



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

**“GENERACIÓN DE UN SISTEMA DE CONTROL Y MONITOREO  
PARA UN BRAZO ROBÓTICO PROTOTIPO BASADO EN LA  
INTERNET DE LAS COSAS Y LA INTEGRACIÓN DE LABVIEW  
Y SOLIDWORKS”**

**Trabajo de titulación**

**Tipo:** Propuesta Tecnológica

Presentado para optar al grado académico de:

**INGENIERO INDUSTRIAL**

**AUTOR:** BYRON IVAN ASQUI RAMIREZ

**DIRECTOR:** Ing. Eduardo Francisco García Cabezas

Riobamba - Ecuador

2020

**©2020, Byron Ivan Asqui Ramirez**

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.

Yo, Byron Ivan Asqui Ramirez declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor (a) asumo 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, 05 de agosto de 2020

A handwritten signature in blue ink, appearing to read 'Byron Ivan Asqui Ramirez', enclosed within a large, loopy circular flourish.

---

**Byron Ivan Asqui Ramirez**  
**C.I: 0603450917**

# ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

## FACULTAD DE MECÁNICA

### CARRERA DE INGENIERÍA INDUSTRIAL

El Tribunal de Trabajo de Titulación certifica que: Propuesta Tecnológica: “**GENERACIÓN DE UN SISTEMA DE CONTROL Y MONITOREO PARA UN BRAZO ROBÓTICO PROTOTIPO BASADO EN LA INTERNET DE LAS COSAS Y LA INTEGRACIÓN DE LABVIEW Y SOLIDWORKS**” de responsabilidad del señor: Byron Ivan Asqui Ramirez, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
Ing. Carlos José Santillán Mariño <b>PRESIDENTE DEL TRIBUNAL</b>	 Firmado electrónicamente por: <b>CARLOS JOSE SANTILLAN MARINO</b>	2020-08-05
Ing. Eduardo Francisco García Cabezas <b>DIRECTOR DEL TRABAJO DE TITULACIÓN</b>	 Firmado electrónicamente por: <b>EDUARDO FRANCISCO GARCIA CABEZAS</b>	2020-08-05
Ing. Jhonny Marcelo Orozco Ramos <b>MIEMBRO DEL TRIBUNAL</b>	 Firmado electrónicamente por: <b>JHONNY MARCELO OROZCO RAMOS</b>	2020-08-05

## **DEDICATORIA**

De manera especial al forjador de mi camino, Dios, quien además es la fuerza renovadora en aquellos momentos difíciles que se presentan en la vida, pero sobre todo porque ha colocado personas extraordinarias en el transcurso de cumplir esta anhelada meta.

A mis padres que además de ser un pilar fundamental en mi vida son el mejor ejemplo de fortaleza y perseverancia, brindándome con gran entusiasmo su cariño, apoyo incondicional e indispensable, de esta manera me permiten cada día seguir con los objetivos planteados. No obstante, también a mi hermano quien con sus ocurrencias y sonrisa me motiva a continuar en cada decisión que tome.

Byron

## **AGRADECIMIENTO**

Mi agradecimiento radica nuevamente en el ser sublime, Dios, padre celestial que me permitió con sabiduría, paciencia y fortaleza llegar a culminar esta etapa importante de mi vida.

A mis padres por la confianza depositada al brindarme su amor, perseverancia, pero sobre todo apoyo moral y económico, esperando con ímpetu el momento oportuno para ver a su hijo subir un peldaño en busca de su profesión.

A mis familiares y amigos que estuvieron acompañándome a lo largo de mis estudios.

Por último, pero no menos importante expreso mi eterna gratitud y agradecimiento a la Escuela Superior Politécnica de Chimborazo por abrirme las puertas de su seno científico para formarme profesionalmente.

Byron

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE ANEXOS .....	xv
RESUMEN.....	xvi
ABSTRACT .....	xvii
<b>1. INTRODUCCIÓN</b>	
1.1. Antecedentes .....	2
1.2. Problematización.....	3
1.3. Justificación .....	3
1.4. Alcance .....	4
1.5. Objetivos .....	5
1.5.1. <i>Objetivo General</i> .....	5
1.5.2. <i>Objetivos Específicos</i> .....	5
<b>2. MARCO TEÓRICO</b>	
2.1. Internet de las cosas (IOT) .....	6
2.1.1. <i>Definición:</i> .....	6
2.1.2. <i>Aplicaciones:</i> .....	6
2.1.3. <i>Arquitectura de capas del IOT</i> .....	7
2.1.4. <i>Niveles de la Internet de las cosas</i> .....	8
2.1.5. <i>Mecanismos de Gestión de la IOT</i> .....	8
2.1.6. <i>Modelo de comunicación de la Internet de las cosas</i> .....	9
2.1.7. <i>Beneficios de la IoT</i> .....	10
2.1.8. <i>Desafíos de la IoT</i> .....	10
2.2. Software Embebido.....	10
2.3. Raspberry Pi.....	11
2.4. Python .....	11
2.5. Programa LabVIEW .....	12
2.5.1. <i>Funcionamiento:</i> .....	12
2.6. Módulo PCA 9685 .....	13
2.7. SOLIDWORKS.....	13

2.8.	<b>Impresión 3D:</b> .....	13
2.9.	<b>Métodos para impresión 3D:</b> .....	14
2.10.	<b>Ácido Poliláctico (PLA):</b> .....	14
2.11.	<b>Modelado Solido 3D y prototipado rápido:</b> .....	15
3.	<b>METODOLOGÍA</b>	
3.1.	<b>Diseño CAD</b> .....	16
3.2.	<b>Análisis Estructural</b> .....	17
3.3.	<b>Análisis Cinemático D-H</b> .....	18
3.4.	<b>Mecanizado</b> .....	18
3.5.	<b>Selección de Hardware</b> .....	19
3.5.1.	<i>Selección de actuadores</i> .....	19
3.5.2.	<i>Selección de controlador</i> .....	22
3.5.3.	<i>Interfaz de potencia</i> .....	23
3.6.	<b>Selección de Software</b> .....	23
3.6.1.	<i>Plataforma de soporte IOT</i> .....	23
3.6.2.	<i>LabVIEW</i> .....	24
3.6.3.	<i>Python</i> .....	24
3.7.	<b>Configuración de Hardware/Software</b> .....	24
3.7.1.	<i>Instalación del sistema operativo en la Raspberry pi</i> .....	24
3.7.2.	<i>Configuración inicial para manipular el módulo PCA</i> .....	25
3.7.3.	<i>Enlace Raspberry – Remote it</i> .....	27
3.7.4.	<i>Enlace Raspberry – Uidots</i> .....	30
3.7.5.	<i>Diseño HMI Python</i> .....	31
3.7.6.	<i>Algoritmo de control Python</i> .....	35
3.7.7.	<i>Enlace LabVIEW – SolidWorks</i> .....	39
3.7.8.	<i>Prueba de conexión LabVIEW – SolidWorks</i> .....	44
3.7.9.	<i>Prueba de conexión LabVIEW – Remote</i> .....	50
3.7.10.	<i>Creación de tableros y eventos en Ubidots</i> .....	53
3.8.	<b>Resultados</b> .....	56
3.8.1.	<i>Diseño CAD</i> .....	56
3.8.2.	<i>Análisis Estructural</i> .....	56
3.8.2.1.	<i>Análisis estático – Base</i> .....	57
3.8.2.2.	<i>Análisis estático – Brazo</i> .....	60
3.8.2.3.	<i>Análisis estático – Antebrazo</i> .....	63
3.8.2.4.	<i>Análisis estático – Muñeca</i> .....	66

3.8.2.5.	<i>Análisis estático – Pinza</i> .....	70
3.8.3.	<i>Análisis Cinemático</i> .....	73
3.8.4.	<i>Implementación del Brazo</i> .....	75
3.8.5.	<i>Programación Python</i> .....	77
3.8.6.	<i>Programación LabVIEW</i> .....	77
3.8.7.	<i>Prueba de Funcionalidad</i> .....	84
3.8.7.1.	<i>Ejecución del sistema</i> .....	84
3.8.7.2.	<i>Funcionamiento del Prototipo</i> .....	88
3.8.7.3.	<i>Consumo de corriente</i> .....	89
4.	<b>GESTIÓN DEL PROYECTO</b>	
4.1.	<b>Costos Directos</b> .....	90
4.2.	<b>Costos Indirectos</b> .....	90
4.3.	<b>Costos Totales:</b> .....	91
5.	<b>CONCLUSIONES</b> .....	92
6.	<b>RECOMENDACIONES</b> .....	94
	<b>GLOSARIO</b>	
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-2:</b> Características, Detalles técnicos y Ventajas de Python .....	12
<b>Tabla 1-3:</b> Parámetros técnicos para la impresión .....	19
<b>Tabla 2-3:</b> Peso en el brazo .....	20
<b>Tabla 3-3:</b> Distancias en el brazo .....	20
<b>Tabla 4-3:</b> Pesos en el antebrazo .....	21
<b>Tabla 5-3:</b> Distancias en el antebrazo .....	22
<b>Tabla 6-3:</b> Especificaciones técnicas Raspberry Pi 3V .....	22
<b>Tabla 7-3:</b> Especificaciones técnicas del Módulo PCA9685 .....	23
<b>Tabla 8-3:</b> Ensamblajes .....	56
<b>Tabla 9-3:</b> Propiedades del Modelo - Base .....	57
<b>Tabla 10-3:</b> Propiedades del Material - Base .....	57
<b>Tabla 11-3:</b> Sujeciones - Base .....	58
<b>Tabla 12-3:</b> Cargas - Base .....	58
<b>Tabla 13-3:</b> Análisis estático tensión nodal - Base .....	58
<b>Tabla 14-3:</b> Desplazamiento estático - Base .....	59
<b>Tabla 15-3:</b> Deformaciones unitarias estática - Base .....	59
<b>Tabla 16-3:</b> Propiedades del Modelo - Brazo .....	60
<b>Tabla 17-3:</b> Propiedades del Material - Brazo .....	60
<b>Tabla 18-3:</b> Sujeciones - Brazo .....	61
<b>Tabla 19-3:</b> Cargas - Brazo .....	61
<b>Tabla 20-3:</b> Análisis estático tensión nodal - Brazo .....	61
<b>Tabla 21-3:</b> Desplazamiento estático - Brazo .....	62
<b>Tabla 22-3:</b> Deformaciones unitarias estática - Brazo .....	62
<b>Tabla 23-3:</b> Propiedades del Modelo - Antebrazo .....	63
<b>Tabla 24-3:</b> Propiedades del Material - Antebrazo .....	64
<b>Tabla 25-3:</b> Sujeciones - Antebrazo .....	64
<b>Tabla 26-3:</b> Cargas - Antebrazo .....	64
<b>Tabla 27-3:</b> Análisis estático tensión nodal - Antebrazo .....	65
<b>Tabla 28-3:</b> Desplazamiento estático - Antebrazo .....	65
<b>Tabla 29-3:</b> Deformaciones unitarias estática - Antebrazo .....	66
<b>Tabla 30-3:</b> Propiedades del Modelo - Muñeca .....	67
<b>Tabla 31-3:</b> Propiedades del Material - Muñeca .....	67
<b>Tabla 32-3:</b> Sujeciones – Muñeca .....	68
<b>Tabla 33-3:</b> Cargas – Muñeca .....	68

