**UNIVERSIDAD DE TARAPACÁ**



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



Área de Ingeniería en Computación e Informática



**Plan de proyecto
War Machine Cube Destroyer**

 **Autor(es): Cristian Bautista M.**

 **Christian Cáceres M.**

 **Felipe López C.**

 **Alan Ortega G.**

 **Asignatura: Proyecto I**

 **Profesor(es): Ricardo Valdivia**

ARICA, DÍA MES AÑO

# Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 16/08/2018 | 1.1 | Desarrollo del panorama general | Alan Ortega |
| 28/08/2018 | 1.3 | Organización del Personal | Alan Ortega |
| 06/09/2018 | 1.5 | Planificación del Proyecto | Alan Ortega |
| 06/09/2018 | 1.7 | Planificación de Recursos | Alan Ortega |
| 06/09/2018 | 1.9 | Referencias | Alan Ortega |

# Tabla de Contenidos

1. Panorama General
	1. Introducción (contexto)
	2. Objetivo General
	3. Restricciones
	4. Entregables
2. Organización del Personal

2.1. Descripción de Roles

2.2. Personal que cumplirá los Roles

2.3. Mecanismos de Comunicación

1. Planificación del Proyecto

3.1. Actividades (nombre, descripción, responsable, producto)

3.2. Asignación de tiempo (carta Gantt Redmine)

3.3. Personal-rol asignado

3.3. Gestión de Riesgos (ver plantilla para el Tratamiento de los Riesgos)

1. Planificación de los Recursos

4.1. Recursos Hardware-Software requeridos

4.2. Estimación de Costos (Hardware, Software, Recursos Humanos)

1. Análisis – Diseño
	1. Especificación de Requerimientos (incluyendo método/algoritmos considerados para resolver el cubo Rubik)
	2. Arquitectura Propuesta (incluyendo aspectos de comunicación)
	3. Diseño de la Interfaz Usuario
2. Implementación
	1. Descripción de los programas implementados (entradas, salidas, procesos)
	2. Diagrama de interacción entre programas
3. Resultados
	1. Estado actual del proyecto
	2. Problemas encontrados y soluciones propuestas
	3. Conclusiones
4. Referencias (estándar IEEE)

Anexos

Anexo A: Código de los programas implementados

Anexo B: Robot (diagrama de construcción, componentes principales)

1. **Panorama General**

# Introducción

Se desarrollará un proyecto que consiste en la construcción de un robot. Este debe tener la capacidad de resolver un cubo rubik’s mediante algoritmos generados por los integrantes. El robot se construirá a base de un kit ev3 lego y deberá ser capaz de mover paso a paso las caras del cubo, logrando que éstas tengan sus colores correspondientes.

# Objetivo General

Construir un robot EV3 capaz de ejecutar algoritmos de resolución para un cubo rubik’s 3x3x3 a través de un programa diseñado para controlar el hardware del WarMachine: Cube Destroyer.

# Objetivos Específicos

* Diseñar y armar un robot que pueda manipular un cubo rubik’s.
* Desarrollar un programa en lenguaje Python que logre realizar los algoritmos traducidos para la resolución de un cubo rubik’s
* Crear una aplicación capacitada para enviar instrucciones remotamente hacia el robot EV3.

# Restricciones

* El tiempo asignado para realizar el proyecto a lo largo del semestre.
* El robot EV3 es el único que podemos utilizar para programar nuestros algoritmos.
* Sólo disponemos de los movimientos que nos proporcionan los motores para la resolución del cubo rubik’s.

# Entregables

* Informe Plan de Proyecto
* Informe Avance I
* Producto Final
* Bitácora Semanal
* Riesgos
* Manual de Usuario
1. **Organización del Personal**

# Descripción de Roles

* Programador: Tiene la responsabilidad de traducir los algoritmos del cubo rubik’s a lenguaje Python, lograr que el robot EV3 realice las acciones correspondientes y también se encarga del avance de la WIKI.
* Constructor: Tiene la responsabilidad de crear una lista con las piezas faltantes del robot EV3 para el armado completo de éste.
* Secretario: Tiene la responsabilidad de dar avance y desarrollo a los informes, carta Gantt y Bitácora Semanal.
* Líder: Tiene la responsabilidad de organizar y otorgar apoyo a las distintas áreas de trabajo dentro del proyecto.

# Personal que cumplirá los roles

* Programador: Alan Ortega G. y Cristian Cáceres M.
* Constructor: Cristian Bautista M.
* Líder: Felipe López C.
* Secretario: Alan Ortega G.

# Mecanismos de Comunicación

Para poder obtener una comunicación adecuada se decidió crear un grupo de WhatsApp establecido por el líder, para poder resolver problemas encontrados en clases y poder solucionarlos, entonces se llegará a la otra sesión con el material adecuado.

