



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE MECÁNICA
CARRERA DE INGENIERÍA INDUSTRIAL

**“DESARROLLO DE UNA ARQUITECTURA SOFTWARE PARA
TELEOPERACIÓN Y MONITOREO DEL ROBOT ANNO RV624”**

Trabajo De Titulación

Tipo: Proyecto Técnico

Trabajo de titulación presentado para optar por el grado académico de:

INGENIERO INDUSTRIAL

AUTORES: JHON WILLIAM MERA REASCOS

CRISTIAN ELIASAR QUICALIQUIN JOGACHO

DIRECTOR: ING. EDUARDO FRANCISCO GARCÍA CABEZAS

RIOBAMBA – ECUADOR

2021

© 2021, Jhon William Mera Reascos - Cristian Elíasar Quicaliquin Jogacho

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, **JHON WILLIAM MERA REASCOS** y **CRISTIAN ELIASAR QUICALIQUIN JOGACHO**, declaramos que el presente trabajo de titulación es de nuestra autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 19 de marzo del 2021



Jhon William Mera Reascos
CI: 2300155682



Cristian Eliasar Quicaliquin Jogacho
CI: 1726749342

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA DE INGENIERÍA INDUSTRIAL

El Tribunal del trabajo de titulación certifica que: El trabajo de titulación: Tipo: Proyecto Técnico, **DESARROLLO DE UNA ARQUITECTURA SOFTWARE PARA TELEOPERACIÓN Y MONITOREO DEL ROBOT ANNO RV624. 2021**, realizado por los señores: **JHON WILLIAM MERA REASCOS Y CRISTIAN ELIASAR QUICALIQUIN JOGACHO**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

FIRMA

FECHA

Ing. Marco Homero Almendariz Puentes
PRESIDENTE DEL TRIBUNAL



Firmado electrónicamente por:
**MARCO HOMERO
ALMENDARIZ
PUENTE**

Ing. Eduardo Francisco García Cabezas
**DIRECTOR DEL TRABAJO
DE TITULACIÓN**



Firmado electrónicamente por:
**EDUARDO
FRANCISCO GARCIA
CABEZAS**

Ing. Jhony Marcelo Orozco Ramos
MIEMBRO DE TRIBUNAL



Firmado electrónicamente por:
**JHONNY MARCELO
OROZCO RAMOS**

DEDICATORIA

Dedico este trabajo de titulación a mi madre María Magdalena Reascos Padilla y a mi padre Pedro Geovanny Mera Ponce por la motivación diaria, la ayuda económica y más que todo por ser mi principal fuente de inspiración para ser una mejor persona, enseñándome valores desde mis primeros años de vida y que el esfuerzo tarde o temprano tendrá su recompensa; a mis hermanos Karol Jamilex Mera Reascos, Ivonne Elisabeth Reascos Padilla por darme ánimos en las etapas más difíciles de mi vida universitaria y por hacerme ver que soy su principal ejemplo a seguir, y a Axel Sebastián Mera Herrera por hacerme sentir que soy su otro padre.

Jhon William Mera Reascos

Este trabajo de titulación se la dedico a mi familia, quienes siempre me apoyaron de una u otro manera. En especial a mi padre, Wilson Quicaliquin, por todo el cariño y amor que me dio para siempre salir adelante, por ese apoyo incondicional junto a mi madre, María Maruja Jogacho, que al igual que mi padre, me brindó todo el cariño, amor y apoyo hasta su último momento, por todo ese sacrificio hecho para poder culminar con esta etapa de mi vida, por las enseñanzas y confianza que siempre tuvieron en mí.

A mis hermanos, por siempre confiar en mí y estar ahí en los momentos complicados de la vida.
A mis Abuelos y Tíos, quienes me dieron consejos y el apoyo en los momentos más adecuados.
A mis Primos por compartir momentos buenos y malos, y que me brindaron su ayuda sin dudarlos.

Cristian Eliasar Quicaliquin Jogacho

AGRADECIMIENTO

Agradezco a los docentes de la carrera de Ingeniería Industrial por todos los buenos y malos momentos que pasé a lo largo de mi carrera universitaria y más que todo por los conocimientos otorgados que hoy en día me ayudan a ser una mejor persona llena de conocimientos.

Un agradecimiento muy especial a Johanna Lisbeth Muñoz Herrera por toda la fuerza, apoyo, optimismo y amor que me brindo todos y cada uno de los días, pilar fundamental para que yo llegue hasta esta instancia de mi vida.

Un agradecimiento especial a mi tutor Ing. Eduardo García y a mi asesor Ing. Jhonny Orozco por brindarnos un poco de su tiempo en los momentos que más lo necesitábamos aportando con consejos acertados y seguros.

Jhon William Mera Reascos

A la Escuela Superior Politécnica de Chimborazo, a la Facultad de Mecánica, Carrera de Ingeniería Industrial, por darme la oportunidad de adquirir conocimientos para mi formación profesional, al Ing. Eduardo García e Ing. Jhony Orozco, por brindarnos su apoyo para culminar con éxitos nuestro trabajo, a todos mis amigos, maestros y compañeros.

A mis padres, quienes siempre han velado por mi bienestar, salud y educación, por haber hecho de mí un hombre con valores, a mis Hermanos, Abuelos, Tíos y Primos. Por ese apoyo incondicional que siempre mostraron.

Por último, pero no menos importante, agradezco a Dios, por darme salud y vida para cumplir con esta anhelada meta, por haber puesto en mi camino a personas con las cuales puede contar siempre para salir adelante.

Cristian Eliasar Quicaliquin Jogacho

TABLA DE CONTENIDO

RESUMEN.....	xvii
SUMMARY	xviii
INTRODUCCIÓN	1
CAPITULO I	
1. DIAGNÓSTICO DEL PROBLEMA	2
1.1 Antecedentes.....	2
1.2 Planteamiento del problema	4
1.3 Justificación.....	5
1.4 Objetivos.....	5
1.4.1 <i>Objetivo general</i>	5
1.4.2 <i>Objetivos específicos</i>	5
CAPITULO II	
2. MARCO TEÓRICO	6
2.1 Historia del robot.....	6
2.2 Tipos de robots.....	7
2.2.1 <i>Robots Industriales</i>	7
2.2.2 <i>Robots militares</i>	8
2.2.3 <i>Robots de entretenimiento</i>	8
2.2.4 <i>Robots en la industria médica</i>	8
2.2.5 <i>Robots de servicio</i>	9
2.2.6 <i>Robots espaciales</i>	9
2.2.7 <i>Robot Manipulador</i>	10
2.3 Actuadores en los Robots.....	10
2.3.1 <i>Actuadores Hidráulico</i>	11
2.3.2 <i>Actuadores Neumáticos</i>	11

2.3.3	Actuadores Eléctricos	11
2.3.3.1	<i>Motores de Paso a Paso</i>	11
2.3.3.2	<i>Motores de Corriente Directa</i>	13
2.3.3.3	<i>Motores de Corriente Alterna</i>	13
2.4	Visual Basic .NET	13
2.4.1	<i>Requisitos previos</i>	14
2.4.2	<i>Lenguaje de programación C#</i>	15
2.4.3	<i>Características de C#</i>	15
2.4.4	<i>Identificación de variables y constantes</i>	17
2.5	Teleoperación	19
2.5.1	<i>Retardos en teleoperación</i>	20
2.5.1.1	<i>Retardo por Protocolo de Comunicación</i>	20
2.5.1.2	<i>Retardo por Velocidad de Transferencia de Datos en la Red</i>	20
2.6	Internet de las cosas	20
2.7	La arquitectura de tres capas	21
2.8	Software de Programación	21
2.8.1	<i>Visual Basic</i>	21
2.8.2	<i>Matlab</i>	21
2.8.3	<i>Arduino</i>	22
2.8.4	<i>Labview</i>	22
2.8.5	<i>Ubidots</i>	22
2.9	Programación en Robots Industriales	23
2.10	Arquitectura de programación de robots industriales	23
2.11	Métodos de programación de robots industriales	24
2.11.1	<i>Programación por guiado</i>	24
2.11.2	<i>Programación textual</i>	24
CAPÍTULO III		
3.	METODOLOGÍA	25

3.1	Evaluación de la situación Actual	26
3.1.1	Hardware.....	26
3.1.1.1	<i>Brazo robótico.....</i>	26
3.1.1.2	<i>Conjunto controlador.....</i>	27
3.1.1.3	<i>Electroválvula y mini bomba.....</i>	29
3.1.1.4	<i>Teach Pendant.....</i>	30
3.1.2	Software.....	30
3.1.2.1	<i>Interfaz.....</i>	30
3.1.2.2	<i>Instalación USB a RS232.....</i>	31
3.1.3	Pruebas realizadas al funcionamiento actual del robot.....	33
3.1.3.1	<i>Manipulación.....</i>	33
3.1.3.2	<i>Respuesta de las pruebas realizadas.....</i>	35
3.2	Requerimientos para la teleoperación y monitoreo.....	37
3.2.1	Requerimiento Funcional	37
3.2.2	Requerimiento hardware.....	37
3.2.3	Requerimiento software.....	37
3.2.4	Requerimiento estructural.....	38
3.3	Diseño del modelo	39
3.3.1	Diseño estructural.....	39
3.3.1.1	<i>Análisis estático.....</i>	42
3.3.1.2	<i>Comparación de costos de fabricación.....</i>	46
3.3.2	Diseño hardware.....	48
3.3.2.1	<i>Conexiones.....</i>	50
3.3.3	Diseño software.....	52
3.3.3.1	<i>Programación.....</i>	52
3.3.3.2	<i>Interfaz.....</i>	59
CAPÍTULO IV		
4.	RESULTADOS OBTENIDOS	65

4.1	Conexión inicial.....	65
4.2	Modos de operación.....	66
4.2.1	<i>Modo control HMI.....</i>	66
4.2.2	<i>Modo HMI-MONITOREO.....</i>	71
4.2.3	<i>Modo control Ubidots</i>	74
4.3	Respuesta de la interacción del Software	75
4.4	Gestión del proyecto	78
4.4.1	<i>Cronograma</i>	78
4.4.2	<i>Costos</i>	78
4.4.2.1	<i>Costos directos e indirectos.....</i>	78
4.4.2.2	<i>Recurso humano</i>	79
4.4.2.3	<i>Recurso Material.....</i>	79
	CONCLUSIONES	80
	GLOSARIO	
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2:	Especificaciones para el equipo.....	14
Tabla 2-2:	Asignación de una variable.	17
Tabla 1-3:	Denominación de los motores paso a paso que contiene el brazo.....	27
Tabla 2-3:	Detalle y especificaciones de los elementos del controlador.....	28
Tabla 3-3:	Pasos para hacer la comunicación de USB a RS232	31
Tabla 4-3:	Funcionamiento de los motores en las pruebas realizadas.	36
Tabla 5-3:	Evaluación para la ponderación del software.	38
Tabla 6-3:	Ponderación del software de programación.....	38
Tabla 7-3:	Características del brazo robótico.....	40
Tabla 8-3:	Propiedades de la bancada.	42
Tabla 9-3:	Propiedades del material de la bancada.	43
Tabla 10-3:	Sujeciones.....	43
Tabla 11-3:	Valores de tensión de la bancada.....	44
Tabla 12-3:	Valores de desplazamiento de la bancada.	45
Tabla 13-3:	Valores de deformación unitaria de la bancada.	46
Tabla 14-3:	Costos de fabricación para una bancada en hierro.....	47
Tabla 15-3:	Costos de fabricación de la bancada en madera.	47
Tabla 16-3:	Rango de moción de cada junta.	49
Tabla 17-3:	Función de los widgets utilizados en los dashboards de Ubidots.....	56
Tabla 1-4:	Evaluación de respuesta del movimiento por juntas.	68
Tabla 2-4:	Evaluación de respuesta de la operación.	68
Tabla 3-4:	Evaluación de respuesta del gripper	69
Tabla 4-4:	Respuesta de la comunicación serial, botón home y velocidad.	70
Tabla 5-4:	Tiempos promedios en desplazarse 180°.....	76
Tabla 6-4:	Velocidad angular de cada juntura al 30% de la velocidad seteada.	76
Tabla 7-4:	Velocidades angulares mínimas.	76
Tabla 8-4:	Velocidades angulares máximas.....	77

Tabla 9-4:	Tiempo promedio de monitoreo	77
Tabla 10-4:	Tiempo promedio de teleoperación.....	78
Tabla 11-4:	Cronograma de actividades del tema de titulación.....	78
Tabla 12-4:	Costos de los elementos adquiridos para en anclaje del brazo.....	79

ÍNDICE DE FIGURAS

Figura 1-2.	Robot industrial KR QUANTEC	7
Figura 2-2.	Robot militar Goga Shutterstock.....	8
Figura 3-2.	Robot Sony Aibo.....	8
Figura 4-2.	Araña robótica.....	9
Figura 5-2.	Robot aspirador IROBOT ROOMBA 606.....	9
Figura 6-2.	Robot Mary Anning	10
Figura 7-2.	Comparativa brazo robótico y brazo humano	10
Figura 8-2.	Elementos de un sistema teleoperado.....	19
Figura 1-3.	Diagrama de flujo de la metodología aplicada.....	25
Figura 2-3.	Estado inicial después del ensamblado	26
Figura 3-3.	Localización de los componentes del brazo	27
Figura 4-3.	Localización de los componentes del conjunto controlador.....	29
Figura 5-3.	Electroválvula.....	29
Figura 6-3.	Teach Pendant	30
Figura 7-3.	Interfaz original.....	31
Figura 8-3.	Propiedades del driver instalado.....	33
Figura 9-3.	Conexión lista para el uso	33
Figura 10-3.	Interfaz de configuraciones para el arranque del robot	34
Figura 11-3.	Pantalla de accionamiento para el movimiento del brazo	35
Figura 12-3.	Movimiento involuntario.....	36
Figura 13-3.	Diagrama de flujo.....	39
Figura 14-3.	Datos para obtener la fuerza máxima a ejercer.	40
Figura 15-3.	Diseño 3D de la bancada para el brazo robótico.	42
Figura 16-3.	Análisis de tensión de la bancada.....	44
Figura 17-3.	Análisis de desplazamiento de la bancada.....	45
Figura 18-3.	Deformación unitaria de la bancada.....	46

Figura 19-3.	Bancada para el brazo robótico.	48
Figura 20-3.	Mantenimiento de los motores 1 y 2.	49
Figura 21-3.	Diagrama de conexión del controlador-fuente.	50
Figura 22-3.	Diagrama de conexión del controlador-motores.	50
Figura 23-3.	Diagrama de conexión del controlador-motores.	51
Figura 24-3.	Diagrama de conexión del controlador- placa de conexión DB25-M.	51
Figura 25-3.	Denominación de las librerías DLL.	53
Figura 26-3.	Programación para movimiento individual de las juntas.	53
Figura 27-3.	Comunicación entre Ubidots y la Interfaz.	54
Figura 28-3.	Dispositivo creado en la plataforma Ubidots	54
Figura 29-3.	Crear una variable con la opción RAW.	55
Figura 30-3.	Dashboards creados en la plataforma.	55
Figura 31-3.	Widgets utilizados para la teleoperación.	56
Figura 32-3.	Interfaz gráfica para teleoperación y monitoreo.	60
Figura 33-3.	Conexión para el puerto serial.	60
Figura 34-3.	Estado de conexión de la interfaz.	60
Figura 35-3.	Selección para modo de conexión.	61
Figura 36-3.	Diagrama de flujo de la programación en el modo Control-HMI.	61
Figura 37-3.	Diagrama de flujo - programación en modo HMI-Monitoreo.	62
Figura 38-3.	Diagrama de flujo de - programación en modo Control Ubidots.	62
Figura 39-3.	Operación trabajar.	63
Figura 40-3.	Mandos para control de forma individual.	63
Figura 41-3.	Mando de accionamiento de mini bomba y gripper.	64
Figura 42-3.	Seteo de velocidad.	64
Figura 1-4.	Estados para conexión de la interfaz.	65
Figura 2-4.	Puesta en marcha para trabajar desde Ubidots.	66
Figura 3-4.	Movimiento de las juntas de forma individual.	67
Figura 4-4.	Movimiento de las juntas de forma colectiva.	67

Figura 5-4.	Botones para la OPERACIÓN en la interfaz.	69
Figura 6-4.	Ubicación de los botones para el gripper en la interfaz.....	70
Figura 7-4.	Ubicación del puerto serial, home y velocidad.	70
Figura 8-4.	Selección del modo HMI-MONITOREO.	71
Figura 9-4.	Monitoreo en tiempo real desde Ubidots.	72
Figura 10-4.	Grafica tiempo vs grado de movimiento junta 6.	72
Figura 11-4.	Grafica tiempo vs grado de movimiento junta 5.	73
Figura 12-4.	Grafica tiempo vs grado de movimiento junta 4.	73
Figura 13-4.	Grafica tiempo vs grado de movimiento junta 3.	73
Figura 14-4.	Grafica tiempo vs grado de movimiento junta 2.	74
Figura 15-4.	Grafica tiempo vs grado de movimiento junta 1.	74
Figura 16-4.	Interfaz de teleoperación.	75

ÍNDICE DE ANEXOS

- ANEXO A:** Condiciones de uso para el robot anno RV624
- ANEXO B:** Manual de uso para la teleoperación y monitoreo.
- ANEXO C:** Planos de la bancada.
- ANEXO D:** Toma de tiempos para un desplazamiento de 180°
- ANEXO E:** Velocidad de cada junta a distintos porcentajes.
- ANEXO F:** Tiempos de respuesta para teleoperar y monitorizar
- ANEXO G:** Grafica 1 de dispersión y errores en las mediciones
- ANEXO H:** Grafica 2 de dispersión y errores en las mediciones

RESUMEN

En el presente trabajo de titulación se describe el proceso para el desarrollo de una arquitectura software para teleoperación y monitoreo del robot ANNO RV624. Al realizar una revisión del Software original del brazo robótico, se determinó que este tiene inconvenientes al comunicarse con el brazo y por esto es necesario una interfaz que sea capaz de interactuar con todo tipo de sistema operativo, además de un Software que nos permita realizar una interfaz gráfica que sea capaz de comunicarse con medios exteriores y la necesidad de una bancada que actúe como soporte para el brazo, convirtiéndose en requerimientos para el desarrollo la arquitectura Software para la teleoperación y monitoreo. Para el modelado de la bancada se usó una herramienta CAD conocido como Solid Works, la misma fue aprobada por medio del análisis estático. Como elementos Hardware del sistema se emplearon los mismos disponibles en el brazo. Para la implementación de la interfaz y carga de los algoritmos, se empleó Visual Basic, también se encarga de emitir las señales o instrucciones hacia el controlador y del enlace entre plataforma IoT Ubidots y el controlador, este es el encargado de procesar la información recibida y transmitirla hacia los drivers de los actuadores eléctricos. Al integrar las DLLs se logró el monitoreo y teleoperación del brazo desde cualquier dispositivo con conexión a internet y poder manipular el mismo desde cientos de kilómetros de distancia. Además, de que no hubo la necesidad de elementos Hardware ajenos a los disponibles en el controlador. Una vez implementado el prototipo de interfaz se obtuvo una funcionalidad del 100% de los datos enviados en relación de movimiento con cada uno de los actuadores, mientras que con la teleoperación y monitoreo se obtuvo un tiempo de 8.37 y 2.62 segundos respectivamente en llegar los datos desde su accionamiento.

Palabras clave: <UBIDOTS>, <SOFTWARE>, <INTERFAZ>, < TELEOPERACIÓN >, <ACTUADORES>

**LUIS ALBERTO
CAMINOS
VARGAS**

Firmado digitalmente por LUIS
ALBERTO CAMINOS VARGAS
Nombre de reconocimiento (DN):
c=EC, I=RIOBAMBA,
serialNumber=0602766974, cn=LUIS
ALBERTO CAMINOS VARGAS
Fecha: 2021.03.31 12:24:39 -05'00'



0899-DBRAI-UTP-2021

SUMMARY

In this degree work, the process for the development of a software architecture for teleoperation and monitoring of the ANNO RV624 robot is described. After reviewing the original software of the robotic arm, it was determined that it has problems when communicating with the arm and therefore it is necessary an interface that is able to interact with all types of operating systems, in addition to software that allows us to make a graphical interface that is able to communicate with external media and the need for a bench that acts as a support for the arm, becoming requirements for the development of the software architecture for teleoperation and monitoring. A CAD tool known as Solid Works was used for the modeling of the bench, which was approved by means of static analysis. As hardware elements of the system, the same as those available in the arm were used. For the implementation of the interface and loading of the algorithms, Visual Basic was used, which is also responsible for issuing the signals or instructions to the controller and the link between the IoT Ubidots platform and the controller, which is responsible for processing the information received and transmitting it to the drivers of the electric actuators. By integrating the DLLs, it was possible to monitor and teleoperate the arm from any device with an internet connection and to manipulate it from hundreds of kilometers away. In addition, there was no need for hardware elements other than those available in the controller. Once the interface prototype was implemented, 100% of the data sent in relation to the movement of each of the actuators was 100% functional, while the teleoperation and monitoring obtained a time of 8.37 and 2.62 seconds, respectively, to receive the data from the actuator.

Keywords: <UBIDOTS>, <SOFTWARE>, <INTERFACE>, <TELEOPERATION>, <ACTUATOR>

INTRODUCCIÓN

La acelerada digitalización de la industria y la sociedad está transformando los métodos de producción y la forma en que trabajamos en general. Representa la profunda conexión entre todas las áreas de la industria y la sociedad. (Standardization, 2018, p. 2)

La robótica como parte de la automática es hoy un área de intensa investigación en todos sus campos. Asimismo, en la industria ocupa un lugar destacado en el proceso de modernización e innovación productiva. La incorporación de robots en industrias de fabricación resulta en productos de mejor calidad y menor precio. Para ello, el diseño de una estructura de software abierta tiene una importancia relevante. (Sławiński et al, 2007)

Generalmente al hacer la adquisición de un brazo robótico, se lo hace con un software propio la cual trae algunas limitantes para el momento de realizar alguna modificación en la interfaz, los brazos robóticos no cuentan con teleoperación, por lo cual es necesario la presencia de un operador para que pueda ser manipulado.

El trabajo de titulación se enfoca en la elaboración de una Arquitectura Software para la manipulación del brazo robótico ANNO Rv624, la misma que contará con el código de programación abierta para nuevas ideas y dar más valor agregado tanto en la mejor de la programación como en la interfaz, además se realiza una interfaz para su teleoperación brindando así la facilidad de realizar la operación desde cualquier otro punto que no sea el laboratorio de la Facultad de Mecánica, de esta manera se incluye otro método de operación para el brazo, esto es controlado por un sistema del software de Visual Studio que por medio de una programación muestra la interfaz gráfica para el operador.

CAPITULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1 Antecedentes

Para el desarrollo del presente proyecto técnico cuyo fin es el desarrollo de una arquitectura software para teleoperación y monitoreo del robot ANNO Rv624, es necesario analizar investigaciones previas realizadas que contemplen similitud con el tema a tratar. A continuación, se citan las investigaciones analizadas.

- En una primera investigación elaborada por (Terán, D. 2015, p. 139) denominado “Diseño e implementación de un sistema de teleoperación y monitoreo para la estación de medición del grado de asentamiento de sedimentos y agua del petróleo”

La implementación de este sistema teleoperado, prueba que estos sistemas reaccionan como cualquier otro sistema cableado, lo cual permite hacer sistemas con mayor cobertura, ya que al manejar comunicaciones a través de protocolo UDP no limita las distancias. El sistema forma una plataforma rápida en tiempo de transmisión de datos entre la estación y el HMI. Según el presente trabajo, se creó un software que permite la supervisión, control y monitoreo en tiempo real, para lo cual se desarrolló un interfaz humano máquina (HMI), la cual está desarrollada en LabVIEW mediante un lenguaje gráfico, gracias a la implementación del software, el sistema automatiza el proceso de medición del grado BS&W del petróleo forma eficiente y de manera remota, sin necesidad de la presencia de un operador en el sitio donde se encuentre la estación.

- En una segunda investigación realizada por (Llumiquinga, C., & Galarza, D., 2018, p. 66) titulada “Teleoperación del Robot Nao para la ejecución de tareas en espacios reducidos usando el dispositivo Kinect v2”

La mejor técnica para intercambiar flujo de datos entre la aplicación realizada en Visual Studio y Python fue mediante protocolo TCP/IP utilizando puerto Socket ya que su implementación requiere de una arquitectura clientes servidores para que la información sea transmitida en toda la red una vez que se establece la comunicación.

En el siguiente trabajo enfocado al internet de las cosas tienen como finalidad la teleoperación de un Robot Nao para la ejecución de tareas en espacios reducidos. El algoritmo usado para teleoperar el robot fueron hechos con los sensores Kinect v2, éste es

un controlador de videojuegos que permite la comunicación entre el usuario y la consola sin necesidad de contacto físico.

- En una tercera investigación realizada por (Hernández, K., & Panchi, F, 2017, p. 165) titulada “Diseño, simulación e implementación de un sistema de teleoperación para un robot móvil tipo Carlike”

Los sistemas teleoperados son maniobrables, pero sus cualidades se ven opacadas por las limitaciones mismas del canal de comunicaciones que no es dedicado como es el internet y también la influencia de las distancias físicas. La eficiencia de un sistema teleoperado también depende la complejidad de las tareas que se requieren.

En el proyecto que se declara, la estación local es una interfaz hombre máquina (HMI) desde el cual, el operador podrá monitorear la estación remota y también comandarla, para la implementación de la estación local se utilizó un GUIDE en Matlab.

- En una cuarta investigación realizado por (Quevedo, W. 2016, p. 71) titulada “Implementar un esquema de teleoperación para un robot manipulador móvil maniobrado a través de dispositivos hápticos, para incrementar la transparencia del sitio remoto, a través de entornos de realidad virtual y realidad aumentada, en el sitio local”

Los resultados mostraron la factibilidad y el buen desempeño del esquema de teleoperación propuesto para realizar tareas complejas en un entorno remoto con un robot manipulador móvil utilizando dos modos de operación: de locomoción, en el cual el operador humano controla el manipulador móvil enviando comandos de velocidad al extremo operativo del robot; y de manipulación, en el cual el operador humano controla el brazo robótico enviando comando de posición al extremo operativo.

En el presente trabajo se ha realizado sistemas que permiten a un operador humano llevar a cabo complejas tareas en entornos remotos. Para el esquema el operador debe estar inmerso en un entorno de realidad aumentada para tener una mayor transparencia del sitio remoto.

- En una quinta investigación realizada por (Burbano, L., 2017, p. 213) en su trabajo titulado “Estudio e implementación en Matlab de un entorno de comunicación basado en protocolos del internet de las cosas para clientes de teleoperación en robótica”

La utilidad de este esquema de comunicación generado a partir de la adaptación de diferentes protocolos permite la integración de varios entornos, dispositivos y aplicaciones a un esquema común de comunicaciones. Por ejemplo, realizar mediante WebSocket la integración de sistemas a los servicios web y hacia una variedad de interfaces de monitoreo; por otro lado, la integración de Matlab a una variedad de aplicaciones, como también con servicios y desarrollos web.