<b>Tabla 34-3:</b> Análisis estático tensión nodal - Muñeca .....	68
<b>Tabla 35-3:</b> Desplazamiento estático - Muñeca .....	69
<b>Tabla 36-3:</b> Deformaciones unitarias estática - Muñeca.....	69
<b>Tabla 37-3:</b> Propiedades del Modelo - Pinza .....	70
<b>Tabla 38-3:</b> Propiedades del Material - Pinza .....	71
<b>Tabla 39-3:</b> Sujeciones - Pinza.....	71
<b>Tabla 40-3:</b> Cargas - Pinza.....	71
<b>Tabla 41-3:</b> Análisis estático tensión nodal - Pinza .....	72
<b>Tabla 42-3:</b> Desplazamiento estático - Pinza.....	72
<b>Tabla 43-3:</b> Deformaciones unitarias estática - Pinza.....	73
<b>Tabla 44-3:</b> Parámetros D-H.....	74
<b>Tabla 45-3:</b> Especificaciones técnicas del servo MG996R.....	75
<b>Tabla 46-3:</b> Especificaciones técnicas del Microservo MG90S.....	76
<b>Tabla 47-3:</b> Interfaz vs Prototipo .....	88
<b>Tabla 48-3:</b> Consumo de Corriente.....	89
<b>Tabla 1-4:</b> Cálculo de costos directos .....	90
<b>Tabla 2-4:</b> Cálculo de costos indirectos .....	90
<b>Tabla 3-4:</b> Cálculo de costos Totales .....	91

## ÍNDICE DE FIGURAS

<b>Figura 1-2.</b> Dispositivos conectados a la IOT .....	6
<b>Figura 2-2.</b> Estructura del IOT .....	7
<b>Figura 3-2.</b> Software embebido.....	11
<b>Figura 4-2.</b> Raspberry Pi .....	11
<b>Figura 5-2.</b> Interfaz del programa LabVIEW.....	12
<b>Figura 6-2.</b> Módulo PCA 9685 .....	13
<b>Figura 7-2.</b> SolidWorks.....	13
<b>Figura 1-3.</b> Brazo Robótico.....	17
<b>Figura 2-3.</b> Mecanizado de las piezas. ....	19
<b>Figura 3-3.</b> Diagrama de fuerzas en el brazo .....	20
<b>Figura 4-3.</b> Diagrama de fuerzas en el antebrazo.....	21
<b>Figura 5-3.</b> Configuración de Raspberry Pi .....	26
<b>Figura 6-3.</b> Instalación de paquete “SMBUS” .....	26
<b>Figura 7-3.</b> Verificación del Módulo PCA.....	27
<b>Figura 8-3.</b> Clonación de la biblioteca “Adafruit” .....	27
<b>Figura 9-3.</b> Instalación de la biblioteca “Adafruit” .....	27
<b>Figura 10-3.</b> Plataforma Remot3 it. ....	28
<b>Figura 11-3.</b> Adición del dispositivo.....	28
<b>Figura 12-3.</b> Instrucciones en Remote.....	28
<b>Figura 13-3.</b> Ejecucion de Instrucciones.....	29
<b>Figura 14-3.</b> Registro en Remote. ....	29
<b>Figura 15-3.</b> Verificación del registro.....	29
<b>Figura 16-3.</b> Plataforma Ubidots.....	30
<b>Figura 17-3.</b> Dispositivos en Ubidots . ....	30
<b>Figura 18-3.</b> Adquisición de instrucciones. ....	30
<b>Figura 19-3.</b> Default token .....	31
<b>Figura 20-3.</b> Dispositivos conectados. ....	31
<b>Figura 21-3.</b> Librería Tkinter. ....	32
<b>Figura 22-3.</b> Código ventana.....	32
<b>Figura 23-3.</b> Metodos de Tkinter. ....	32
<b>Figura 24-3.</b> Código etiqueta de visualización.....	33
<b>Figura 25-3.</b> Código entrada de información. ....	33
<b>Figura 26-3.</b> Código Imágenes.....	34
<b>Figura 27-3.</b> Código Botones.....	34

<b>Figura 28-3.</b> Código Scale. ....	35
<b>Figura 29-3.</b> Librerías. ....	36
<b>Figura 30-3.</b> Variables. ....	36
<b>Figura 31-3.</b> Funciones. ....	37
<b>Figura 32-3.</b> Conexión Cliente/Servidor. ....	37
<b>Figura 33-3.</b> Metodos de Socket. ....	38
<b>Figura 34-3.</b> Control de PCA. ....	38
<b>Figura 35-3.</b> Reemplazo de Códigos. ....	39
<b>Figura 36-3.</b> Llamado al hilo “ubidots_web” ....	39
<b>Figura 37-3.</b> Complementos de Solidworks ....	40
<b>Figura 38-3.</b> Ensamblaje del Brazo. ....	40
<b>Figura 39-3.</b> Inserción de motores. ....	40
<b>Figura 40-3.</b> Simulación. ....	41
<b>Figura 41-3.</b> Adición del ensamblaje ....	41
<b>Figura 42-3.</b> Motores rotatorios en LabVIEW ....	42
<b>Figura 43-3.</b> Colocación de axis ....	42
<b>Figura 44-3.</b> Axis en LabVIEW ....	42
<b>Figura 45-3.</b> Propiedades en My Computer ....	43
<b>Figura 46-3.</b> Configuración de axis. ....	43
<b>Figura 47-3.</b> Propiedades de Ensamblaje ....	44
<b>Figura 48-3.</b> Creación de VI en LabVIEW ....	45
<b>Figura 49-3.</b> Interfaz de mando en LabVIEW ....	45
<b>Figura 50-3.</b> Propiedades del mando numérico. ....	45
<b>Figura 51-3.</b> Bucle while loop. ....	46
<b>Figura 52-3.</b> Funciones de Express en LabVIEW ....	46
<b>Figura 53-3.</b> Conexiones con la función Line. ....	47
<b>Figura 54-3.</b> Otras conexiones. ....	47
<b>Figura 55-3.</b> Verificación de conflictos ....	48
<b>Figura 56-3.</b> Simulación en marcha. ....	48
<b>Figura 57-3.</b> Verificación de movimientos. ....	49
<b>Figura 58-3.</b> Parada de la Simulación. ....	49
<b>Figura 59-3.</b> Crear un VI. ....	50
<b>Figura 60-3.</b> Creación de la interfaz. ....	50
<b>Figura 61-3.</b> Bucle while loop. ....	51
<b>Figura 62-3.</b> Conexiones en el diagrama de bloques. ....	51
<b>Figura 63-3.</b> Interfaz gráfica de prototipo. ....	52
<b>Figura 64-3.</b> Dirección pública en Remote ....	52

<b>Figura 65-3.</b> Verificacion.....	53
<b>Figura 66-3.</b> Tablero en Ubidots.....	53
<b>Figura 67-3.</b> Widget en Ubidots.....	54
<b>Figura 68-3.</b> Verificación de Widget .....	54
<b>Figura 69-3.</b> Eventos en Ubidots.....	54
<b>Figura 70-3.</b> Configuración de Eventos .....	55
<b>Figura 71-3.</b> Reporte del Evento .....	55
<b>Figura 72-3.</b> Diagrama sistemas de referencia del brazo .....	73
<b>Figura 73-3.</b> Brazo Robótico con 6 grados de libertad .....	76
<b>Figura 74-3.</b> Programación Python. ....	77
<b>Figura 75-3.</b> Primera sección Flat Sequence Structure. ....	77
<b>Figura 76-3.</b> Diseño de la Interfaz .....	78
<b>Figura 77-3.</b> Bucle repetitivo .....	78
<b>Figura 78-3.</b> Primera sección Flat Sequence Structure, parte 1. ....	79
<b>Figura 79-3.</b> Primera sección Flat Sequence Structure, parte 2. ....	79
<b>Figura 80-3.</b> Segunda sección Flat Sequence Structure, parte 1. ....	80
<b>Figura 81-3.</b> Segunda sección Flat Sequence Structure, parte 2. ....	80
<b>Figura 82-3.</b> Segunda sección Flat Sequence Structure, parte 3. ....	81
<b>Figura 83-3.</b> Tercera sección Flat Sequence Structure, parte 1.....	81
<b>Figura 84-3.</b> Tercera sección Flat Sequence Structure, parte 2.....	82
<b>Figura 85-3.</b> Tercera sección Flat Sequence Structure, parte 3.....	82
<b>Figura 86-3.</b> Cuarta sección Flat Sequence Structure, parte 1. ....	83
<b>Figura 87-3.</b> Cuarta sección Flat Sequence Structure, parte 2. ....	83
<b>Figura 88-3.</b> Interfaz grafica del prototipo.....	84
<b>Figura 89-3.</b> Verificación de conflictos. ....	85
<b>Figura 90-3.</b> Ejecución de la simulación.....	85
<b>Figura 91-3.</b> Dirección publica en Remote It.....	85
<b>Figura 92-3.</b> Interfaz grafica en LabVIEW.....	86
<b>Figura 93-3.</b> Tableros en Ubidots .....	86
<b>Figura 94-3.</b> Interfaz grafica del prototipo.....	86
<b>Figura 95-3.</b> Brazo robótico prototipo. ....	87
<b>Figura 96--3.</b> Interfaz grafica en LabVIEW.....	87
<b>Figura 97-3.</b> Tableros en Ubidots. ....	87

## **ÍNDICE DE ANEXOS**

**ANEXO A: CÓDIGO PYTHON**

**ANEXO B: MANUAL DE USUARIO**

**ANEXO C: MOMENTO DE INERCIA**

## RESUMEN

La propuesta se orientó a la integración de la tecnología IOT con los Softwares: LabVIEW y SolidWorks con el fin de generar un sistema de control y monitoreo para un brazo robótico. Se tomó como referencia un modelo de brazo de 6 grados de libertad de la biblioteca de Thingiverse, al mismo que se aplicó modificaciones y se las validó mediante un análisis estático con el software SolidWorks previo a su mecanizado con tecnología de impresión 3D. Como elemento hardware principal, se utilizó una Raspberry Pi3 empleada como punto de conexión con la plataforma IOT Remote para generar el acceso remoto al equipo, como soporte local de Python en donde se desarrollaron algoritmos para gestionar las señales de control hacia los actuadores del prototipo y la generación de una interfaz gráfica de alto nivel para interacción con el usuario. Para la inducción de movimiento al dispositivo robótico se emplearon servomotores MG996R de 11 kgfxcn y un microservo MG90S 2,2 kgfxcn en torque, en conjunto con la interfaz de potencia PCA9685. Se logró por medio de la habilitación del módulo SOFT-MOTION enlazar LabVIEW y SolidWorks, plataformas empleadas para el monitoreo virtual del brazo robótico. Además, se empleó LabVIEW como receptor y emisor de información de la plataforma IOT y como fuente de información para el entorno de simulación en SolidWorks. Mediante pruebas de funcionalidad se estableció que los ángulos emitidos por la interfaz gráfica poseen un margen de error de 1 a 2 grados, el consumo de corriente del sistema fue de aproximadamente de un amperio, en el análisis estático se determinó que el material empleado en la construcción del brazo una deformación de 0,00013 - 0,020 imperceptible a una fuerza de 100N. Se recomienda no efectuar movimientos bruscos ni sobrecargar el brazo podrían generarse averías o bloqueo en los actuadores.

PALABRAS CLAVES: <INTERNET DE LAS COSAS>, <LABVIEW (SOFTWARE)>, <SOLIDWORKS (SOFTWARE)>, <RASPBERRY PI (HARDWARE – SOFTWARE)>, <PYTHON (SOFTWARE)>.



25-08-2020

0257-DBRAI-UPT-2020

## **ABSTRACT**

The proposal was oriented to the integration of the IOT technology with the Softwares: LabVIEW and SolidWorks to generate a control and monitoring system for a robotic arm. A 6-degree-of-freedom arm model was taken as a reference from the Thingiverse library, to which modifications were applied and validated through static analysis with SolidWorks software before machining with 3D printing technology. As the main hardware element, a Raspberry Pi3 was used as a connection point with the IOT Remote platform to generate remote access to the equipment, as local Python support where algorithms were developed to manage the control signals to the prototype actuators and the generation of a high-level graphical interface for interaction with the user. For the induction of movement to the robotic device, MG996R servo motors of 11 kgfxcn and a MG90S microservo 2.2 kgfxcn in torque were used, in conjunction with the PCA9685 power interface. Through the enabling of the SOFT-MOTION module, it was possible to link LabVIEW and SolidWorks, platforms used for the virtual monitoring of the robotic arm. Besides, LabVIEW was used as a receiver and transmitter of information from the IOT platform and as a source of information for the SolidWorks simulation environment. Through functionality tests, it was established that the angles emitted by the graphic interface have a margin of error of 1 to 2 degrees, the current consumption of the system was approximately one ampere, in the static analysis it was determined that the material used in the arm construction a deformation of 0.00013 - 0.020 imperceptible at a force of 100N. It is recommended not to make sudden movements or overload the arm, which could cause failure or blockage of the actuators.

