**UNIVERSIDAD DE TARAPACÁ**



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



Área de Ingeniería en Computación e Informática



**Informe Final:**

**Robot Rubiktron**

**Autor(es): -Matías Aguayo Méndez**

**-Leonel Alarcón Bravo**

**-Camilo Mamani Cáceres**

**-José Vásquez Gutiérrez**

**-Gonzalo Vega Mujica**

**Asignatura: Proyecto I**

**Profesores: -Ricardo Valdivia**

**-Diego Aracena**

**Arica – Chile**

**2017**

**Historia de revisión**

|  |  |  |
| --- | --- | --- |
| Fecha | Versión | Descripción |
| 31/08/2017 | 0.5 | Borrador del desarrollo del proyecto con sus respectivos puntos y factores. |
| 07/09/2017 | 1.0 | Se agregó la sección de “Gestión de riesgos” y se finalizó el informe inicial. |
| 14/09/2017 | 1.5 | Se modificaron y detallaron algunos puntos en concreto, corrigiendo errores. |
| 28/09/2017 | 1.5.1 | Descripción de los patrones inversos |
| 09/11/2017 | 2.0 | Actualización de los datos del presente documento con respecto al avance del proyecto |
| 23/11/2017 | 3.0 | Finalizar detalles y ordenar el documento para la presentación final. |

1. **Introducción**

Para comenzar, en estos últimos tiempos existe un famoso juego interactivo llamado cubo Rubik y mucha gente lo ha armado llegando a haber competencias sobre esto, pero así como hay mucha gente que sabe armarlo, también hay otros quienes les gusta formar diseño con el cubo, por lo que existen los patrones que logran esto mediante un cubo rubik en estado armando, es por eso que nuestro proyecto trata de ensamblar un robot que pueda armar patrones de estos cubos.

**1.1 Alcance del proyecto**

**1**. En un comienzo se armara el robot.

**2.** Se analizaran códigos y se modificaran para poder llevar a cabo nuestro problema.

**3.** Buscar patrones para que el robot pueda realizarlos

**4.** Poner en práctica el robot y ver el funcionamiento de este.

**5.** Resolver los errores presentados y llevar a cabo su presentación final.

**6.** Desarrollar conocimientos de como programar las instrucciones para que el robot la realice.

**7.** Adquirir experiencia de trabajar con un equipo distinto a lo acostumbrado.

**1.2 Objetivo general**

Crear un robot que pueda armar patrones en un cubo rubik.

**1.3 Objetivos específicos**

- Construir el robot.

- Desarrollar el software para realizar el funcionamiento correcto del robot.

- Manipular remotamente el robot.

**1.4 Entregables del Proyecto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificación Entregable** | **Descripción Entregable** | **Fecha de entrega** | **Lugar de entrega** | **Condiciones satisfacción** |
| Informe inicial y presentación de “Plan de Proyecto” | Se entregara un informe donde se detallaran los distintos tipos de factores involucrados en el proyecto. | 12/09/2017 | Sala de clases | El informe estará completo con sus respectivos factores.  Realizar una presentación óptima. |
| Informe Final y presentación de el “Trabajo de proyecto” | Documento donde se presentaran todos los puntos e importantes donde se detallaran cada aspecto del desarrollo del Proyecto. | 30/11/2017 | Plataforma Redmine | El presente documento estará detallado y bien organizado con sus correspondientes puntos. |
| Aplicación móvil | Aplicación donde se escogerán los patrones para que el robot los pueda realizar. | 30/11/2017 | Sala de clases | Aplicación quede totalmente funcional. |
| Manual de usuario | Documento en el cual se presentan todas las instrucciones para manipular el robot y la aplicación de buena forma | 30/11/2017 | Sala de clases | Manual organizado, bien presentado.  Instrucciones claras. |