Hoy en la actualidad se habla mucho acerca del internet de las cosas, las empresas manufactureras e industriales, se enfrentan al reto de buscar e implantar nuevas técnicas organizativas y de producción que les permitan competir en un mercado global. El proyecto denominado estudio e implementación en Matlab de un entorno de comunicación basado en protocolos del internet de las cosas para clientes de teleoperación en robótica, muestra un entorno de comunicación en Matlab, basado en protocolos de Internet de las Cosas (IoT). El diseño se sustenta en los requerimientos de la Teleoperación de robots, es decir el control remoto de los mismos.

1.2 Planteamiento del problema

Con la evolución de la industria, los robots en sus inicios fueron usados para trabajos muy simples, sin explotar al máximo la capacidad con la que contaban, con el paso del tiempo este uso se ha ido actualizando y tecnificando en todo el mundo ya que un robot es muy versátil por sus usos en varios campos de la industria. Hoy en día la adquisición de un robot industrial es realmente accesible, con la ayuda del internet podemos escoger y realizar la compra desde la plataforma de nuestro proveedor de preferencia.

Al hacer la adquisición de un robot se lo debe hacer con un software y sistema operativo propio, esto hace que el costo de la adquisición sea elevado. Para hacer el uso del robot es necesario que el operador se familiarice con el software que en muchos de los casos vienen con un único idioma de interacción dependiendo del país en la sea adquirida, el software que es proporcionado por el fabricante en muchos casos sirve únicamente para configuraciones locales. En este sentido, el principal problema que aqueja a la parte industrial es la falta de un software propio que se adapte a la teleoperación y control de un robot, apropiada para el proceso al que se va a destinar, por lo tanto, es prioritaria la elaboración de un software para teleoperación y monitoreo propio.

El Grupo de Investigación de la Facultad de Mecánica – ESPOCH “Tecnologías de la información, comunicación y procesos industriales AUTOPRO” no cuenta con un software propio que sea capaz de adaptarse a los sistemas de teleoperación y monitores de los brazos robot, que se capaz de reducir los costos de adquisición y ampliar las formas de configuraciones.

Es por ello por lo que el presente trabajo considera importante estudiar y desarrollar un software para la teleoperación y monitoreo de un robot, la misma que pueda minimizar los costos de adquisición, haciendo posible que solo sea necesario adquirir el robot para adaptarlo al software de teleoperación y monitoreo nuestro. También abre la posibilidad de que el robot sea construido en el país, para luego vincularlo al software de teleoperación y monitoreo propio. Se aplicarán los

parámetros necesarios con el fin de mantener el robot en la máxima disponibilidad y confiabilidad, garantizando así una funcionabilidad óptima.

1.3 Justificación

La teleoperación de robots implica manejar un robot a distancia para realizar alguna tarea dada. De esta forma se permite que un operador humano pueda “transportar” su capacidad y destreza hacia ambientes de trabajos remotos y/o peligrosos, minimizando los posibles riesgos asociados al ambiente, o más aún, se puede alcanzar lugares inaccesibles para el hombre. (Gómez et al, 2006)

La innovación en la industria va mejorando cada día más, y así hace necesario la inserción de recursos robóticos a los procesos y así optimizar los mismo. En base a lo antes expuesto, el presente trabajo se llevará acabo con el objetivo principal de dar una alternativa para la innovación y manejo del Robot ANNO RV624, mediante la elaboración de un software que ayudará en la teleoperación y monitoreo. Esto permitirá dar la opción de elaborar el robot en empresas nacionales y adaptarlo al software para su interacción. Lo que hace posible mejorar el proceso de trabajo, teniendo como perspectiva una mayor productividad, menor costo de adquisición y mayor utilidad.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar una arquitectura software para teleoperación y monitoreo del robot ANNO RV624.

1.4.2 Objetivos específicos

- Diagnosticar y evaluar la interacción del operador y el software que incluye el robot.
- Determinar los requerimientos de funcionalidad, hardware y software para la implementación del sistema de teleoperación y monitoreo.
- Implementar una interfaz que se relacione con una plataforma online para la teleoperación y monitoreo del robot.
- Evaluar mediante pruebas de funcionalidad la respuesta del sistema de teleoperación y monitoreo al interactuar con el robot.

CAPITULO II

2. MARCO TEÓRICO

2.1 Historia del robot

Con el pasar de los siglos los humanos han creado máquinas similares a las partes del cuerpo humano y que tenga la capacidad de imitar la figura y movimientos de un ser animado. En la Grecia antigua se construían estatuas que se operaban con sistemas hidráulicos. Los antiguos egipcios unían brazos mecánicos a las estatuas de sus dioses, estos eran activados por sacerdotes aclamando que el movimiento de éstos eran inspiración de sus dioses. En 1921 la palabra robot fue usada por primera vez por el escritor checo Karel Capek al estrenar en el teatro nacional de Praga su obra Rossums Universal Robot (R.U.R). (Barrientos et al, 2007, pp. 5-8)

En octubre de 1945 las tres leyes de la robótica fueron enunciadas por Isaac Asimov en la revista Galaxy Science Fiction.

- Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
- Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
- Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley.

1954 la transmisión mecánica fue sustituida por otra eléctrica haciendo uso de la tecnología electrónica y del servocontrol y desarrollando así el primer telemanipulados con servocontrol bilateral. En este mismo año la primera patente de un dispositivo robótico fue solicitada por el inventor británico C. W. Kenward.

1958 se desarrolló el dispositivo denominado Handy-Man, consiste en dos brazos mecánicos teleoperados mediante un maestro del tipo denominado exoesqueleto.

1961 en estado Unidos se patentó el dispositivo de transferencia de artículos programada.

1968 tras la firma de acuerdos con Kawasaki, inició la construcción de robots tipo Unimate. El crecimiento de la robótica en Japón aventajó en breve a los Estados Unidos gracias a Nissan.

1972 se firmó la primera asociación robótica del mundo, la Asociación de Robótica Industrial.

1973 se construyó el primer robot con accionamiento totalmente eléctrico, el robot IRb6

1974 se formó el Instituto de Robótica de América (RIA), la misma que en 1984 cambió su nombre por el de Asociación de Industrias Robóticas, bajo las mismas siglas (RIA).

1982 se desarrolla el concepto de robot SCARA (Selective Compliance Assembly Robot Arm) la cual busca un robot con un número reducido de GDL (3 o 4).

Los años siguientes los desarrollos de la robótica han aumentado su movilidad, destreza y autonomía de sus acciones. En la actualidad la mayor parte de los robots son con base estática, y son utilizadas en aplicaciones industriales, así como en otros tipos de aplicaciones que han hecho evolucionar en gran medida tanto la concepción de los robots como su propia morfología.

Gracias al avance de la tecnología se espera una gran mejora en el diseño, aplicación y morfología de los robots.

2.2 Tipos de robots

Gracias a los avances de la tecnología en la actualidad, diferentes tipos de robots se usan en diversas áreas mientras se busca ir mejorando continuamente su diseño, investigación y eficiencia para ser usados tanto en el ámbito doméstico, comercial, médico, militar y por supuesto como ayuda en las áreas de matemática, tecnología, ingeniería y ciencia. (Slawiński et al, 2007, pp. 1-10)

2.2.1 Robots Industriales

Los robots industriales son manipuladores diseñados para mover materiales, partes y herramientas y realizar una serie de tareas programadas en ambientes de producción y manufactura.



Figura 1-2. Robot industrial KR QUANTEC
Fuente: (Revistaimg, 2019)

2.2.2 Robots militares

Son robots autónomos o manejados a través de control remoto que han sido diseñados para aplicaciones militares como transporte y búsqueda, o rescate y ataque. En esta clasificación podemos encontrar diversos tipos de drones, en especial de espionaje y recolección de datos e imágenes



Figura 2-2. Robot militar Goga Shutterstock
Fuente: (Rigor, 2019)

2.2.3 Robots de entretenimiento

Este tipo de robots son algunos de los más sofisticados, con un diseño de alto desempeño, pero además sensibilidad y gracia para interactuar con las personas. Podemos encontrar desde robots utilizados como juguetes hasta robots que ayudan a la enseñanza de conocimientos.



Figura 3-2. Robot Sony Aibo
Fuente: (ELCOMERCIO, 2017)

2.2.4 Robots en la industria médica

Estos robots son utilizados en medicina e instituciones médicas como hospitales, centros de rehabilitación, clínicas, centros odontológicos u oftalmológicos, entre otros.

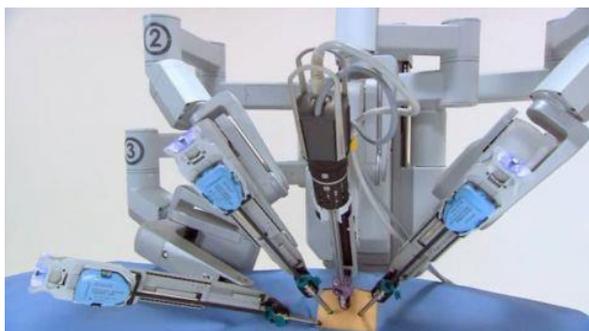


Figura 4-2. Araña robótica
Fuente: (PortalPolitico, 2018)

2.2.5 Robots de servicio

Los robots de servicio son un enorme salto en mejorar la productividad en prácticamente cualquier tarea. Aquí encontramos la posibilidad de automatizar todo tipo de trabajo que requiere eficiencia y mayor rapidez, como tomar una orden en un restaurante o llevar servicio a la habitación en un hotel.



Figura 5-2. Robot aspirador IROBOT ROOMBA 606
Fuente: (Showroom, 2019)

2.2.6 Robots espaciales

La Administración Nacional de Aeronáutica y Espacio, NASA, utiliza diversos tipos de robots para misiones en el espacio. Algunos de ellos son usados en exploración de terrenos y ambientes como Marte o la Luna.



Figura 6-2. Robot Mary Anning
Fuente: (Woollaston, 2020)

2.2.7 Robot Manipulador

Un robot manipulador consiste en una estructura mecánica con una secuencia de partes sólidas interconectadas mediante articulaciones con lo cual forman un brazo robótico para la movilidad y un efector para la ejecución de tareas requeridas mediante la programación que se le haya desarrollado. Su constitución puede asemejarse a la anatomía de las extremidades superiores del cuerpo humano.

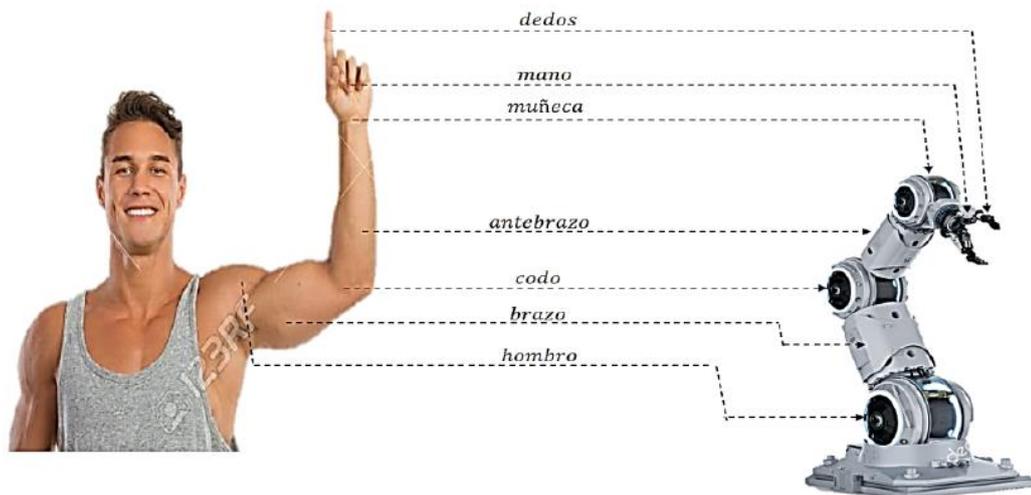


Figura 7-2. Comparativa brazo robótico y brazo humano
Fuente: (Muela Guaicha, 2020)

2.3 Actuadores en los Robots

Los actuadores son los músculos de los robots y comprenden lo siguiente:

- Una fuente de alimentación.
- Un amplificador de potencia.

- Un servomotor.
- Un sistema de transmisión.

2.3.1 Actuadores Hidráulico

Los actuadores hidráulicos también son dispositivos de potencia de fluidos para robots industriales. Ellos utilizan fluidos de alta presión, como aceite, para transmitir fuerzas al punto de aplicación deseado. Un actuador hidráulico es muy similar en aspecto a uno accionado neumáticamente. Estos actuadores están diseñados para operar con presiones mucho más altas (normalmente entre 70 y 170 bar). Son apropiados para aplicaciones de alta potencia.

2.3.2 Actuadores Neumáticos

El actuador neumático es uno de los dos tipos de dispositivos de potencia dada por fluidos para robots industriales, el otro es hidráulico. Los actuadores neumáticos utilizan aire comprimido para su accionamiento y se usan extensamente para realizar los movimientos típicos de abrir y cerrar las mordazas en el sujetador (gripper) de un robot, o bien para su accionamiento de brazos de robot sencillos en aplicaciones donde el control continuo de movimientos no sea una preocupación. Las especificaciones de estos sistemas son empleadas, por ejemplo, en los sujetadores de robots.

2.3.3 Actuadores Eléctricos

Por lo general, se habla de los actuadores eléctricos como aquéllos donde un motor eléctrico acciona los eslabones del robot por medio de alguna transmisión mecánica como por ejemplo los engranes. En los primeros años de la robótica industrial, los robots hidráulicos eran los más comunes, pero las recientes mejoras en el diseño de motores eléctricos implicaron que la mayoría de los robots nuevos fueran de construcción completamente eléctrica. El primer robot comercial de accionamiento eléctrico fue introducido por ABB en 1974.

2.3.3.1 Motores de Paso a Paso

Estos motores son empleados en algunos robots en el más pequeño y término medio del rango industrial y con los robots de enseñanza y pasatiempo. También son ampliamente usados en otras aplicaciones industriales y tienen las ventajas de que no necesariamente requieren un sistema de retroalimentación y de los costos asociados. Sin embargo, son compatibles con muchos dispositivos de retroalimentación, si esto se requiriera, y son usados en configuraciones de servocontrol total en robots industriales de servicio medio. Puesto que se trata de motores de control digital, no necesitan el gasto adicional de equipos de conversión de digital a analógica cuando se conectan a un sistema de control por computadora. Normalmente, la flecha del motor

gira en forma incremental en pasos iguales en respuesta a un tren de impulsos de entrada programado. Como el rotor indexa una cantidad específica para cada pulso de control, todo error de posicionamiento es no acumulativo. Para conocer la posición final del rotor, todo lo que se requiere es contar el número de pulsos transmitidos al devanado de fase de la bobina del estator. El número de pulsos por unidad de tiempo determina la velocidad del motor. El rotor puede estar hecho para indexar lentamente, haciendo una pausa después de cada incremento, o puede moverse rápidamente generando un movimiento continuo que se llama “slewing” (hasta llegar a una velocidad de giro constante). El máximo par torsor dinámico en un motor de paso ocurre con bajas frecuencias de pulsos. Por lo tanto, puede fácilmente acelerar una carga. Una vez que se haya alcanzado la posición requerida y se terminen los impulsos de comando, la flecha se detiene sin necesidad de embragues o frenos. Los momentos de rotación o ángulos de paso reales de la flecha pueden obtenerse normalmente desde 1.8° hasta 90° , dependiendo de la selección particular del motor. Así, con un ángulo de paso nominal de 1.8° , una corriente de 1000 pulsos dará un desplazamiento angular de 1800° o cinco revoluciones completas. También tienen una capacidad de velocidad baja sin necesidad de reducción de engranaje. Por ejemplo, si el motor mencionado anteriormente es accionado por 500 pulsos por segundo, girará a 150 RPM. Otra ventaja del motor a pasos es que la inercia del motor a menudo es baja, y también si más de un motor a pasos es accionado desde la misma fuente éstos se mantendrán perfectamente sincronizados. Algunas desventajas son que los motores a pasos tienen menor salida y eficiencia en comparación con otros, y que las entradas y circuitos de accionamiento tienen que estar cuidadosamente diseñados en relación con el par torsor y velocidades requeridas. Hay varios tipos de motores a pasos, por ejemplo, los de reluctancia variable, los de imanes permanentes y los híbridos. Existen tres tipos de motores a paso:

- **Motor a pasos de reluctancia variable:** “Reluctancia magnética” o simplemente “reluctancia” es el término análogo de resistencia eléctrica. Igual que una corriente sólo sucede en un lazo cerrado, un flujo magnético ocurre sólo alrededor de una trayectoria cerrada, aunque tal vez ésta pueda ser más variada que el de una corriente.
- **Motor a paso de imanes permanentes:** El método básico de operación de un tipo de motor de imanes (o magnetos) permanentes es similar al de reluctancia variable, hay dos bobinas, A y B, cada una de las cuales produce cuatro polos, pero que están desplazadas una de la otra por medio paso polar. El rotor es de construcción de imanes permanentes y tiene cuatro polos, ilustra. Cada polo está enrollado con devanado de campo y las bobinas que hay en los pares opuestos de polos están en serie. La corriente es suministrada desde una fuente de CD hasta el devanado a través de interruptores.
- **Motor a pasos híbridos:** Los motores a pasos híbridos combinan las características de los motores de reluctancia variable y de imanes permanentes, teniendo un imán permanente

interconectado en cubiertas de hierro que están cortadas para tener dientes, El rotor se coloca en la posición de reluctancia mínima en respuesta a que se estén energizando un par de bobinas del estator. Los ángulos de paso normales son 0.9° y 1.8° . De la descripción anterior resulta entonces obvio que la velocidad con la que se aplican los impulsos determina la velocidad del motor, el número total de impulsos determina el desplazamiento angular y el orden en el que se energizan las bobinas en la primera instancia determina la dirección de rotación.

2.3.3.2 Motores de Corriente Directa

Aunque mucha gente podría nunca haber tenido suficiente experiencia con sistemas neumáticos o hidráulicos, todos han tenido contacto casi diario con motores eléctricos. Ya sean motores CD (frecuentemente operados por baterías) para arrancar el auto u operar un juguete de niños, o motores CA (operados por la red eléctrica) para encender la licuadora, etc., sus capacidades son bien conocidas. El primer robot industrial y comercial eléctricamente accionado fue introducido en 1974 por la gigantesca corporación sueca ASEA. Tradicionalmente, los expertos en robótica empleaban motores de DC (corriente directa) eléctricamente accionados para robots no sólo por la disponibilidad de versiones potentes, sino porque son fácilmente controlables con dispositivos electrónicos relativamente sencillos.

2.3.3.3 Motores de Corriente Alterna

Los motores CA no han sido extensamente empleados en aplicaciones robóticas debido a los problemas que surgen en el control de velocidad. En principio, entre más alta sea la frecuencia de la corriente alterna aplicada al motor, más rápidamente girará. Hasta hace poco, en gran medida fue muy impráctico suministrar frecuencias variables en forma simultánea a un número de accionamientos de ejes. (Saha, 2008, pp. 38-49)

2.4 Visual Basic .NET

Visual Basic es uno de los lenguajes de programación de alto nivel que pertenecen al paquete .NET (otros lenguajes son C#, C/C++, etc.). Con Visual Basic .NET se pueden escribir tanto programas convencionales como para Internet. Las aplicaciones podrán mostrar una interfaz gráfica al usuario, o bien una interfaz de texto, como hacen denominas aplicaciones de consola.

Programa: Un programa no es más que una serie de instrucciones dadas al ordenador en un lenguaje entendido por él, para decirle exactamente lo que queremos que haga. Si el ordenador no entiende alguna instrucción, lo comunicará generalmente mediante mensajes visualizados en la pantalla. (Ceballos, 2017, pp. 4-10)

Ventanas, eventos y mensajes

Pero para entender el proceso de desarrollo de una aplicación debe familiarizarse con el entorno Windows, el cual incluye tres conceptos clave: ventanas, eventos y mensajes.

Una ventana es una región rectangular con sus propios límites. Hay varios tipos de ventanas:

- De Explorador de Windows
- De documento dentro de su programa de proceso de textos
- Un cuadro de diálogo que emerge para recordarle una cita
- Controles como: un botón, los iconos, cuadros de texto, botones de opción, barras de menús.

Un evento es una acción reconocida por un formulario o un control. Los eventos pueden producirse mediante acciones del usuario, como hacer clic con el mouse (ratón) o presionar una tecla, mediante programación o incluso como resultado de acciones de otras ventanas.

Cada vez que se produce un evento se envía un **mensaje** al sistema operativo. El sistema procesa el mensaje y lo transmite a las demás ventanas. Entonces, cada ventana puede realizar la acción apropiada, basándose en sus propias instrucciones para tratar ese mensaje en particular (por ejemplo, volverse a dibujar cuando otra ventana la ha dejado al descubierto). (Ceballos, 2013)

2.4.1 Requisitos previos

Visual estudio es un entorno de desarrollo que puede resultar extremadamente potente, pero para sacar el máximo partido de sus funcionalidades es preciso que la máquina sobre la que está instalado disponga de una mínima configuración de hardware y software. (Putier, 2018, pp. 41, 42)

Tabla 1-2: Especificaciones para el equipo.

Sistema operativo	Windows 7 SP1 Windows 8.1 (con Update 2919355) Windows Server 2012 R2 (con Update 2919355) Windows 10 (versión 1507 y +) Windows Server 2016
Procesador	Procesador de 1,8 GHz

Memoria RAM	2 GB de RAM (2,5 GB en caso de ejecución sobre una máquina virtual)
Espacio en disco disponible	De 1 a 40 GB, en función de las funcionalidades instaladas
Video	Tarjeta de video que soporte una resolución de 1280x768

Fuente: (PUTIER, 2018)

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

2.4.2 Lenguaje de programación C#

C# (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el **lenguaje nativo de .NET**.

2.4.3 Características de C#

Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.

- El código escrito en C# es **autocontenido**, lo que significa que no necesita de ficheros adicionales a la propia fuente tales como ficheros de cabecera o ficheros IDL
- El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::).

Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico **decimal** que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción **foreach** que permita recorrer colecciones con facilidad y es ampliable a tipos

definidos por el usuario, la inclusión de un tipo básico **string** para representar cadenas o la distinción de un tipo **bool** específico para representar valores lógicos.

Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales, sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente **propiedades** (similares a campos de acceso controlado), **eventos** (asociación controlada de funciones de respuesta a notificaciones) o **atributos** (información sobre un tipo o sus miembros).

Gestión automática de memoria: Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, dado que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active –ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en el fuente–, C# también proporciona un mecanismo de liberación de recursos determinista a través de la instrucción **using**.

Seguridad de tipos: C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

Instrucciones seguras: Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un **switch** ha de terminar en un **break** o **goto** que indique cuál es la siguiente acción para realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.

Sistema de tipos unificado: A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada

System.Object, por lo que dispondrán de todos los miembros definidos en esta clase (es decir, serán “objetos”)

Extensibilidad de tipos básicos: C# permite definir, a través de **estructuras**, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia.

Extensibilidad de operadores: Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos.

Compatible: Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados **Platform Invocation Services (PInvoke)**, la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. (González, 2002, pp. 21-26)

2.4.4 Identificación de variables y constantes

Las variables se dividen en enteros, reales y booleanos. Las variables enteras contienen números enteros, las variables reales contienen números reales y las variables booleanas contienen dos estados. (verdadero o falso)

En todos los lenguajes de programación el primer paso es la definición de las variables que haces que la computadora pueda reconocerlas. La variable es como un contenedor y contiene lo que indicamos. (Cuccaro, 2021)

Lógica de programación

Las variables: son de vital importancia para la programación. Después de haberlos identificado, se debe definir al inicio del programa. Las variables contienen datos y tienen un nombre único. En cambio, la asignación consiste en dar un cierto valor a la variable.

El valor asignado a la variable varía durante el programa, mientras que el valor de una constante permanece igual a lo largo del programa.

Tabla 2-2: Asignación de una variable.

Boolean	Puede contener valores de álgebra booleana (verdadero o falso).
Byte	Puede contener valores naturales.
Currency	Puede contener valores monetarios incluidos 4 lugares decimales.
Date	Puede contener información sobre la fecha y la hora incluida.
Double	Puede contener valores decimales de doble precisión
Integer	Puede contener valores naturales firmados
Long	Puede contener valores naturales con signo con respecto al rango de cambio anterior.
Object	Puede contener una referencia al objeto.
Single	Puede contener valores decimales de precisión simple.
String	Puede contener cadenas alfanuméricas.
Variant	Puede contener cualquier tipo.

Fuente: (Cuccaro, 2021)

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Alcance de las variables Cuando declara una variable en un procedimiento, sólo el código de dicho procedimiento puede tener acceso o modificar el valor de la variable; tiene un alcance que es local al procedimiento. A veces, sin embargo, se necesita usar una variable con un alcance más general, como aquella cuyo valor está disponible para todos los procedimientos del mismo módulo o incluso para todos los procedimientos de toda la aplicación, se conocen como globales. Visual Basic le permite especificar el alcance de una variable cuando la declara.

El alcance de una variable define qué partes del código son conscientes de su existencia, es decir, el espacio de la aplicación donde la variable es visible y por lo tanto se puede utilizar.

Las condiciones: se utilizan para realizar operaciones complejas que deben cumplir ciertos requisitos y tomar dos caminos diferentes. La condición se expresa con SI... entonces... lo demás significa: SI (condición) entonces... De lo contrario...

2.5 Teleoperación

La teleoperación hace referencia al trabajo que se puede realizar a distancia, es decir, el elemento humano se encuentra separado del robot. Teleoperar es permitir a un operador en un lugar específico ejecutar una tarea en otro lugar, posiblemente separados por grandes distancias; mientras que, un sistema de teleoperación permite que cualquier dispositivo sea teleoperado por un elemento humano. (Alencastres et al, 2003, p. 4)

Según (Quevedo, 2016, pp. 11-13) un sistema de teleoperación consta de los siguientes elementos:

- Operador o teleoperador: es el encargado de realizar a distancia el control de la operación. Su acción puede ir desde un control continuo hasta una intervención intermitente, con la que únicamente se ocupa de monitorizar y de indicar objetivos y planes cada cierto tiempo.
- Dispositivo teleoperado: podrá ser un manipulador, un robot, un vehículo o dispositivo similar. Es la máquina que trabaja en la zona remota y que está siendo controlada por el operador.
- Interfaz: conjunto de dispositivos que permiten la interacción del operador con el sistema de teleoperación. Se considera al manipulador maestro como parte del interfaz, así como a los monitores de vídeo, o cualquier otro dispositivo que permita al operador mandar información al sistema y recibir información de este.
- Control y canales comunicación: conjunto de dispositivos que modulan, transmiten y adaptan el conjunto de señales que se transmiten entre la zona remota y la local. Generalmente se contará con uno o varias unidades de procesamiento.
- Sensores: conjunto de dispositivos que recogen la información, tanto de la zona local como de la zona remota, para ser utilizada por el interfaz y el control.