**Keywords:** <INTERNET OF THINGS>, <LABVIEW (SOFTWARE)>, <SOLIDWORKS (SOFTWARE)>, <RASPBERRY PI (HARDWARE - SOFTWARE)>, <PYTHON (SOFTWARE)>.

## 1. INTRODUCCIÓN

El perfeccionamiento tecnológico de los últimos años ha influenciado en la industria principalmente por el manejo de sistemas automatizados que facilitan tareas garantizando seguridad y calidad en la producción de manera confiable.

Los sistemas automatizados basan su temática en monitoreo y control, principalmente buscando evaluar el correcto funcionamiento de las actividades establecidas en el diseño además de dar seguimiento al proceso de acuerdo a resultados reales que permitan encontrar soluciones en problemas de forma inmediata.

Las Tecnologías de Información y Comunicación (TIC) es una integración de herramientas que se emplean para diversas acciones como: adquirir, procesar, administrar, compartir, almacenar y registrar información haciendo uso de soportes tecnológicos.

La variabilidad de tecnologías ha permitido integrar una serie de programaciones que conlleva a satisfacer un fin determinado, siendo el caso para “El Internet de las cosas” (IoT), algunas veces denominado "Internet de los objetos", mismo que hace referencia a la conexión a través de plataformas para diferentes dispositivos intercambiando datos.

El IoT a simple vista parece una declaración arriesgada, sin embargo, hay que tener en cuenta el impacto que la Internet ha tenido sobre diversos aspectos como: educación, comunicación, empresas, ciencia, gobierno y la humanidad en general, por ende, representa la próxima evolución de Internet, lo que significa un enorme salto en la capacidad de: reunir, analizar y distribuir datos para convertirlos en información, conocimiento y en última instancia, sabiduría. (Evans, 2011, p. 2)

El Internet de las cosas nace para facilitar información en la cadena de suministro, extendiéndose a todo tipo de objetos –físicos y digitales–, a los animales, a las personas y a los entornos o ambientes. La IoT se define como un término popular para describir escenarios en los que la Internet, la conectividad y la capacidad de procesamiento de información se extienden a una variedad de objetos, dispositivos, sensores y elementos cotidianos. De este modo, se incorpora a la dimensión de las TIC, ofreciendo capacidades de operación en todo momento y en cualquier lugar, transformando objetos tradicionales (pasivos) en inteligentes (activos). (Barrio, 2018, pp. 20-21)

## **1.1. Antecedentes**

La Industria 4.0 mejor conocida como la Cuarta Revolución Industrial, destina su origen en tres etapas que van desde la década de los 80 con un sistema bajo control CAD-CAM (Diseño Asistido por Computadora- Fabricación Asistida por Computadora), posteriormente en la década de los 90 se presenta la segunda etapa de acuerdo al surgimiento del Internet conjuntamente con tecnologías asociadas. Finalmente, la tercera etapa se concluye en el nuevo milenio a través de conceptos basados en el IoT ligado a dispositivos móviles y su capacidad de conexión. (Interempresas,2019)

El término Internet of Things (IoT) fue instaurado por primera vez en el año 1999, por el MIT (Instituto de Tecnología de Massachusetts) y Kevin Ashton, para referirse a un mundo conectado por completo a internet, lo cual incluye cosas como: casas, autos, fábricas, automóviles, electrodomésticos, robots industriales, entre otros dispositivos.

En el año 2010 los robots generaron logros positivos debido a que se emplearon de diversas formas para labores de manejo industrial, montaje y manipulación, logro alcanzado a través de la digitalización de los procesos productivos en las fábricas.

Los sensores y sistemas de información trabajaron para transformar los procesos con eficiencia, permitiendo controlar a distancia las tareas con una interconexión digital de objetos cotidianos con internet, es decir la capacidad de conectar dispositivos embebidos con capacidades limitadas de CPU, memoria y energía.

Los robots desarrollados para realizar labores de manejo, montaje y manipulación industrial, desde el punto de vista de la clausura resultan de mayor interés aquellos que son: manipuladores controlados a distancia, robots (estacionarios y móviles), tecnología visual, de sensores, de soporte físico y lógico relacionado con el equipo. (Da Costa, 2015)

El internet de las cosas en la actualidad tiene aplicaciones en casi cualquier área, por ejemplo, en la industria nuclear se ha creado, y se sigue creando, una amplia gama de manipuladores y equipos especializados que realizan por control a distancia tareas como: inspección, mantenimiento, reparación y renovación.

Las ventajas descubiertas a lo largo del surgimiento de la IoT son varias, sin embargo, se mencionan las más importantes como: transformación de los entornos de trabajo, forma de relacionarse en las fábricas, mejora continua de los procesos productivos, conexión y enlace remoto, monitoreo del proceso independiente del lugar en que se encuentre y sobre todo reducción de errores humanos.

## **1.2. Problematización**

Las diferentes técnicas de control y monitoreo de procesos continuamente evalúan el desempeño de las tareas; sin embargo, su implementación en una industria se hace difícil de adquirir debido a los costos de inversión que representa, por ende, se considera el volumen de producción, vida útil del producto, las ventas a cubrir, entre otros parámetros. Cabe señalar que los costos se incrementan a nivel que se utiliza licencias de softwares.

La ejecución manual de distintas tareas en los procesos productivos se ha visto afectada generalmente en fallos como: impericia humana, calibración de equipos, precisión de tareas, datos erróneos o poco confiables, bajo rendimiento relacionado a productividad y ciclos de trabajo lentos, entre otros, los cuales provocan grandes pérdidas dentro de una empresa.

El internet de las cosas fundamenta su propósito tanto en una utilidad personal como social, permitiendo la conexión, supervisión y control remoto de cualquier sistema, sea este productivo o de distinta índole, gestionando eficientemente la comunicación, tanto entre máquinas como entre personas, de esta manera presenta plataformas gratuitas o pagadas a bajos costos, adaptables a programas licenciados como LABVIEW, causando una gran reducción de costes para su implementación. (Nadal, 2018)

El presente trabajo será elaborado acorde al almacenamiento, recuperación y visualización de datos en tiempo real, conjuntamente con un diseño de brazo robot ejecutado por análisis mecánico en función a posiciones y orientaciones en el espacio tridimensional, demostrando los beneficios, ventajas y alcances de emplear las plataformas digitales que se presentan en la IoT. (Vera, 2017)

La manipulación tele operada es necesario ponerla en marcha debido a su enfoque para salvaguardar la integridad de la persona como supervisor u operario del proceso. (Coque, 2013)

## **1.3. Justificación**

La investigación consistió en desarrollar un sistema de control y monitoreo a través del diseño de un prototipo de brazo robótico con 6 grados de libertad, basado en la tecnología internet de las cosas, cuya aplicabilidad es el control a distancia de un proceso industrial.

El sistema automatizado se basó en la utilización de la tecnología “open source”, cuyo propósito fue desarrollar un sistema económico, seguro y flexible, aportando en distintos campos como: investigación, innovación tecnológica, responsabilidad social, de forma significativa a la producción de las PYMES en el país y mejora de los procesos de manufactura. Además, la IOT

es un pilar fundamental para industria 4.0 la cual es utilizada para enlazar los procesos a través del internet con la programación de la interfaz visual y de control en el software LabVIEW.

La estructura y configuración de los sistemas embebidos Raspberry pi se manejaron con tecnología de tipo open source ofreciendo soporte además de fácil adaptabilidad a la tecnología del Internet de las cosas.

La proliferación del prototipo en su vinculación a un hardware y software estableció sistemas de control, monitoreo, registro y tratamiento de información remota con la habilidad de comunicar información a todos, permitiendo nuevas capacidades de procesamiento en datos, siendo un gran impacto social y económico.

#### **1.4. Alcance**

El proyecto está encaminado a desarrollar un sistema de control y monitoreo remoto mediante un prototipo de brazo robótico para un proceso industrial. El objetivo principal radica en innovar eficazmente los procesos productivos de acuerdo a las PYMES, enlazando los procesos a través del internet de las cosas y la programación con interfaz visual y de control en el software LabVIEW., para aportar en campos como: investigación, innovación tecnológica, mejorar la productividad de una empresa en relación a notificación de datos, rentabilidad, entre otros., se toma en cuenta también la implementación del prototipo de forma física.

## **1.5. Objetivos**

### ***1.5.1. Objetivo General***

Generar un sistema de control y monitoreo para un brazo robótico prototipo basado en el internet de las cosas y la integración de LabVIEW y SolidWorks.

### ***1.5.2. Objetivos Específicos***

- Implementar un sistema de supervisión y control local escalable basado en Python y Raspberry Pi para manipulación - configuración del brazo robótico prototipo.
- Realizar la simulación virtual remota del brazo robótico prototipo mediante el enlace entre el software de programación LabVIEW y el SolidWorks.
- Enlazar los sistemas local y remoto por medio del Internet de las Cosas para un control y monitoreo simultáneo.
- Evaluar el sistema generado mediante pruebas para valoración de tiempos de respuesta y precisión.

## 2. MARCO TEÓRICO

### 2.1. Internet de las cosas (IOT)

#### 2.1.1. Definición:

Internet de las cosas (IoT) es la red de objetos físicos que se conectan a Internet usando diversas tecnologías y que tienen capacidades de conexión e interacción con el entorno, capacidades que les permiten tomar decisiones y comunicarse con el mundo. (López, 2019, p. 21)



**Figura 1-2.** Dispositivos conectados a la IOT

Fuente: <https://bit.ly/2QSJdzB>

La Figura 1-2 muestra el alcance de la IoT en comunicación, transmisión de información, pero sobre todo conectividad entre diferentes objetos.

#### 2.1.2. Aplicaciones:

Según (Alvear et al., 2017, pp. 244-256), el campo de aplicaciones de IoT se ha extendido por diversas áreas de estudio, permitiendo usar esta tecnología en la mejora de recursos naturales, tecnológicos e incluso mejorar ciertas actividades de la vida cotidiana, por mencionar se tiene:

- *Conteo de personas en edificios inteligentes:* Nace de la necesidad de implementar un sistema de conteo de personas dentro de un edificio, para mejorar las condiciones medio ambientales y de seguridad dentro de lugares donde se tiene un alto número de ocupantes.
- *Parqueo inteligente y reducción de tráfico vehicular:* Uno de los pilares del concepto Smart City es el manejo inteligente del tráfico. La solución se centra en el uso de WSNs y una adaptación a esta tecnología empleando visión artificial que da como resultado una VSN (Visual Sensor Network). Un nodo VSN es aquel que está conectado a una cámara adquiriendo, procesando y capturando imágenes de un lugar específico transmitiendo a un nodo central.

- *Sistemas de vigilancia y control en hogares:* IoT requiere tan solo de una cámara y ciertos algoritmos para poder realizar el análisis, lo que se requiere específicamente es capturar imágenes.
- *Medicina:* En el campo de la medicina mediante la visión artificial se puede clasificar o visualizar una imagen en alta resolución para opera desde lugares remotos sin necesidad de que un doctor deba trasladarse al lugar de la cirugía.
- *Inspección de Calidad:* La inspección puede involucrar sectores de: verificación en dimensiones y fallas de un elemento, cámaras monitoreando la producción en serie de una fábrica, el control se lo realiza fuera o dentro de las instalaciones, este proceso puede ser revisado mediante cualquier dispositivo móvil con acceso a las direcciones públicas de la empresa.
- *IoT con Visión Artificial en la Educación:* Desde los niveles iniciales hasta las aulas universitarias involucrar a los estudiantes con las TICs. En el caso específico de la IoT se pueden desarrollar herramientas didácticas de aprendizaje que cuenten con una plataforma en internet con resultados reales del estudiante para ser visualizados.