1. **Organización del proyecto**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** | **Descripción** | **Involucrados** | **Responsable** |
| Programación | Modificar el código de instrucciones para el robot. | Matías Aguayo Méndez  Leonel Alarcón Bravo  Camilo Mamani Cáceres  José Vásquez Gutiérrez  Gonzalo Vega Mujica | José Vásquez Gutiérrez |
| Diseño | Se armara el robot y en caso de la falta de piezas se realizaran adaptaciones del diseño. | Matías Aguayo Méndez  Leonel Alarcón Bravo  Camilo Mamani Cáceres | Matías Aguayo Méndez |
| Documentación | Realización de los informes, bitácoras y organización del proyecto. | Gonzalo Vega Mujica  José Vásquez Gutiérrez | Gonzalo Vega Mujica |

* 1. **Mecanismos de comunicación**

Para poder tener una buena comunicación, hemos creado un grupo de Facebook, este nos permitirá estar en contacto cada vez que haya una idea o para resolver los problemas que se nos presenten, tanto con relación al proyecto, o dentro del equipo.

Además, se llevaran a cabo una junta cada semana, para poder ver los procesos que llevamos en el proyecto, y así de esta manera tener una mejor claridad con respecto a todas las opiniones formuladas por cada integrante del equipo.

Para finalizar, también se ha creado un grupo en la plataforma Discord, ya que este nos servirá por si alguno de los integrantes del equipo no lograra estar presente en el lugar de la reunión del equipo, así de esa manera a pesar de no estar presente en el lugar, también sabremos lo que opina sobre el avance y posibles cambios durante el desarrollo del proyecto.

1. **Planificación de los procesos de gestión**

Se contará con un cubo rubik adquirido mediante contactos entre los integrantes del equipo, también se contarán con 5 personas las cuales serán los integrantes del equipo y dentro de este se delegarán distintos roles y actividades para realizar el proyecto de forma adecuada.

Considerando el bajo nivel de dificultad del proyecto, además del tiempo estimado que nos llevará el realizarlo, calculamos cerca de unos $500.000 como el valor aproximadamente del robot lego, además agregar el valor del cubo rubik el cual es $8.000, por ultimo añadir el costo humano que equivaldrá a $3.000 \* hora trabajada y tener en cuenta de las 5 personas trabajando en el proyecto. Cabe destacar que nuestro equipo de trabajo a vista de que concluimos ciertas tareas con anticipación en relación a lo planificado a nuestra carta Gantt por lo tanto nos propusimos el desafío y meta de agregar los patrones inversos para así aprovechar al máximo el potencial del robot, también esto fue explicado con más especificaciones en \***6.4 Módulos Implementados.**

Por lo tanto, el costo aproximado final es de $1.588.000 + $500.000

Costo final total: $2.088.000 clp.

* 1. **Gestión de Riesgos**
  2. CATASTRÓFICO
  3. CRÍTICO
  4. MARGINAL
  5. DESPRECIABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **RIESGOS** | **PROBABILIDAD DE OCURRENCIA** | **NIVEL DE IMPACTO** | **ACCIONES REMEDIALES** |
| Falta del personal | 10% | 2 | Repartir todas las funciones y responsabilidades de dicha persona en el resto de integrantes, para así cumplir con su labor y evitar posibles atrasos. |
| Inexperiencia en programación | 20% | 2 | Poder cumplir con todas las áreas iniciales para así tener un aumento gradual en la dificultad de la programación del cual dará solución a nuestro proyecto. |
| Lentitud en toma de decisiones | 10% | 3 | Se consideraran cierto tiempo al inicio de cada clase en la que definiremos nuestras aspiraciones a cumplir en la jornada de trabajo |
| Baja motivación | 10% | 2 | El equipo se ayudará mutuamente con la intensión de mejorar el ánimo necesario para continuar. |
| Accidentes/Enfermedades | 5% | 3 | Se trabajará con normalidad, realizando antes la tarea del compañero faltante si es requerida pronto. Cuando el compañero ausente vuelva, se le pondrá al día con el trabajo. |
| Oposición comunitaria | 15% | 2 | Se tomara en cuenta la opinión de cada integrante, procurando siempre seguir el camino más favorable para el equipo y el trabajo. |
| Fallas de los servicios básicos entregados | 10% | 2 | Se harán los reclamos necesarios para que la facultad cumpla con los servicios básicos. En lo que se soluciona se avanzará en alguna otra parte del trabajo. |
| Indisponibilidad de las herramientas de desarrollo | 5% | 1 | Se avanzara con otras partes del proyecto, dejando el ensamblaje del robot para otro momento. |
| Irresponsabilidad del personal | 15% | 2 | Se comunicará a dicha persona para que cumpla de buena forma su labor a la que fue asignada y se responsabilice de esta, además de que se le supervisará de mejor manera. |
| Falta de complementación durante el trabajo grupal | 10% | 2 | Los integrantes del equipo deberán realizar ciertas reuniones cortas en las cuales se conversará para llegar a ideas y aspiraciones similares y que sean beneficiosas. |
| Pérdida de piezas del robot | 10% | 4 | Se avanzara en lo posible en otras actividades de ensamblaje mientras se reponen las piezas faltantes. |