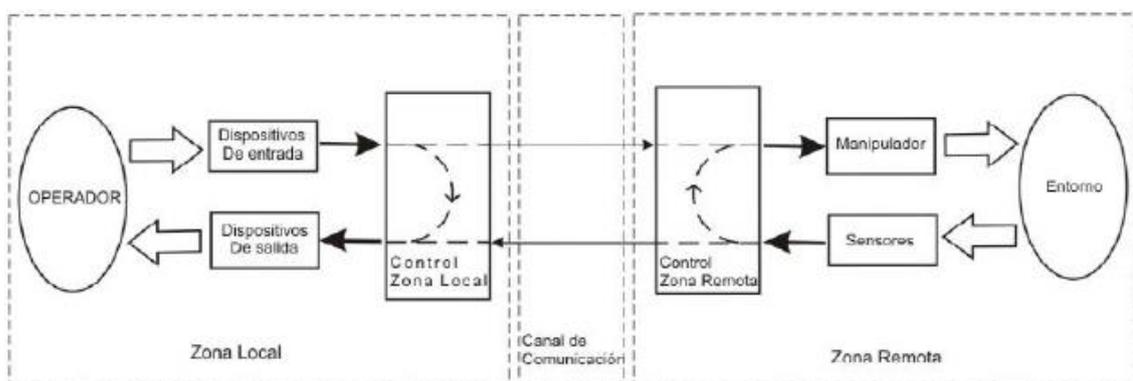


Figura 8-2. Elementos de un sistema teleoperado

Fuente: (Bogado Torres, 2007)

2.5.1 Retardos en teleoperación

La teleoperación presenta la mayor dificultad al momento del envío y recepción de datos entre el maestro y el dispositivo, causando una inestabilidad en el sistema teleoperado. El retardo en la teleoperación se produce principalmente por dos factores.

- Retardo por protocolo de comunicación
- Retardo por velocidad de transferencia de datos en la red

2.5.1.1 Retardo por Protocolo de Comunicación

Características de un protocolo de comunicación

Solicitud: Elemento que es emitido por el usuario para solicitar algún servicio y proporcionar los parámetros necesarios.

Indicaciones: Elemento que emite el proveedor del servicio para indicar que ha sido invocado un procedimiento por el usuario del servicio.

Respuesta: Elemento que el usuario del servicio emite para confirmar o completar algún procedimiento invocado previamente mediante una indicación a ese usuario.

Confirmación: Elemento que el usuario de servicio emite para confirmar o completar algún procedimiento iniciado previamente mediante una solicitud por parte del usuario del servicio.

2.5.1.2 Retardo por Velocidad de Transferencia de Datos en la Red

- **El ancho de banda en la red:** Es aquel que determina la cantidad de datos que pueden ser transportados en un tiempo determinado, en consecuencia, la velocidad.
- **La cantidad de usuarios:** Se refiere a la cantidad de usuarios que envían información por una red magnifica el tráfico de datos que exista en dicha red, y esto genera un mayor retardo en la comunicación.
- **La distancia física:** Esta influye de manera directa en el retardo, debido a que si la distancia es mayor se necesitan más dispositivos dentro del canal de comunicación.

2.6 Internet de las cosas

Es una red de objetos físicos conectados a través de internet, los cuales logran interactuar vía sistemas embebidos, redes de comunicación, mecanismos de computación de respaldo y aplicaciones típicamente en la nube. Permite a los objetos comunicarse entre sí, acceder a

información de internet, capturar, almacenar y recuperar datos e interactuar con usuarios humanos, así como con otros sistemas y aplicaciones. (Quiñonez, 2019)

2.7 La arquitectura de tres capas

Cada arquitectura cumple con funciones específicas y disponen sus propios protocolos que son lenguajes de comunicación entre las capas, constan de tres capas que son Percepción, capa Red y capa Aplicación. (Bahamonde, 2020)

- **Capa de Percepción:** Es donde se ubican sensores, las cuales se encargan de recoger la información del entorno físico.
- **Capa de Red:** se encarga de conectar los elementos a elementos inteligentes, o bien a elementos que se puedan conectar a la red o servidores.
- **Capa de Aplicación:** Es la capa de aplicaciones del usuario permite ver aplicaciones domésticas y logísticas para las empresas, que optimicen los recursos y el tiempo de procesado

2.8 Software de Programación

2.8.1 Visual Basic

Es uno de los guajes de programación de alto nivel que pertenecen al paquete .NET (otros lenguajes son C#, C/C++, etc.). Con Visual Basic .NET se pueden escribir tanto programas convencionales como para internet. Las aplicaciones podrán mostrar una interfaz gráfica a los usuarios, o bien una interfaz de texto, como hacen las denominadas aplicaciones de consola.

2.8.2 Matlab

Es una de las muchas sofisticadas herramientas de computación disponibles en el comercio para resolver problemas de matemáticas, tales como Maple, Mathematica y MathCad. A pesar de lo que afirman sus defensores, ninguna de ellas es “la mejor”. Todas tienen fortalezas y debilidades. Cada una permitirá efectuar cálculos matemáticos básicos, pero difieren en el modo como manejan los cálculos simbólicos y procesos matemáticos más complicados, como la manipulación de matrices. Por ejemplo, MATLAB es superior en los cálculos que involucran matrices, mientras que Maple lo supera en los cálculos simbólicos. El nombre de MATLAB es una abreviatura de Matrix Laboratory, laboratorio matricial. En un nivel fundamental, se puede pensar que estos programas son sofisticadas calculadoras con base en una computadora. Son capaces de realizar las mismas funciones que una calculadora científica, y muchas más. Si usted tiene una computadora en su escritorio, descubrirá que usar MATLAB en lugar de su calculadora incluso para la más simple de las aplicaciones matemáticas, por ejemplo, para el balance de su chequera.

En muchas clases de ingeniería, la realización de cálculos con un programa de computación matemático como MATLAB sustituye la programación de computadoras más tradicional. Esto no significa que el lector de no deba aprender un lenguaje de alto nivel como el C++ o Fortran, sino que los programas como MATLAB se ha convertido en una herramienta estándar para ingenieros y científicos. (Moore, 2007)

2.8.3 Arduino

Lo más importante a la hora de entender Arduino es que no es únicamente una placa con un microcontrolador, como se suele creer, Arduino es el resultado de la suma de los siguientes elementos:

- Una placa de hardware libre.
- Un software libre, gratis y multiplataforma.
- Un lenguaje de programación libre.

Con estos tres elementos podemos construir la definición de Arduino. Básicamente sería la siguiente: Arduino es un dispositivo de hardware libre que monta un microcontrolador en una placa de circuito impreso con los elementos necesarios para su funcionamiento y que dispone de un entorno de programación libre junto con un lenguaje de programación propio. (Moreno et al, 2018, pp. 29-31)

2.8.4 Labview

LabVIEW es el acrónimo de Laboratory Virtual Instrument Engineering Workbench. Es un lenguaje y a la vez un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla.

LabVIEW es una herramienta de programación gráfica. Originalmente este programa estaba orientado a aplicaciones de control de instrumentos electrónicos usadas en el desarrollo de sistemas de instrumentación, lo que se conoce como instrumentación virtual. Por este motivo los programas creados en LabVIEW se guardarán en ficheros llamados VI y con la misma extensión, que significa instrumento virtual (Virtual Instruments). También relacionado con este concepto se da nombre a sus dos ventanas principales: un instrumento real tendrá un Panel Frontal donde estarán sus botones, pantallas, etc. y una circuitería interna. (Lájara, 2008, pp. 21, 22)

2.8.5 Ubidots

Nacida como una empresa de servicios de ingeniería en 2012, Ubidots entregó soluciones integrales de IoT en conjunto con su socio y empresa cofundadora Netux , para monitorear,

controlar y automatizar de forma remota los procesos para clientes de atención médica, así como para startups bien financiadas y Fortune 1,000. en el sureste de Estados Unidos y en toda América Latina. Entre 2012 y 2014, Ubidots logró innumerables proyectos conectados a Internet en los sectores de salud, energía / servicios públicos, fabricación, transporte y venta minorista, aprendiendo las muchas pequeñas características de la habilitación de IoT y la nube con las que los expertos en transformación digital no pueden hablar a menos que se hayan ensuciado las manos. el campo. Con una sólida columna vertebral y una firme determinación de convertirse en una empresa dirigida por productos, Ubidots se unió a Boston MassChallenge Accelerator en 2013, convirtiéndose en un start up tecnológico global, ganando apoyo y empresas estadounidenses, y dejando atrás el negocio de servicios locales. Desde el pivote de Ubidots en 2014, Ubidots se ha hecho conocido dentro de los círculos de hardware, software, ingeniería integrada y fabricantes como la plataforma asequible, confiable y más utilizable en el ecosistema de habilitación de aplicaciones de IoT. (Ubidots, 2016)

2.9 Programación en Robots Industriales

La programación de robots industriales tiene como objetivo indicar la secuencia de acciones a realizar mientras se cumple una tarea que el dispositivo se proponga a ejecutar, se centra en la estructuración de sentencias de control definidas para un sistema robótico. (Escandón, 2013, pp. 8)

Principales puntos de acción.

- Movimientos a puntos predefinidos.
- Manipulación de Objetos.
- Interacción y sincronización con el entorno.

2.10 Arquitectura de programación de robots industriales

Dependiendo de las órdenes programadas y enviadas al robot se refieran a los movimientos propiamente dichos de robot, al estado en el que deben permanecer los objetos manipulados o al objetivo general a conseguir, la programación textual puede ser clasificada en tres niveles.

Nivel Robot: En este nivel se deben especificar cada movimiento que el robot deberá realizar para alcanzar su objetivo, así como: velocidad, orientación, activación o desactivación de la herramienta, entre otros. Los algoritmos que el robot seguirá deben ser interpretados fácilmente por los controladores que a su vez darán movimiento a los actuadores.

Nivel objeto: En este nivel de programación la complejidad se reduce ya que las órdenes descritas no están dirigidas al robot como tal, sino que describen el accionar general del sistema vasados únicamente en cómo deben situarse los objetos inmiscuidos en el proceso.

Nivel tarea: El nivel tarea comprendería únicamente una única sentencia que ya no especifique ni los objetos a moverse ni como el robot lo hará sino únicamente especifica cual es el objetivo que el robot deberá cumplir.

2.11 Métodos de programación de robots industriales

El criterio de sistema empleado para indicar la secuencia de acciones a realizar por el robot ilustra de mejor manera los métodos implementados. También establece de forma muy general que un robot puede ser programado de dos formas diferentes el denominado por programación guiada o mediante un procedimiento textual.

2.11.1 Programación por guiado

También conocida como de aprendizaje asistido consiste en una manipulación directa de los actuadores del robot por medio de una maqueta de este o llevándolo a las diferentes posiciones gracias a un controlados de posición embebido, el posicionamiento de los diferentes puntos o trayectorias debe ser registrado por el robot para la posterior repetición.

Una variación resuelve el problema de manipular físicamente el robot, la cual se realiza por medio de un maniquí o un doble que posea la misma configuración del robot, pero es mucho más ligero. Este tipo de programación es muy utilizada por robots de pintura integrados en una etapa final de una línea de fabricación industrial. Esta programación también es conocida como programación guiada pasiva por maniquí.

La variación que utiliza una botonera o mando analógico es quizás la más utilizada, por medio del cual controla los actuadores del robot para llevarlo a los puntos requeridos o por las trayectorias deseadas. A diferencia de la programación guiada, esta no registra automáticamente por el sistema de control del robot, sino que almacenan únicamente los expresados por el programador.

2.11.2 Programación textual

La programación textual es una alternativa a los métodos de programación por guiado, consiste en indicar la tarea al robot por medio del uso de un lenguaje de programación determinado y normalmente específico para cada marca de robot.

CAPÍTULO III

3. METODOLOGÍA

Para la ejecución de una estructura software de teleoperación y monitoreo del robot ANNO Rv624 se siguió la siguiente metodología, lo que primero se hizo fue evaluar la situación actual en lo que se refiere tanto a Hardware como a Software, posterior a esto se hizo un planteamiento de los requerimientos funcionales, hardware, software y estructural a la necesidad de tener una bancada que soporte al brazo robótico, luego se hizo el diseño del modelo el cual consta del diseño estructural, hardware, software, continuando con las respectivas pruebas de funcionamiento y culminando con una validación del tema.

La imagen presentada a continuación detalla de una manera más grafica la metodología utilizada.

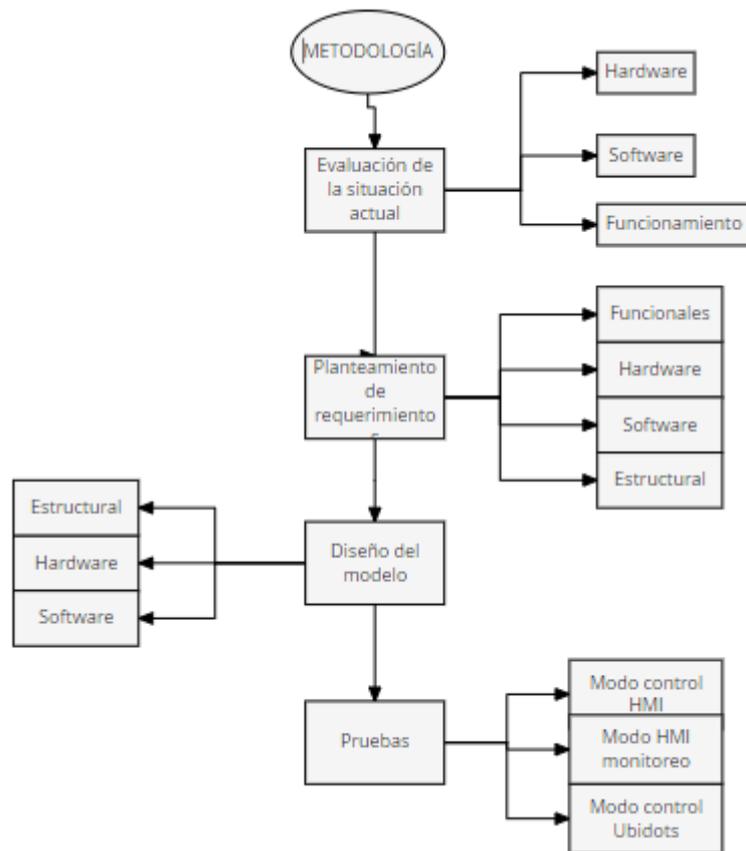


Figura 1-3. Diagrama de flujo de la metodología aplicada
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

3.1 Evaluación de la situación Actual

A continuación, se detalla los elementos encontrados tanto de hardware como de software y su funcionamiento conjunto.

3.1.1 Hardware

Son partes externas del robot además de los componentes que se vincula a él para el correcto funcionamiento, son partes visibles tangibles y que se pueden manipular manualmente.

3.1.1.1 Brazo robótico

Al momento del ensamble el robot adopta una posición que deberá ser depurada para así dar instrucciones para su nuevo posicionamiento.

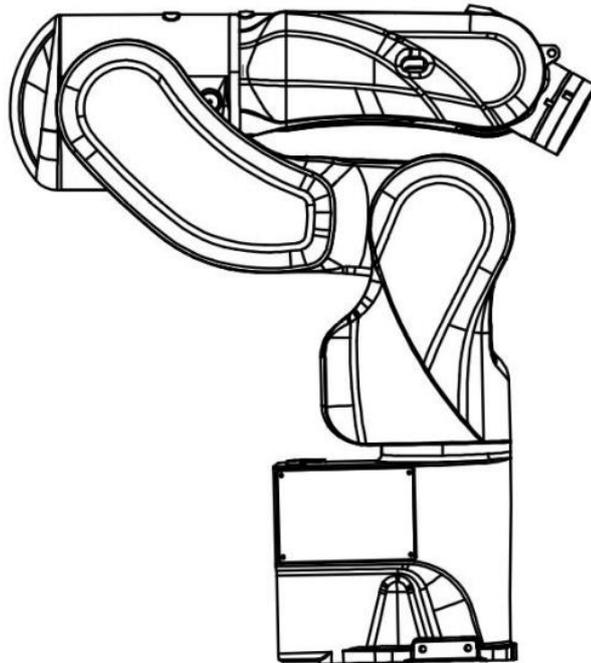


Figura 2-3. Estado inicial después del ensamblado
Fuente: manual usuario

El brazo robótico está compuesto por seis motores paso a paso además de un gripper que está situado en la parte superior haciendo actuar las tenazas de sujeción, cabe recalcar que la estructura del brazo es de aluminio. Cada uno de los motores tienen su denominación que a continuación se lo detallara:

Tabla 1-3: Denominación de los motores paso a paso que contiene el brazo

Parte	Denominación	Características
Motor paso a paso 1	J1	Nema 34, 50w, 1.5-4.4 A, 24V, 500KHz
Motor paso a paso 2	J2	Nema 34, 100w, 1.5-4.4 A, 24V, 500KHz
Motor paso a paso 3	J3	Yizhi 8192P, 50W, 2.2-4.4A, 24-36V, 500KHz
Motor paso a paso 4	J4	Yizhi 8192P, 50W, 2.2-4.4A, 24-36V, 500KHz
Motor paso a paso 5	J5	Yizhi 60AIM25, 7A, 36V, 200W
Motor paso a paso 6	J6	Yizhi 8192P, 50W, 2.2-4.4A, 24-36V, 500KHz

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

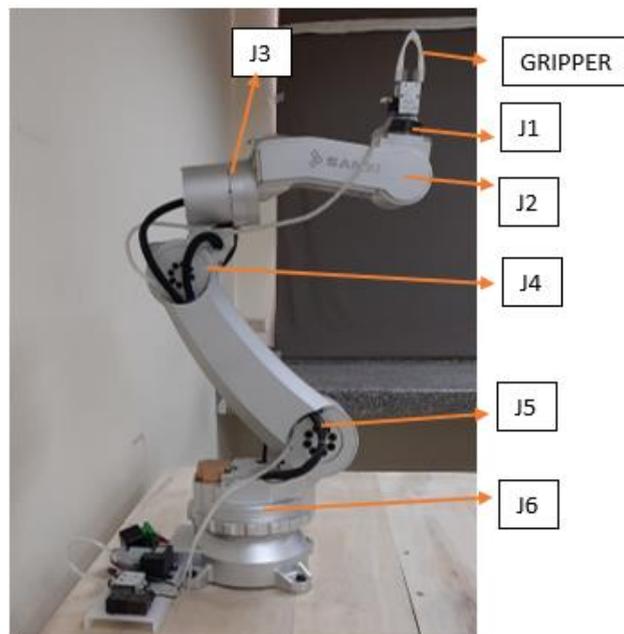


Figura 3-3. Localización de los componentes del brazo

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Al arrancar con el encendido siempre se lo hace teniendo en cuenta las situaciones en las que se encuentra, con un movimiento brusco podría ocasionar algún tipo de accidente.

3.1.1.2 Conjunto controlador

Es un conjunto de elementos electrónicos dispuestos de tal forma para el trabajo coordinado en el correcto envío de señal digital o analógica hacia sus elementos actuadores, en este caso el brazo

robótico. Se podría decir que es la parte principal o el CPU del robot ANNO RV624, el cual está compuesto por los siguientes elementos:

Tabla 2-3: Detalle y especificaciones de los elementos del controlador

Elemento	Detalle
Controlador de movimiento de seis ejes	Controlador de movimiento para el control de robots especialmente con motores de paso a paso o servomotores; marca ANNO dispuesto a ser utilizado con cualquiera de los sistemas operativos.
Fuente	Fuente de poder necesaria para la alimentación de los elementos del controlador.
Placa de relés	Esta placa nos va a permitir controlar varias cargas con un mismo procesador, son mucho más compactas y útiles. Cabe hacer notar que esta parte se encuentra por debajo del regulador de voltaje. (para mejor comprensión en la figura 4-3)
Regulador de Voltaje	Con el regulador de voltaje nos aseguramos de que los picos límites de voltaje causados por las corrientes variables de energía puedan ser evitados y por ende el correcto funcionamiento de los elementos.
Placa de conexión para puerto paralelo	Esta placa nos permite la comunicación entre las señales del procesador y la electroválvula que accionara al gripper del brazo.

Fuente: Manual del robot

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

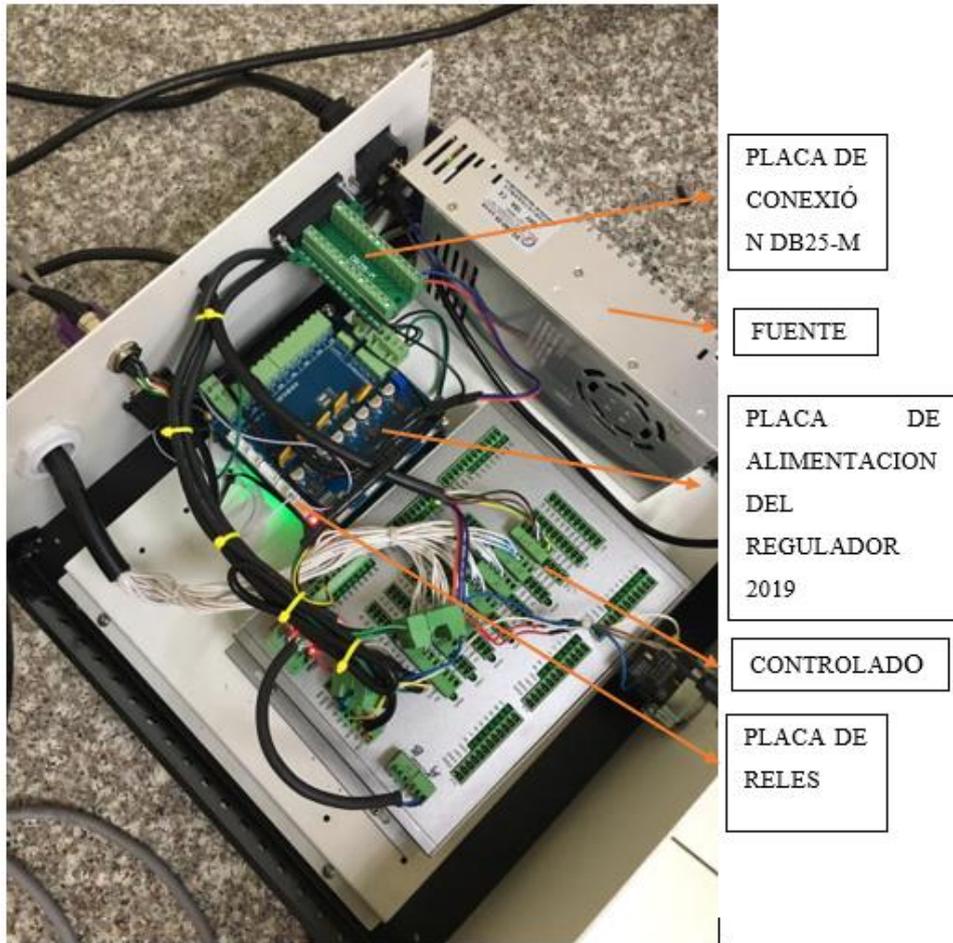


Figura 4-3. Localización de los componentes del conjunto controlador
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

3.1.1.3 Electroválvula y mini bomba

Es un elemento que permitirá el accionamiento del Gripper para poder sujetar las cosas deseadas.

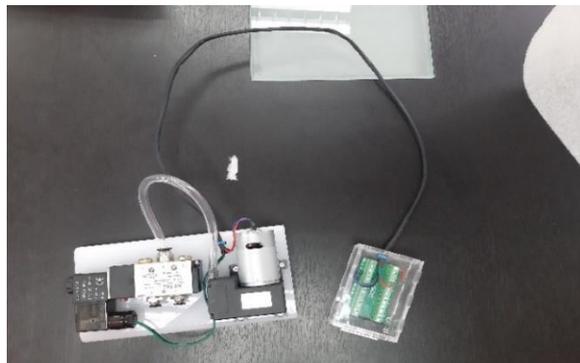


Figura 5-3. Electroválvula
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

La electroválvula irá conectada a la fuente del controlado y al Gripper que se encuentra como último actuador del robot. Sin este elemento, el último actuador quedará inútil a menos que el sistema de sujeción sea reemplazado.

3.1.1.4 Teach Pendant

Uno de los elementos importantes es la teach pendant que es la pantalla de interfaz del robot, esta puede ser usada para controlar el robot sin necesidad de una computadora, facilitando la comunicación operador robot.

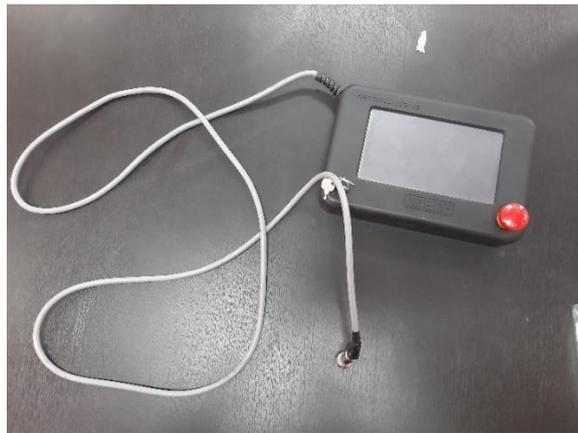


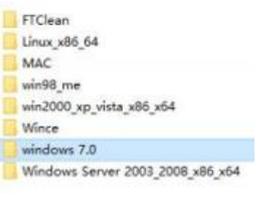
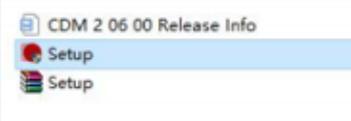
Figura 6-3. Teach Pendant
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

A pesar de estar pensando para facilitar la operación; en el estado actual, tiene inconveniente ya que no puede usarse para controlar el brazo debido a la mala configuración y para trabajar con la computadora es obligatorio conectar la pantalla, de lo contrario no podremos interactuar con el robot.

3.1.2 Software

3.1.2.1 Interfaz

La interfaz es la pantalla principal con la que va a trabajar el operador del brazo robótico, esta interfaz cuenta con los elementos necesario para la programación de la trayectoria del robot, así como también con los sistemas de alarma respectiva, además de una serie de alternativas con las cuales el operador puede navegar. La interfaz viene con el idioma mandarín, lo que hace difícil la interacción con el operador.

<p>3. Seleccione la carpeta USB a RS232.</p>	
<p>4. Seleccionar sistema operativo (solo hasta Windows 7).</p>	
<p>5. Ejecutar la aplicación "Setup".</p>	
<p>6. Ir al administrador de dispositivos.</p>	
<p>7. En otros dispositivos reconocer el USB Serial Port (COM3) que después de haber ejecutado la aplicación se convertirá en un controlador de bus serie universal listo para recibir y enviar señales.</p>	

Fuente: Manual del robot

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Para ver el funcionamiento del robot es necesario tener el sistema operativo adecuado, en este caso Windows 7 para el correcto funcionamiento del software. Entendiendo de que este sistema operativo ya está obsoleto y en algunos casos hasta fuera de servicio. Se tuvo que buscar un driver que reconozca de RS232 a USB o viceversa, compatible para cualquier sistema operativo; y el driver encontrado es el PL2303_Prolific_GPS_1013_20090319.