### 2.1.3. Arquitectura de capas del IOT

La arquitectura del IOT para (Liñán et al., 2015, p. 3) se basa en:



**Figura 2-2.** Estructura del IOT

Fuente: <https://bit.ly/35kkPMm>

- Una capa del dispositivo: Contiene los dispositivos (sensores, actuadores, dispositivos RFID) y pasarelas (gateways) que se usan para recolectar las lecturas del sensor para su procesamiento posterior.

- Una capa de red: Proporciona el transporte necesario y las capacidades de red para enrutar los datos de IoT a los sitios de procesamiento.
- Una capa de soporte: Es una capa intermedia (middleware) que sirve para esconder la complejidad de las capas inferiores.
- Una de aplicación: Para dar servicios específicos y genéricos tales como: almacenamiento bajo formas diferentes (sistemas de manejo de base de datos y/o sistemas de computación en la nube), así como otros servicios, por ejemplo, traducción.

#### **2.1.4. Niveles de la Internet de las cosas**

La IoT posee una arquitectura macro de tres niveles o capas. (Parra et al., 2017, p. 11)

- El nivel básico: Despliega diferentes sensores, cada uno de ellos es una fuente de información, que capturan diversos contenidos y formatos de información.
- El nivel intermedio: Integra varias redes cableadas e inalámbricas para transferir la información de las cosas con precisión.
- El nivel superior: Es la aplicación que consta de tres capas (capa de tecnología, capa de middleware, capa de aplicación) y se compone de aplicaciones que exportan toda la funcionalidad de los sistemas para el usuario final.

#### **2.1.5. Mecanismos de Gestión de la IOT**

Según (Rodríguez, 2013, pp. 53-55) la literatura existente relativa a las formas de implementación de la gestión de la IoT se puede dividir en cuatro grupos de acuerdo al mecanismo o estándar de gestión empleado:

- *SNMP (Simple Network Management Protocol)*: El empleo de SNMP en los elementos de la IoT es una variante atrayente debido al planteamiento óptimo de gestionar sus elementos, instalando un agente por cada dispositivo, un gestor local en cada Gateway y un gestor remoto en la infraestructura de red IP existente.
- *TMN (Telecommunication Network Management)*: La característica fundamental de este tipo de arquitectura radica en combinar la robustez de CMIP con la interoperabilidad de

CORBA. Los sistemas se destacan por aportar modularidad, abstracción, posibilidad de reutilización del software, nombrado de recursos y localización.

CMIP es el protocolo que define la información de gestión para la comunicación entre las aplicaciones de gestión y agentes del modelo de gestión OSI (Open System Interconnection).

CORBA es un framework que entra en el grupo de sistemas de gestión distribuida conocidos como DOT (Distributed Object Technology).

- *WBEM (Web Based Enterprise Management)*: La gestión Web provee un conjunto de estándares y tecnologías para la gestión de Internet desarrollados con el fin de unificar los sistemas de gestión de redes, usuarios al igual que aplicaciones existentes. Los componentes fundamentales que forman parte de la arquitectura son la aplicación de gestión (WBEM-Client) y el intermediario entre el gestor además del hardware del dispositivo.
  
- *Gestión de IoT haciendo uso de PBM (Policy Based Management) que corresponde a la gestión autónoma*: Este tipo de gestión se destaca por modificar el rol del operador, para controlar el sistema directamente, realizando funciones asociadas a la descripción de políticas. La gestión autónoma ha sido propuesta para solventar el problema de dispositivos complejos. Las cuatro funcionalidades prefijadas de la gestión autónoma son: autoconfiguración, auto reparación, auto optimización y autoprotección.

#### **2.1.6. Modelo de comunicación de la Internet de las cosas**

Desde el punto de vista operativo, es útil pensar en cómo se conectan y comunican los dispositivos de la IoT. En marzo de 2015, el Comité de Arquitectura de Internet (IAB) dió a conocer un documento para guiar la creación de redes en objetos inteligentes (RFC 7452),<sup>39</sup> que describe un marco de cuatro modelos de comunicación comunes como:

- Comunicaciones: Dispositivo- Dispositivo
- Comunicaciones: Dispositivo a la nube
- Modelo: Dispositivo a puerta de enlace
- Modelo: Intercambio de datos a través de back – end. (Rose et al., 2015, pp. 19-24)

### **2.1.7. Beneficios de la IoT**

- Al ser una tecnología en desarrollo su implementación permitirá la creación de nuevos empleos y negocios.
- Los clientes serán los más beneficiarios de acuerdo a la gama de ofertas y servicios en tiempo real que se muestran.
- Envío de datos en tiempo real de acuerdo a la producción o actividad que se requiere controlar.
- En la actualidad la mayoría de dispositivos integran servicio de conexión a internet, por tal motivo su comunicación entre sí, es más sencilla.
- La IoT se puede implementar en cualquier proceso industrial, de servicio o simplemente en objetos a controlar.

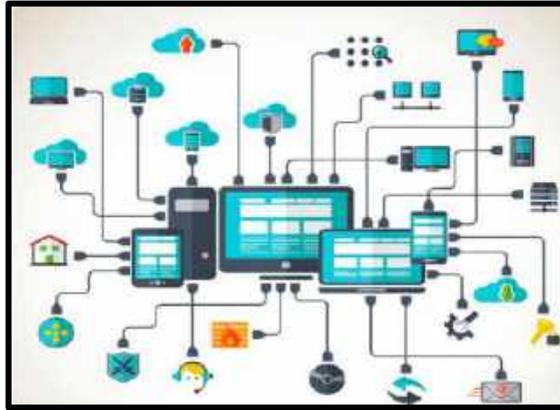
### **2.1.8. Desafíos de la IoT**

Los desafíos para IoT de acuerdo a (López y Cárdenas, 2019, p. 76) son:

- Banda infinita o mayor ancho de banda en las infraestructuras de red. En el momento en que tengamos más dispositivos, mayor será el tráfico que debemos manejar con la red actual, por ende, se requiere altas demandas de conectividad.
- La seguridad y la privacidad también son grandes inconvenientes. Los dispositivos relacionados con el IoT, tienen un alto grado de vulnerabilidad. Estos dispositivos recopilan información personal como el número de contacto, nombres, correos, dirección de residencia, número de cuentas bancarias, entre otros.
- Regulación debido a los vacíos legales que se tiene con ciertos aparatos acorde de su implementación en las tareas cotidianas de las personas.

## **2.2. Software Embebido**

El software embebido es un componente de un «producto completo». En el software embebido, el consumidor o usuario del dispositivo no interactúa directamente con el software y, en muchos casos, como en el caso de los frenos antibloqueo, es posible que no tenga ni idea de que el software existe. El software no embebido, por otro lado, se carga normalmente en una máquina por el usuario, pudiendo interactuar con él a través de una interfaz y siendo fácilmente configurable por el mismo. (Llaneza, 2018, p. 57)



**Figura 3-2.** Software embebido

Fuente: <https://bit.ly/2SKkmQV>

### 2.3. Raspberry Pi

Raspberry Pi es un ordenador de placa reducida o placa única (SBC – Single Board Computer) de bajo coste (poco más de 30 Euros según modelo) desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. (López, 2017, p. 63)



**Figura 4-2.** Raspberry Pi

Fuente: <https://cutt.ly/orwbJHT>

### 2.4. Python

Python es un lenguaje de programación moderno, de propósito general, orientado a objetos de alto nivel. (Johansson, 2016, p. 8)

**Tabla 1-2:** Características, Detalles técnicos y Ventajas de Python

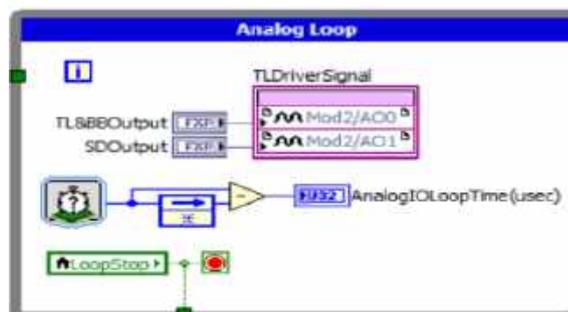
CARACTERÍSTICAS GENERALES DE PYTHON	DETALLES TÉCNICOS	VENTAJAS
Lenguaje limpio y simple. ➤ Código fácil de leer e intuitivo ➤ Sintaxis minimalista fácil de aprender ➤ Capacidad de mantenimiento, se adapta bien al tamaño de los proyectos.	Escrito dinámicamente: no es necesario definir el tipo de variables, argumentos de función o tipos de retorno.	Documentación estrechamente integrada con el código.
Lenguaje expresivo: ➤ Menos líneas de código ➤ Menos errores ➤ Más fácil de mantener.	Gestión automática de memoria: no es necesario asignar y desasignar explícitamente memoria para variables y matrices de datos. No hay errores de pérdida de memoria.	Una gran biblioteca estándar y una gran colección de paquetes complementarios.

Fuente: Johansson Robert, 2016.

## 2.5. Programa LabVIEW

### 2.5.1. Funcionamiento:

Un programa LabVIEW consiste en una serie de funciones unidas mediante cables. Los datos ‘circulan’ o ‘fluyen’ por los cables. Una función solo podrá ejecutarse cuando tenga disponibles todos los datos que le sirven como entradas. Esta forma de ejecutar un programa favorece el paralelismo para sistemas multiprocesador y multihilo, parecida al funcionamiento de los circuitos hardware e incluso al lenguaje VHDL. (Lajara y Pelegri, 2011, p. 25)

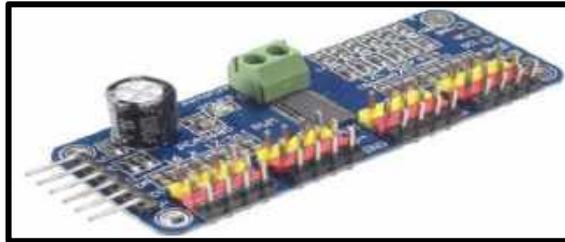


**Figura 5-2.** Interfaz del programa LabVIEW

Fuente: <https://bit.ly/35x2Cv4>

## 2.6. Módulo PCA 9685

El PCA9685 es un controlador LED de 16 canales, optimizado para aplicaciones de retroiluminación de color rojo / verde / azul / ámbar (RGBA). Cada salida tiene su Controlador PWM individual de frecuencia fija con resolución de 12 bits (4096 pasos) a una frecuencia programable de 24 a 1526 Hz, su ciclo de trabajo es ajustable de 0% a 100% para permitir que el LED se configure en un valor de brillo específico. (Semiconductors, 2015, p. 1)



**Figura 6-2.** Módulo PCA 9685

Fuente: <https://cutt.ly/BrwnSjq>

## 2.7. SOLIDWORKS

SolidWorks es un software CAD, se trata de un programa que permite realizar el proceso completo de diseño mecánico, desde la concepción de la idea por el diseñador hasta la realización de los planos técnicos necesarios para su fabricación. (Rodríguez, 2015, pp. 14-15)



**Figura 7-2.** SolidWorks

Fuente: <https://cutt.ly/TttXMLv>

## 2.8. Impresión 3D:

La impresión 3D funciona de modo completamente distinto: la pieza se crea en un solo paso, capa por capa, a un ritmo medio de unos dos centímetros de altura por hora. El objeto creado puede constar de mecanismos internos (como un rodamiento de bolas), formas tejidas, entrelazadas, o incluso de huecos y curvas.

Si bien existen numerosos procesos de impresión 3D, todos tienen un factor común en la producción de objetos a base de superposición por capas sucesivas. Esta técnica recibe el nombre

de fabricación "aditiva", pues se lleva a cabo mediante la adición de materia: el objeto cobra forma a medida que las capas se solidifican.