**4.0 Análisis**

**4.1 Modelo de Diseño General**

Nuestro propósito en un inicio fue fijar un camino claro el cual seguir durante todo el proceso de creación. Básicamente, planteamos las actividades de la siguiente manera:

Paso 1: Inicio

Paso 2: Revisión y búsqueda de las piezas y herramientas para el proyecto.

Paso 3: Construcción del robot.

Paso 4: Conexión del robot a un computador

Paso 5: Creación y elaboración de algoritmos en lenguaje NXC.

Paso 6: Prueba de movimientos del robot.

Paso 7: Desarrollo del software para celular controlador del robot.

Paso 8: Exportación de una aplicación para celular.

Paso 9: Pruebas de manipulación remota del robot.

Paso 10: Elaboración del manual de usuario.

**4.2 Descripción de la arquitectura**

Las herramientas de trabajo para elaborar el proyecto son relativamente sencillas de conseguir ya que la mayoría serán proporcionadas por la misma institución, como es el caso de los computadores y las piezas de LEGO. Elementos como los instructivos a seguir o el cubo Rubik serán buscados y conseguidos por el propio equipo de trabajo.

En un principio, la idea es una vez conseguido las herramientas comenzar inmediatamente a ensamblar las piezas de LEGO para armar el robot. Luego, usando conocimientos previos de programación, nos la ingeniaremos para elaborar algoritmos en el lenguaje de programación NXC, los cuales tendrán como función principal el control de los movimientos del robot. Se realizarán pruebas de movimiento para ir encontrando unos algoritmos más correctos.

Pasando ya la última parte del proyecto, veremos ya los entregables finales de este. Se trabajará en la aplicación para celular y la interfaz de usuario, con la que el consumidor tendrá completo control sobre el robot. Finalmente, y después de hacer todas las pruebas necesarias, se pasará a realizar el manual de usuario que servirá como guía para el consumidor, lo que le enseñará a usar correctamente nuestro producto.

**4.3 Documentación de diseño de Interfaz de Usuario**

Para facilitar nuestro trabajo pensamos en utilizar una herramienta sencilla y ya conocida por todos los miembros del equipo, ***AppInventor2***.

*AppInventor2* es un entorno de desarrollo de software especializado en la elaboración de aplicaciones destinadas a dispositivos Android. Esta plataforma resulta fácil de manejar ya que utiliza el lenguaje de programación visual *Blockly*, que consiste en consiste en, a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación, parecido a como si fueran piezas de un rompecabezas. El sistema es gratuito y se puede utilizar fácilmente desde web.

En esta plataforma se creará y diseñará la interfaz de usuario, para posteriormente pasar a exportarla como una aplicación que corra en distintos dispositivos Android.