 PL2303_Prolific_GPS_1013_20090319	
Tipo de archivo:	Aplicación (.exe)
Descripción:	Setup.exe
Ubicación:	C:\Users\usuario\Desktop\Trabajo de Titulació
Tamaño:	2,91 MB (3.057.807 bytes)
Tamaño en disco:	2,91 MB (3.059.712 bytes)
Creado:	jueves, 24 de diciembre de 2020, 9:58:38
Modificado:	jueves, 19 de marzo de 2009, 14:16:02
Último acceso:	Hoy, 2 de marzo de 2021, hace 4 minutos

Figura 8-3. Propiedades del driver instalado.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

3.1.3 Pruebas realizadas al funcionamiento actual del robot.

Antes de poner en marcha y conocer el funcionamiento del robot con su software original, es recomendable leer y tener en cuenta las condiciones de uso que esta detallado en el ANEXO A.

3.1.3.1 Manipulación

Para realizar pruebas al robot se debe hacer un previo estudio del software, cómo es su funcionamiento, cómo va conectado el brazo con el controlador y con la computadora. Se debe tener en cuenta que el robot trabaja con 220V de corriente alterna por lo que se debe buscar la disponibilidad de ese tipo de suministro de corriente.



Figura 9-3. Conexión lista para el uso
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Una vez realizada la correcta conexión, el previo estudio y que la computadora reconozca al controlador se procede de la siguiente manera.

- 1) Abrir el ejecutable del robot: este es el software original que en su mayor parte está en chino, pero existe una opción para poner el lenguaje al idioma inglés y tener una mejor comprensión.
- 2) Seleccionar el puerto de comunicación y cerrar el puerto.
- 3) Configurar la velocidad de movimiento que en este caso tiene de 100, 75, 50, 20, 10 y 5%
- 4) Luego depurar el sistema de algún comando que este ejecutado presionando los siguientes botones:
 - a. Pause 30H
 - b. Exit 10H
 - c. Go zero 15H



Figura 10-3. Interfaz de configuraciones para el arranque del robot
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

5) Seleccionar la opción TEACH en donde podremos manipular cada uno de los motores.

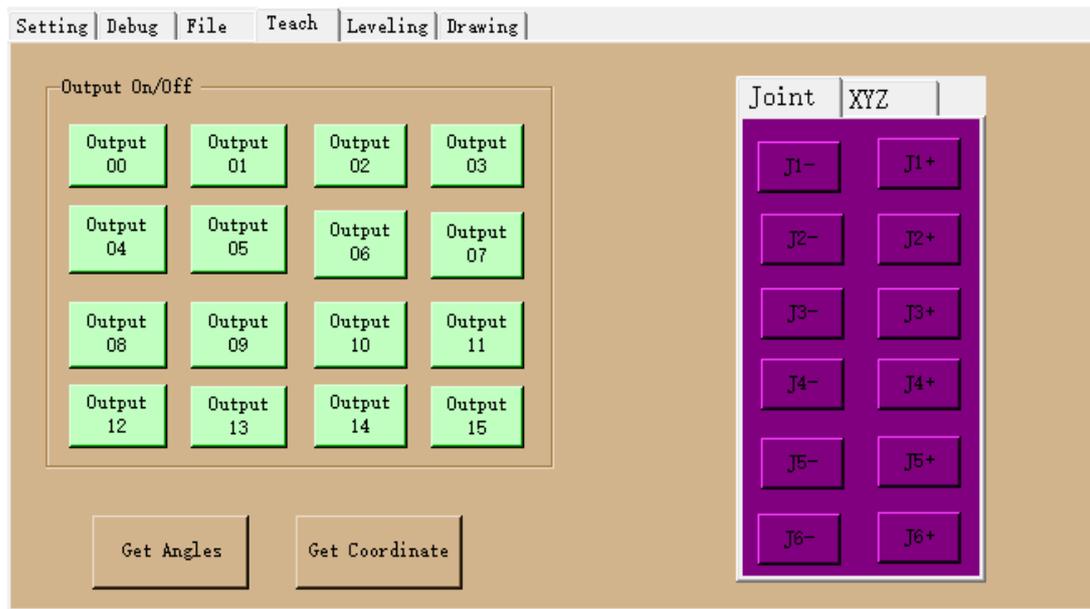


Figura 11-3. Pantalla de accionamiento para el movimiento del brazo

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Se puede mover cada uno de los motores con la denominación ya antes establecida (J1, J2, J3, J4, J5, J6), presionando los botones con más y menos para girar en un sentido o en otro depende como se lo requiera.

6) Existe la opción de realizar un movimiento coordinado, este movimiento coordinado permite mover más de un motor al mismo tiempo. Se debe pulsar en la opción que dice XYZ para los movimientos preestablecidos.

3.1.3.2 Respuesta de las pruebas realizadas.

De la revisión realizada a la situación actual se tiene que en conjunto, el software actual no tiene un buen funcionamiento, cuando se prende el controlador hace que los motores se accionen de manera involuntaria como se puede observar en la figura 11-3, en la interacción las señales que recibe el controlador tiene muchas interferencias y esto hace que el brazo robótico se mueva de manera irregular y descoordinada, para corregir el problema de interferencia es necesario reiniciar toda las conexiones incluyendo el software, se da casos en las que el controlador no reacciona a ninguna de las señales que enviadas y se es necesario reiniciar todo nuevamente.



Figura 12-3. Movimiento involuntario.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Los movimientos preestablecidos no tienen ningún sentido ya que cuando se le envía la señal se mueve muy poco cada uno de los motores que comprenden al movimiento mencionado, puede ser que sea parte de los requerimientos del proveedor, pero no se ve ninguna utilidad.

A continuación, se detalla el funcionamiento de cada uno de los motores después de haber sido enviada una señal desde el software.

Tabla 4-3: Funcionamiento de los motores en las pruebas realizadas.

Motor	Funcionamiento		Detalle
	Bueno	Malo	
J1		X	No recibe la señal adecuadamente desde el software, hay veces que no recibe la señal por lo que se podría decir que su funcionamiento es malo.
J2		X	La señal que se le envía no la reconoce adecuadamente, hay señales que si las recibe mientras otras no; funcionamiento malo.
J3		X	No recibe la señal de manera adecuada; este motor al igual que el 1 y 2 son los que

			se mueven de manera involuntaria cuando el controlador se prende.
J4	X		Su funcionamiento es bueno, dando una respuesta rápida a las señales enviadas.
J5	X		Tiene un funcionamiento bastante bueno.
J6	X		Comprende totalmente la señal que se le envía.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Se debe recalcar que el funcionamiento de la electroválvula que va a accionar al Gripper es adecuado cuando se le envía la señal.

3.2 Requerimientos para la teleoperación y monitoreo

El sistema de teleoperación y monitoreo requiere cuatro partes elementales para su correcto funcionamiento los cuales son funcionales, hardware, software y estructural que se detallan a continuación.

3.2.1 Requerimiento Funcional

Para la operación, teleoperación y monitoreo es necesario operar al robot de manera rápida, eficiente y optima, al igual que el sistema de teleoperación y monitoreo; prestaciones que nos permitan operar desde cualquier computadora sin importar el sistema operativo que tenga y teleoperar y monitorizar desde cualquier dispositivo que tenga una conexión estable a internet.

3.2.2 Requerimiento hardware

Con la necesidad de controlar el brazo robótico, se optó por utilizar los mismos elementos que tiene el robot con la consideración de que se les dio un mejor uso, teniendo así un controlador de excelentes características y un brazo robótico muy útil que se le puede sacar mucho provecho.

3.2.3 Requerimiento software

De la necesidad de tener una interfaz compatible para cualquier sistema operativo sin importar el modelo de computadora que se tenga, una interfaz que sea entendida por la mayoría de las personas y en español, plataformas que nos den la facilidad de realizar la conexión con el internet

de las cosas específicamente con la plataforma Ubidots y el software de programación, un software de programación compatible con el controlador que se tiene.

Tabla 5-3: Evaluación para la ponderación del software.

Evaluación			
	Disponibilidad de adquisición	Reconocimiento del puerto serial	Conexión a ubidots
Matlab	3	4	1
Visual Studio	3	4	4
Labview	3	3	4

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Para la evaluación se tomó en cuenta pruebas realizadas para el puerto serial de cada uno de los softwares, se investigó alternativas de conexión con la plataforma Ubidots y la disponibilidad para adquirir el software.

Tabla 6-3: Ponderación del software de programación.

Ponderación para elección de software de programación										
	Disponibilidad de adquisición	Peso	Subtotal	Reconocimiento del puerto serial	Peso	Subtotal	Conexión a ubidots	Peso	Subtotal	Total
Matlab	3	20%	0,6	4	40%	1,6	1	40%	0,4	2,6
Visual Studio	3	20%	0,6	4	40%	1,6	4	40%	1,6	3,8
Labview	3	20%	0,6	3	40%	1,2	4	40%	1,6	3,4

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Para la ponderación se decidió dar los siguientes pesos:

Disponibilidad de adquisición: se le dio un peso del 20% dado a que adquirir cualquier de los softwares mencionados es difícil, pero se lo puede hacer sin relevancia.

Reconocimiento del puerto serial: se le dio un peso del 40%, dado a que es muy necesario que el software tenga la conexión serial ya que el controlador envía ese tipo de señal.

Conexión a Ubidots: se le dio un peso de 40%, dado a que es muy importante trabajar en esta plataforma para la teleoperación y monitoreo.

Teniendo como resultado de la ponderación al software para programación VISUAL STUDIO con un 3.8.

3.2.4 Requerimiento estructural

Maniobrar, teleoperar y monitorizar, son acciones en la cual se necesita más de dos personas para poder llevar a cabo, teniendo dicha necesidad nace la idea de un requerimiento estructural el cual

debe servir para la creación de una bancada que ayude al anclaje del brazo robótico para reducir un recurso muy importante como es el talento humano. Teniendo herramientas como Solidwork que permite realizar el diseño, planos y análisis estático para comparar entre materiales y elegir la mejor opción tanto en funcionalidad como en costo.

Una vez realizado las pruebas correspondientes para observar y diagnosticar el desenvolvimiento del brazo robótico se procederá a realizar la programación, interfaz y monitorización con el internet de las cosas (IoT); la propuesta es la siguiente.

3.3 Diseño del modelo

3.3.1 *Diseño estructural*

La metodología aplicada para realizar la bancada se la detalla en el siguiente gráfico.

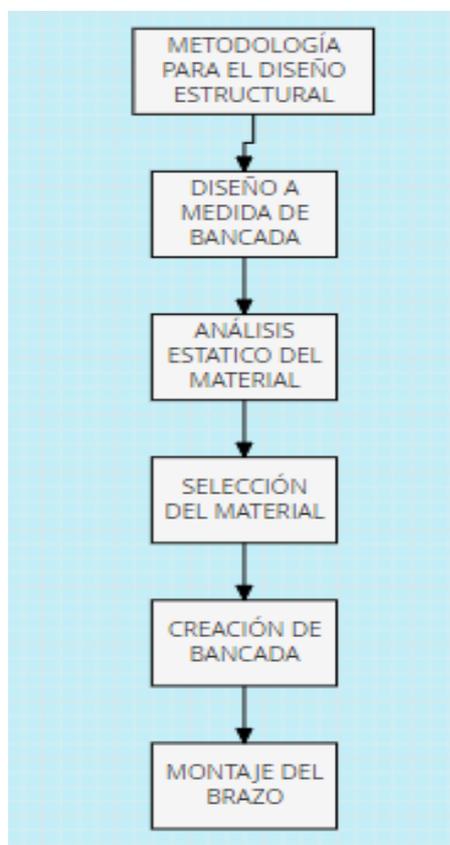


Figura 13-3. Diagrama de flujo.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 7-3: Características del brazo robótico.

características del brazo	
Peso del cuerpo	25 KG
Carga máxima	3 KG
Largo del brazo	782 mm
Repetibilidad	-0.05mm, +0,05mm
Medio ambiente	0-45°C, 20-80% de humedad relativa

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Para determinar la fuerza máxima que va a ejercer el brazo, que es cuando esta extendido horizontalmente en su totalidad se hizo una sumatoria de reacciones.

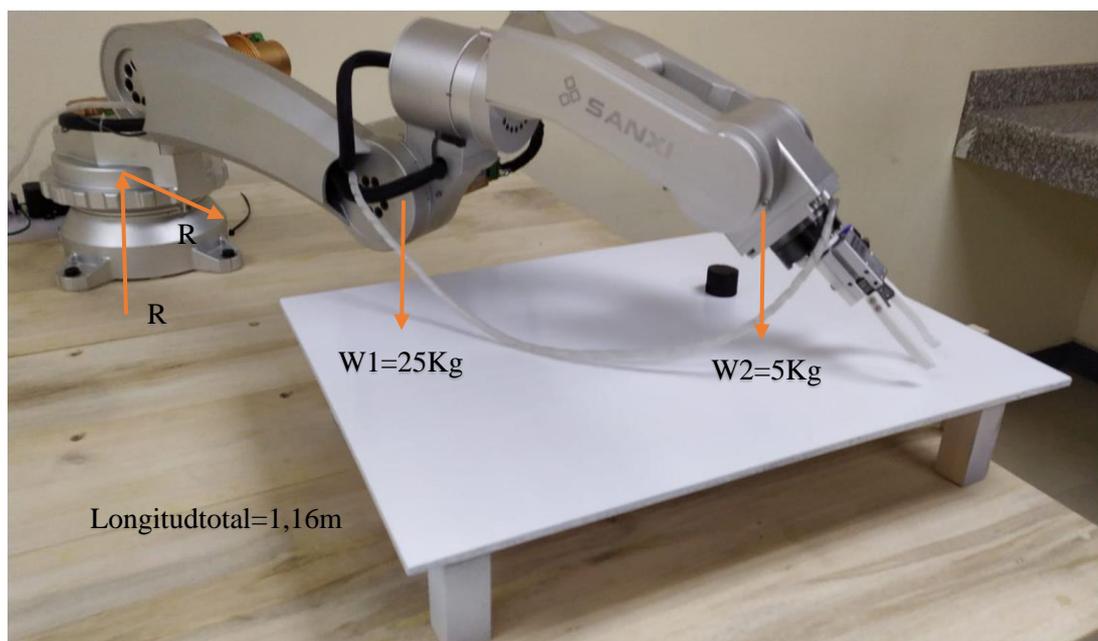


Figura 14-3. Datos para obtener la fuerza máxima a ejercer.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Datos:

$$W1=25kG \quad W2=5Kg \quad Lt=1,16m \quad g=9,8m/s^2$$

$$F1 = W1 * 9.8 \frac{m}{s^2}$$

$$F1 = W1 * 9.8 \frac{m}{s^2}$$

$$F1 = 25Kg * 9.8 \frac{m}{s^2}$$

$$F1 = 5Kg * 9.8 \frac{m}{s^2}$$

$$F1 = 245 N$$

$$F1 = 49 N$$

$$\sum F_y = 0$$

$$T = \sum F * d$$

$$R_y = F1 + F2$$

$$T = F1 * d1 + F2 * d2$$

$$R_y = 245N + 49N$$

$$T = 245N * 0,58m + 49N * 1.16m$$

$$R_y = 294N$$

$$T = 198.94N.m$$

$$\sum F_x = 0$$

$$R_x = 0$$

Teniendo en cuenta el peso del brazo, carga máxima, largo del brazo y el medio ambiente ideal para el brazo, se diseñó el siguiente modelo de bancada para que soporte al brazo robótico en condiciones de trabajo optimas.

Una vez determinada el requerimiento estructural para el brazo robótico, se procedió con la modelación de la estructura de la bancada en el Software CAD denominada Solid Works, debido a que esta ofrece varias facilidades al momento de crear diseños 3D, así como también permite la evaluación de la estructura mediante análisis estructurales.

Con la finalidad de presentar un diseño que satisfaga con el confort y calidad para el operario. Para esto fue necesario la búsqueda de alternativas para profundizar el comportamiento de los materiales a través de evaluaciones previas.

Una vez diseñado el modelo se realizó los planos con las medidas correspondientes, con las vistas respectivas para una mayor perspectiva para su construcción con el fin de descartar posibles errores. Revisar el ANEXO C donde se detalla las cotas de la bancada.

Al terminar con la construcción de la banca, se procede al montaje del brazo rótico con la bancada como se muestra en la figura 18-3.

3.3.1.1 Análisis estático

Análisis estático es la que se encarga de evaluar las fuerzas que se aplica a un cuerpo que se mantiene en equilibrio. Es utilizada para determinar el estado de un cuerpo cuando es sometida a fuerzas conocidas. Es después del análisis estático que se pudo determinar que el material con el que se trabajó es idóneo para la construcción de la bancada.

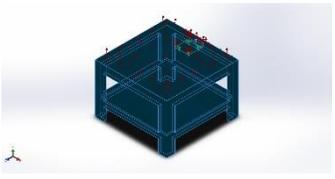
En la siguiente figura se muestra la elaboración del modelo 3D de la bancada.



Figura 15-3. Diseño 3D de la bancada para el brazo robótico.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la tabla la siguiente tabla se detallan las propiedades volumétricas de la bancada.

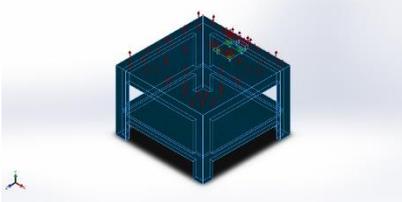
Tabla 8-3: Propiedades de la bancada.

NOMBRE DE DOCUMENTO Y REFERENCIA	TRATADO COMO	PROPIEDADES VOLUMÉTRICAS
Línea dividida 1 	Cuerpo sólido	Masa: 70.4272 kg Volumen: 0.202683 m ³ Densidad: 159.99 kg/m ³ Peso: 317.787 N

Fuente: Solid Works 2019

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 9-3: Propiedades del material de la bancada.

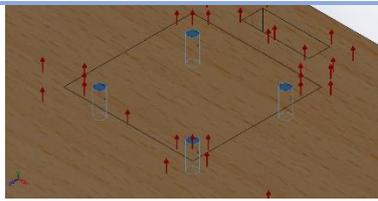
REFERENCIA DE MODELO	PROPIEDADES
	<p>Nombre: Roble</p> <p>Tipo de modelo: Isotrópico elástico lineal</p> <p>Criterio de error predeterminado: Desconocido</p> <p>Fueza de influencia: 2e+07 N/m²</p> <p>Módulo elástico: 3e+09 N/mm²</p> <p>Coefficiente de Poison: 0.29</p> <p>Densidad de masa: 159.99 N/mm³</p> <p>Módulo de corte: 3e+08 N/mm²</p>

Fuente: Solid Works 2019

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

A continuación, se detalla los parámetros para el análisis estático, la cual es el brazo que va a soportar la bancada.

Tabla 10-3: Sujeciones.

NOMBRE DE SUJECIÓN	IMAGEN DE SUJECIÓN	DETALLES DE SUJECIÓN
Fijo-1		<p>Entidades: 4 cara(s)</p> <p>Tipo: Geometría fija</p>

Fuente: Solid Works 2019

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

a) Análisis de tensiones

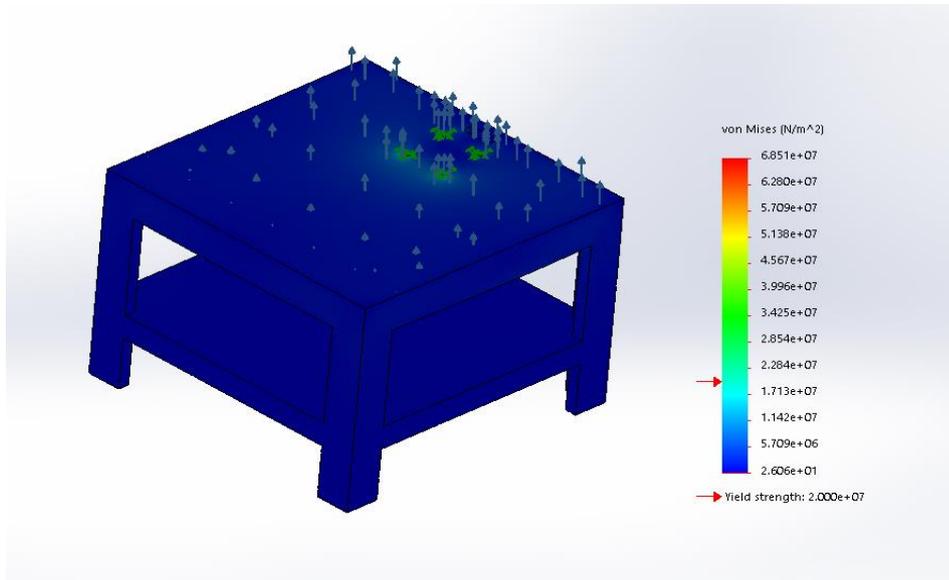


Figura 16-3. Análisis de tensión de la bancada.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 11-3: Valores de tensión de la bancada.

Nombre	Tipo	Mínimo	Máximo
Tensión	Von Mises	2.606e+01 N/m ² Node: 7438	6.851e+07 N/m ² Node: 86313

Fuente: Solid Works 2019

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la figura 15-3 se muestra el análisis estático de tensiones de la fuerza aplicada en la bancada, dando como resultado un límite elástico de 2.000e+07 N/m², en la misma se muestra una barra en la cual se observa los límites permitidos que soporta las tensiones, partiendo desde lo ideal que se encuentra en color azul con un valor de 2.606e+01 N/m², las demás tensiones son aceptadas hasta llegar al su límite de aceptación de 6.851e+07 N/m², al superar el valor máximo se debe cambiar de material para evitar fallos en la estructura. Al aplicar los puntos críticos en la cuales existe mayor tensión al momento de ejercer las fuerzas, se verifica que el material utilizado para realizar la bancada es el correcto, debido a que tiene una resistencia a la deformación para tensiones con un valor por abajo del límite principal.

a) Análisis de desplazamiento

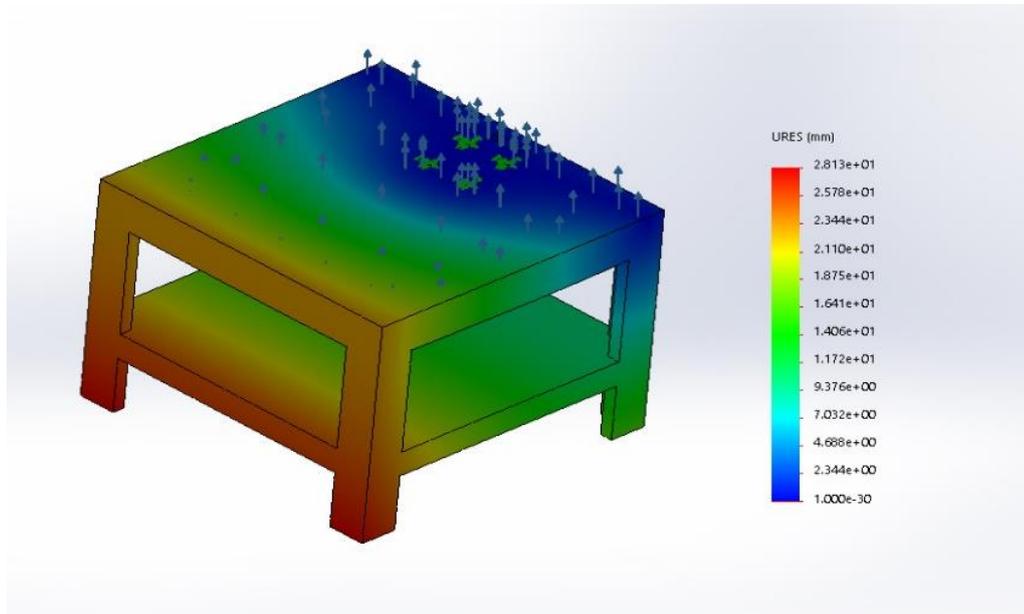


Figura 17-3. Análisis de desplazamiento de la bancada.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 12-3: Valores de desplazamiento de la bancada.

Nombre	Tipo	Mínimo	Máximo
Desplazamiento	Ures	0.000e+00 mm	2.813e+01 mm
		Node: 1	Node: 6936

Fuente: Solid Works 2019

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Al realizar el análisis para desplazamiento de la bancada expuesta a las mismas cargas, se obtiene como resultado un límite mínimo de 0.000e+00 mm y un máximo de 2.813e+01 mm. Se muestra que, si el límite máximo es superado, se debe cambiar de materia para la construcción, que esto podría ocasionar daños en la estructura. La figura 16-3 muestra que la sección que más está expuesta a desplazamiento es la parte delantera.

b) Deformación unitaria de la bancada

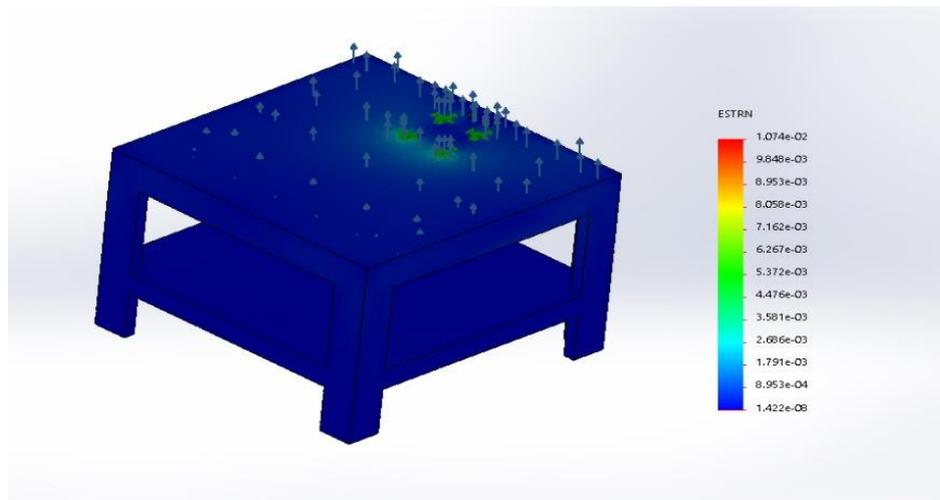


Figura 18-3. Deformación unitaria de la bancada.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 13-3: Valores de deformación unitaria de la bancada.

Nombre	Tipo	Mínimo	Máximo
Deformación unitaria	ESTRN	1.422e-08	1.074e-02 Element: 15366

Fuente: Solid Works 2019

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la estructura base de la figura 17-3 se muestran los límites para la deformación unitaria del material teniendo un valor mínimo $1.422e-08 \text{ N/m}^2$ en color azul y un valor máximo de $1.074e-02 \text{ N/m}^2$ en color rojo, el análisis presenta en todos los tubos un color azul indicando que está dentro del rango permitido, mediante la barra de resultados se evalúa que el material es adecuado para la construcción.