La impresión 3D se lleva a cabo en una impresora 3D. La máquina puede tener tamaños y aspectos variados con diversos programas informáticos, fundamentales para el proceso, pues permiten preparar el archivo 3D del objeto que se va a fabricar y controlar en la impresión. Una impresora 3D, por lo tanto, es una máquina capaz de fabricar un objeto físico a partir de un modelo 3D. (Berchon y Luyt, 2016, pp. 3-4)

## 2.9. Métodos para impresión 3D:

- **Por adición:** Inyecta polímeros que se agregan o suman por capas, método también denominado por polimerización.
- **Por sintonización láser:** Este método combina distintas capas de metal que son provistas por un suministrador que entrega el metal a un dispositivo láser que funde las capas y forma el objeto.
- **Por compactación:** Formando una masa de polvo que se compacta por estratos.
- **Por estilográfica:** Este método utiliza una masa de resina fotosensible que se solidifica al ser expuesta a la luz ultravioleta. (Mejía, 2016, p.30)

## 2.10. Ácido Poliláctico (PLA):

El ácido poliláctico (PLA), es un biopolímero termoplástico cuya molécula precursora es el ácido láctico. Debido a su biodegradabilidad, propiedades de barrera y biocompatibilidad, este biopolímero ha encontrado numerosas aplicaciones ya que presenta un amplio rango inusual de propiedades, desde el estado amorfo hasta el estado cristalino; propiedades que pueden lograrse manipulando las mezclas entre los isómeros D (-) y L (+), los pesos moleculares, y la copolimerización.

Las propiedades físicas y mecánicas que hacen interesante al ácido poliláctico son el mecanismo de biodegradabilidad, reabsorción y las aplicaciones en diferentes campos. El PLA puede ser tan duro como el acrílico o tan blando como el polietileno, rígido como el poliestireno o flexible como un elastómero.

Las resinas de PLA pueden ser sometidas a esterilización con rayos gamma y es estable cuando se expone a los rayos ultravioleta. Al PLA se le atribuyen también propiedades de interés como la suavidad, resistencia al rayado y al desgaste. (Serna et al., 2003, pp. 16-26)

### **2.11. Modelado Sólido 3D y prototipado rápido:**

El modelado sólido 3D facilita el diseño de formas complejas y su representación gráfica, el prototipado rápido permite la fácil materialización de una maqueta real o de un modelo funcional.

La incorporación de prototipado rápido, tecnología proveniente del sector industrial, en el proceso de enseñanza – aprendizaje, posibilita la transferencia de experiencias de la actividad industrial al aula, por lo que se constituye en una innovación educativa, que además está en línea con el concepto actual de formación basada en competencias. (Torres et al., 2012, pp. 25-32)

### 3. METODOLOGÍA

El desarrollo de la propuesta tecnológica ha sido ejecutado por la aplicación lógica y metódica de diversos fundamentos bibliográficos detallados en el marco teórico. La parte práctica estuvo respaldada por distintas técnicas de estudio como: Diseño CAD, Análisis Estructural, Análisis Cinemático D-H, Mecanizado, Selección de Hardware, Selección de Software, Configuración de Hardware/Software, entre otros.

Otros aportes evidenciados fueron:

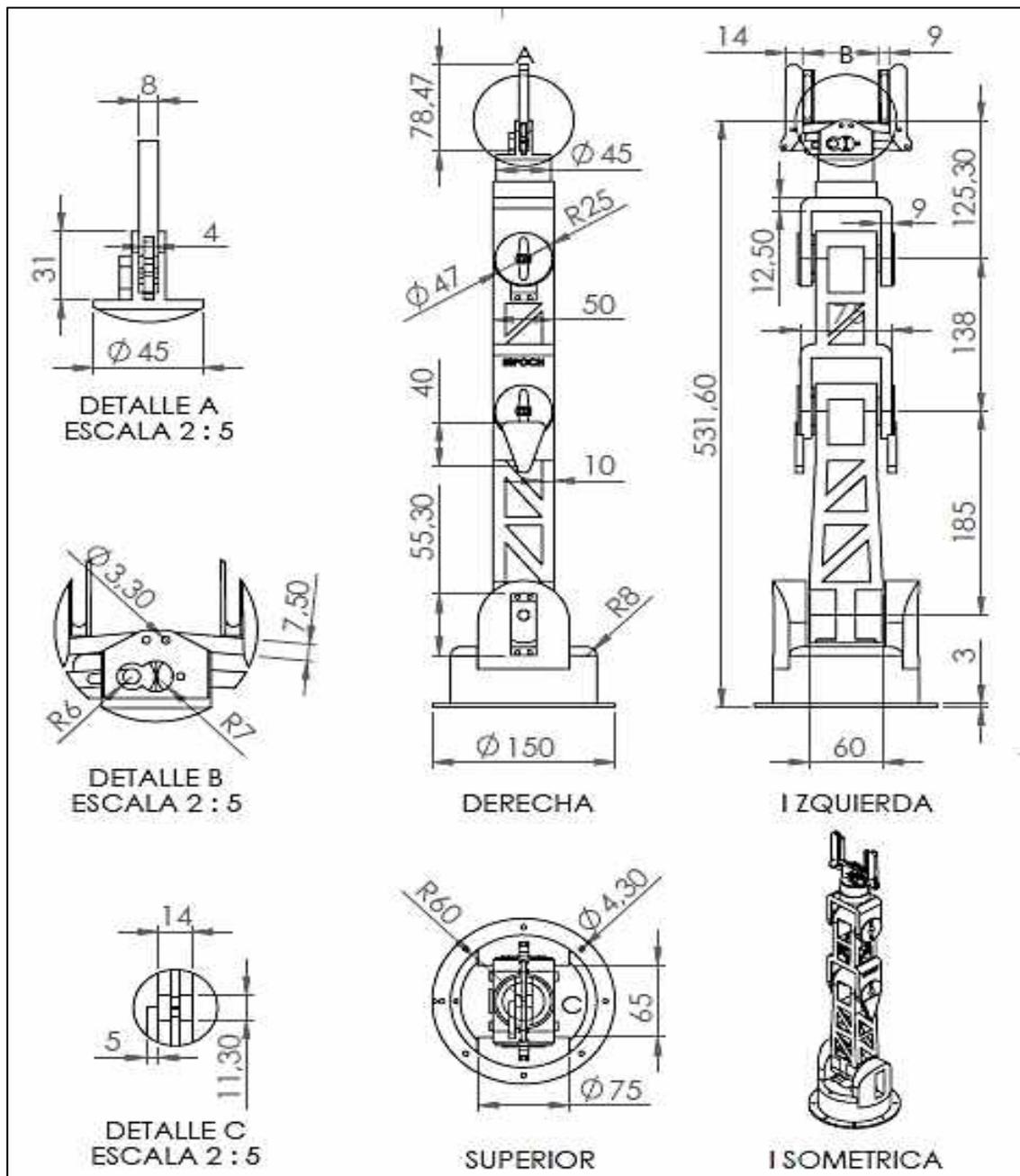
- La cinemática en su forma directa e inversa, aportó al estudio del movimiento del brazo sin considerar las fuerzas que lo provocan.
- El controlador del prototipo se decidió bajo el criterio de dos principales características la naturaleza de la planta y las condiciones operacionales que incluye: seguridad, costo, confiabilidad, precisión, peso y tamaño. (Coque, 2013)
- La simulación del brazo robot estuvo dada por el software SolidWorks controlada desde LabVIEW mostrando de manera virtual el panel interactivo para realizar pruebas preliminares sobre el elemento creado en SolidWorks. (Hernández, 2016)
- Los parámetros de comunicación de los dispositivos se ejecutaron al conectarse a una señal wifi mediante el ingreso a la aplicación creada desde la terminal, obteniendo las direcciones virtuales y concatenando con LabVIEW.

El sistema de control y monitoreo para un brazo robótico prototipo basado en la integración de la internet de las cosas fue desarrollado con éxito bajos los parámetros descritos anteriormente.

#### 3.1. Diseño CAD

El empleo del Diseño CAD en el proyecto tuvo modelos bidimensionales y tridimensionales de componentes físicos para la simulación de una red industrial, la búsqueda se concluyó con un brazo robótico de la biblioteca de figuras 3D imprimibles en la página [www.thingiverse.com](http://www.thingiverse.com), sistema enfocado en la utilización de las plataformas IOT mas no en el diseño.

El procedimiento del modelo inició con las medidas básicas que caracterizan para redibujarlo en un nuevo archivo utilizado en el monitoreo del sistema, debido a que el archivo descargado se encuentra en formato “STL” y posee una gran cantidad elementos que ralentizan el análisis de movimientos.



**Figura 1-3.** Brazo Robótico.

Fuente: Asqui Byron, 2020.

En la Figura 1-3 se observa las vistas del ensamblaje, identificando los componentes del modelo y señalando las medidas básicas del brazo robótico prototipo.

### 3.2. Análisis Estructural

El Análisis CAE (Ingeniería asistida por computadora) tuvo participación en la simulación de elementos finitos con la ayuda del software SolidWorks, con criterio de máxima tensión de Von Mises, haciendo alusión a la teoría de fallo para comprobar la ductilidad del material PLA, empleado en la impresión del brazo prototipo.

Se comprobó que el material está bajo tensión, es decir, es igual o inferior al rendimiento del mismo material, bajo una tensión simple este cederá y fallará, pero aplicando una carga de 100N con un factor de seguridad de 1,2 a uno de los eslabones del brazo no cedió, por tanto, es un material de alta ductilidad apto para el soporte de los servomotores añadidos.

### 3.3. Análisis Cinemático D-H

El análisis Cinemático D-H está basado en el modelo Denavit-Hartenberg de la cinemática directa, consiste en localizar el último grado de libertad con respecto a un sistema de coordenadas. Los sistemas de coordenadas en cada articulación posibilitan pasar de un eslabón a otro mediante cuatro transformaciones básicas que dependen de los parámetros referidos al tamaño y forma del eslabón y tomando en cuenta que sean relativos a un eslabón y a su antecesor. El resultado arrojó un patrón que permiten relacionar el sistema de referencia del eslabón  $i$  con respecto al sistema  $i-1$ :

- Rotación alrededor del eje  $Z_{i-1}$  un ángulo  $\theta_i$
- Traslación a lo largo de  $Z_{i-1}$  una distancia  $d_i$
- Traslación a lo largo de  $X_i$  una distancia  $a_i$
- Rotación alrededor del eje  $X_i$  un ángulo  $\alpha_i$

Para finalizar se creó una tabla con los parámetros de D-H, los cuales permitieron obtener las matrices homogéneas para realizar el cálculo correspondiente localizando el extremo del robot con su orientación y posición.

### 3.4. Mecanizado

El mecanizado del brazo robótico prototipo consistió en la impresión 3D de las piezas del modelo. El proceso se llevó a través del software laminador de modelos tridimensionales Ultimaker Cura, que se encuentra disponible en <https://ultimaker.com/es/software/ultimaker-cura>. Al instalarlo en el ordenador se realizó las configuraciones acordes a la impresora además de modificar los parámetros de impresión según la pieza.

A continuación, la Tabla 1-3 describe los principales parámetros técnicos de impresión utilizados para la fabricación del prototipo.

**Tabla 1-3:** Parámetros técnicos para la impresión

Características:	Descripción	
Material:	PLA	
Velocidad:	40-50 mm/s	
Relleno:	30-80%	
Altura de capa:	0,2 mm	
Patrón de relleno:	rejilla	
Soportes:	En todos	

**Realizado por:** Asqui Byron, 2020.

Una vez ingresadas todos los parámetros de impresión se importó y segmentó las piezas para generar un archivo de código G que fue interpretado por la impresora 3D para crear un objeto tridimensional mediante la superposición de capas sucesiva de material, repitiendo la operación con cada una de las piezas del brazo en mención.



