabriel Martínez |
| 21/10/2017 | 0.8 | Se agregaron nuevos entregables además de corregir lo anteriormente implementado. | Gabriel Martínez  Byron Yavi |
| 16/11/2017 | 1.0 | Modificación de requerimientos conexión remota | Byron Yavi  Jorge Fernández |

**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. |
| Patrones Inversos | Se agregará una opción en la conexión remota para rearmar el cubo una vez ejecutado un patrón. | 20-11-2017 | Jornada de clases respectiva. | Funciona sin fallas. |

**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.

**8.1-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 Gonzales  Gabriel 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.

***9-Plan de Integración***

A continuación, se expondrán los pasos realizados para le ejecución del programa de ejecución del proyecto.

Gráficamente El plan de integración consta del siguiente esquema.

Ejecución de tareas

Evaluación del trabajo

Proceso de asignación de cargos y tareas

***9.1Proceso de Asignación de cargos***

Antes de cualquier cosa el equipo de trabajo acuerda cuáles serán los entregables y los encargados de cada uno de los trabajos, este proceso esta seguido de una investigación la cual tiene como objetivo definir las tareas generales del proyecto, “véase entregables v.1”.

***9.2-Ejecución de tareas (implementación)***

Una vez realizada la asignación de cargos se pone en marcha la ejecución de las tareas, esta se lleva a cabo siguiendo en estricta medida los plazos y actividades acordadas previamente en la carta Gantt, un breve esquema de este es el expuesto a continuación.

Fecha Entregables Carta Gantt

Cumple o no lo pedido

Actividad en pleno proceso

Actividad Inicial

Dentro de lo acordado entre los miembros del equipo de proyecto, se ha decidido establecer una lista de requerimientos, esta lista sirve para determinar si una tarea fue cumplida con éxito o no, en efecto, esta es la forma de evaluar el trabajo realizado en cada tarea, a continuación, se mostrará la lista y los requerimientos de esta y una especificación de cada requerimiento según tareas.

Lista de Evaluación

La nota final de la lista de evaluación representa la satisfacción del equipo respecto a una tarea del proyecto, esta es representada con una nota del 1 al 7, cada miembro del equipo deberá evaluar la tarea según su criterio, es decir si son 5 miembros solo se promedia su nota de satisfacción con las de los 4 integrantes restantes y si la nota es >4 la tarea se da por finalizada.

En términos generales, la evaluación de una tarea refleja la opinión del grupo respecto a esta, es por esto que se puede indicar la siguiente tabla con el punto de satisfacción correspondiente y la forma en la que este puntaje debe ser interpretado.

|  |  |
| --- | --- |
| Nota | Opinión General |
| 1 | Pésimo |
| 2 | Malo |
| 3 | Disconforme |
| 4 | Aceptable |
| 5 | conforme |
| 6 | Optimo |
| 7 | Excelente |

Tarea: Armado del robot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre Tarea | Nota  Ejecución | Nota  Implementación | Nota  Fecha entrega | Nota final |
| Armado robot | 5 | 4 | 4 | 4,3 |
| Código de patrones | 5 | 5 | 6 | 5,3 |

***10-Descripción de la Arquitectura***

Dentro del ámbito del equipo técnico e implementador se puede describir la Arquitectura como una serie de acciones ordenadas en forma secuencial que logran el correcto funcionamiento de la estructura ocupada para realizar una tarea específica, en este caso el robot y su acción de armar patrones.

***10.1-Arquitectura y Acciones:***

***Brazo del robot:*** Tiene la función de cambiar de cara el cubo, Su función y la cantidad de veces usada puede depender de cada patrón.

***Base del robot:*** Es una Pieza que presta como soporte al cubo, es una parte importante en la ejecución de patrones.

***Tipo de Soporte y contacto al piso:*** La estructura y las partes en donde se encuentran posicionadas cada uno de los elementos del robot no están ensamblados de forma aleatoria, se doto al robot de 4 soportes, 2 pares en la parte superior y 2 en la parte inferior, a si mismo el diseño del robot contribuye a distribuir el peso de manera uniforme logrando un equilibrio en la estructura.