La interfaz de usuario planeada para el producto final debe ser lo más accesible y vistosa posible para cualquier persona, por lo mismo consideramos que los siguientes elementos son los más importantes a tener en cuenta y que deben estar presentes de lo que será nuestra interfaz:

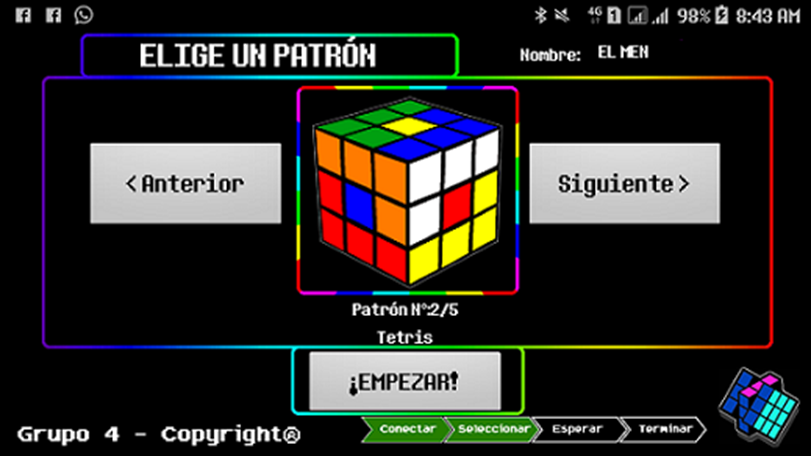
1) Botón para conectar el celular con el robot.

2) Vista del patrón a armar.

3) Botón para ver el siguiente/anterior patrón.

4) Botón para comenzar a armar el patrón actual.

Tomando en cuenta estos aspectos, logramos hacer un primer diseño para nuestra interfaz de usuario, la cual se parece a lo siguiente:

****

*Primer diseño de la Interfaz de Usuario*

**4.4 Especificación de requerimientos**

Nuestro proyecto no solo requerirá el uso de nuestros conocimientos previos de programación y tiempo invertido en este, para concretarlo harán falta materiales tangibles en los cuales poder trabajar. Los principales requerimientos de nuestros trabajadores son los siguientes:

- Piezas de LEGO

- Computador

- Manual de instrucciones del robot

- Celulares Android

- Cubo Rubik

Además, es importante mencionar que contemplamos que nuestro producto final haga uso y requerimiento de elementos externos para funcionar, algunos de estos provenientes por parte del mismo consumidor. Los principales requerimientos que necesitará el cliente para usar nuestro producto son:

- Celular Android

- Manual de usuario

- Cubo Rubik

**5.0 Diseño**

**5.1 Modelo del Diseño Específico**

Paso 1: Inicio:  
 Se comienza con los pasos iniciales para empezar el proyecto.

Paso 2: Revisión y búsqueda de las piezas y herramientas para el proyecto:  
 Se buscan y se ordenan las piezas del set de Lego Mindstorms según el manual del proyecto  
 [MindCuber](http://mindcuber.com/), para mayor eficiencia al construir el robot.

Paso 3: Construcción del robot:  
 Se comienza la construcción del robot, con las partes ya ordenadas y sin saltarse los pasos del  
 manual, En caso de alguna pieza rota o algún error, reponerla.

Paso 4: Conexión del robot a un computador:  
 Se conecta el robot a el computador para hacer pruebas de movimiento con respecto a  
 la plataforma donde el cubo estará y el brazo que moverá el cubo.

Paso 5: Creación y elaboración de algoritmos en lenguaje NXC:  
 Se comienza a crear el programa para que ejecute algoritmos para armar patrones en un cubo  
 mezclando los movimientos del brazo y de la plataforma.

Paso 6: Prueba de movimientos del robot:  
 Con los algoritmos de movimiento ya diseñados, se comienza a hacer pruebas con un algoritmo de  
 un patrón ya generado basándose en los movimientos programados para el robot.

Paso 7: Desarrollo del software para celular controlador del robot:  
 Se investigan formas de conectividad para conectar el celular a el robot, luego de encontrar varias   
 se plantea por cual decidirse, en este caso se usa App Inventor 2, luego de investigar la forma de   
 trabajo de la plataforma, se comienza la codificación del programa, con pruebas de conexión,   
 interpretación de mensajes en el robot y todas las pruebas necesarias.

Paso 8: Creación y elaboración de algoritmos de patrones inversos en lenguaje NXC:  
 Se comienza a crear el programa para que ejecute algoritmos para armar patrones inversos en un   
 cubo mezclando los movimientos del brazo y de la plataforma.