**Figura 2-3.** Mecanizado de las piezas.

**Fuente:** Asqui Byron, 2020.

### 3.5. Selección de Hardware

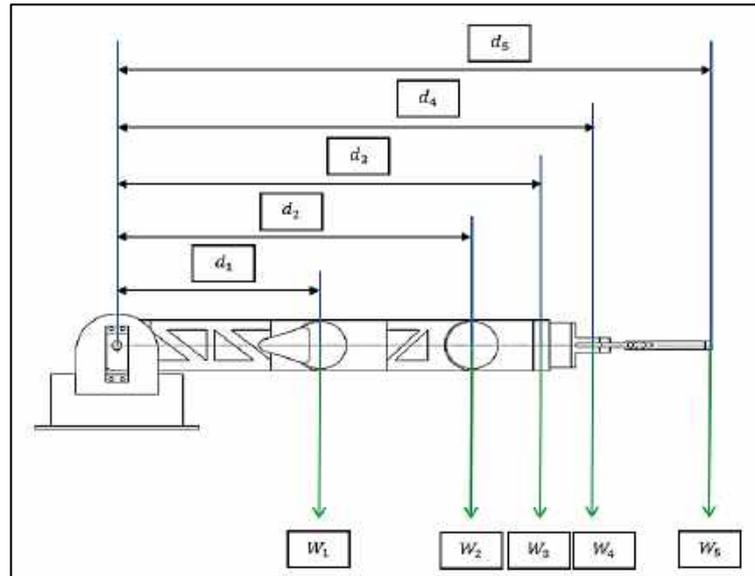
#### 3.5.1. Selección de actuadores

Los puntos críticos del brazo prototipo se detallaron en un Diagrama de Cuerpo Libre como se muestra en la Figura 3-3, acorde a las 6 articulaciones, lugares considerados zonas críticas para la manipulación de los eslabones que conforman el brazo. El análisis estático de fuerzas usó cálculo del torque producido por cada eslabón, considerando las siguientes variables:

- Peso del eslabón
- Distancia de cada eslabón

### Cálculo Torque Brazo

Identificación de pesos que actúan en la articulación.



**Figura 3-3.** Diagrama de fuerzas en el brazo

Fuente: Asqui Byron, 2020.

**Tabla 2-3:** Peso en el brazo

Descripción		Peso(g)
W1	Actuador	55
W2	Actuador	55
W3	Actuador	13,4
W4	Actuador	13,4
W5	Estructura	436

Realizado por: Asqui Byron, 2020.

**Tabla 3-3:** Distancias en el brazo

Descripción	Distancia (mm)
$d_1$	185
$d_2$	323
$d_3$	385
$d_4$	432
$d_5$	540

Realizado por: Asqui Byron, 2020.

El torque responde a la fórmula:

$$\tau = W * d$$

Donde:

$\tau = \text{Torque}$

$W = \text{Peso}$

$d = \text{Distancia}$

$$\tau_1 = W_1 * d_1 = 0,055kg * 18,5cm = 1,0175 Kg cm$$

$$\tau_2 = W_2 * d_2 = 0,055kg * 32,3cm = 1,7875 Kg cm$$

$$\tau_3 = W_3 * d_3 = 0,0134kg * 38,5cm = 0,5159 Kg cm$$

$$\tau_4 = W_4 * d_4 = 0,0134kg * 43,2cm = 0,57888 Kg cm$$

$$\tau_5 = W_5 * d_5 = 0,436kg * 54,0cm = 23,544 Kg cm$$

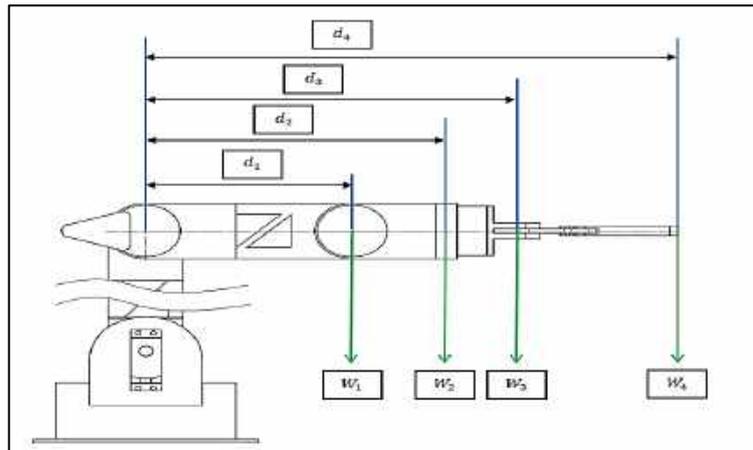
$$\tau_{Brazo} = \tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5 = 27,44 Kg cm$$

$$\tau_{Servo} = \frac{\tau_{Brazo}}{2} = 13,72 Kg cm$$

El análisis mostró los torques máximos generados por los actuadores en cada tramo cuando el prototipo se encuentra en su máxima extensión, dando como resultado que el servomotor soporta 13,72 kg cm.

#### Cálculo Torque Antebrazo

Elaboración del diagrama de cuerpo libre.



**Figura 4-3.** Diagrama de fuerzas en el antebrazo

Fuente: Asqui Byron, 2020.

**Tabla 4-3:** Pesos en el antebrazo

Descripción	Peso(g)
W1 Actuator	55
W2 Actuator	13,4
W3 Actuator	13,4
W4 Estructura	247

Realizado por: Asqui Byron, 2020.

**Tabla 5-3:** Distancias en el antebrazo

Descripción	Distancia (mm)
$d_1$	138
$d_2$	200
$d_3$	247
$d_4$	355

Realizado por: Asqui Byron, 2020.

Cálculos:

$$\tau_1 = W_1 * d_1 = 0,055 * 13,8 = 0,759 \text{ Kg cm}$$

$$\tau_2 = W_2 * d_2 = 0,0134 * 20,0 = 0,268 \text{ Kg cm}$$

$$\tau_3 = W_3 * d_3 = 0,0134 * 24,7 = 0,331 \text{ Kg cm}$$

$$\tau_4 = W_4 * d_4 = 0,247 * 35,5 = 8,7685 \text{ Kg cm}$$

$$\tau_{\text{Antebrazo}} = \tau_1 + \tau_2 + \tau_3 + \tau_4 = 10,13 \text{ Kg cm}$$

$$\tau_{\text{Servo}} = \tau_{\text{Antebrazo}} = \mathbf{10,13 \text{ Kg cm}}$$

Una vez efectuado el análisis de las posiciones críticas en las articulaciones, se tomó en cuenta los torques 13,72 y 10,13 kg cm correspondientes al brazo y antebrazo respectivamente, para la compra de los actuadores.

### 3.5.2. Selección de controlador

El controlador seleccionado fue la Raspberry Pi debido a sus características de ordenador completo multifuncional, pequeño, muy versátil para control y un costo de adquisición bajo en comparación con otros dispositivos.

La variedad de módulos del controlador resultó idónea para el prototipo además que se empleó como servidor de archivos en la manipulación de una gran cantidad de datos para el monitoreo de un proceso específico en tiempo real.

**Tabla 6-3:** Especificaciones técnicas Raspberry Pi 3V

Características:	Descripción	
Procesador:	Broadcom BCM2837BO	
Memoria:	1 GB LPDDR2 SDRAM	
Conectividad:	LAN inalámbrica de 2,4 GHz y 5GHz	
Video y sonido:	1 x HDMI	
Multimedia:	Decodificación H	
Número de pines:	40 pines GPIO	
Potencia de entrada:	5V/2.5 A DC	

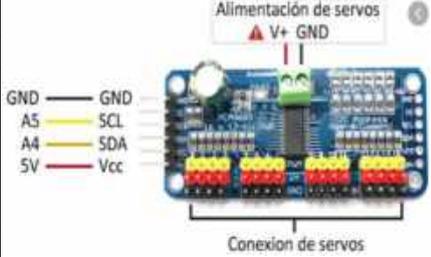
Realizado por: Asqui Byron, 2020.

### 3.5.3. Interfaz de potencia

La interfaz de potencia se encargó de indicar los dispositivos intermedios entre el micro controlador y aparato a manipular, por lo tanto, en esta sección se da a conocer el módulo PCA9685 como el circuito de potencia.

El módulo PCA9685 permitió aumentar el número de salidas PWM necesarios para los servomotores ya que la Raspberry Pi 3 no contaba con estas señales. Este controlador con un circuito compacto en pines de alimentación y comunicación I2C, facilitó el control de los actuadores.

**Tabla 7-3:** Especificaciones técnicas del Módulo PCA9685

Características:	Descripción	
Resolución:	12 bits	
Salidas configurables:	“Push-pull” “drenador abierto”	
Número de canales:	16	
Resistencias:	220 Ohms	
Jumpers soldables:	6	
Control:	Leds y servos	
Número de placas:	1	

Realizado por: Asqui Byron, 2020.

La Tabla 3-3 describe las características técnicas del módulo PCA9685, siendo el parámetro más destacado que posee hasta 16 señales PWM, controlando 16 servos, únicamente con 2 pines, lo que llevó a ser la mejor opción para el brazo prototipo.

## 3.6. Selección de Software

### 3.6.1. Plataforma de soporte IOT

El Soporte de “Internet de las cosas” se basó en dos plataformas virtuales que otorgaron acceso remoto desde cualquier parte del globo terráqueo al ordenador, además de funciones en el prototipo para: editar información, cargar información, eliminar información, modificar nuevos componentes, enlaces, etc. Dichas plataformas se denominan remote it y ubidots.

Remote It se usó como punto de conexión entre el dispositivo Raspberry Pi 3 y el Software Labview. Ubidots dentro del desarrollo del prototipo proporcionó el monitoreo constante.

### **3.6.2. LabVIEW**

El software LabVIEW actuó como enlace al IOT ya que interpretó el flujo de datos que proporcionaban los dispositivos inteligentes (Raspberry Pi 3) y las plataformas de computación en la nube (Remote it, Ubidots).

La integración final de los responsables de la conectividad a la red y al servidor (software LabVIEW, plataformas) se llevó a cabo por protocolos de comunicación como Wi-Fi, Ethernet, telefonía celular, entre otros.

Los estándares que gestionan la comunicación entre los dispositivos y la nube son varios, sin embargo, primordialmente se buscó garantizar un medio confiable con datos en tiempo real a través de la interoperabilidad.

### **3.6.3. Python**

Python fue un lenguaje de programación ideal en los programas de diseño para prototipos y aplicaciones que no requieren cálculos complejos, cabe señalar que el lenguaje Python 2 y 3 se encuentran preinstalados en el sistema operativo Raspbian.

El lenguaje al ser multiparadigma presentó diversas características como: automatizar tareas repetitivas, desarrollar análisis de datos, crear aplicaciones de escritorio, construir páginas web, entre otras. La principal ventaja se evidenció en la multiplataforma (función que permite ejecutarse en cualquier otro sistema) y el código abierto (significa que el lenguaje se desarrollará en cualquier tipo de proyecto).

## **3.7. Configuración de Hardware/Software**

### **3.7.1. Instalación del sistema operativo en la Raspberry pi**

El dispositivo Raspberry Pi en su participación para el desarrollo exitoso del prototipo final se fundamentó en accesorios que garantizaron su correcta funcionabilidad como:

- Memoria micro SD de 16Gb clase 10
- Cargador 5V/2,5 A
- Monitor HDMI
- Teclado USB
- Mouse USB

El sistema operativo en Windows integra un procedimiento metódico, por ende, debe ser instalado correctamente ya que las afecciones se muestran en el funcionamiento imposibilitando su ejecución. A continuación, se detalla el proceso:

- Descargar la Imagen Raspbian
- Descargar e instalar el Programa Win32DiskImager
- Insertar la tarjeta SD a nuestro computador
- Abrir el programa Win32DiskImager, el cual reconocerá automáticamente nuestra tarjeta SD
- Pulsar el icono de búsqueda y seleccionamos la Imagen Raspbian
- Pulsamos clic izquierdo sobre el botón “Write”
- Una vez finalizada la instalación se podrá extraer e insertar la micro SD a nuestra raspberry.