***Acciones de la arquitectura***

El Robot realiza un movimiento de inicio para verificar que el cubo se encuentra en posición.

Posición inicial del brazo

El robot realiza movimientos de agarre y posición inicial, lo que implica realizar una acción una y otra vez dependiendo del patrón realizado.

Movimiento iterativo del brazo.

El robot realiza un movimiento de agarre al cubo luego este esta en posición para ejecutar el patrón.

Movimiento de Agarre

***11-Modelo de Implementación***

El presente Diagrama expresa en orden de ejecución y prioridad del proyecto las faces que le siguieron al mismo, a continuación, presentaremos una tarea y su desarrollo, luego mostraremos un diagrama que expone de forma mas general el modelo de implementación.

***11.1 Ejemplo de implementación del modelo “Divide para conquistar Tarea”.***

Tarea Principal

Modularización del trabajo

Modularización del trabajo

Sub Tarea

Sub Tarea

11.2***Implementación del modelo en perspectiva general***

Proyecto

“Sub Tareas”

Modularización

Gestión

Investigación

Entregables

Tareas

***4-Modulos de Implementación Final***

***12.1 Diagrama, módulo de comunicación***

Robot Conectado

Comunicación

Bluetooth

Móvil

“Envía el nombre del patrón”

El robo ejecuta el programa “nombre del patrón”

El programa esta listo para ejecutar otro patron

El robot finaliza la ejecución

***12.2-Implementación Ámbito de Programación (Movimiento robot)***

***Descripción:*** El Código implementado este hecho en el programa NXC.

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);

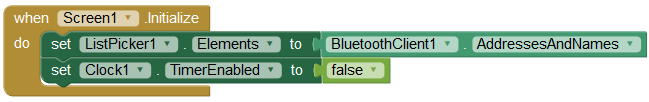
Off(OUT\_C);

Wait(100);

}

***12.3-Implementación Ámbito de Conexión Remota (Comunicación robot App)***

Descripción: La conexión remota fue desarrollada en el entorno de desarrollo App inventor2, el cual proporciona las funciones necesarias para realizar esta tarea, a continuación, mostraremos las funciones utilizadas para esto, y una breve descripción para ver el código fuente completo véase anexo.

****** 1-Se inicializa la conexión vía bluetooth

****** 2-Ejecuta el programa con el nombre “patrón 1”

***13-Análisis de carta Gantt***

En el transcurso del proyecto hemos hecho uso de la carta Gantt proporcionada por la página web entregada como herramienta de administración del proyecto (véase bibliografía), a continuación, expondremos un análisis desde un punto de vista como equipo técnico e implementador.

***13.1- Análisis causa de modificaciones en la carta Gantt***

En la primera parte del proyecto se plantearon una lista de entregables las cuales en el transcurso del tiempo fueron añadidas a la carta Gantt, esta lista de entregables en el transcurso del proyecto fue modificada es decir fueron añadidas o eliminadas algunas tareas.

(Carta Gantt v 1.0) (Carta Gantt v 2.0)

Entregables B

Entregables A

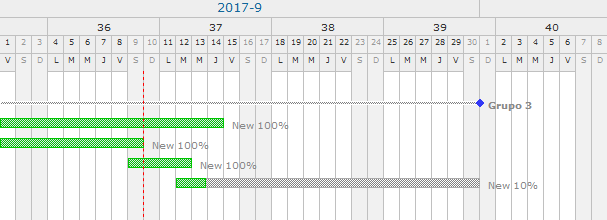
*Tabla de entregables (Tareas Sub tareas)*

|  |  |
| --- | --- |
| Cantidad entregables A | Cantidad Entregables B |
| 7 Tareas 4 sub tareas | 10 tareas 5 sub tareas |

Conclusión Causa de modificaciones: Existen factores que en menor o mayor medida afectan en el avance planificado del proyecto, como, por ejemplo, la falta de piezas para el armado del robot, la ausencia de un integrante del grupo, son solo algunos eventos que significan el retraso en el avance normal del proyecto, es decir, si no se cumplen las tareas en la fecha estipulada esto implica mayor tiempo para terminar dicha tarea lo que hace que otras tareas se retrasen y a su vez implica una modificación en los entregables de la carta Gantt, así mismo habrán algunas sub tareas que no se podrán llevar a cabo por cuestiones de tiempo, esto nuevamente implica una modificación de la carta Gantt.