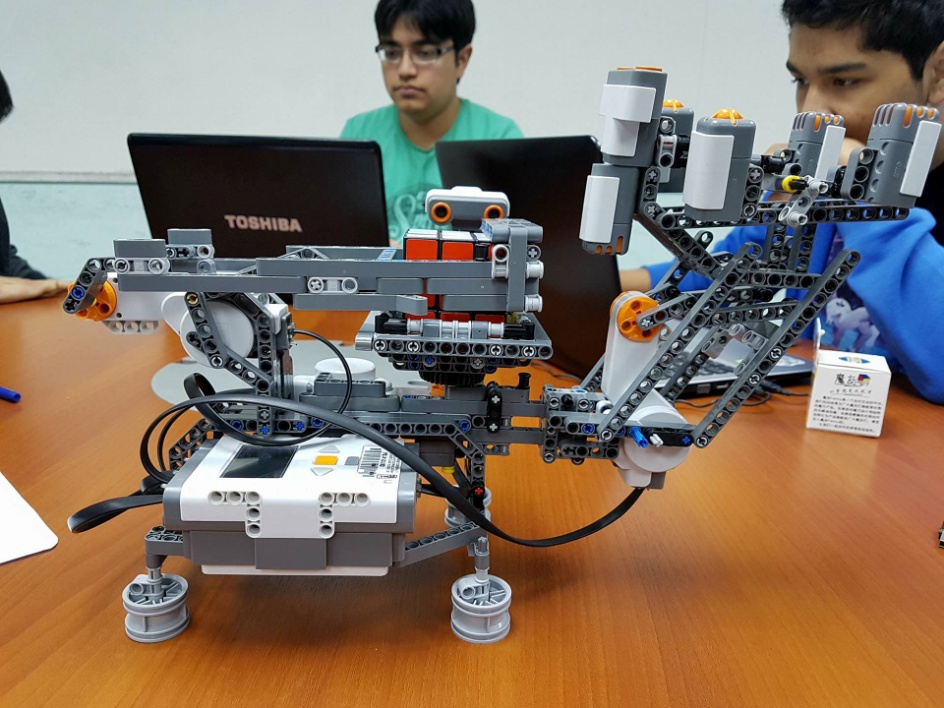
Paso 9: Exportación de una aplicación para celular:  
 Con la aplicación ya hecha, se exporta la aplicación a un archivo .apk que es el instalable de  
 el sistema operativo Android, para uso en cualquier dispositivo que tenga dicho sistema.

Paso 10: Pruebas de manipulación remota del robot:  
 Con el programa para el robot codificado y descargado, y la aplicación de celular creada e instalada,   
 se hacen las pruebas de conexión finales, y se hace pruebas con todos los algoritmos de los  
 patrones.

Paso 11: Elaboración del manual de usuario:  
 Se crea el manual de usuario, para que el usuario lea y comprenda el uso de la aplicación y  
 cómo usar dicha aplicación para armar los patrones.

**5.2 Modelo de interacción**

**-Diagrama de flujo.**



**5.3 Descripción de la Arquitectura.**

A continuación daremos la descripción de los movimientos en específicos y estos serán agrupados y descritos mediante la relación que tengan entre ellos

* **Levantar y Bajar**: Para ejecutar estas acciones se requiere el uso del Motor ‘’A’’, el cual permitirá la movilidad de los brazos del robot, de esta forma al accionar el motor correspondiente con una velocidad del 75% y con un tiempo de xxx segundos, los brazos se detendrán cuando el tiempo ingresado haya finalizado de esta forma los brazos subirán o bajaran dependiendo de qué acción se quiera realizar, además estos movimientos están muy relacionados y asociados al movimiento de “Giro” ya que si el brazo del robot se encuentra abajo y se hace un giro se llevara a cabo el giro de una cara del cubo, por el contrario si el brazo se encuentra arriba se hace el giro del cubo completo.
* **Giro**: Para llevar a cabo el giro se deberá tener los brazos del robot abajo y además se usara el Motor ‘’B’’, ya que este poseerá el control de la plataforma donde se encuentra el cubo rubik, de esta manera al girar la plataforma al lado derecho o izquierdo, según se requiera, los brazos detendrán el cubo mientas la plataforma girará la base del cubo y así realizar el movimiento de girar una de las caras. Por otra parte si los brazos del robot se encuentran levantados se realizara un giro completo del cubo, la razón de esto es porque al girar la plataforma se moverá el cubo completo sin cambiar ninguna de sus caras.
* **Vertical**: esta acción consiste en que gracias al uso del motor ’’A” le dan la movilidad a los brazos del robot el cual hará el movimiento de tomar el cubo y gracias a que el brazo se levanta el cubo logra hacer un movimiento vertical completo haciendo que la cara del cubo que está en la base del robot cambie y así poder hacer este movimiento las veces que sea necesario mientras se solicite para cumplir con los requisitos de cada patrón.