3.3.1.2 Comparación de costos de fabricación

Después de concluir con el análisis estático realizado, se procedió a realizar un análisis de costos de fabricación entre el material escogido que es la madera y otro tipo de material que cumpla con similares especificaciones teniendo lo siguiente.

Tabla 14-3: Costos de fabricación para una bancada en hierro.

Costo de fabricacion de la bancada (Acero)			
Elemento	Cantidad	Costo Unitario	Subtotal
Perfil 4 pul	2,5	\$ 18,00	\$ 45,00
Plancha de hierro Ductil 6 lineas 1.3X1.3m	1	\$ 83,00	\$ 83,00
Mano de obra	1	\$ 80,00	\$ 80,00
Elementos varios	1	\$ 30,00	\$ 30,00
Total			\$ 238,00

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Después de haber cotizado en algunos centros ferreteros los datos reflejados en la tabla son los datos más económicos encontrados, teniendo como costo final para la fabricación de la bancada en material Acero ASTM A36 de 238\$.

Tabla 15-3: Costos de fabricación de la bancada en madera.

Costo de fabricacion de la bancada (Madera)			
Elemento	Cantidad	Costo Unitario	Subtotal
Tablón de madera Piwi 20cmX5cm	13	\$ 7,00	\$ 91,00
Tabla Machimbrada de Eucalipto	13	\$ 0,90	\$ 11,70
Mano de obra	1	\$ 47,30	\$ 47,30
Elementos varios	1	\$ 10,00	\$ 10,00
Total			\$ 160,00

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

La tabla refleja la cotización más económica teniendo como costo final de 160\$ con un tipo de madera que es muy resistente y cumple con las especificaciones que necesitamos.

Después de haber analizado las dos opciones para la bancada se llegó a la conclusión de optar por la bancada de madera, que cumple con las especificaciones requeridas y el costo es mucho menor.



Figura 19-3. Bancada para el brazo robótico.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

3.3.2 Diseño hardware

Teniendo disponible el robot ANNO Rv624 se lo acopló a las necesidades antes especificadas, se le hizo algunos cambios y arreglos para que su funcionamiento sea el óptimo.

Fueron utilizados los mismos elementos del controlador debido a que su funcionamiento es el adecuado, los accesorios adicionales como el Gripper, la mini bomba y la electroválvula también fueron utilizados, los motores del brazo robótico fueron sometidos a mantenimiento debido a su incorrecto funcionamiento detallando que el motor uno y dos fueron intervenidos y arreglados satisfactoriamente.



Figura 20-3. Mantenimiento de los motores 1 y 2.
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Podemos observar en la figura que se encontró dos tornillos los cuales eran el motivo por el cual el funcionamiento de los motores era incorrecto cuando se le enviaba una señal desde la interfaz original. Estos daños pudieron ser ocasionados por un mal montaje al momento del ensamble por parte de los proveedores o también por un excesivo Angulo cuando se envió una señal al motor, se debe tener mucho cuidado y consideración con el rango que tiene cada motor, esto se lo puede verificar en la siguiente tabla.

Tabla 16-3: Rango de moción de cada junta.

Rango de moción	
J1	+170° a -155°
J2	+90° a -120°
J3	+125° a -85°
J4	+160° a -190°
J5	+30° a -200°
J6	+360° a -360°

Fuente: (ANNO, 2018)

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

3.3.2.1 Conexiones

A continuación, se muestra las conexiones del controlador con la fuente de alimentación, el Gripper y los actuadores.

a) Conexión controlador-fuente

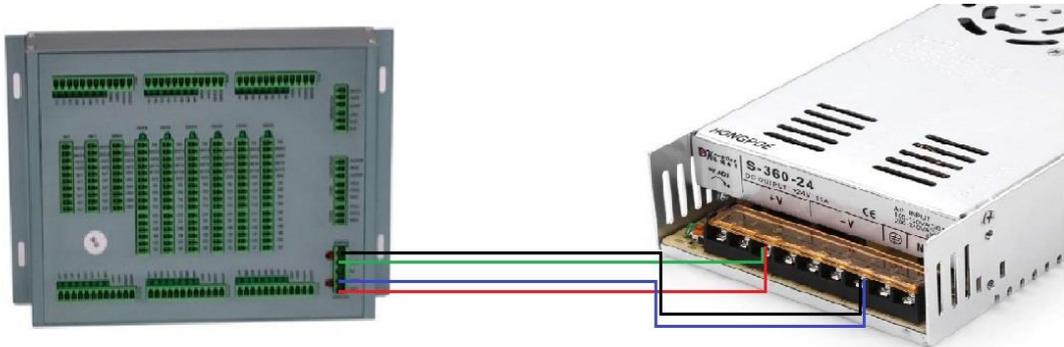


Figura 21-3. Diagrama de conexión del controlador-fuente.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

La figura 20-3 muestra la conexión del controlador hacia la fuente, en el controlador tiene dos entradas para voltaje y dos para GND, los dos pines de voltaje del controlador están conectados al positivo de la fuente y los dos pines del GND del controlador están conectados al negativo de la fuente.

b) Conexión controlador-motores

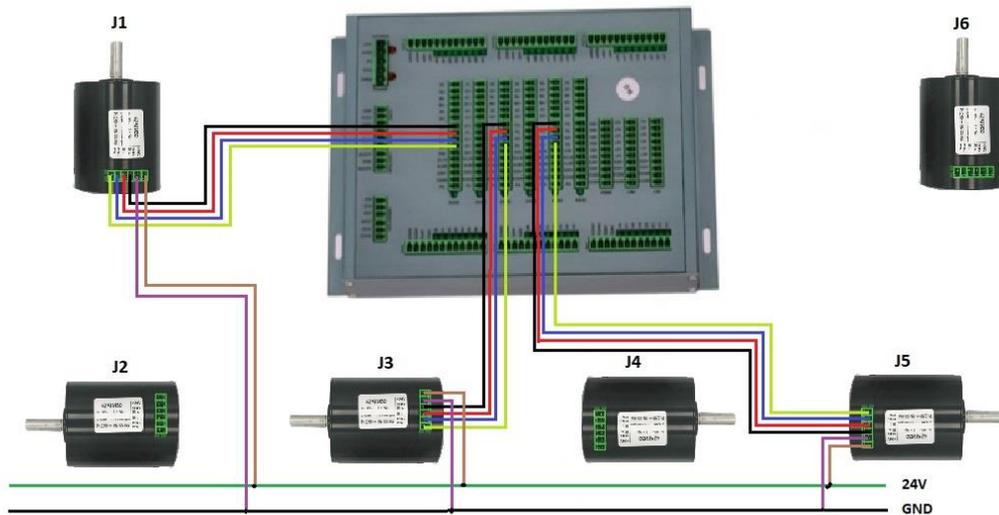


Figura 22-3. Diagrama de conexión del controlador-motores.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la figura 21-3 se muestra las conexiones de los motores correspondientes a las juntas 1, 3 y 5, cada uno tiene conectado tanto sus pulsos positivos como los negativos, la fuente a 24V y el GND.

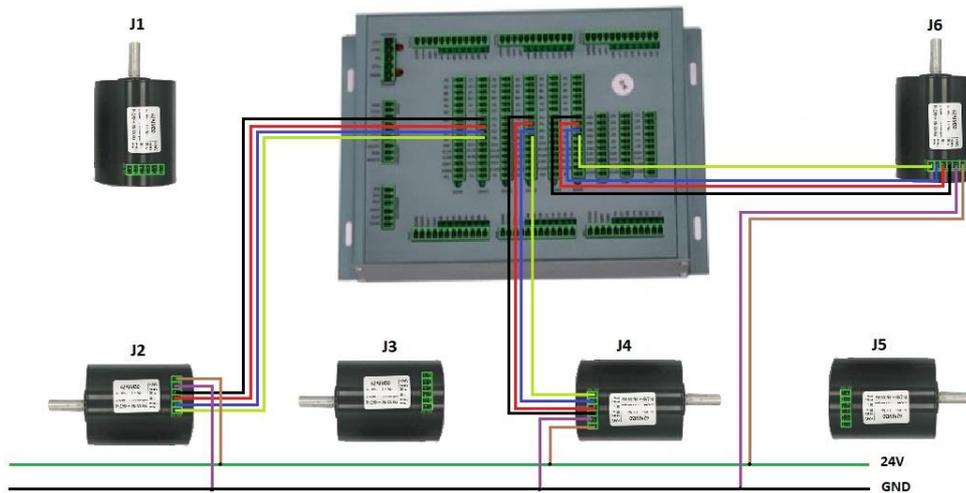


Figura 23-3. Diagrama de conexión del controlador-motores.
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la figura 22-3 se muestra las conexiones de los motores correspondientes a las juntas 2, 4 y 6, cada uno tiene conectado tanto sus pulsos positivos como los negativos, la fuente a 24V y el GND.

a) Conexión controlador- PLACA DE CONEXIÓN DB25-M

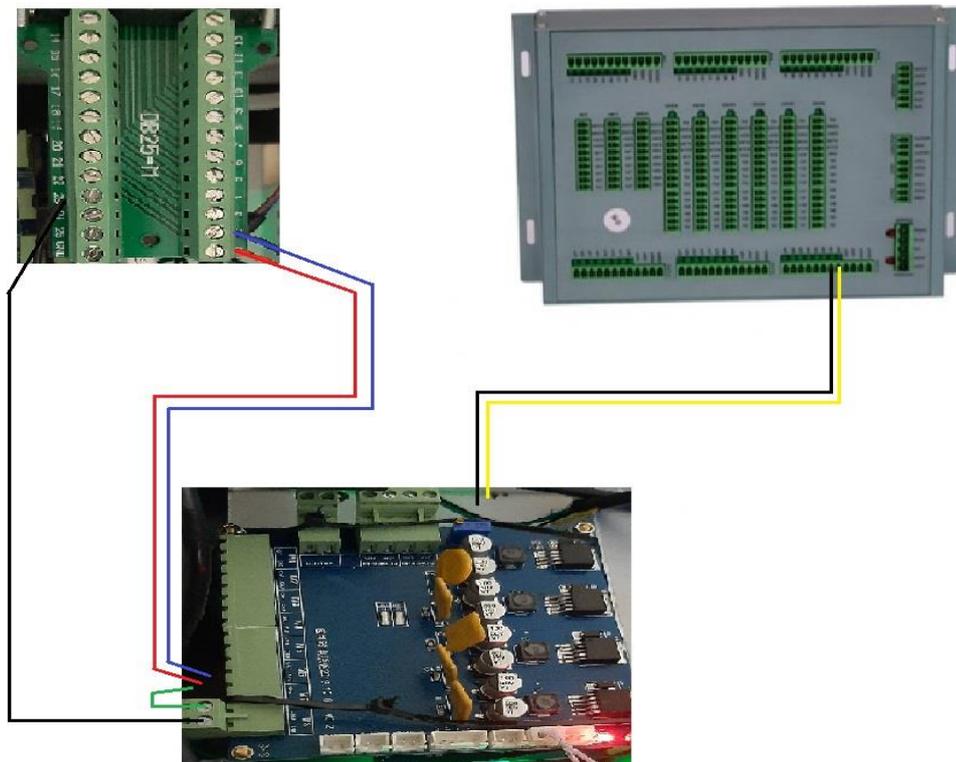


Figura 24-3. Diagrama de conexión del controlador- placa de conexión DB25-M.
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la figura 23-3 se muestra la conexión de la electroválvula y la mini bomba las cuales sirven para el accionamiento del Gripper, los pines “00” y “01” son digitales, esto quiere decir que recibe valor de cero o uno. El pin “00” es el que gestiona el encendido o apagado de la electroválvula mientras que el pin “01” el encendido o apagado de la mini bomba.

3.3.3 Diseño software

Para la programación, el software elegido es Visual Studio elección ponderada de acuerdo con nuestras necesidades en la tabla 6-3 se lo detalló, software el cual nos permite programar en varios lenguajes como se especifica en el marco teórico. En Visual Studio se realizó la programación para el movimiento de cada una de las juntas del brazo, se eligió este software por las facilidades de programación que otorga, así como también brinda opciones de trabajar con varias librerías para usar y conectar dependiendo del tipo de programación que se requiera, también la facilidad que brinda para la comunicación con el controlador haciendo uso de librerías como: openPort, closeport, initRunStatus, runJointAngle, runXYZ, runSingleJointAngle, las cuales permiten realizar la comunicación con el puerto serial del controlador y así poder enviar y recibir los datos que sean necesarios para el desarrollo del tema.

Para la programación de la interfaz se lo hace con lenguaje C#, debido a que las librerías obtenidas realizan la comunicación de mejor manera con el controlador. Además, que existe un vínculo directo con las librerías de Ubidots en este lenguaje de programación.

3.3.3.1 Programación

A continuación, se presenta las partes más relevantes del código para la programación de la interfaz y para hacer la comunicación en la teleoperación y monitoreo con Ubidots.

Para poder establecer comunicación entre el brazo robótico y el controlador, se declaran las librerías ya antes mencionadas y que se muestran en la figura 20-3, una vez declaradas en las líneas de programación se deben copiar en la carpeta en las cuales se ha creado el proyecto de lo contrario no se realiza el llamado y genera error al momento de compilar el código.

```

19 namespace robotanno_ubidot_net
20 {
21
22     6 referencias
23     public partial class Form1 : Form
24     {
25         const string rutadll = "RobotAnnoStuVer.dll";
26         //Decalarando las librerias dll
27         [DllImport(rutadll)]
28         static extern int openPort(string port, int baudios);
29         [DllImport(rutadll)]
30         static extern int closePort();
31         [DllImport(rutadll)]
32         static extern int initRunStatus();
33         [DllImport(rutadll)]
34         static extern int runJointAngle(double j1, double j2, double j3, double j4, double j5, double j6);
35         [DllImport(rutadll)]
36         static extern int runXYZ(double x, double y, double z, int rx, int ry, int rz);
37         [DllImport(rutadll)]
38         static extern int runSingleJointAngle(int indexJ, double coord);
39         //indexJ = 1 para la junta 1
40         [DllImport(rutadll)]
41         static extern int setSpeed(int s);
42         //La velocidad varia de 1-100 relacionado al 1%-100%
43         [DllImport(rutadll)]

```

Figura 25-3. Denominación de las librerías DLL.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Para que los motores se muevan de forma individual se hace la programación por separado para esto se hace uso de las variables ya declaradas con anterioridad, en la figura 21-3 se muestra la codificación para que la junta aumente de grado, esto se lo realiza con la librería dll declarada al inicio <<runSingleJointAngle>> esta librería permite se muevan un solo ángulo de las juntas o articulaciones. También de forma simultánea se envía los datos a la plataforma Ubidots para poder visualizar los grados que se mueve la articulación, el valor que adquiera la variable <<home_jun1>> será la misma que se verá reflejada en las variables colocadas en la plataforma Ubidots <<Lec_j1>>.

```

412 //funciones para aumentar por grados el desplazamiento de las juntas
413 1referencia
414 private void btMasJ1_Click(object sender, EventArgs e)
415 {
416     home_jun1 = home_jun1 + 1;
417     lb_j1.Text = home_jun1.ToString();
418     int valor = runSingleJointAngle(1, home_jun1);
419     if (valor == 0)
420     {
421         lb_envio_pos.Text = "Enviado";
422         if (modo_general == 1)
423             Envio_datos_Ubidots(home_jun1, Lec_j1);
424     }
425     else
426         lb_envio_pos.Text = "Sin Conexión";
427     Thread.Sleep(250);
428 }
429

```

Figura 26-3. Programación para movimiento individual de las juntas.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Al estar en el modo <<Control Ubidots>> esta actúa de acuerdo con la programación descrita en la figura, es decir que, al instante se abre una segunda ventana, al dar inicio se prepara para recibir las órdenes que sean enviadas desde la plataforma Ubidots. Los valores serán almacenados en la variable <<Recibir_datos_ubidots>>. Una vez que los datos sean almacenados, estos son

enviados a la interfaz, permitiendo así que las juntas se muevan de acuerdo con los grados que haya sido movido desde la plataforma.

```
100 private async void bt_iniciar_ClickAsync(object sender, EventArgs e)
101 {
102     lb_est_ubid.Text = "Conectado";
103     bt_iniciar.Enabled = false;
104     bt_parar_Ubidots.Enabled = true;
105
106
107     _canceller = new CancellationTokenSource();
108     await Task.Run(() =>
109     {
110         do
111         {
112             int valor_trabajar = Convert.ToInt32(Recibir_datos_Ubidots(con_trabajar));
113             int valor_pausar = Convert.ToInt32(Recibir_datos_Ubidots(con_pausar));
114             int valor_parar = Convert.ToInt32(Recibir_datos_Ubidots(con_parar));
115             if (valor_trabajar == 0 && valor_pausar == 0 && valor_parar == 0)
116             {
117                 double valor_j1 = Recibir_datos_Ubidots(var_j1);
118                 Actualizar_Interfaz(lb1Ubidots, valor_j1);
119                 Envio_datos_Robot(valor_j1, 1);
120                 if (_canceller.Token.IsCancellationRequested)
121                     break;

```

Figura 27-3. Comunicación entre Ubidots y la Interfaz.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Para conectar al internet de las cosas y enviar los datos de cada una de las juntas del brazo robótico, se utiliza la plataforma UBIDOTS, y la librería correspondiente que nos permite la comunicación de la plataforma con el software Visual Studio, para ello se detalla cada uno de los pasos seguidos desde la creación de las variables hasta la conexión de las librerías con Visual Studio.

- Crear cuenta en la plataforma Ubidots.
- En la pestaña DEVICE, seleccionamos la opción DEVICES.
- Seleccionamos “add new device” y escoger “blank devices”; aquí pide agregar el nombre del dispositivo, en nuestro caso se denomina Brazo Robótico. Una vez creado el dispositivo lo abrimos.

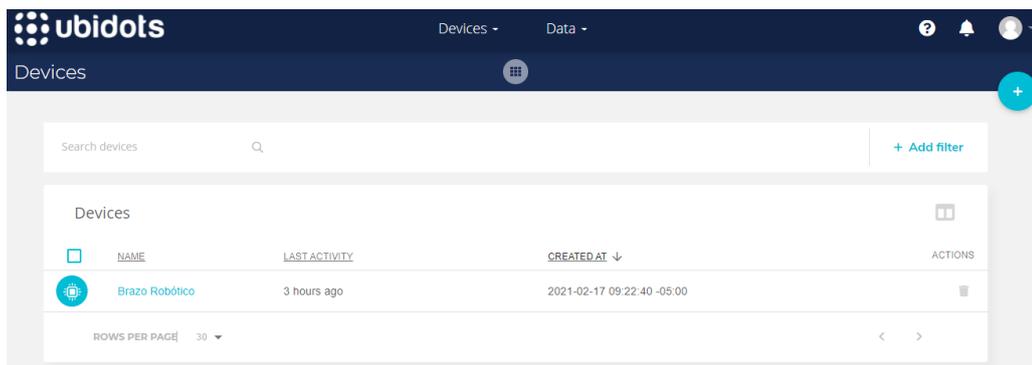


Figura 28-3. Dispositivo creado en la plataforma Ubidots
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

- d) Se procede a añadir variables; las que sean necesarias, en nuestro caso lo que queremos monitorear y teleoperar son los pasos de cada uno de los motores correspondientes al brazo robótico y los botones de operación, para lo cual hemos creado un total de 17 variables.

Se debe seleccionar “add variable” y escoger el tipo “Raw”.

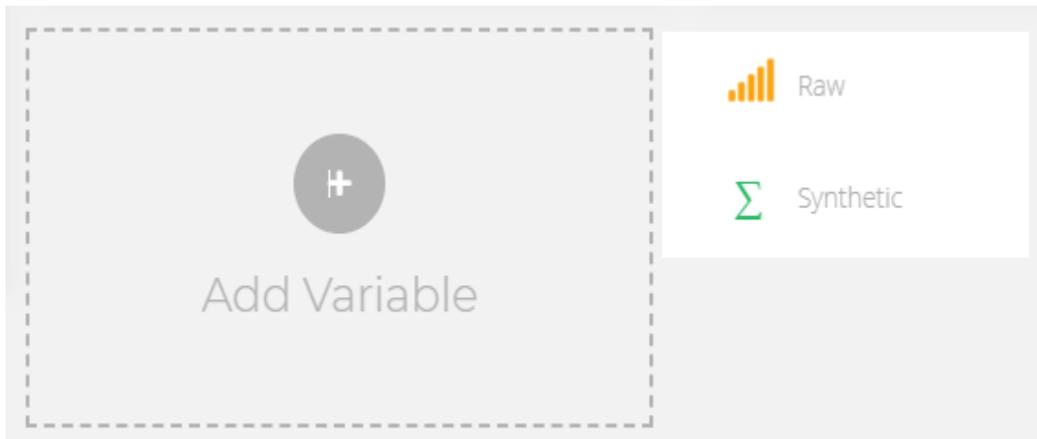


Figura 29-3. Crear una variable con la opción RAW

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

- e) En la pestaña DATA seleccionamos “dashboard” y luego “add new dashboard” y se le da un nombre en nuestro caso hemos creado tres dashboard denominado TELEOPERACIÓN POR JUNTAS, el otro denominado MONITOREO y el ultimo TELEOPERACION CONTINUA Y GRIPPER.

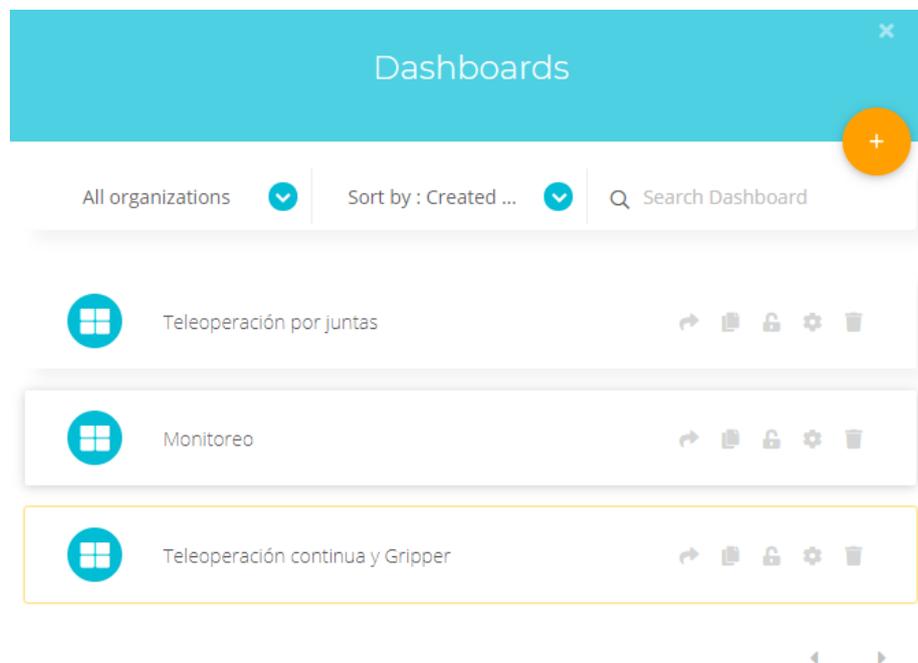


Figura 30-3. Dashboards creados en la plataforma

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

A continuación, se debe agregar un nuevo widget; en la plataforma existen diferentes tipos de widget como indicadores, histogramas, mapas, switch, batería, reloj, etc.

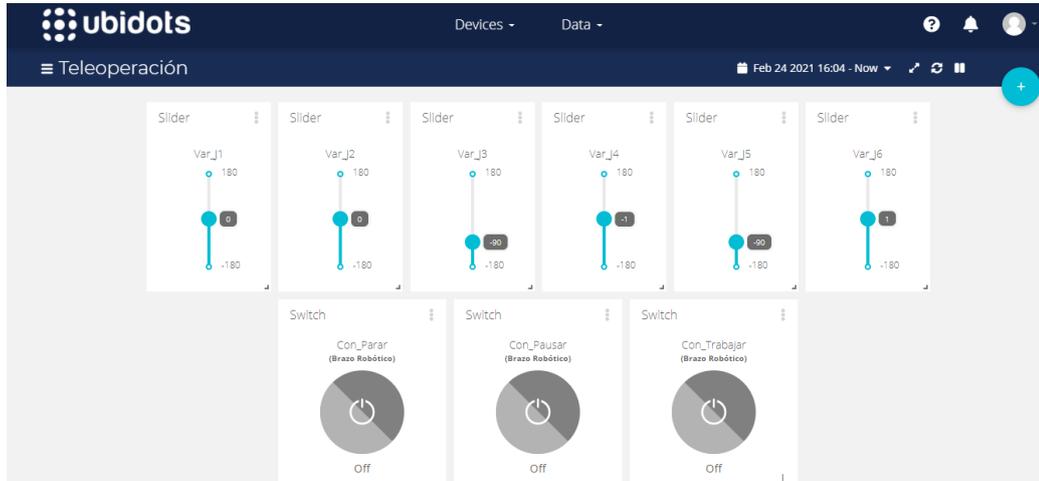
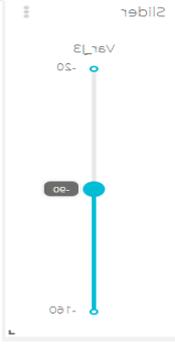


Figura 31-3. Widgets utilizados para la teleoperación
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Luego de seleccionar el widget que se necesite se debe asignar una variable de las que se crearon anteriormente. Al momento que se asigna una variable en consecuencia sale un cuadro para modificar datos, pero en ocasiones se le debe dejar por defecto, aunque esto depende del widget que se vaya a utilizar. En nuestro caso escogimos los siguientes widgets:

Tabla 17-3: Función de los widgets utilizados en los dashboards de Ubidots.