***13.2-La carta Gantt y su evolución en el proyecto***

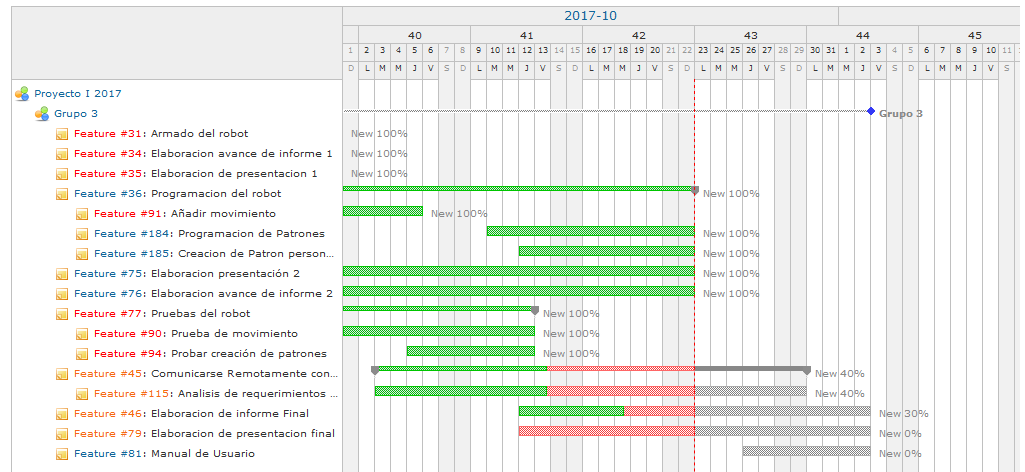
A continuación, se expondrán imágenes de la carta Gantt en orden cronológico, con fecha, y descripción de tareas y un análisis paralelo al avance del proyecto.



Fecha: 9 de septiembre 2017

Descripción: El proyecto estaba ya formulado, se comenzó la puesta en marcha, en la carta Gantt fueron agregados los entregables del proyecto, algunos de estos eran, por ejemplo.

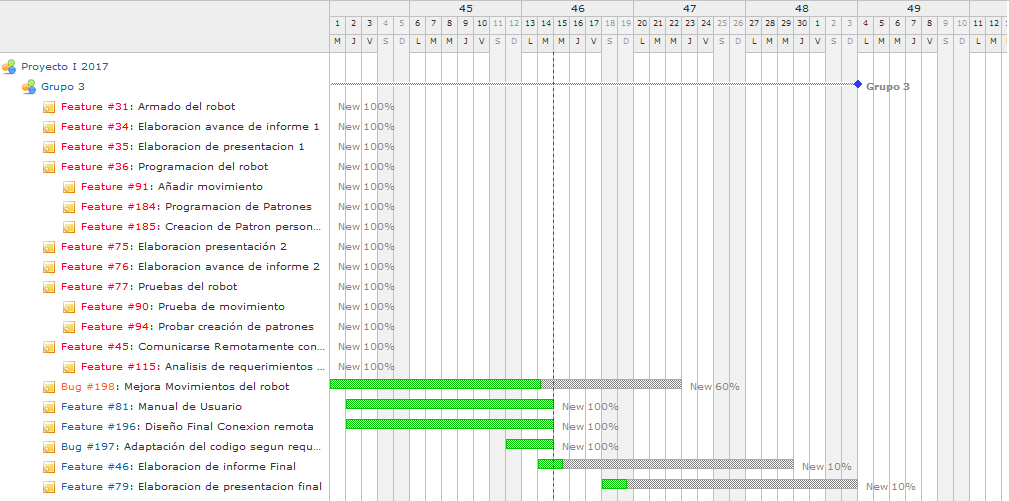
|  |
| --- |
| Armar el Robot |
| Probar Movimiento |
| Detectar Problemas |