**6.0 Implementación**

**6.1 Plan de Integración:**

Nuestro plan de integración para miembros nuevos consiste en explicar los detalles básicos de cómo trabajar, ya sea como trabaja el robot, con qué programa codificamos el programa del robot y el programa de conectividad remota. Luego consiste en explicar en detalle las operaciones básicas del robot, y como generamos el algoritmo para un patrón, que es en base a estos movimientos (levantar, giro\_base,etc) y luego explicar cómo trabaja la aplicación remota, la cual controla al robot.

**6.2 Descripción de la arquitectura:**

**Sub levantar()**  
{ OnFwd(OUT\_C,50);   
Wait(330);  
Off(OUT\_C); }

**Sub vertical (int n)**{ repeat(n)  
 { RotateMotor(OUT\_C,70,-80);  
 RotateMotor(OUT\_C,70,80); }  
}

**sub giro (int t)**{ if (t==1) //DERECHA  
 { RotateMotor(OUT\_A,75,-305);   
 RotateMotor(OUT\_A,75,35); }  
 else //IZQUIERDA  
 { RotateMotor(OUT\_A,75,305);  
 RotateMotor(OUT\_A,75,-35);}  
}

**Sub bajar()**{ OnRev(OUT\_C,30); Wait(550);  
Off(OUT\_C);  
Wait(200); }

Interpretación del N° Del   
algoritmo a trabajar

**Algoritmo 1:**  
giro(0);  
vertical(2);  
giro(1);  
vertical(1);  
….

Código general con todas estas subtareas

**Algoritmo 2:**  
giro(0);  
vertical(2); giro(0);  
levantar();  
….

Algoritmos restantes…

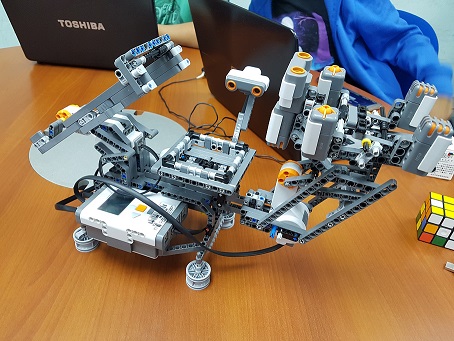
**6.3 Modelo de Implementación:**

La forma de implementar el proyecto se divide en 3 módulos:

|  |  |  |
| --- | --- | --- |
| 1.- Armado del Robot | 2.- Programa del Robot NXT | 3.- Programa de conectividad remota |
| El equipo de Diseño trabaja en construir el robot según las instrucciones del manual para su uso. | Se comienza con las subrutinas mencionadas anteriormente (bajar, levantar, giro\_base, etc.), el cual al recibir un mensaje desde un programa de conectividad remota, el programa del robot lo interpreta como el n° del algoritmo a desarrollar, y según el n°, se ejecutará el conjunto de instrucciones con las subrutinas para lograr armar el robot y terminar enviando una señal de estado a el programa remoto para avisar que se terminó. | Se creará un programa en la plataforma App Inventor 2 la cual tiene facilidad de conectar un dispositivo Android al Robot NXT.  Con una combinación de ventanas y botones, y gráficos (imágenes de los patrones, etc.) se hace una interfaz amigable para el usuario, con botones explicativos para trabajar con el robot. |

**6.4 Módulos Implementados:**

* Armado del robot:



* Creación del Programa NXT:



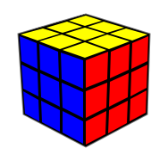
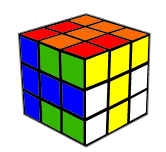
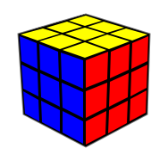
* Creación del Programa de conectividad Remota:



El programa está completo, envía las señales correspondientes y el robot interpreta correctamente dicho mensaje enviado.