La Configuración Inicial, debe ser realizada en el dispositivo, para garantizar su funcionamiento como se describe en el procedimiento:

- Conectar los accesorios al dispositivo y procedemos a encender el dispositivo
- Visualizar una ventana de bienvenida donde se selecciona la opción “Nex” para continuar las configuraciones
- Configurar las opciones de Ubicación
- Cambiar la contraseña del sistema operativo
- Eliminar el borde de pantalla
- Seleccionar una red Wifi e ingresar la contraseña
- Actualizar el sistema operativo y sus librerías
- Esperar la ventana emergente que indique “System is up to date” y presionar “OK”
- Seleccionar “Restart” para reiniciar el dispositivo con las configuraciones efectuadas.

### ***3.7.2. Configuración inicial para manipular el módulo PCA***

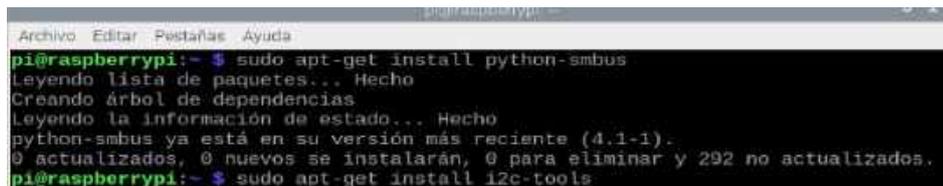
La configuración inicial se llevó a cabo a través de la selección en menú, luego “preferencias” y finalmente clic en “Configuración de Raspberry Pi”. Cuando la ejecución es correcta se visualizará la Figura 5-3, eligiendo entre las opciones la pestaña “Interfaces” para activar la opción “I2C”.



**Figura 5-3.** Configuración de Raspberry Pi  
**Fuente:** Asqui Byron, 2020.

La instalación del SMBUS (Bus Administrador del Sistema) se realizó en la terminal del dispositivo, ingresando el comando “sudo apt-get install Python-smbus”. SMBUS fue esencial para habilitar la biblioteca de Python necesaria en el acceso I2C de la Raspberry Pi.

Al término de proceso anterior se efectuó la búsqueda de la dirección I2C del PCA9685 a través del programa llamado “I2C-tools” ejecutado con el comando “sudo apt-get install i2c-tools”, como lo muestra la Figura 6-3.



**Figura 6-3.** Instalación de paquete “SMBUS”  
**Fuente:** Asqui Byron, 2020.

Al estar las herramientas y biblioteca de la I2C habilitadas correctamente se conectó el módulo PCA9685 a la Raspberry pi con las siguientes condiciones:

- GND (PCA9685) con Pin 9 (GND Raspberry)
- VCC (PCA9685) con Pin 1 o 2 (VCC Raspberry)
- SDA (PCA9685) con Pin 3 (SDA Raspberry)
- SCL (PCA9685) con Pin 5 (SCL Raspberry)

La verificación del módulo PCA conectado al dispositivo se realizó insertando el comando “sudo i2cdetect -y 1” para visualizar una tabla de direcciones con los números “40” y “70” en la columna “0”, la Figura 7-3 es un claro ejemplo.

```
pi@raspberrypi:~$ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40: 40  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70: 70  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

**Figura 7-3.** Verificación del Módulo PCA

Fuente: Asqui Byron, 2020.

El paquete “Git” se obtuvo a partir de la biblioteca “Adafruit”, bajo el comando “sudo apt-get install git build-essential python-dev”, además se clonó la biblioteca del repositorio mediante el comando “git clone https://github.com/adafruit/Adafruit\_Python\_PCA9685.git”

```
pi@raspberrypi:~$ git clone https://github.com/adafruit/Adafruit_Python_PCA9685.git
Clonando en 'Adafruit_Python_PCA9685'...
remote: Enumerating objects: 32, done.
remote: Total 32 (delta 0), reused 0 (delta 0), pack-reused 32
Desempaquetando objetos: 100% (32/32), listo.
pi@raspberrypi:~$
```

**Figura 8-3.** Clonación de la biblioteca “Adafruit”

Fuente: Asqui Byron, 2020.

Finalmente se utiliza el comando “ls” para visualizar los archivos de la carpeta “Pi”. Los comandos “cd Adafruit\_Python\_PCA9685/” con “ls” sirvieron para acceder al archivo de instalación, identificado “setup” digitamos “sudo python setup.py install” para instalar la biblioteca.

```
pi@raspberrypi:~/Adafruit_Python_PCA9685$ ls
Adafruit_Python_PCA9685  Downloads  MagPi  Public  Videos
Desktop                 'Instalando Arduino'  Music  sketchbook
Documents               'Instalando PCA9685'  Pictures  Templates
pi@raspberrypi:~/Adafruit_Python_PCA9685$ cd Adafruit_Python_PCA9685/
pi@raspberrypi:~/Adafruit_Python_PCA9685$ ls
Adafruit_PCA9685  examples  ez_setup.py  LICENSE  README.md  setup.py
pi@raspberrypi:~/Adafruit_Python_PCA9685$ sudo python setup.py install
```

**Figura 9-3.** Instalación de la biblioteca “Adafruit”

Fuente: Asqui Byron, 2020.

### 3.7.3. Enlace Raspberry – Remote it

El enlace Raspberry con la plataforma Remote It se creó a partir de una cuenta en Remote It, para agregar un nuevo dispositivo, las instrucciones se adquirieron del registro, de acuerdo al siguiente proceso:

1. *Crear una cuenta en Remote IT:* Ingresar a la página Remote It, llenar el formulario con su verificación para enlazar diferentes dispositivos a través del internet con algún tipo de comunicación que puede ser: SSH, TCP, HTTP., etc., todo acorde a la necesidad del usuario, para la investigación el monitoreo de datos remotos fue el enlace TCP.



**Figura 10-3.** Plataforma Remot3 it.

Fuente: Asqui Byron, 2020.

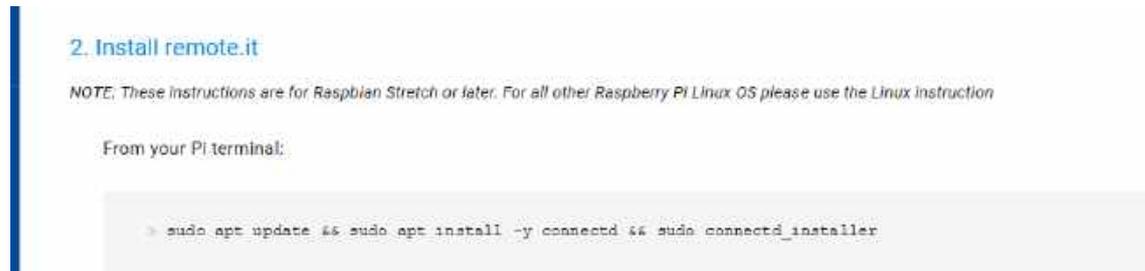
2. *Agregar un nuevo dispositivo:* Seleccionar “ADD DEVICE”.



**Figura 11-3.** Adición del dispositivo.

Fuente: Asqui Byron, 2020.

3. *Adquirir instrucciones para el registro de un dispositivo:* Seleccionar el sistema operativo de la Raspberry Pi y seguir los pasos que se muestran en pantalla.



**Figura 12-3.** Instrucciones en Remote.

Fuente: Asqui Byron, 2020.

4. *Ejecutar las instrucciones de Remote It:* Abrir la terminal, determinar la versión del sistema operativo, actualizar la lista de paquetes del repositorio de Linux mediante el comando “sudo apt update”, descargar e instalar el paquete de Remote con “sudo apt install -y connectd”, registrar y configurar el dispositivo con “sudo connectd\_installer” además de ingresar usuario y contraseña.

```

pi@raspberrypi:~$ sudo connectd_installer
remote.it connection installer Version: 2.3.17 lib_v2.1.17
Modified: April 16, 2019 (library) April 17, 2019
Build date: Thu Apr 18 18:21:59 PDT 2019
curl with SSL support installed.

Checking the daemon for compatibility...

Using architecture arm-linux-pi...

Checking your network for compatibility...

Network connection OK...

```

**Figura 13-3.** Ejecucion de Instrucciones.  
Fuente: Asqui Byron, 2020.

5. *Registrar el dispositivo en Remote:* Asignar un nombre al dispositivo para la plataforma, elegir la primera opción que se muestra en pantalla para añadir un nuevo servicio a la cuenta de remote, seleccionar el protocolo TCP e ingresar el puerto “9999”.

```

===== Installed remote.it Services =====
Service Name | Protocol | LAN Address | Port | Application
=====
Device Name: Raspberry Platform: arm-linux-pi
Hardware ID: b8:27:eb:e9:16:9c-kLHMKELcePIMgREQaaIS

===== Main Menu =====

1) Attach/reinstall a remote.it Service to an application
2) Attach/reinstall a remote.it Service to a LAN application
3) Remove a remote.it Service from an application
4) Remove all remote.it Services, then exit
5) Exit

'application' is any TCP service (e.g. SSH, VNC, HTTP, etc.)
running on this device. 'LAN application' is a TCP service
running on another device on this LAN.