Elemento	función	Imagen
SLIDERS		
Slider 1	Regula la posición manualmente y vía internet del motor 1 cumpliendo la teleoperación; su rango oscila desde 0° a 113°	

Slider 2	Regula la posición manualmente y vía internet del motor 2 cumpliendo la teleoperación; su rango oscila desde 0° a 124°	
Slider 3	Regula la posición manualmente y vía internet del motor 3 cumpliendo la teleoperación; su rango oscila desde -160° a -20°	
Slider 4	Regula la posición manualmente y vía internet del motor 4 cumpliendo la teleoperación; su rango oscila desde -190° a 85°	
Slider 5	Regula la posición manualmente y vía internet del motor 5 cumpliendo la teleoperación; su rango oscila desde -130° a 170°	
Slider 6	Regula la posición manualmente y vía internet del motor 6 cumpliendo la teleoperación; su rango oscila desde +360° a -360°	

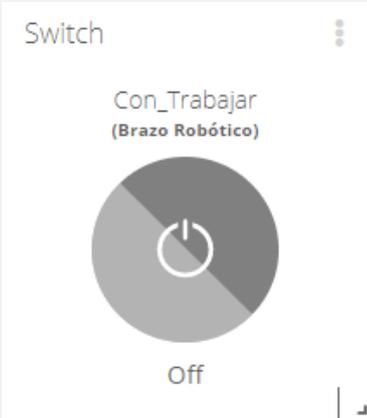
SWITCHS		
Switch Trabajar	<p>Este switch nos permite controlar de forma teleoperada una secuencia preestablecida del brazo robótico desde internet, la acción que hace el brazo cuando se presiona este switch consiste en agarrar un objeto de una posición determinada y dejarlo en otro totalmente distinta, de manera repetitiva.</p>	
Switch Pausa	<p>Este switc nos permite pausar la acción antes mencionada del brazo robótico independientemente de la posición en la que se encuentre, con el beneficio de que cuando se desabilite este switch el robot seguirá con sus labores normales.</p>	
Switch Paro	<p>Este switc nos permite parar completamente la acción que esté realizando el brazo robótico, cabe recalcar que cuando este switch es accionado el brazo robótico irá inmediatamente a su posición home.</p>	

TABLA DE VARIABES

Variables Name	Esta tabla nos permite visualizar el grado en que se encuentra cada una de las juntas y más que todo nos permite monitorizar en tiempo real estos datos enviados desde la interfaz grafica	 <table border="1" data-bbox="922 206 1279 568"> <thead> <tr> <th>VARIABLE NAME ↑</th> <th>LAST VALUE</th> </tr> </thead> <tbody> <tr> <td>Lec_J1</td> <td>0.00</td> </tr> <tr> <td>Lec_J2</td> <td>0.00</td> </tr> <tr> <td>Lec_J3</td> <td>-90.00</td> </tr> <tr> <td>Lec_J4</td> <td>0.00</td> </tr> <tr> <td>Lec_J5</td> <td>-90.00</td> </tr> <tr> <td>Lec_J6</td> <td>0.00</td> </tr> </tbody> </table>	VARIABLE NAME ↑	LAST VALUE	Lec_J1	0.00	Lec_J2	0.00	Lec_J3	-90.00	Lec_J4	0.00	Lec_J5	-90.00	Lec_J6	0.00
VARIABLE NAME ↑	LAST VALUE															
Lec_J1	0.00															
Lec_J2	0.00															
Lec_J3	-90.00															
Lec_J4	0.00															
Lec_J5	-90.00															
Lec_J6	0.00															

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

- Sliders (6): uno para cada junta del brazo robótico y los parámetros que se modificaron fueron los limites en un rango determinado dependiendo de las especificaciones del proveedor.
 - Switch (3): uno para accionar el brazo robótico a un movimiento prestablecido que le hemos denominado TRABAJAR, otro para pausar la operación y por último uno para parar completamente la operación.
 - Tabla de variable (1): aquí podremos monitorear las posiciones de cada una de las juntas enviadas desde la interfaz gráfica en tiempo real
- f) A continuación, se procede a realizar la conexión con Visual Studio mediante una librería llamada “Ubidots Library C#” la cual nos permite comunicarnos directamente entre Ubidots y Visual Studio.

Luego de haber realizado la comunicación Ubidots-Visual Studio y viceversa ya podremos manipular desde cualquier ordenador y desde cualquier parte no necesariamente donde se encuentre el brazo y podremos monitorizar ya sea desde la interfaz gráfica o bien desde la plataforma de Ubidots teniendo como resultado la misma respuesta con cualquiera de los métodos que se requiera o se pretenda utilizar.

3.3.3.2 Interfaz

La siguiente imagen refleja la distribución y el contenido de la interfaz ejecutable del software en donde se procederá a detallar para que sirve cada uno de los elementos que constituyen a la misma.

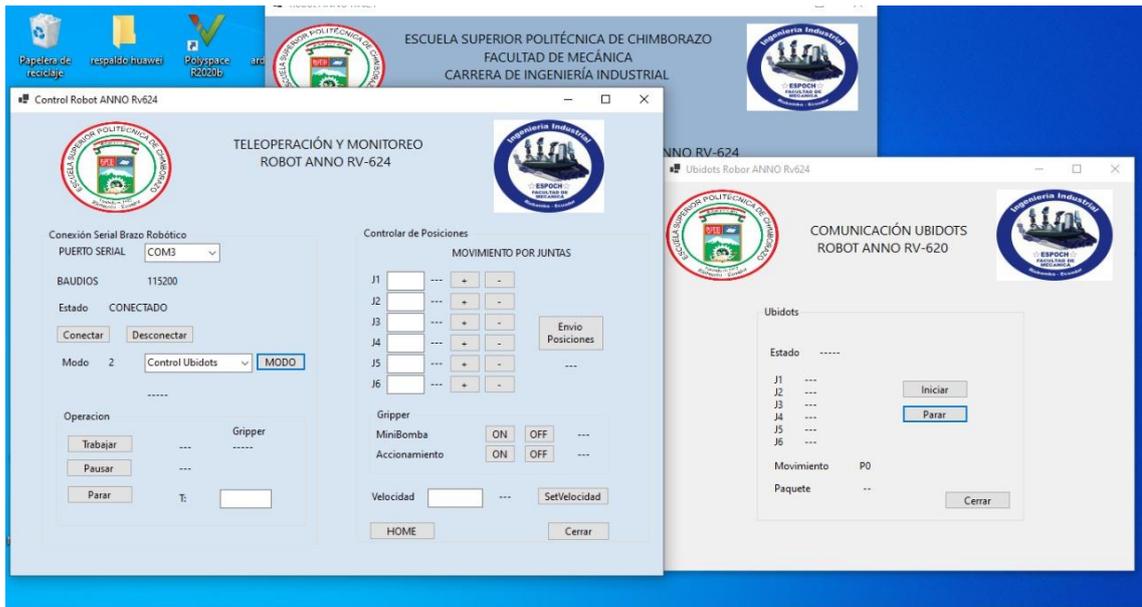


Figura 32-3. Interfaz gráfica para teleoperación y monitoreo.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

a) Puerto de comunicación:

En el apartado de puerto serial el usuario tiene la opción de escoger el puerto en la cual se encuentre conectado el brazo robótico a la computadora.

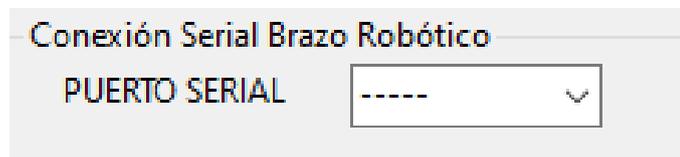


Figura 33-3. Conexión para el puerto serial.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

b) Estado:

Una vez hecha la conexión con el puerto correspondiente, pide que se conecte a la interfaz mediante el botón <<Conectar>>. Al no hacer este paso el usuario no puede utilizar la interfaz, cuando la interfaz se encuentre conectado correctamente en la opción de <<Estado>> le aparecerá Conectado.



Figura 34-3. Estado de conexión de la interfaz.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

c) Modo de conexión:

Existen tres modos de estar conectado en la interfaz, el usuario escoge la opción con la que desea trabajar.

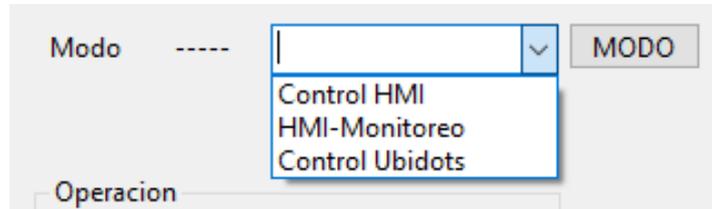


Figura 35-3. Selección para modo de conexión.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Control HMI: cuando se selecciona este modo el operador puede manipular el brazo robótico desde la interfaz gráfica, pudiendo operar ya se por juntas o el movimiento preestablecido que tiene la interfaz, además, de modificar la velocidad de movimiento y accionar el gripper. En la figura se muestra el diagrama de flujo de este modo.

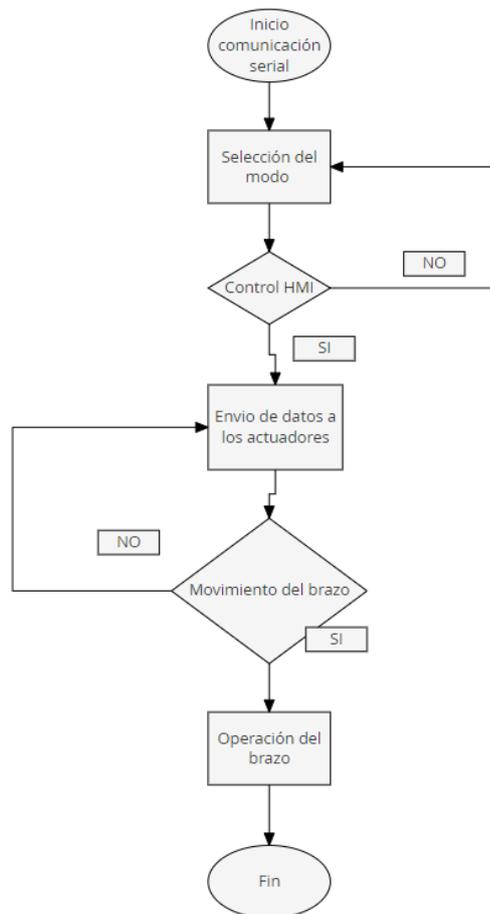


Figura 36-3. Diagrama de flujo de la programación en el modo Control-HMI.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

HMI-Monitoreo: cuando se selecciona este modo además de hacer lo que hace el modo anterior, permite visualizar las posiciones en las que se encuentra cada junta del brazo mediante la plataforma Ubidots, haciendo un monitoreo en tiempo real. En la figura se muestra el diagrama de flujo para este modo.

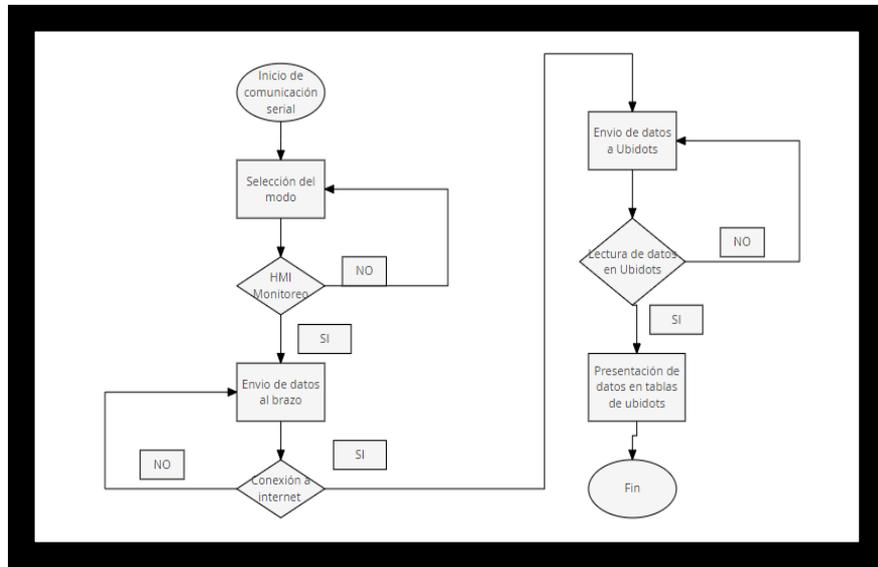


Figura 37-3. Diagrama de flujo - programación en modo HMI-Monitoreo.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Control Ubidots: cuando se selecciona este modo se puede hacer uso de la facultar de la teleoperación, esto quiere decir que se va a poder manipular el brazo desde cualquier dispositivo que tenga acceso a internet, dando las mismas prestaciones que en el modo Control HMI. En la figura se muestra el diagrama de flujo para este modo.

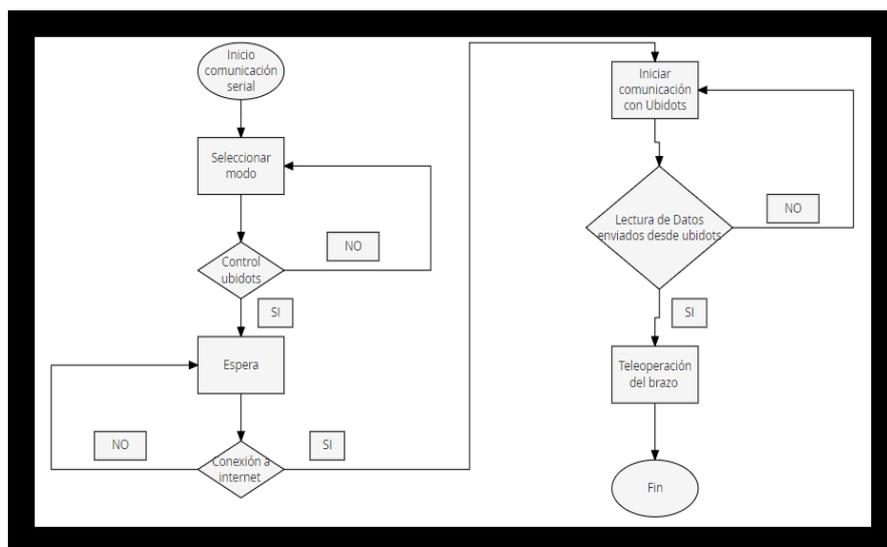


Figura 38-3. Diagrama de flujo de - programación en modo Control Ubidots.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

d) Operación:

En este apartado se muestra la opción para que el brazo robótico realice un movimiento preestablecido y secuencial, una vez que se da la opción de trabajar se debe oprimir la opción <<Pausa>> o <<Parar>> para detener el brazo robótico.

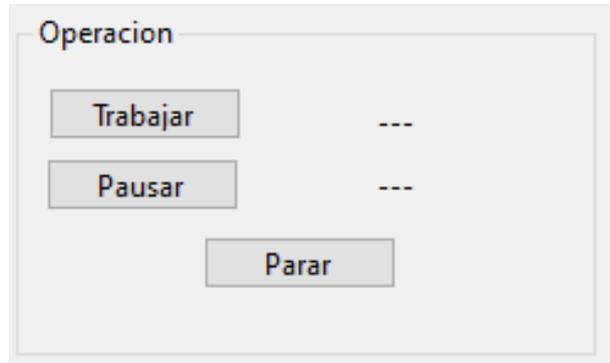


Figura 39-3. Operación trabajar.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

e) Movimiento por juntas:

Como se muestra en la figura, este apartado permite controlar el brazo de forma manual, ya sea enviando los ángulos a las que se desea mover o moviendo grado por grado a través de los botones <<+>> y <<->>



Figura 40-3. Mandos para control de forma individual.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

f) Gripper:

Para el funcionamiento del gripper se lo hace con el accionamiento de la Mini bomba y la electroválvula que son accionadas de forma separada.



Figura 41-3. Mando de accionamiento de mini bomba y gripper.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

g) Velocidad y Home

En este último apartado de la interfaz permite poner la opción de velocidad a la cual se requiere mover las juntas, así como la posición de enviar a home al brazo robótico, sin importar en la posición en la que se encuentre.



Figura 42-3. Seteo de velocidad.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

CAPÍTULO IV

4. RESULTADOS OBTENIDOS

A continuación, se detalla los resultados obtenidos de la puesta en marcha del funcionamiento del software para teleoperación y monitoreo.

4.1 Conexión inicial

La siguiente figura muestra la conexión total realiza en la interfaz, para su funcionalidad se procede a abrir el ejecutable de la interfaz y a dar las configuraciones iniciales para su inicio.

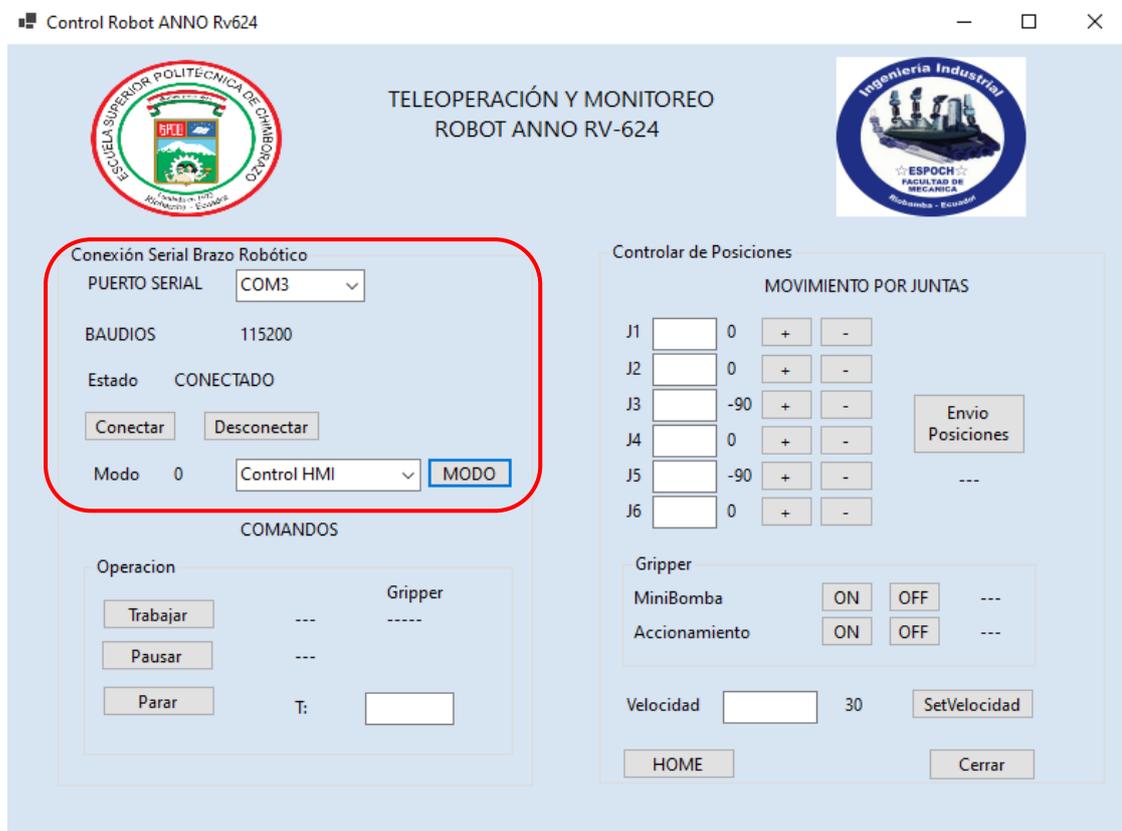


Figura 1-4. Estados para conexión de la interfaz

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

La figura muestra la configuración inicial de la interfaz para interactuar con el robot, una vez verificado el puerto, el estado de la conexión, y el modo en la cual se va a trabajar, las demás configuraciones aparecerán automáticamente, en el caso de las juntas siempre aparecerá el valor de las posiciones en home y la velocidad al 30%.

Para la conectividad entre la interfaz y la plataforma Ubidots se debe tener en cuenta el modo de conexión, una vez escogido el modo de conexión aparece una segunda ventana como muestra en la figura 28-3.

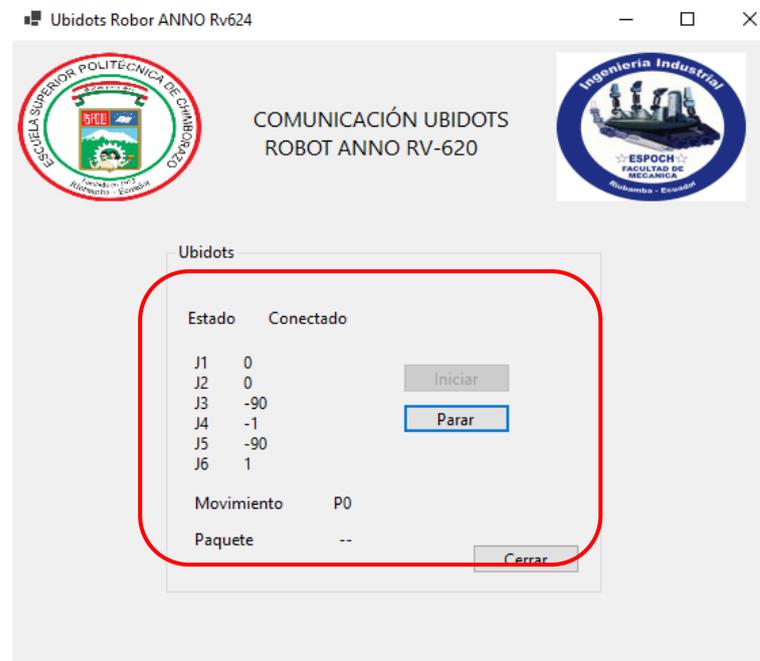


Figura 2-4. Puesta en marcha para trabajar desde Ubidots.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Al estar conectado a la plataforma Ubidots muestra como resultado la ventana en la cual aparece especificado los ángulos en las que se encuentra cada junta del brazo, una vez que se da en el botón iniciar está lista para la teleoperación desde la plataforma, como muestra de que se está recibiendo información los ángulos que sean movidos desde la plataforma se verán reflejados en dicha ventana.

4.2 Modos de operación

A continuación, se detallará el funcionamiento de cada uno de los tres modos de operación que tiene la interfaz.

4.2.1 Modo control HMI

En la figura se muestra el funcionamiento en modo CONTROL HMI, esto se lo puede hacer de dos maneras, la primera es utilizando los botones de aumento o disminución de ángulo de forma individual.



Figura 3-4. Movimiento de las juntas de forma individual.
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la figura se aprecia el movimiento de una sola junta, es así como se puede manipular las juntas una a la vez.

La segunda manera permite manipular varias juntas a la vez y en el ángulo que se desee como se muestra en la figura.



Figura 4-4. Movimiento de las juntas de forma colectiva.
 Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

A continuación, se detallará el funcionamiento de la interfaz en el modo Control HMI.

Tabla 1-4: Evaluación de respuesta del movimiento por juntas.

Movimiento por juntas				
Botón	Funcionabilidad			Observación
	Malo	Bueno	Excelente	
J1			X	La capacidad de respuesta es inmediata y efectiva.
J2			X	Tiene una respuesta inmediata después de su accionamiento
J3			X	Excelente capacidad de respuesta.
J4			X	Respuesta inmediata.
J5			X	Trabaja con mucha eficiencia.
J6			X	Su velocidad de respuesta es muy alta.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En general el movimiento por juntas y su capacidad de respuesta es excelente para todas las juntas del motor, tenemos que resaltar que con el software de fábrica no teníamos respuesta de dos de sus motores.

Tabla 2-4: Evaluación de respuesta de la operación.

Operación				
Botón	Funcionabilidad			Observación
	Malo	Bueno	Excelente	
Trabajar		X		La Capacidad de respuesta se la cataloga como buena ya que se debe esperar un tiempo aproximado de dos a tres segundos para que comience a operar como está programado.

Pausar		X		El pausar es muy útil y recomendable ya que tiene una respuesta muy rápida cuando se presiona el botón.
Parar		X		Excelente capacidad de respuesta cuando se presiona este botón, además que se puede regresar a su posición de inicio una vez parada la operación.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la operación en general su funcionamiento es el correcto y el esperado, lo cual garantiza que su funcionalidad para operar va a ser la adecuada.

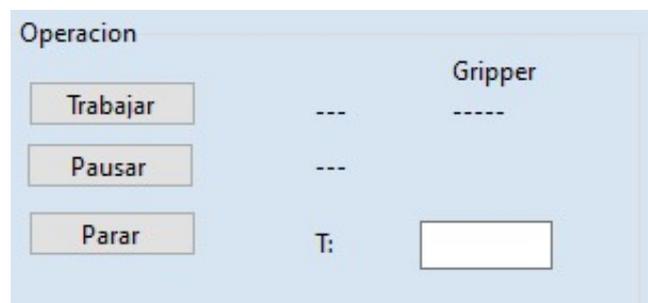


Figura 5-4. Botones para la OPERACIÓN en la interfaz.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 3-4: Evaluación de respuesta del gripper

Gripper				
Botón	Funcionamiento			Observación
	Malo	Bueno	Excelente	
Gripper			X	El gripper es el funcionamiento en conjunto entre la mini bomba y el accionamiento por lo que se cataloga con una excelente respuesta.
Mini Bomba			X	Excelente respuesta cuando se prende solo la mini bomba
Accionamiento			X	El accionamiento es muy rapido y eficaz cuando se presiona de manera independiente.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En general el funcionamiento del gripper es óptimo y adecuado por lo que no se podrá tener problemas a futuro al utilizar esta en la interfaz.

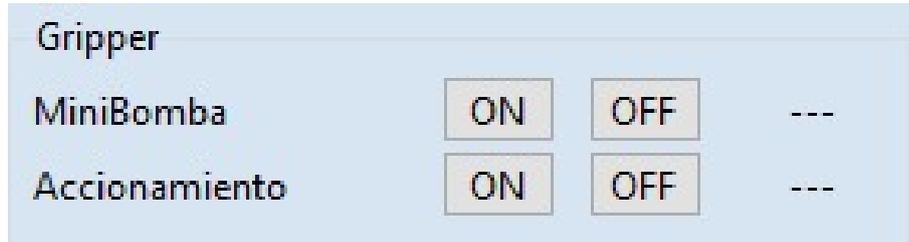


Figura 6-4. Ubicación de los botones para el gripper en la interfaz.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 4-4: Respuesta de la comunicación serial, botón home y velocidad.

Comunicación serial, home y velocidad				
Botón	Funcionamiento			Observación
	Malo	Bueno	Excelente	
Puerto de comunicación serial			X	Reconoce de manera muy rápida el puerto serial al que se va a conectar el controlador del brazo robótico.
Home			X	Excelente respuesta cuando se acciona este botón.
Velocidad			X	Una respuesta inmediata cuando se le cambia a la velocidad que se necesite.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

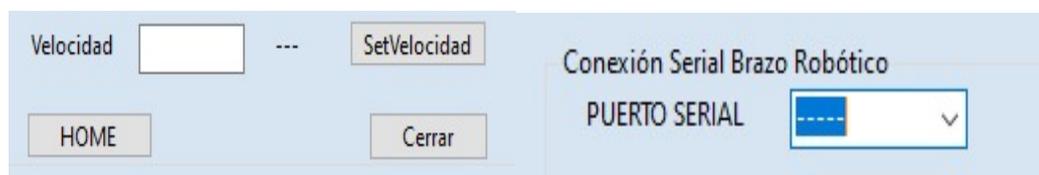


Figura 7-4. Ubicación del puerto serial, home y velocidad.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Haciendo un recuento general de la interfaz en el modo Control HMI se podría decir que su funcionamiento es el esperado y que no va a tener ningún inconveniente de uso cuando se esté ejecutando.

4.2.2 Modo HMI-MONITOREO

Cuando en nuestra interfaz se conecta a el modo HMI-MONITOREO los datos son enviados a la plataforma Ubidots en donde podremos visualizar en tiempo real el grado o ángulo de cada una de las juntas.