* Creación e implementación de los patrones inversos:

**Patrones** **inversos**: Cabe mencionar que en un comienzo nos complicó la creación de algoritmos para crear los patrones, pero una vez ya implementado el primero supimos como trabajar con ellos y en función de mayor eficiencia y en pos de aprovechar el tiempo nos separamos el trabajo en la creación de patrones creando al menos 1 patrón cada integrante del equipo, por esta razón y más relacionadas al trabajo practico del proyecto nuestro avance fue tan eficiente en relación a lo planificado en la carta Gantt que luego de finalizar todos los patrones decidimos aprovechar mejor el potencial del robot y nuevamente en pos de aprovechar el tiempo decidimos ir más allá de los requerimientos del proyecto y plantearnos el desafío de implementar los patrones inversos a estos, se implementaran patrones que vuelva al estado original al cubo rubik, para ello los códigos se invertirán y se modificaran de acuerdo al patrón que fue armado ya que no era algo tan simple como poner el código de vuelta ya que no había movimiento inverso a la función vertical por lo tanto tuvimos problemas pero luego de mucho análisis y trabajo pudimos solucionarlo, de esta forma el cubo quedara por defecto y se podrán realizar los demás patrones sin tener que perder tiempo en resolver dicho patrón, una de las razones que nos motivó y nos llevó a la realización de estos fue lo último, aprovechar el tiempo y tener más eficiencia en realizar nuestro proyecto por lo mismo aprovechando más el potencial de nuestro robot logramos implementar los patrones inversos y así realizar un mejor uso y desempeño del mismo.

*   

**6.5 Reporte de Revisión:**

|  |  |
| --- | --- |
| **Revisión** | **Descripción detallada** |
| 0.01b | Prueba inicial con el movimiento básico (mover motores) |
| 0.1b | Creación del algoritmo de giro\_base y giro\_vertical, el cual es para girar el cubo y mover la plataforma de base |
| 0.2b | Creación del algoritmo de levantar, bajar y hacer pruebas de todos los movimientos |
| 0.5b | Creación de un algoritmo de prueba (Patrón 1) para hacer la prueba definitiva de los algoritmos anteriores |
| 0.8b | Creación de Algoritmos de los patrones 2, 3, 4 |
| 0.9b | Creación de Algoritmos de los patrones 5, 6 |
| 1.0 | Creación del algoritmo del patrón 7 y optimizar código del programa NXT |
| 1.1 | Implementación del modo “Deshacer” el cual con un patrón ya hecho y el cubo bien posicionado, el robot desarma el patrón |
| 1.2 | Implementado el algoritmo de Deshacer de los patrones 1 y 2 |
| 3.0 | Agregado patrón 8, Implementado y probado todos los algoritmos de armar y deshacer patrón. |

**7.0 Aspecto Generales**

**7.1 Problemas Encontrados y Soluciones Propuestas**

1. **Problema**: Se tiene algunas complejidades con relación a los códigos inversos de los patrones, esto quiere decir que al devolver al estado original del cubo desde el patrón armado no queda como se espera.

**Solución propuesta**: Revisar bien el código y posicionar el cubo de tal forma que esta no genere problemas al revertirlo.

1. **Problema**: Uno de los problema fue que en algunos momentos la base giraba más de lo requerido y esto ocurrió el movimiento que generaba el motor del robot no será del todo preciso por ende este se trababa al tratar de realizar el siguiente movimiento del patrón.

**Solución propuesta**: Modificar el algoritmo del código de la base el cual después de cambiar la velocidad de la rotación de la plataforma.