1. **Planificación del Proyecto**

# Actividades

* Armar robot EV3.
* Instalar sistema operativo en la tarjeta SD para trabajar con el robot EV3.
* Configurar conexión WIFI mediante un adaptador USB Tenda.
* Instalar programas necesarios para la programación y comunicación con el robot EV3.
* Traducir algoritmos de resolución del cubo rubik’s al lenguaje de máquina.
* Implementar una aplicación para la comunicación con el robot mediante un dispositivo Android.

# Asignación de Tiempo



# Personal-rol Asignado

|  |  |
| --- | --- |
| Responsable a cargo | Actividad |
| Alan Ortega | Instalación de programas, configuración Wifi, Desarrollo del Plan de Proyecto y programación en Python |
| Felipe López | Presentación y lista de riesgos |
| Cristian Cáceres | Traducción de algoritmos del cubo Rubik’s, avance en redmine (WIKI) e instalar Sistema Operativo EV3DEV Stretch Beta |
| Christian Bautista | Construcción robot EV3 y diseño de la interfaz |

# Gestión de Riesgos

|  |
| --- |
| CATEGORIA DE RIESGOS |
| 1 | CATASTROFICO |
| 2 | CRÍTICO |
| 3 | MARGINAL |
| 4 | DESPRECIABLE |

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgos | Probabilidad de ocurrencia | Nivel de impacto | Acción remedial |
| Perdida de información | 60% | 1 | Hacer un respaldo de la información correspondiente continuamente en los dispositivos del equipo. |
| Enfermedades/Accidentes | 40% | 3 | Repartir tareas del integrante en cuestión al equipo y el integrante cuando se reintegre necesitara adelantar trabajo. |
| Falta de herramientas para desarrollar el proyecto | 30% | 3 | Buscar mediante otros medios estas herramientas. |
| Falta de piezas para el robot ev3 | 70% | 2 | Comprar las piezas faltantes para el robot en el mercado. |
| Daño a tarjeta de SD del robot | 60% | 2 | Hacer el respaldo debido en algún dispositivo del equipo. |
| Falla del software necesario | 30% | 3 | Buscar mediante internet otras opciones similares al software necesario. |
| Errores de programación | 50% | 3 | Buscar errores en internet para su debida resolución. |
| Abandono de integrantes | 30% | 2 | Tomar las tareas del integrante y repartir deberes a los integrantes del equipo. |

1. **Planificación de Recursos**

# Recursos Hardware-Software Requeridos

Hardware:

* Robot Lego EV3
* Adaptador WIFI Tenda
* Cubo Rubik’s
* Memoria SD

Software:

* Sistema Operativo EV3DEV – STRETCH BETA
* Visual Studio Code
* Putty (SSH)
* Python 3.7.0

# Estimación de Costos

* Robot Lego EV3 - $490.209 CLP
* Cubo Rubik’s - $15.000 CLP
* Micro/Adapater SD ADATA 8GB - $3.550 CLP
* Adaptador WIFI Tenda - $6.000 CLP

# Recursos humanos

Para llevar a cabo el proyecto, se debió gestionar un aproximado de sueldo para el equipo, el cual llego al monto de $2.592.000 CLP por el periodo de desarrollo del proyecto, debido a que la hora de trabajo por cada integrante esta evaluada en $9.000 CLP, son 4,5 horas de trabajo a la semana y 16 semanas en las cuales se debe realizar el proyecto, el sueldo para cada integrante es de $648.000 CLP.

1. **Análisis – Diseño**

# Especificación de Requerimiento

* El usuario dispondrá de una interfaz en un computador el cual contiene imágenes descriptivas de cada movimiento en específico con su debido algoritmo de resolución y este estará conectado con el brick del robot vía wifi para que se puedan comunicar como es debido.
* Se dispondrá de un manual de usuario con el cual se podrá proceder a utilizar la interfaz correctamente, señalando los algoritmos para armar el cubo.
* Se implementará un robot que estará capacitado con motores para poder realizar los movimientos correspondientes al efectuar el armado del cubo mediante un brazo y una base.
* Algoritmos de Resolución:

BR – Movimiento con el brazo

GSH – Giro Suelto Horario

GSAH – Giro Suelto Anti-Horario

GPH – Giro Presionado Horario

GPAH – Giro Presionado Anti-Horario

Sexy Move Derecha:

2BR – 1GPAH – 1GSH – 1BR – 1GPH – 2GS – 1BR – 1GPH – 2GS – 1BR – 1GPAH – 1BR – 1GSH

Sexy Move Izquierda:

2BR – 1GPH – 1GSAH – 1BR – 1GPAH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 1BR – 1GSAH

T Derecha:

1GPH – 1BR – 1GSH – 1BR – 1GPH – 1GSH – 1BR – 1GPH – 1GSH – 1BR – 1GPAH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPAH – 1GSAH – 2GS – 1BR – 1GPAH – 1GSH – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 2GS – 1BR – 1GPH – 2GS – 1BR – 2GS