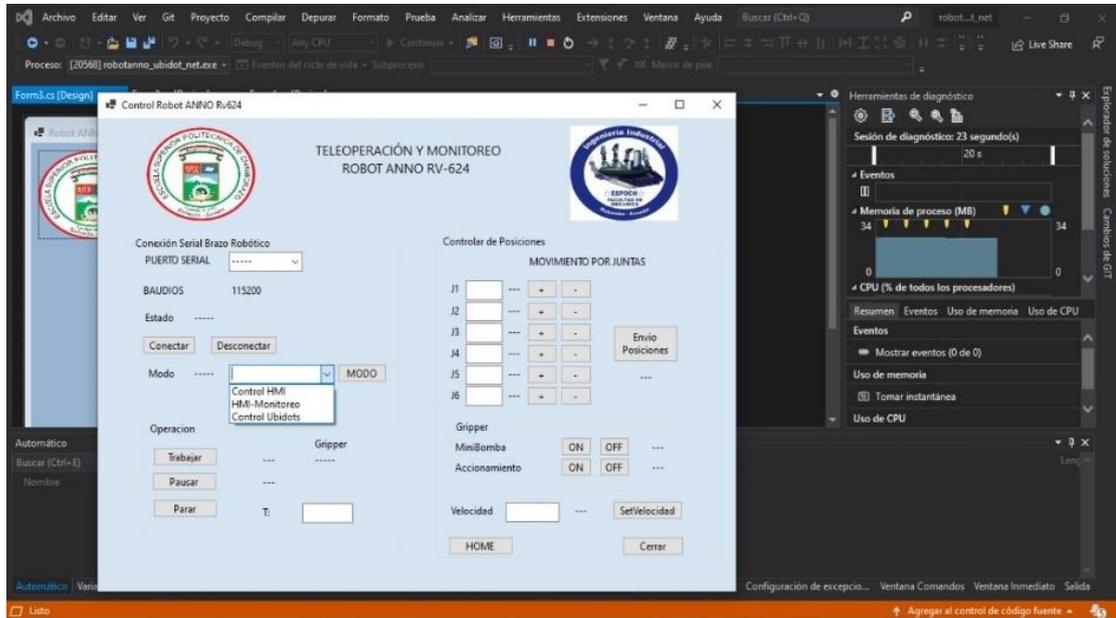


Figura 8-4. Selección del modo HMI-MONITOREO.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Cuando ya es seleccionado el modo de operación se puede manipular ya siendo enviado las posiciones manualmente o moviendo cada junta paso a paso y los datos se reflejarán en la plataforma Ubidots con lo cual estaremos haciendo un monitoreo en tiempo real y sin tener la necesidad de estar presente con el brazo robótico, lo único que se necesitara es una conexión estable a internet para poder monitorizar estos datos y estar pendiente del correcto funcionamiento del brazo robótico. Se cataloga como un funcionamiento excelente por el solo hecho que se podrá monitorizar donde y cuando sea.

Tabla de variables	
NOMBRE DE LA VARIABLE	ÚLTIMO VALOR
Lec_J1	0,00
Lec_J2	0,00
Lec_J3	-90,00
Lec_J4	0,00
Lec_J5	-90,00
Lec_J6	0,00

Figura 9-4. Monitoreo en tiempo real desde Ubidots.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Además de tener esa facilidad de monitoreo en tiempo real, la plataforma tiene la ventaja de otorgar graficas de funcionamiento con su respectiva fecha y hora, es decir que vamos a poder determinar si por algún caso se dañe algunos de los motores integradores del brazo, la fecha y hora especifica de cuando se realizó dicho daño, por lo que será de gran ayuda para poder determinar las causas o el culpable que estuvo a cargo de su funcionamiento. Además, que este reporte nos ayudara a determinar de mejor manera cuál de los motores hace más esfuerzo en el momento de la operación, con esto nosotros podremos acudir a programar mantenimientos en tiempos adecuados para tener un mejor funcionamiento del brazo robótico.



Figura 10-4. Grafica tiempo vs grado de movimiento junta 6.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la gráfica antes mencionada se puede detallar el funcionamiento de los grados movidos en una determinada fecha para la junta o el motor número 6.



Figura 11-4. Grafica tiempo vs grado de movimiento junta 5.
Realizado por: Quicaliqin Cristian, Mera Jhon, 2021

En la gráfica antes menciona se puede detallar el funcionamiento de los grados movidos en una determinada fecha para la junta o el motor número 5.



Figura 12-4. Grafica tiempo vs grado de movimiento junta 4.
Realizado por: Quicaliqin Cristian, Mera Jhon, 2021

En la gráfica antes menciona se puede detallar el funcionamiento de los grados movidos en una determinada fecha para la junta o el motor número 4.



Figura 13-4. Grafica tiempo vs grado de movimiento junta 3.
Realizado por: Quicaliqin Cristian, Mera Jhon, 2021

En la gráfica antes mencionada se puede detallar el funcionamiento de los grados movidos en una determinada fecha para la junta o el motor número 3.



Figura 14-4. Grafica tiempo vs grado de movimiento junta 2.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la gráfica antes mencionada se puede detallar el funcionamiento de los grados movidos en una determinada fecha para la junta o el motor número 2.



Figura 15-4. Grafica tiempo vs grado de movimiento junta 1.

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la gráfica antes mencionada se puede detallar el funcionamiento de los grados movidos en una determinada fecha para la junta o el motor número 1.

Las gráficas antes mencionadas reflejan los movimientos de cada uno de los motores cuando el robot está trabajando de manera continua por un tiempo determinado de 10 minutos.

4.2.3 Modo control Ubidots

Cuando seleccionamos el modo Control Ubidots podemos hacer el uso de la facultad de la teleoperación, que a continuación se la ira detallando.

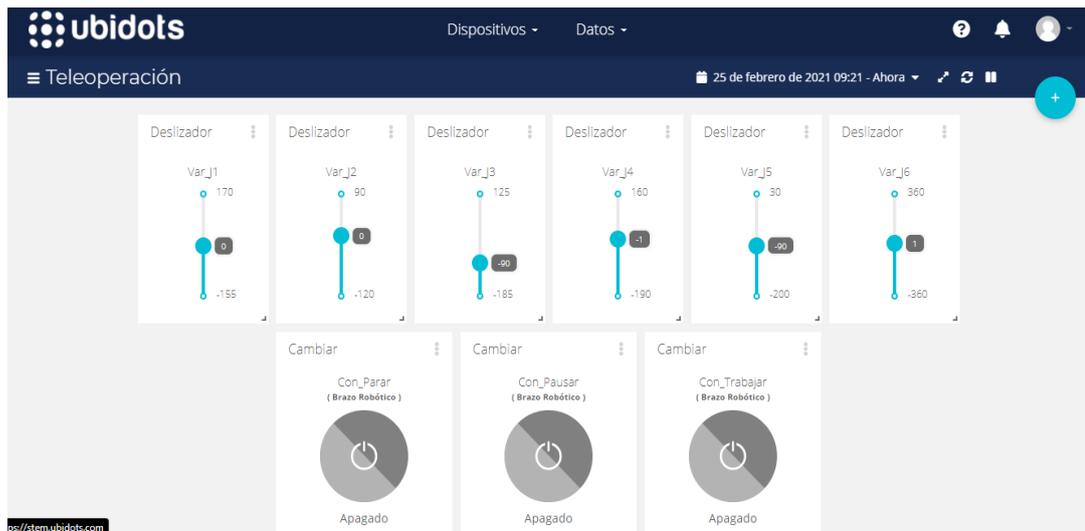


Figura 16-4. Interfaz de teleoperación.
Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

En la figura antes mostrada se tiene los controles que servirán para cumplir con la teleoperación, en donde se tiene 6 widgets llamados Slider los cuales nos van a servir para desplazar manualmente cada una de las respectivas juntas, que cave recalcar que están especificadas con un rango de valor máximo y valor mínimo con la única finalidad de no enviar datos erróneos al brazo robótico y así cuidar cada uno de los motores que podrían sufrir daños por malos usos en sus ángulos.

Para la teleoperación del modo OPERACIÓN que tiene la interfaz se tiene tres Switches denominados Trabajar, Parar y Pausar.

- Trabajar: cuando activemos este Switch desde la teleoperación el brazo robótico hará el mismo proceso que con el botón trabajar de la interfaz, con el único cambio que estaríamos operando desde cualquier dispositivo con acceso a internet.
- Pausar: cuando activemos este Switch vamos a poder pausar el movimiento que esté realizando el brazo con la ventaja de poder reiniciar la operación en el mismo punto donde se lo pausó.
- Parar: cuando accionemos este Switch la operación que esté haciendo el brazo robótico se cancelara de manera inmediata y sus motores se irán a la posición inicial o conocida como HOME.

4.3 Respuesta de la interacción del Software

Después de haber realizado las pruebas en los diferentes modos de interacción que tiene la interfaz se detalla lo siguiente.

Se hizo una toma de tiempos para determinar las velocidades en rad/s a la que se movería cada una de las juntas, se detalla esto ya que en la interfaz se puede settear en porcentajes del 10% al

80%. La toma de tiempos se la hizo con una velocidad seteada de 30% en la interfaz, se hizo un total de 50 tomas por cada junta y el desplazamiento fue de 180°. En el ANEXO D se detalla la toma completa de tiempos.

Obteniendo como resultado los siguientes tiempos promedios.

Tabla 5-4: Tiempos promedios en desplazarse 180°.

Tiempos Promedios (Segundos)					
Juntura 1	Juntura 2	Juntura 3	Juntura 4	Juntura 5	Juntura 6
7,15	4,30	5,72	7,22	6,37	14,37

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Una vez obtenido los tiempos promedio se procedió a calcular la velocidad angular con la siguiente ecuación.

$$\omega = \frac{\theta}{t} \rightarrow \omega = \frac{180^\circ}{t} \rightarrow \omega = \frac{\pi}{t}$$

Donde:

ω : Velocidad Angular (rad/s)

θ : Desplazamiento Angular (rad)

t : Tiempo (s)

Obteniendo como resultado las siguientes velocidades angulares.

Tabla 6-4: Velocidad angular de cada junta al 30% de la velocidad seteada.

Velocidad Angular 30% (rad/s)					
Juntura 1	Juntura 2	Juntura 3	Juntura 4	Juntura 5	Juntura 6
0,4396	0,7305	0,5491	0,4350	0,4935	0,2186

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Posterior a esto se procedió a realizar los cálculos para las velocidades mínima (10%) y para la máxima (80%). Obteniendo como resultado las siguientes velocidades angulares.

Tabla 7-4: Velocidades angulares mínimas.

Velocidad Angular Mínima 10% (rad/s)					
Juntura 1	Juntura 2	Juntura 3	Juntura 4	Juntura 5	Juntura 6
0,1465	0,2435	0,1830	0,1450	0,1645	0,0729

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Tabla 8-4: Velocidades angulares máximas.

Velocidad Angular Máxima 80% (rad/s)					
Juntura 1	Juntura 2	Juntura 3	Juntura 4	Juntura 5	Juntura 6
1,1723	1,9479	1,4644	1,1601	1,3160	0,5830

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Determinando que la juntura más rápida en moverse es la numero 2 con una velocidad de 1,9479 rad/s, mientras que la juntura que se desplaza a menor velocidad es la 6 con una velocidad de 0,0729 rad/s.

La tabla completa de velocidades se muestra en el ANEXO E.

Además, se realizó otra toma de datos para determinar el tiempo que se demora la interfaz en enviar las señales de cada una de las juntas y monitorizarlas en Ubidots. Se hizo un total de 30 tomas diferentes para determinar el tiempo promedio en que se demora monitorizar. La toma de los tiempos completos se muestra en el ANEXO F mientras que la gráfica de dispersión de las tomas y los errores se muestra en el ANEXO G.

Tabla 9-4: Tiempo promedio de monitoreo

Numero de tomas	tiempo promedio
30	2,62

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

El tiempo en que se demora monitorizar los datos desde Ubidots es de 2,62 segundos, siendo este un tiempo muy rápido de respuesta.

Para determinar los tiempos que se demora en enviar la señal desde Ubidots al robot, también se realizó una toma de datos, en este caso se hizo un total de 50 tomas diferentes para poder determinar el tiempo promedio en el que la señal de la teleoperación tiene respuesta. Al igual que el caso anterior la toma completa de los tiempos se presenta en el ANEXO F mientras que la gráfica de dispersión de las tomas y los errores se muestra en el ANEXO G.

Tabla 10-4: Tiempo promedio de teleoperación.

Numero de tomas	tiempo promedio
50	8,37

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

Se obtuvo un tiempo promedio de 8,37 segundos que se demora en llegar la señal desde la teleoperación al brazo robótico. Estos tiempos influirán mucho dependiendo de la calidad de conexión a internet que se tenga y la estabilidad de esta.

4.4 Gestión del proyecto

4.4.1 Cronograma

La duración de las actividades durante la elaboración de proyecto de titulación fue acorde a como se detalla en la siguiente tabla.

Tabla 11-4: Cronograma de actividades del tema de titulación.

Actividad	Mes 1				Mes 2				Mes 3				Mes 4			
	Semanas				Semanas				Semanas				Semanas			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Capacitación en el uso del brazo robótico ANNO Rv 624																
Etapa de Análisis y Recolección de la Información Necesaria																
Etapa de Selección de los medios para la elaboración de la interfaz																
Etapa de diseño del prototipo de la interfaz para control y monitoreo																
Etapa de implementación del prototipo de la interfaz para control y monitoreo																
Etapa de Análisis de resultados y pruebas																
Etapa de redacción del informe final																

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

4.4.2 Costos

4.4.2.1 Costos directos e indirectos

La siguiente tabla muestra el costo total para la implementación de la interfaz para control y monitoreo del Brazo Robótico ANNO Rv624.

Tabla 12-4: Costos de los elementos adquiridos para en anclaje del brazo.

N°	Detalle	Cantidad	Valor Unitario (\$)	Total (\$)
1	Bancada hecha en madera de PIWI	1	160,00	160,00
2	Pegamento Epóxico	1	3,00	3,00
3	Juego de llaves exagonales	1	3,00	3,00
4	Perno de 4plg	4	0,60	2,40
5	Arandelas	16	0,25	4,00
6	Llave mixta 14mm	1	4,50	4,50
			Total (USD)	176,90

Realizado por: Quicaliquin Cristian, Mera Jhon, 2021

4.4.2.2 Recurso humano

El principal recurso humano son Cristian Eliasar Quicaliquin Jogacho y Jhon William Mera Reascos, autores del presente proyecto. Gracias a la metodología empleada y el compromiso con el trabajo, permitieron alcanzar los objetivos propuesto y la meta del trabajo de titulación.

4.4.2.3 Recurso Material

El software y plataforma utilizados fueron presentados anteriormente, dando algunas alternativas para la programación y también para la teleoperación, estas pueden ser utilizadas de acuerdo con su comodidad y requerimientos. Los costos fueron aceptados por los estudiantes autores del trabajo, con el fin de presentar una alternativa para la creación de un software que pueda servir para crear una interfaz de teleoperación y monitoreo de un brazo robótico, a su vez para servir de base para futuros trabajos a implementar en la mejora de la teleoperación y monitoreo, para lograr un impacto significativo en la mentalidad de los estudiantes y en el área de investigación.

CONCLUSIONES

Al utilizar el Software desarrollado se tiene un control mejorado del Robot ANNO Rv 624, permitiendo una mejor interacción con el humano, este a su vez presenta tres modos para la manipulación del Robot las cuales son: Control HMI, HMI-Monitoreo y Control Ubidots, mismas que ofrece diferentes modos de trabajo sin salir de la zona de confort del operador.

Al realizar el diagnóstico se determinó que en el software original trae muchas limitantes al momento de realizar la operación, también presenta fallos en la comunicación con algunas de las juntas como son la junta 2, 1 y en ocasiones en la junta 3, estos errores hacen que muchas de las veces sea imposibles manipular el robot.

Con la utilización de plataformas hardware y software se logró desarrollar una interfaz prototipo para la teleoperación y monitoreo del Robot ANNO Rv624, esto fue desarrollado por medio de la programación en Visual Basic que en combinación con las librerías DLL facilitan el manejo y operación de las juntas del brazo robótico para poder dar movimiento de este.

Se realizó una interfaz para monitoreo del robot haciendo uso de la plataforma Ubidots, en este caso por la facilidad de diseño, haciendo como soporte de la aplicación el Software Visual Basic y como él envió de datos al controlador del brazo robótico, el mismo que se encarga de procesar la información recibida y gestionar las señales de control hacia los drivers de los actuadores o también llamadas juntas que son utilizadas para generar el movimiento.

Se hicieron pruebas de la interacción del Robot ANNO Rv624 consiguiendo así una funcionalidad del 100% en envió de datos para la ejecución de los movimientos, configuración desde la interfaz gráfica y monitoreo mediante la plataforma Ubidots.

En cuanto al monitoreo y la teleoperación al momento de enviar información desde la plataforma Ubidots hacia la interfaz y viceversa, se tuvo una demora de un tiempo promedio de 2.62 segundos para el monitoreo y 8.37 segundos para la teleoperación debido a la conexión, estabilidad y calidad de la red.

La comprensión para utilizar el nuevo software es mucho mejor, debido a que toda su interfaz esta detallada en español idioma entendido por la mayoría de los operarios a nivel nacional, así dejamos atrás lo tedioso que es manejar un idioma desconocido para interactuar con el robot.

Gracias a la adaptación de una bancada para que nos ayude con el soporte del brazo robótico se hace más fácil el trabajo de la persona que lo va a operar, ya que ahora el brazo esta empernado a la mesa y ya no es necesario más de una persona para poner en marcha su funcionamiento.

En la sección operación el robot tiene un método de trabajo continuo el cual consiste en agarrar un objeto de un lugar determinado y llevarlo a otro totalmente distinto hasta que el usuario tenga la necesidad de pausar la operación con opción a reiniciar en el mismo punto de donde se pauso o en tal caso parar de forma tajante y que vuelva a su posición de home o inicio.

El monitoreo en tiempo real de los grados de cada una de las juntas a través de Ubidots es de mucha ayuda ya que se puede determinar graficas en donde vamos a poder visualizar la fecha exacta y como fueron fluctuando cada una de sus juntas dependiendo la operación que se esté ejecutando con el brazo robótico.

RECOMENDACIONES

Al utilizar la interfaz original deben cumplir con todas las recomendaciones dadas por el distribuidor como son: hacer uso del software con sistema operativo Windows 7 o inferior, para mejorar esto se recomienda instalar el driver antes mencionado.

Siempre al iniciar o finalizar la operación del brazo tener en cuenta la posición en la que se encuentra el brazo, debido a que una vez que se inicie la operación con el software original, esa será su posición inicial y pasará a ser el home hasta tener un nuevo seteo.

Tomar como base la programación de la interfaz del presente proyecto e implementar otro tipo de monitoreo o teleoperación, ya que al haber utilizado un software de libre uso da la posibilidad de innovar y continuar con la mejora continua.

Para teleoperar de manera óptima a través de la plataforma Ubidots, se recomienda tener una buena señal de red, ya que si pierde conexión con la red podría ocasionar daños en los actuadores u ocasionar algún tipo de accidente.

Antes de utilizar la interfaz para la operación, teleoperación y monitoreo es recomendable leer el manual de uso que se deja plasmado en este trabajo de titulación, además de las condiciones de uso planteadas por los fabricantes del brazo.

GLOSARIO

C#: (leído en inglés "C Sharp" y en español "C Almohadilla") es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella. (González Seco, 2002)

Controlador: Controladores En un sistema de control el controlador es el "cerebro" del circuito, esté es el que toma la decisión de llevar una determinada acción a cabo. (Jiménez Ceciliano, 2014)

DLL: Son códigos realizados que pueden ser cargados o usados por otros programas, estas librerías pueden ser compartidas entre varios programas.

Sistema informático: es un sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático. El hardware incluye computadoras o cualquier tipo de dispositivo electrónico, que consisten en procesadores, memoria, sistemas de almacenamiento externo. (Quezada Haro, 2019)

BIBLIOGRAFÍA

ALENCASTRES MIRANDA, Moisés, et al. *Teleoperating Robots in Multiuser Virtual Environments*. México : ENC, 2003. p. 4

ANNO. *Robotic Arm Body*. [En línea] 12 de enero de 2018. [Consulta: 11 de febrero de 2021]. Disponible en: <http://www.robotanno.com/en/educational-robotic-arm/850b-robotic-arm-body.html>.

BAHAMONDE CHICAIZA, Darwin Javier. Diseño de la Red Internet de la Cosas (IOT) en la Empresa Roger Sport. Quito. [En línea] (Trabajo de titulación) Universidad Politécnica Salesiana, Pichincha-Ecuador, 2020. [Consulta: 2020-12-20] Disponible en: <https://dspace.ups.edu.ec/handle/123456789/18932>

BARRIENTOS, Antonio; et al. *Fundamentos de Robótica*. Madrid : McGraw-Hill, 2007. ISBN 8448108159, pp. 5 - 8.

BOGADO TORRES, Juan Manuel. Control bilateral de robots teleoperados por convergencia de estados. [En línea] (Tesis Doctoral) Universidad Politécnica de Madrid, Madrid-España, 2007. [Consulta: 2020-12-20]. Disponible en: <https://1library.co/document/y9g3djrj-control-bilateral-robots-teleoperados-por-convergencia-estados.html>

BURBANO VALLEJO, Leonardo Francisco. 2017. Estudio e implementación en matlab de un entorno de comunicación basado en protocolos del internet de las cosas para clientes de teleoperación en robótica. [En línea] (Trabajo de titulación) Escuela Politécnica Nacional, Pichincha, Ecuador. 2017. p. 213. [Consulta: 2020-12-11]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/17470>

CEBALLOS SIERRA, Francisco Javier. *Enciclopedia de microsoft visual basic. interfaces gráficas y aplicaciones para internet con windows forms y asp.NET*. Madrid-España: RA-MA, 2013. ISBN 9788499642659, pp. 4 - 10.

CEBALLOS SIERRA, Javier. *Visual Basic .NET Lenguaje y aplicaciones*. Madrid-España : RA-MA, 2017. ISBN 9788494009013.

CUCARO, Olga María Stefania. *Fácil programación con Visual Basic (VB): Del diagrama de flujo al programa en Visual Studio VB*. 2021. ISBN 9791220261470.

EL COMERCIO. *Tecnología Sony, Robótica.* [En línea] 01 de noviembre de 2017. [Consulta: 14 de noviembre de 2020.] Disponible en: <https://www.elcomercio.com/guaifai/tecnologia-sony-robotica-perro-aibo.html>.

ESCANDÓN CUEVA, Rodrigo Andrés. 2013. Interfaz de control para un Brazo Robot Articulado Basado en Software de Desarrollo Integral. [En línea] (Trabajo de titulación) Universidad Politécnica Salesiana Sede Matriz Cuenca, Cuenca Ecuador, 2013. p. 8. [Consulta: 2020-01-15]. Disponible en: <https://dspace.ups.edu.ec/handle/123456789/4646>

GÓMEZ DE GABRIEL, Jesús Manuel; et al. *Teleoperación y Telerrobótica.* Madrid : Pearson Education, 2006. ISBN 8483222965.

GONZÁLEZ SECO, José Antonio. *El lenguaje de programación del C#.* Madrid-España, 2002, pp. 21-26.

HERNÁNDEZ GANÁN, Karla Daniela, & PANCHI GUAMANGALLO, Freddy Manuel. 2017. Diseño, simulación e implementación de un sistema de teleoperación para un robot móvil tipo Carlike [En línea] (Trabajo de titulación). Escuela Politécnica Nacional, Pichincha, Ecuador. 2017. p. 165. [Consulta: 2020-12-10]. Disponible: <http://bibdigital.epn.edu.ec/handle/15000/18975>

JIMÉNEZ CECILIANO, Pablo Javier. Conceptos de controladores PID e implementación en un horno de fundición de acero [En línea] diciembre 2014. [Consulta: 2021-03-03]. Disponible en: https://www.researchgate.net/publication/291072181_Conceptos_de_controladores_PID_e_implementacion_en_un_horno_de_fundicion_de_acero

LÁJARA VIZCAÍNO, José Rafael; & SEBASTIÁ, José Pelegí. *LabVIEW entorno gráfico de programación.* Mexico : Marcombo-España, 2008. ISBN 978-970-15-1133-6, pp. 21, 22.

LLUMIQUINGA PUMISACHO, Christian Mauricio, & GALARZA GUERRERO, David Sebastián. Teleoperación del Robot Nao para la ejecución de tareas en espacios reducidos usando el dispositivo Kinect v2. [En línea] (Trabajo de titulación). Universidad Politécnica Salesiana, Pichincha, Ecuador. 2018. p. 66. [Consulta: 2020-12-10]. Disponible en: <https://dspace.ups.edu.ec/handle/123456789/15287>

UBIDOTS. *Ubidots.* [En línea] 7 de Mayo de 2016. [Consulta: 15 de Enero de 2021.] <https://ubidots.com/about/>.

MOORE, Holly. *MATLAB para ingenieros.* Naucalpan de Juarez-México : Pearson Educación, 2007. ISBN 9789702610823.

MORENO MUÑOZ, Alfredo; & CÓRCOLES CÓRCOLES, Sheila. *Arduino Curso Práctico*. Madrid-España : RA-MA Editorial, 2018. ISBN 9788499647456, pp. 29-31.

MUELA GUAICHA, Stalin Rene. Teleoperación con realimentación de fuerza para una formación de robots manipuladores aéreos basados en algoritmos de control distribuido. [En línea] (Trabajo de titulación) Escuela Politécnica Nacional, Pichincha, Ecuador. 2020. [Consulta: 2020-12-12] Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/21068>

PORTAL POLITICO. *Ciencia y tecnología*. [En línea] 07 de agosto de 2018. [Consulta: 14 de noviembre de 2020.] Disponible en: <https://www.portalpolitico.tv/ciencia-y-tecnologia/desarrollan-robots-milimetricos-y-flexibles-con-potencial-en-la-medicina>.

PUTIER, Sébastien. *C# 7 y Visual Studio 2017: Los fundamentos del lenguaje*. Barcelona-España: Ediciones ENI, 2018. ISBN 9782409013485, pp. 41, 42

QUEVEDO PÉREZ, Washington Xavier. Implementar un esquema de tele-operación para un robot manipulador móvil maniobrado a través de dispositivos hápticos, para incrementar la transparencia del sitio remoto, a través de entornos de realidad virtual y realidad aumentada, en el sitio local [En línea] (Trabajo de titulación) Universidad de las Fuerzas Armadas ESPE, Tungurahua, Ecuador. 2016. p. 71. [Consulta: 2020-12-10] Disponible en: <http://repositorio.espe.edu.ec/handle/21000/11977>

QUEZADA HARo, Patricio Alexander. Elaboración e implementación de un sistema automatizado de gestión de inventarios y accesos del personal en las bodegas de almacenamiento de la empresa vilema imev en el cantón guano [En línea] (Trabajo de titulación) Escuela Superior Politécnica de Chimborazo, Chimborazo-Ecuador. 2019. [Consulta: 2021-03-01]. Disponible en: <http://dspace.espech.edu.ec/handle/123456789/11397>

Quiñonez Muñoz, Oswaldo. *Internet de las cosas (IoT)*. España : Ibukku LLC, 2019. ISBN 9781640864320.