1. **Problema:** El brazo del robot no ejecutaba el 100% de un movimiento deseado, el cual era el primer movimiento de cada patrón, lo cual generaba que se trabara al inicio de este y por consecuencia se debía detener el proceso del armado de dicho patrón.

**Solución propuesta:** Nosotros manualmente hacíamos el primer dicho movimiento el cual generaba confusión al robot, por lo tanto, la solución propuesta fue agregarle este movimiento al inicio de cada algoritmo de cada patrón, por ende nos ahorraríamos el movimiento que nosotros realizábamos manualmente, y así el 100% de los movimientos los realizaría el robot y por consecuencia terminaría con esta confusión.

1. **Problema**: si bien en un inicio tuvimos bastante avance practico lo que en la creación de patrones, armado, ensamblando, otros, pero en lo que respecta a la documentación la eficiencia de avance no es similar a la que tenemos en el ámbito practico.

**Solución:** Dedicarnos netamente unas clases (2 bloques/ 4 horas pedagógicas) completa en la cual discutimos y nos separamos por equipos ciertos trabajos de documentación enfatizando la eficiencia de creación y avance del mismo de los archivos de documentación que se deben presentar.

1. **Problema**: Por causa de apoyo a dos ferias presenciales en las cuales nuestro robot participó en ambas, sufrimos desperfectos de diseño en el robot en cual nos tomó de sorpresa y creo un desfase de tiempo para solucionar cierto problema que no teníamos previsto en la carta Gantt.

**Solución**: Tuvimos que quitar participantes de pequeño grupos que estaban realizando ciertas tareas acordadas previamente para arreglar el ensamblaje del robot y reparación de este mismo, esto fue en función de no atrasarnos en la programación previstas por nosotros mismos.

**7.2 Conclusión**

-A lo largo de la realización de nuestro proyecto nos encontramos con variados problemas de todos los ámbitos, ya sean prácticos como en el ensamblaje del robot, falla de piezas y motores, movimientos y acciones que no tenían sentido e incluso desperfectos en el diseño del mismo perjudicado por personas externas a nuestro proyecto y equipo como también por otro lado en el ámbito de la documentación del mismo ya que como mencionamos antes nuestro avance practico ya sea en elaboración de patrones y demases siempre fue mayor en comparación a la documentación del mismo que si bien el en avance cumplíamos no fue tan equitativo mejor de lo planeado como en lo practico pero a pesar de estos pudimos dialogar, discutir y separarnos bien el trabajo entre los integrantes del equipo para así poder cumplir no solo de manera eficaz sino también de manera eficiente la elaboración y desarrollo de actividades según nuestra planificación de la carta Gantt.

Cabe destacar y mencionar que adquirimos mucha experiencia a lo largo del proyecto, ya sea en el ámbito de la programación en sí de conocer un lenguaje nuevo del cual trabajábamos de forma paralela con una herramienta externa que en este caso es el robot y elaboración de algoritmos para crear los patrones con los limitados movimientos que podía genera el mismo como también en el ámbito del trabajo en equipo en el cual clase a clase teníamos que trabajar y desempeñarnos usando todos los conocimientos previamente adquiridos en ramos anteriores los cuales nos sirvió de mucho al desempeñarnos en este proyecto, cada problema que sucedía en las sesiones de trabajo nos llevó a trabajar como equipo compartiendo ideas y dialogando siempre en función de lo que sea mejor para el equipo y además fue de mucha importancia empezar a relacionarnos ya con otros aspectos como la robótica y trabajar directamente con herramientas y máquinas de trabajo como estas.

**7.3 Trabajo a futuro**

El trabajo y pretensiones a futuro que consideramos importante es la de realizar y llevar un avance y cambio considerable con el proyecto sería adaptar y modificar el ensamblaje de él, de tal forma que podamos realizar más patrones de mayor complejidad, además este irá acompañado de un trabajo de soporte y mantención de la aplicación para que el trabajo de conexión y manipulación inalámbrica sea siempre eficaz y vaya mejorando según vayan saliendo errores o posibles mejoras.