Fecha 24 de octubre 2017

Descripción: El objetivo del grupo y su metodología estaban puestas a prueba, este fue el momento en el que el que la ejecución del proyecto, algunos entregables eran.

|  |  |  |
| --- | --- | --- |
| Comunicación remota | Creación de patrones | Análisis de problemas |
| Pruebas del Robot | Programación APP | Diseño APP |



Fecha 14 de noviembre 2017

Descripción: El proyecto esta prácticamente terminado, y se están puliendo algunas cosas relacionadas al producto, La app y se están realizando los últimos informes.

Algunos Entregables son

|  |
| --- |
| Preparación para la presentación final |
| Elaboración del Informe Final |
| Conexión remota y códigos vinculados |
| Manual de Usuario |
| Mejora en movimientos de robot |

Conclusión Evolución del proyecto respecto a la Carta Gantt: Dentro del ámbito del desarrollo del proyecto se hace imprescindible una herramienta de organización como la es la carta Gantt.

***14-Problemas Encontrados y posibles soluciones***

En este apartado se expondrán algunos problemas encontrados, una descripción general las causas y las consecuencias de estos problemas y sus posibles soluciones, los siguientes problemas están expuestos en orden cronológico.

*14.1-Falta de piezas en el armado del robot*

**Problema**: En el proceso de armado de robot faltaron piezas para finalizar el armado del mismo.

**Posibles soluciones**: Comprar Piezas restantes, Conseguir piezas con otros grupos, ver si existían piezas extra disponibles.

**Consecuencias**: Retraso significativo en el proyecto.

*14.2-Desface en los movimientos del robot*

**Problema**: Al ejecutar un patrón el robot ejecuta movimientos con normalidad, pero durante el desarrollo del armado del patrón se detecta que la base se va desviando a medida que el programa de armado es ejecutado, esto produce que en algún momento determinado el robot haga un movimiento fallido y no se logre la culminación del patrón.

**Posibles Soluciones**: Rearmar el robot, Copiar un modelo que funcione perfectamente, Buscar otro diseño, Modificar el código y crear una función que compense el desfase.

**Consecuencias**: El no armado de un patrón afectaría significativamente al producto final, este no cumpliría con lo pedido.

*14.3- Problema de conexión Remota*

**Problema:** Durante la programación de conexión remota existió una incertidumbre en cuanto a cómo sería el proceso de comunicación, es decir, se adaptaría el código de conexión remota al programa de patrones o viceversa.

**Posibles soluciones**: que el código de conexión remota se adapte al programa o viceversa, ver cual resulta menos complicado y llevar a cabo dicha idea.

**Consecuencias**: El no funcionamiento de la aplicación remota haría que el proyecto no cumpla con lo pedido.

**15-Reflexión general del proyecto**: Durante el transcurso del proyecto el equipo de trabajo se encontró con muchos impedimentos para llevar a cabo la ejecución del proyecto según las fechas estipuladas por los entregables, sin embargo, creemos que hemos sido capaces de afrontar estos problemas trabajando en equipo, aportando ideas y finalmente encontrar la solución al problema, en algunos casos nos vimos frente a situaciones en donde no sabíamos por dónde comenzar por ejemplo, la conexión remota, este tipo de situaciones nos enseñó a valorar la investigación previa la ejecución, y además nos demostró que un proyecto bien ejecutado es un proyecto bien planificado, y a pesar de tener una agenda y fechas establecidas es muy importante tener un plan de contingencia, es decir un plan de respaldo que sirva como soporte si es que llegase a acontecer algún evento imprevisto que afecte al avance normal del proyecto.