REVISTAIMG. *Novedades tecnológicas*. [En línea] noviembre de 2019. [Consulta: 14 de noviembre de 2020.] Disponible en: <https://www.revistaimg.com/kr-quantec-conoce-al-primer-robot-industrial-del-mundo/>.

RIGOR, ACADEMIC. *The Conversation*. [En línea] 11 de abril de 2019. [Consulta: 14 de noviembre de 2020.] Disponible en: <https://theconversation.com/detengamos-a-los-robots-asesinos-antes-de-que-existan-114666>.

SAHA, Subir Kumar. *Introducción a la Robótica.* Mexico D. F. : McGraw-Hill, 2008. ISBN: 9786071503138, pp. 38 - 49.

SHOWROOM. *Robots de aseo.* [En línea] 01 de julio de 2019. [Consulta: 14 de noviembre de 2020.] Disponible en: https://as.com/showroom/2018/04/13/portada/1523605115_515517.html.

SLAWIŃSKI, Emanuel, et al. *Estructura Abierto de Software para un Robot Industrial.* 3, San Juan, 2007, Vol. 4. ISBN 16977912, pp. 1 - 10.

STANDARDIZATION INTERNATIONAL ORGANIZATION. *La nueva revolución industrial.* Suiza, 2018. ISSN 23107987. p. 2

TERÁN MERCHÁN, Daniel Lizardo. Diseño e implementación de un sistema de teleoperación y monitoreo para la estación de medición del grado de asentamiento de sedimentos y agua del petróleo [En línea] (Trabajo de titulación) Universidad de las Fuerzas armadas ESPE, Pichincha, Ecuador. 2015. p. 139. [Consulta: 2020-12-10]. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/11178>

WOOLLASTON, Victoria. *Los mejores robots espaciales del pasado, presente y futuro.* [En línea] 03 de Agosto de 2020. [Consulta: 12 de Diciembre de 2020.] Disponible en: <https://www.pocket-lint.com/es-es/gadgets/noticias/147685-robots-espaciales>.

ANEXOS

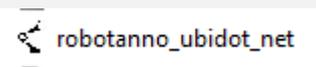
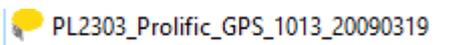
ANEXO A: Condiciones de uso para el robot anno RV624

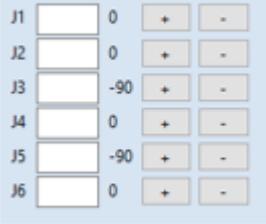
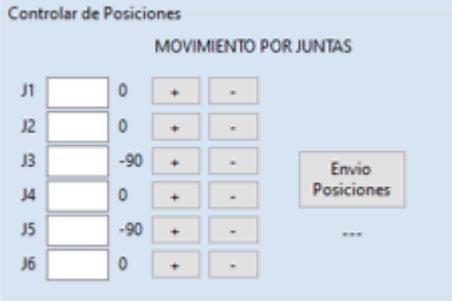
CONDICIONES DE USO	
SECCION	DETALLE
Elementos peligrosos	<p>a) La caja del controlador cuenta con un botón de paro de emergencia que puede ser utilizada al inmediatamente en caso de fallo o cualquier otra causa. Si el robot no se puede detener a tiempo, puede causar lesiones.</p> <p>b) Una vez que se solucione el paro de emergencia, el botón de paro debe volver a su posición original para que se pueda alimentar nuevamente los motores de paso.</p>
Precauciones	<p>Al utilizar Robot Anno los siguientes elementos deben comprobarse antes del brazo del robot. Si hay alguna anomalía, debe repararse o tomarse a tiempo.</p> <ol style="list-style-type: none">1) Si los cables eléctricos están conectados correctamente.2) Si la cubierta del cable eléctrico está dañada.3) Si el interruptor de parada de emergencia está en estado liberado.4) Si existe alguna anomalía o ruido anormal en la acción del brazo mecánico. <p>El propietario y el operador del brazo robótico deben ser responsables de su propia seguridad, además de utilizar el equipo de seguridad correspondiente.</p> <p>No modifique el brazo robótico sin autorización.</p> <p>Cuando operen dos o más personas, designe claramente a la persona responsable de la supervisión.</p>
Entorno de uso inconveniente	<p>No coloque el brazo robótico en un entorno hostil. La suciedad, los desechos y las altas temperaturas pueden dañar los componentes internos.</p>

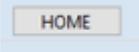
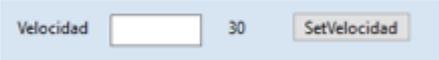
	<p>Después de usar el brazo robótico, desenchufe el cable de alimentación y coloque el brazo robótico en un lugar seco y con temperatura ambiente.</p> <p>No se puede utilizar en las siguientes situaciones RobotAnno Brazo robotico:</p> <ol style="list-style-type: none"> 1) Si hay cerca sustancias inflamables 2) Ambiente con potencial explosivo 3) En agua u otros líquidos 4) En un ambiente con gas corrosivo 5) En lugares con temperaturas mayor a 40 grados Celsius.
<p>Operativos de seguridad.</p>	<p>Al controlar el movimiento del brazo:</p> <ol style="list-style-type: none"> 1) Antes de operar el brazo robótico, utilice una velocidad de anulación más baja para depurar el movimiento del brazo robótico. 2) Antes de presionar el botón de encendido, considere la tendencia de movimiento del brazo robótico. 3) Es necesario considerar evitar la trayectoria de movimiento del brazo robótico y confirmar que la línea no está interferida. 4) El área alrededor del brazo robótico debe estar limpia, libre de aceite, agua e impurezas. <p>Operación de producción:</p> <ol style="list-style-type: none"> 1) Antes de la puesta en marcha y funcionamiento, debe conocer todas las tareas que realizará el brazo robótico según lo programado. 2) Es necesario conocer la posición y el estado de todos los interruptores, sensores y señales de control que afectarán el movimiento del brazo robótico. 3) Es necesario conocer la posición del botón de parada de emergencia en el dispositivo de control del brazo robótico y prepararse para situaciones de peligro.

	<p>4) Nunca asuma que el brazo robótico ha completado su programa sin moverlo porque el brazo robótico puede estar esperando una señal de entrada.</p>
<p>Mantenimiento y almacenamiento de rutina</p>	<p>El brazo robótico es seguro de usar y puede adaptarse al medio ambiente en la mayor medida posible. Siga las instrucciones cuando lo use.</p> <ol style="list-style-type: none"> 1) Nunca tire a la fuerza del eje del brazo robótico, de lo contrario podría causar lesiones personales y daños al equipo. 2) Evite estar debajo 20 grados Celsius o más de 40 grados Celsius. 3) Evite colocarlo a la luz solar directa durante mucho tiempo. 4) Evite estar en un ambiente fangoso y polvoriento. 5) Manténgase alejado de entornos con fuertes vibraciones. 6) Mantener alejado de un ambiente con electricidad estática.

ANEXO B: Manual de uso para la teleoperación y monitoreo.

<p>Manual de uso del software de teleoperación y monitoreo del robot ANNO Rv624</p>	
<p>Ejecutar la aplicación robotanno_ubidots_net, esta se encontrará en la carpeta del trabajo de titulación</p>	
<p>Conectar el controlador a la computadora mediante el cable USB. (si no reconoce el dispositivo instalar el driver PL2303</p>	

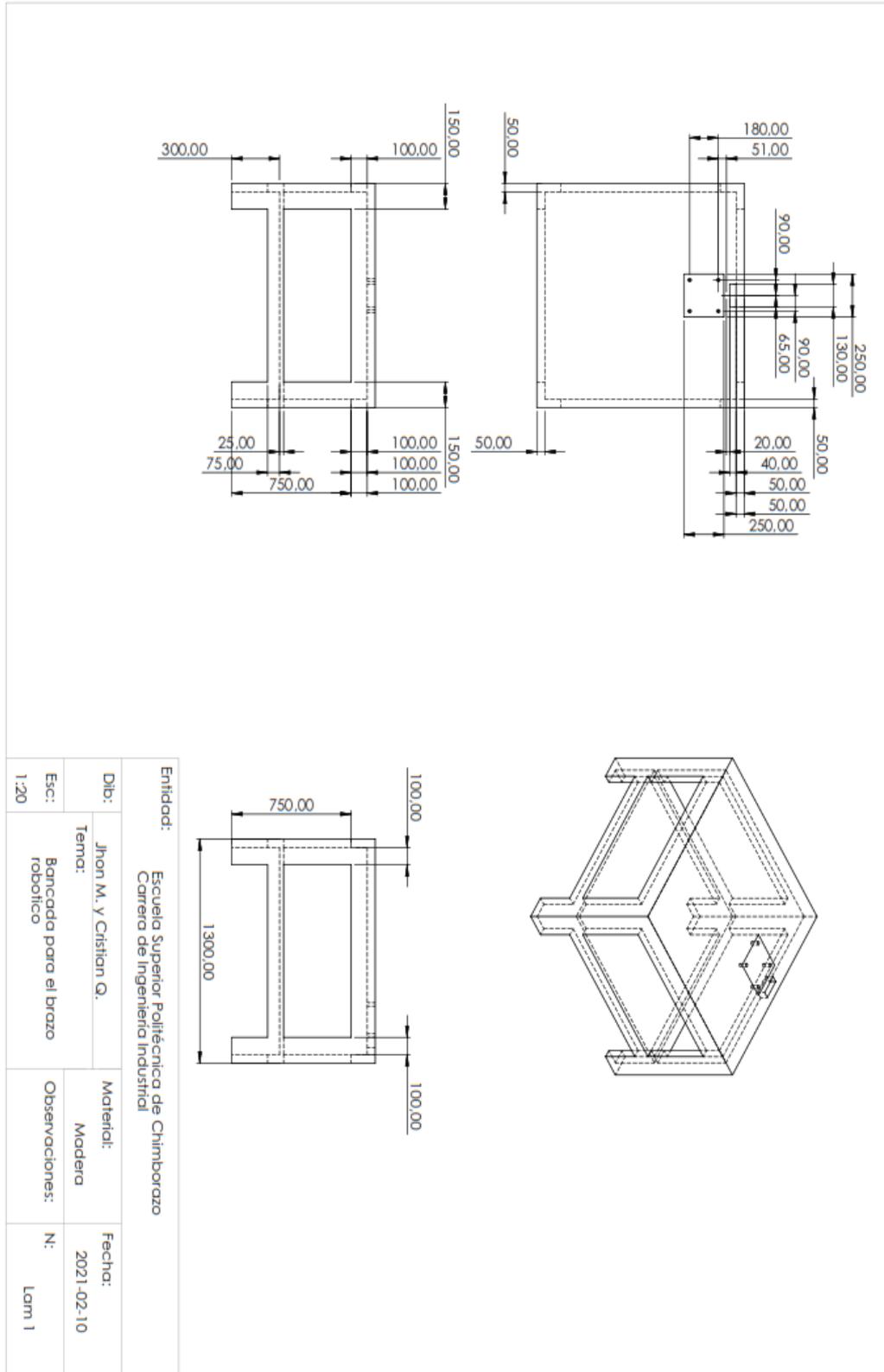
<p>Prolific GPS 1013 20090319 que se encuentra en la misma carpeta.)</p>	
<p>Seleccionar el puerto de comunicación.</p>	<p>Conexión Serial Brazo Robótico</p> <p>PUERTO SERIAL <input type="text" value="-----"/></p>
<p>Presionar el botón Conectar, y la interfaz estará lista para funcionar.</p>	<p>Estado -----</p> <p><input type="button" value="Conectar"/> <input type="button" value="Desconectar"/></p>
<p>Seleccionar el modo de operación y presionar el botón MODO.</p>	<p>Modo ----- <input type="text" value=""/> <input type="button" value="MODO"/></p> <p><input type="checkbox"/> Operacion</p> <p>Control HMI HMI-Monitoreo Control Ubidots</p>
<p>Modo CONTROL-HMI: Cuando selecciona el modo control-hmi el usuario puede manipular la interfaz e interactuar con el robot</p>	
<p>Se puede interactuar con el robot de manera manual accionando la junta que requiera mover presionando los botones más y menos dependiendo la dirección de giro que necesite.</p>	
<p>Enviar a una posición determina de los seis motores, esto se puede hacer escribiendo en el text box el grado al que se requiere que los motores giren y luego presionar ENVIO POSICIONES.</p>	
<p>La interfaz está programada para que el robot haga una acción determinada en un ciclo repetitivo presionando el botón trabajar además de que se puede parar y pausar cuando se necesite.</p>	<p>Operacion</p> <p><input type="button" value="Trabajar"/></p> <p><input type="button" value="Pausar"/></p> <p><input type="button" value="Parar"/></p>

<p>Se puede accionar manualmente el gripper con los botones on/off</p>															
<p>El brazo robótico esta seteado a una posición de inicio o HOME, cuando se presiona este botón el brazo inmediatamente se va a una posición inicial.</p>															
<p>Se puede cambiar a la velocidad que se desee que los motores funcionen escribiendo la velocidad deseada en un porcentaje de 10-80% y presionando el botón SETVELOCIDAD</p>															
<p>Modo HMI-MONITOREO</p>															
<p>Cuando este en este modo se podra hacer exactamente lo mismo que en el modo Control-HMI con la diferencia que monitorizara en tiempo real el angulo de cada una de las juntas y se presentara en una tabla en la plataforma Ubidots</p>	 <table border="1" data-bbox="863 958 1270 1391"> <thead> <tr> <th>NOMBRE DE LA VARIABLE</th> <th>ÚLTIMO VALOR</th> </tr> </thead> <tbody> <tr><td>Lec_J1</td><td>0,00</td></tr> <tr><td>Lec_J2</td><td>0,00</td></tr> <tr><td>Lec_J3</td><td>-90,00</td></tr> <tr><td>Lec_J4</td><td>0,00</td></tr> <tr><td>Lec_J5</td><td>-90,00</td></tr> <tr><td>Lec_J6</td><td>0,00</td></tr> </tbody> </table>	NOMBRE DE LA VARIABLE	ÚLTIMO VALOR	Lec_J1	0,00	Lec_J2	0,00	Lec_J3	-90,00	Lec_J4	0,00	Lec_J5	-90,00	Lec_J6	0,00
NOMBRE DE LA VARIABLE	ÚLTIMO VALOR														
Lec_J1	0,00														
Lec_J2	0,00														
Lec_J3	-90,00														
Lec_J4	0,00														
Lec_J5	-90,00														
Lec_J6	0,00														
<p>Modo CONTROL UBIDOTS: en este modo se ejecutara la teleoperación</p>															
<p>Moviento por juntas, se lo realizara a traves de los Sliders que estan en la plataforma Ubidots</p>															
<p>Tambien se puede teleoperar el movimiento de ciclo repetitivo.</p>															

Ademas de poder accionar el gripper de forma teleoperada.



ANEXO C: Planos de la bancada.



ANEXO D: Toma de tiempos para un desplazamiento de 180°

		Muestra de tiempos para un desplazamiento de π rad (180°)					
		Juntura					
		Juntura 1	Juntura 2	Juntura 3	Juntura 4	Juntura 5	Juntura 6
N ú m e r o d e T o m a	1	7,12	4,26	5,65	7,30	6,43	14,55
	2	7,02	4,33	5,73	7,11	6,31	14,16
	3	7,18	4,25	5,86	7,20	6,30	14,50
	4	7,28	4,39	5,70	7,20	6,28	14,35
	5	7,15	4,29	5,69	7,15	6,36	14,22
	6	7,11	4,31	5,70	7,22	6,33	14,36
	7	7,23	4,35	5,71	7,25	6,37	14,19
	8	7,14	4,25	5,73	7,19	6,35	14,35
	9	7,22	4,29	5,80	7,18	6,39	14,44
	10	7,13	4,32	5,76	7,25	6,38	14,36
	11	7,11	4,26	5,66	7,19	6,39	14,43
	12	7,19	4,35	5,68	7,20	6,39	14,37
	13	7,20	4,37	5,72	7,19	6,41	14,41
	14	7,09	4,33	5,67	7,21	6,37	14,50
	15	7,11	4,32	5,73	7,20	6,37	14,36
	16	7,12	4,26	5,79	7,21	6,41	14,39
	17	7,15	4,30	5,64	7,23	6,37	14,46
	18	7,22	4,28	5,69	7,25	6,41	14,39
	19	7,08	4,36	5,65	7,24	6,38	14,48
	20	7,14	4,28	5,73	7,26	6,40	14,38
	21	7,11	4,27	5,81	7,21	6,41	14,26
	22	7,17	4,29	5,65	7,20	6,34	14,36
	23	7,14	4,28	5,70	7,20	6,35	14,35
	24	7,16	4,26	5,77	7,20	6,38	14,29
	25	7,08	4,28	5,68	7,21	6,39	14,37
	26	7,21	4,29	5,72	7,25	6,42	14,26
	27	7,10	4,26	5,76	7,26	6,34	14,42
	28	7,15	4,27	5,68	7,20	6,40	14,29
	29	7,22	4,33	5,76	7,21	6,36	14,37
	30	7,16	4,29	5,71	7,25	6,43	14,36
	31	7,11	4,38	5,69	7,21	6,31	14,42
	32	7,19	4,27	5,71	7,24	6,30	14,42
	33	7,07	4,35	5,67	7,25	6,28	14,38
	34	7,09	4,31	5,69	7,26	6,37	14,38
	35	7,12	4,28	5,73	7,20	6,41	14,36
	36	7,08	4,33	5,76	7,24	6,31	14,41
	37	7,21	4,26	5,68	7,29	6,38	14,29
	38	7,18	4,36	5,71	7,24	6,32	14,35
	39	7,09	4,29	5,66	7,28	6,36	14,36
	40	7,14	4,37	5,71	7,26	6,34	14,29
	41	7,16	4,26	5,69	7,21	6,39	14,38
	42	7,10	4,28	5,73	7,19	6,38	14,42
	43	7,05	4,27	5,77	7,25	6,34	14,30
	44	7,16	4,26	5,81	7,19	6,41	14,39
	45	7,12	4,28	5,69	7,28	6,37	14,41
	46	7,24	4,30	5,81	7,19	6,40	14,35
	47	7,17	4,31	5,79	7,20	6,35	14,44
	48	7,21	4,28	5,72	7,20	6,38	14,39
	49	7,14	4,35	5,76	7,21	6,29	14,37
	50	7,18	4,28	5,74	7,26	6,39	14,36
	Promedio	7,15	4,30	5,72	7,22	6,37	14,37

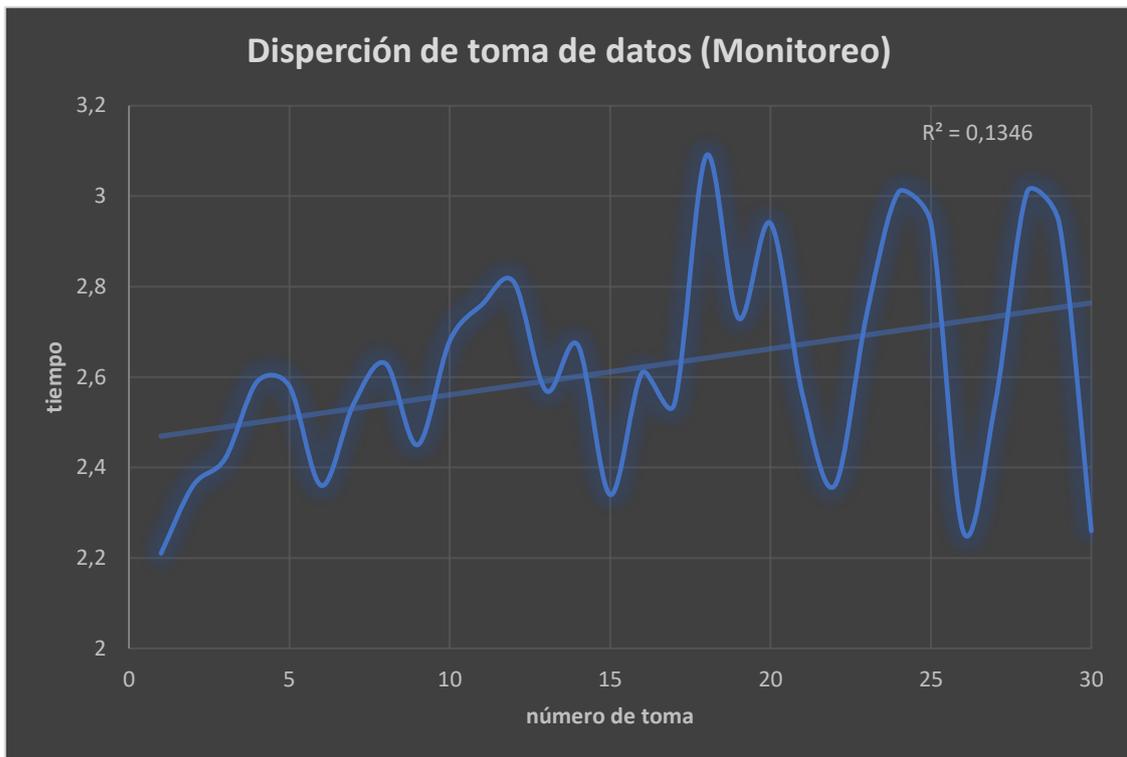
ANEXO E: Velocidad de cada junta a distintos porcentajes.

		Velocidad en rad/s de cada junta a distintos porcentajes						
		Juntura 1	Juntura 2	Juntura 3	Juntura 4	Juntura 5	Juntura 6	
Porcentaje de velocidad en la interfaz	10%	0,1465	0,2435	0,1830	0,1450	0,1645	0,0729	Vmin
	20%	0,2931	0,4870	0,3661	0,2900	0,3290	0,1458	
	30%	0,4396	0,7305	0,5491	0,4350	0,4935	0,2186	
	40%	0,5862	0,9740	0,7322	0,5801	0,6580	0,2915	
	50%	0,7327	1,2174	0,9152	0,7251	0,8225	0,3644	
	60%	0,8793	1,4609	1,0983	0,8701	0,9870	0,4373	
	70%	1,0258	1,7044	1,2813	1,0151	1,1515	0,5102	
	80%	1,1723	1,9479	1,4644	1,1601	1,3160	0,5830	Vmax

ANEXO F: Tiempos de respuesta para teleoperar y monitorizar.

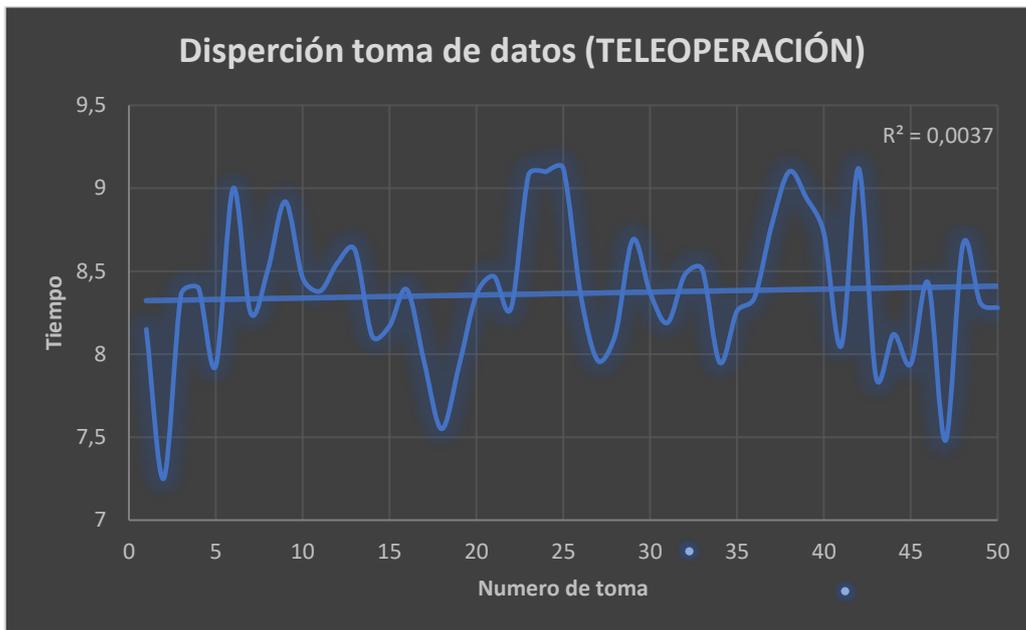
Tiempo de respuesta en teleoperar		Tiempo de respuesta en monitorizar	
Numero de toma	Tiempo	Numero de toma	Tiempo
1	8,15	1	2,21
2	7,25	2	2,36
3	8,36	3	2,42
4	8,4	4	2,59
5	7,93	5	2,58
6	9	6	2,36
7	8,25	7	2,54
8	8,51	8	2,63
9	8,92	9	2,45
10	8,46	10	2,68
11	8,38	11	2,76
12	8,55	12	2,81
13	8,63	13	2,57
14	8,11	14	2,67
15	8,17	15	2,34
16	8,39	16	2,61
17	7,95	17	2,54
18	7,55	18	3,09
19	7,93	19	2,73
20	8,36	20	2,94
21	8,47	21	2,56
22	8,28	22	2,36
23	9,08	23	2,74
24	9,1	24	3,01
25	9,12	25	2,94
26	8,36	26	2,26
27	7,96	27	2,54
28	8,12	28	3,01
29	8,69	29	2,94
30	8,37	30	2,26
31	8,19	Promedio	2,62
32	8,48		
33	8,51		
34	7,95		
35	8,26		
36	8,34		
37	8,78		
38	9,1		
39	8,94		
40	8,73		
41	8,05		
42	9,12		
43	7,86		
44	8,12		
45	7,94		
46	8,43		
47	7,48		
48	8,66		
49	8,31		
50	8,28		
Promedio	8,37		

ANEXO G: Grafica 1 de dispersión y errores en las mediciones



e=	0,1956
E=	0,0747
E%=	7%

ANEXO H: Grafica 2 de dispersión y errores en las mediciones



e=	0,3209
E=	0,0384
E%=	4%



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS PARA EL
APRENDIZAJE Y LA INVESTIGACIÓN**



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 06/04/2021

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Jhon William Mera Reascos Cristian Eliasar Quicaliquin Jogacho
INFORMACIÓN INSTITUCIONAL
Facultad: Mecánica
Carrera: Ingeniería Industrial
Título a optar: Ingeniero Industrial
f. Analista de Biblioteca responsable: Lic. Luis Caminos Vargas Mgs.

**LUIS ALBERTO
CAMINOS
VARGAS**

Firmado digitalmente por LUIS
ALBERTO CAMINOS VARGAS
Nombre de reconocimiento
(DN): c=EC, l=BIOSAMBA,
serialNumber=0602766974,
cn=LUIS ALBERTO CAMINOS
VARGAS
Fecha: 2021.04.06 11:32:57
-05'00'



0899-DBRAI-UTP-2021