T Izquierda:

1GPAH – 1BR – 1GSAH – 1BR – 1GPAH – 1GSAH – 1BR – 1GPAH – 1GSAH – 1BR – 1GPH – 2GS – 1BR – 1GPH – 2GS – 1BR – 1GPH – 1GSH – 2GS – 1BR – 1GPH – 1GSAH – 1BR – 1GPH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 2GS

L Invertida:

1BR – 1GPAH – 1GSH – 1BR – 1GPAH – 1GSAH – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 2GS – 1BR – 1GPH – 1GS – 1BR – 1GPH – 1BR

Aristas Superior:

1GSH – 1BR – 1GPH – 1BR – 1GPH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 2GS – 1BR – 1GPH – 2GS – 1BR – 2GPH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 2BR

Esquinas Superior:

2BR – 1GPAH – 1GSH – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 1BR – 1GPH – 1GS – 1BR – 1GPAH – 1BR – 1GPH – 2GS – 1BR – 1GPH – 1BR – 1GPAH – 1BR – 1GSAH

Final:

1GSH – 1BR – 1GPH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPAH – 2GS – 1BR – 1GPH – 1GSH

# Arquitectura Propuesta



# Diseño de la Interfaz de Usuario



1. **Implementación**

# Descripción de los Programas Implementados

Se utiliza el programa Visual Studio Code el cual nos permite conectar el brick del robot EV3 mediante una conexión wifi para enviar los códigos de los algoritmos implementados a partir del lenguaje Python y las bibliotecas correspondientes para la utilización de las funciones enviadas por el usuario mediante la aplicación remota para el movimiento del robot que permitan a éste lograr la ejecución de los distintos algoritmos de resolución.

Las funciones implementadas reciben como parámetros, velocidad y segundos, los que nos permiten modificar la potencia y la duración de cada movimiento.

LARGEMOTOR (OUTPUT\_” ”): Nos permite asignar a una variable el objeto motor específico del robot (el cual se desea utilizar) ya sea el motor “A” o “B” (motores que realizan los movimientos, en nuestro caso).

ON\_FOR\_SECONDS (VELOCIDAD, SEGUNDOS): Función que nos permite modificar la velocidad (que también aumenta la potencia del motor) y segundos correspondientes que necesita cada movimiento en específico.

Ejemplo:



En el ejemplo anterior podemos apreciar una función definida como GSH() la cual realiza el algoritmo llamado “Giro Suelto” en sentido horario, “m” es una variable a la cual se le asigna el objeto motor B (que pertenece a la base del robot) y luego se utiliza la función on\_for\_seconds (descrita anteriormente) dando como parámetros la velocidad (50) y los segundos (0.53).

Esta función es implementada en todas las funciones de cada algoritmo, ya que es la más fácil y eficaz para poder programar todos los movimientos del robot.

# Diagrama de Interacción entre Programas

Aplicación Remota

(Computador)

Ejecuta

Código Python

Ingresa

Selección de Movimiento

Conecta

Vía WIFI (SSH)

Con

Robot EV3

Ejecuta

Algoritmo

1. **Resultados**

# Estado Actual del Proyecto

El robot actualmente se encuentra armado y realiza algunos movimientos programados (brazo y giro de base). Nos encontramos desarrollando los algoritmos de resolución del cubo y además se creó una interfaz que posteriormente se implementará. Una vez lograda la conexión remota entre el robot y la interfaz de usuario podremos seguir a la fase final del proyecto.

# Problemas Encontrados y Soluciones Propuestas

* Problemas con la programación del robot, específicamente en los grados, estos funcionan correctamente pero luego de su uso reiterado no completa el recorrido como es debido.

Solución: Utilizar otro tipo de funciones y variar con la velocidad del motor.

* Incompatibilidad de conexión remota entre nuestra interfaz hecha en app inventor y el ev3.

Solución: Cambiar nuestro dispositivo remoto a un computador y crear una interfaz que sea compatible.

# Conclusiones

Como grupo determinamos los problemas encontrados durante el proyecto, los cuales deberán ser resueltos para poder proseguir a la fase final. Estos problemas nos ayudaron a tomar decisiones rápidas e inesperadas ya que implicaban realizar un cambio brusco en esta etapa del proyecto, pero una vez solucionados los problemas, se finalizarán muchas tareas pendientes.

1. **Referencias**

<https://python-ev3dev.readthedocs.io/en/ev3dev-stretch/motors.html#ev3dev.motor.Motor.on_for_degrees>

Anexo A:



Anexo B:

El robot dispondrá de las siguientes piezas, las cuales permiten que se arme el cubo rubik’s:

* Base o plataforma: Permite girar completamente (o lo solicitado) de forma horizonal el cubo rubik’s.
* Brazo o gancho: Permite girar verticalmente las veces que sean necesarias el cubo rubik’s.