15.1**-Conclusión:** El proyecto Mindcuber llevado a cabo por el Grupo 3 de proyecto I del segundo semestre del 2017, cumple a fecha de 14 de noviembre del 2017 con todos los requerimientos pedidos para el producto final, siendo los más relevantes, El armado de patrones de un cubo rubik 3x3, y la ejecución de este mediante la conexión remota, de esta manera el producto está terminado y se concluye que el proyecto a finalizado.

Diagrama General Principio y fin del proyecto

Desarrollo del proyecto

Mantenimiento

Planificación del proyecto

Descripción General:

|  |  |  |
| --- | --- | --- |
| Planificación | Desarrollo | Mantenimiento |
| Se constituyen las tareas y encargados | Se ejecuta el plan de proyecto | Se realizan actualizaciones y/o mejoras del producto. |
|  |  |  |

16-***Trabajo Futuro:*** Desde Este Punto, una vez culminado el proyecto queda realizar algunas mejoras en el producto, a continuación, algunas actividades agendadas.

16.1-***Mejorar el diseño de conexión remota***: El actual diseño de la aplicación de conexión remota no está mal, pero dentro del tiempo restante es posible una mejora en cuanto a diseño.

16.2-***Segunda opción de comunicación remota***: La conexión remota se realizó mediante la ejecución de un programa enviado por la app, es posible realizar una aplicación que englobe todos los programas y solo se envíe un mensaje con un patrón especifico y este se ejecute en el mismo programa que los otros, esto es posible mediante el uso de un “while” y un “switch” en el código.

16.3-***Reducción del factor de riesgo de falla en armado de un patrón***:

En el proceso de ejecución de un patrón el robot tiene un pequeño porcentaje a fallar mediante la ejecución esto debido a factores propios del robot, es posible generar un diseño o un código que reduzca aún mas este factor de riesgo.

16.4-***Codigo Eficiente de Transitividad de un patrón a otro***:

Para realizar esta mejora es necesario el código adaptado a conexión remota con selección unitaria de patrones, en lo que se basa esto es en crear una función que sea capaz de detectar la posición del brazo del robot moviéndola hasta la posición más óptima para la realización de un patrón x.

16.5-***Conexión Remota interactiva***:

La aplicación de conexión remota pone en marcha un patrón x, sin embargo, es posible implementar más funciones en esta aplicación que hagan más dinámica la interacción entre el usuario y el producto.

16.6-***Robot con censores de proximidad***:

Una apuesta algo arriesgada consiste en modificar el diseño del robot para que este haga uso de los censores de proximidad, este detectará el desfase de la base del robot y en caso de existir riesgo inminente de desfase este ejecutará un método el cual realizará un movimiento de restablecimiento de la base.

***17-Video demostración funcionamiento del proyecto***

En el siguiente video se presenta el funcionamiento del robot y la elaboración del patrón “6T”, cabe señalar que la ejecución se lleva a cabo mediante la aplicación del producto.

Para acceder al video presione ctrl y haga clic en el vínculo.

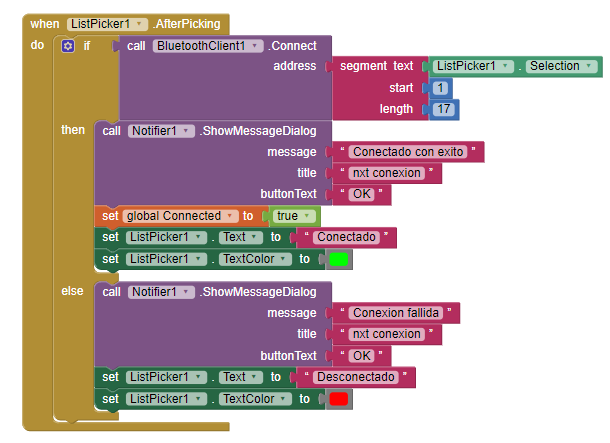
<https://www.youtube.com/watch?v=n60c9FM4piU&feature=youtu.be>

**18-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();**  **}** |

**N°2 Código fuente de conexión remota App inventor 2**

****