Choose a menu selection (1 - 5):

```

**Figura 14-3.** Registro en Remote.  
Fuente: Asqui Byron, 2020.

6. *Verificar el registro en Remote:* Ingresar en la plataforma de Remote, a partir de este momento la cuenta mostrará el dispositivo dado de alta, al igual que el servicio creado con la dirección y el puerto de conexión de la raspberry pi.



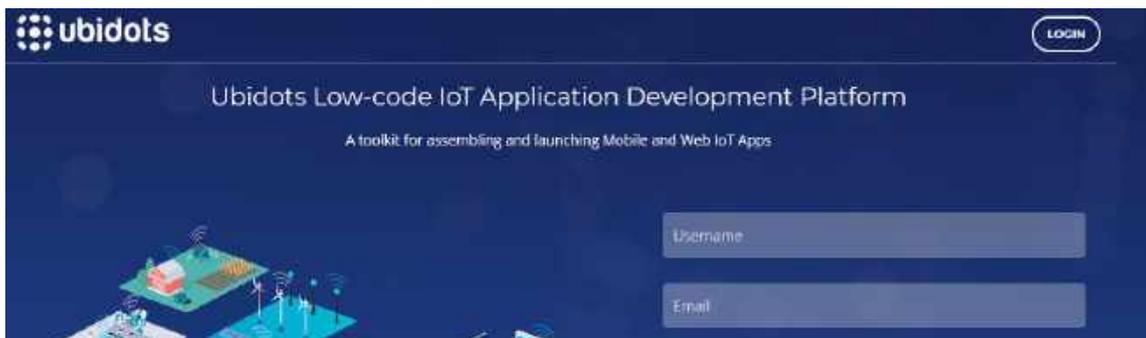
**Figura 15-3.** Verificación del registro.  
Fuente: Asqui Byron, 2020.

### 3.7.4. Enlace Raspberry – Uidots

La aplicación Ubidots es una plataforma IOT que envía y recepta datos a la nube, se conecta con diferentes plataformas, configura tableros y alertas, el usuario podrá tomar decisiones en cualquier parte del mundo.

*Procedimiento:*

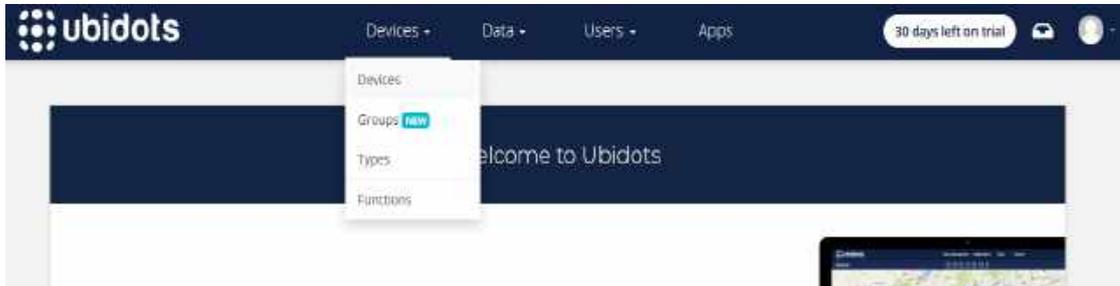
1. *Crear una cuenta en Ubidots:* Ingresar a la página de Ubidots, llenar un formulario, acceder a los beneficios de la plataforma.



**Figura 16-3.** Plataforma Ubidots.

Fuente: Asqui Byron, 2020.

2. *Agregar un nuevo dispositivo:* Registrar el dispositivo, seleccionar en devices.



**Figura 17-3.** Dispositivos en Ubidots .

Fuente: Asqui Byron, 2020.

3. *Adquirir instrucciones para el registro de un dispositivo:* Identificar y seleccionar el tipo de dispositivo, elegir el enlace, mismo que redireccionará a otra página con las respectivas instrucciones de registro.



**Figura 18-3.** Adquisición de instrucciones.

Fuente: Asqui Byron, 2020.

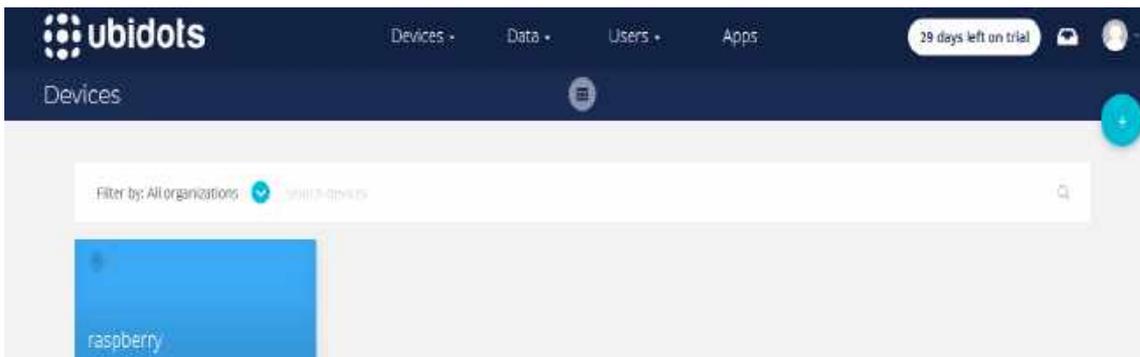
4. *Ejecutar las instrucciones:* Dirigirse a la terminal del dispositivo, actualizar los repositorios e instalar la biblioteca “requests”, crear un archivo python con el programa Thony Pytonn IDE que está instalado por defecto, ingresar un código, por ejemplo, de Ubidots, el dispositivo requerirá una clave temporal para acceder a la plataforma, ir a la opción API credentials y escoger el token, modificar el código pyhton e iniciar dicho archivo.



**Figura 19-3.** Default token .

Fuente: Asqui Byron, 2020.

- 5.- *Verificar el registro:* La cuenta Ubidots presentará el dispositivo dado de alta, además del flujo continuo de datos, por ende, se determina el registro de la Raspberry realizado correctamente.



**Figura 20-3.** Dispositivos conectados.

Fuente: Asqui Byron, 2020.

### 3.7.5. *Diseño HMI Python*

La interfaz gráfica cumple la función de proporcionar un entorno visual amigable y sencillo de utilizar, facilitando la comunicación del usuario con el equipo, razón por la cual es indispensable verificar la versión de Python. En el prototipo se generó códigos interpretados para la versión 2.7 de Python.

#### **Librerías**

Las librerías tienen el siguiente proceso:

- Crear un archivo Python con editor o intérprete.
- Programar el desarrollo de la interfaz gráfica, importando la librería Tkinter (paquete estándar instalada por defecto en el intérprete Python del dispositivo).

```
from Tkinter import *
import Tkinter as tk
import tkMessageBox
```

**Figura 21-3.** Librería Tkinter.

Fuente: Asqui Byron, 2020.

La Figura 21-3 enseña la librería “Tkinter” y el módulo “tkMessageBox” para mostrar mensajes al usuario.

## Ventanas

Una ventana es un área visual rectangular con cuadros de texto, botones, imágenes, etc., permitiendo interactuar con el prototipo. La Figura 22-3 indica un bloque códigos utilizados para la creación de la ventana.

```
root = Tk()
root.title("Control de Brazo Robot")
root.minsize(800,650)
root.resizable(width=False, height=False)
#Instrucciones
#Instrucciones
#Instrucciones
root.mainloop()
```

**Figura 22-3.** Código ventana.

Fuente: Asqui Byron, 2020.

Los métodos de Tkinter cumplen las siguientes funciones:

- Title: Utilizada para cambiar el nombre de la ventana
- Minsize: Sirve para minimizar la ventana con ciertas dimensiones
- Resizable: Habilita/deshabilita el redimensionamiento de la ventana

La Figura 23-3 presenta códigos “withdraw”, “deiconify” y “destroy” para tareas como: ocultar, visualizar y cerrar la ventana cuando el programa está en ejecución. Para cuadros de dialogo se tiene las opciones: “showinfo”, “showwarning” y “askquestion”.

```
root.withdraw()
root.deiconify()
root.destroy()

tkMessageBox.showinfo("Informacion", "Texto")
tkMessageBox.showwarning("Advertencia", "Texto")
tkMessageBox.askquestion("Interrogacion", "Texto")
```

**Figura 23-3.** Metodos de Tkinter.

Fuente: Asqui Byron, 2020.

## Etiquetas

La etiqueta es una casilla para textos, destinado a orientar al usuario sobre la interfaz gráfica.

```
var3 = StringVar()
etiqueta3 = Label(root, textvariable=var3,width="18", relief=FLAT,bg="bisque", pady=5)
var3.set("Password")
etiqueta3.grid(column=1, row=4)
```

**Figura 24-3.** Código etiqueta de visualización.

Fuente: Asqui Byron, 2020.

La Figura 24-3 presenta una variedad de códigos con sus funciones detalladas a continuación:

- “StringVar”: Utilizada para declarar una variable tipo cadena
- “Label”: Permite mostrar un texto en una ventana
- “set”: Asigna un valor a una variable de control
- “grid”: Permite posicionar los widgets en una celda específica

## Caja de texto

La caja de texto se utiliza para introducir información por pantalla, almacenándola en una variable. El control tiene un ligero cambio en el código anterior debido que reemplaza la función “Label” por “Entry”, además el método “set” permite que el cuadro inicie sin ningún texto en su interior.

```
var4 = StringVar()
etiqueta4 = Entry(root, textvariable=var4 ,width="18", show="*")
etiqueta4.grid(column=1, row=5)
```

**Figura 25-3.** Código entrada de información.

Fuente: Asqui Byron, 2020.

## Imágenes

Una imagen es una representación visual de un objeto real o imaginario. La función “Label” además de mostrar textos en la pantalla tiene capacidad de publicar imágenes siempre y cuando se ligue el widget a un objeto de tipo “PhotoImage” o “BitmapImagen”.

Si se emplea “PhotoImage” se debe proporcionar el nombre de la imagen al atributo “file”, si la imagen no se encuentra en el mismo directorio del archivo “.py” se coloca la ruta completa, admite formatos “GIF” y “PNG” entre otros.

La Figura 26-3 presenta los tipos de códigos para ingresar imágenes a la aplicación.

```
##Fondo de pantalla
img = PhotoImage(file="FondoT1.png")
widget = Label(root, image=img)
widget.place(x=0,y=0,relwidth=1.0,relheight=1.0)

##Imágenes en celdas específicas
img2 = PhotoImage(file="Logo.png")
img2 = img2.subsample(2,2)
widget2 = Label(root, image=img2)
widget2.grid(column=1, row=1)
```

**Figura 26-3.** Código Imágenes.

Fuente: Asqui Byron, 2020.

## Botones

Widget Button (Botón) es el componente en el cual se desarrolla aplicaciones, asociando una variedad de funciones, al hacer clic sobre el componente.

La creación de este botón es similar a la de una etiqueta, en este caso el comando “Button” asociará las instrucciones del código Python.

```
Binf = Button(root,
              text="Modo de uso",
              bg='lightPink1',
              activebackground='SeaGreen1',
              relief=GROOVE,
              width="15",
              command=info)
Binf.grid(column=1,row=3)
```

**Figura 27-3.** Código Botones.

Fuente: Asqui Byron, 2020.

La Figura 27-3 revela una variedad de parámetros para la creación de un botón, las principales son:

- “text”: Utilizado para mostrar un texto dentro de un botón.
- “command”: Acciona una función cuando el operador presione el componente.

## Scale

El widget Escala se maneja por el control deslizante en las aplicaciones de Python para establecer algún valor entero o flotante dentro de un rango especificado.

La Figura 28-3 tiene las instrucciones en la creación de un widget escala.

```
angulo1 = Scale(root,
                 command = serv1,
                 from_=0,
                 to = 160,
                 orient = HORIZONTAL,
                 length = 200,
                 troughcolor = 'mint cream',
                 bg='LightSkyBlue1',
                 activebackground='SeaGreen1',
                 width=30,
                 cursor='dot',
                 label = 'Posicion Base')
angulo1.grid(column=3, row=3)
angulo1.set(0)
```

**Figura 28-3.** Código Scale.

Fuente: Asqui Byron, 2020.

Los parámetros principales del control deslizante son:

- “command”: Acciona una función cuando el operador presione el componente.
- “from”: Representa un extremo del rango de valores del widget
- “to”: Representa el otro extremo del rango valores
- “label”: Utilizada para mostrar un texto dentro del widget

En este apartado el Diseño de la Interfaz Gráfica HMI permite comunicarse con el sistema mostrando los datos en tiempo real para que los usuarios controlen el prototipo de brazo robótico.

### 3.7.6. Algoritmo de control Python

El algoritmo de control detalla adecuadamente una estrategia de desarrollo en las instrucciones de forma ordenada y secuencial para guiar un proceso a la solución de un problema; por ende, la creación de este algoritmo en la investigación figuró como la infraestructura lógica para la manipulación de datos y control de actuadores.

### Librerías

Las librerías por lo general promueven la correcta compilación y ejecución del código de programación respondiendo al siguiente proceso:

- Ubicarse en el archivo Python de la interfaz gráfica
- Añadir las líneas de código de la Figura 29-3.

```
import time
import Adafruit_PCA9685
import socket
import threading
from numpy import interp
import requests
import math
import random
```

**Figura 29-3.** Librerías.

Fuente: Asqui Byron, 2020.

En la Figura 29-3 se observa variedad de librerías con funciones individuales detalladas a continuación:

- **Adafruit\_PCA9685:** Controlar el módulo PCA
- **time:** Agregar un tiempo de espera.
- **socket:** Intercambiar cualquier flujo de datos
- **threading:** Efectuar operaciones simultaneas.
- **interp:** Escalamiento/conversor de datos
- **requests:** Realizar peticiones HTTP, FTP
- **math:** Ejecutar cálculos matemáticos.
- **random:** Generar datos aleatorios.

## Variables

Las variables son ubicaciones de memoria que almacenan información. Además, las variables de Python no necesitan una declaración explícita, según el tipo de variable, el intérprete reserva la memoria y decide que almacenar en la memoria reservada.

```
val=@
DEVICE_LABEL = "Raspberry"
dt1=str(@)
```

**Figura 30-3.** Variables.

Fuente: Asqui Byron, 2020.

En la imagen se muestra algunas maneras de asignar información a una variable, las cuales permitirán controlar procesos, sentencias y el intercambio de flujo de datos con las plataformas IOT.

## Funciones

Una función es un bloque de código ejecutable en cualquier momento, en otras palabras, es una forma conveniente de dividir los códigos para entender y ahorrar tiempo en la creación de un algoritmo de control.

```

#####Funcion
def unir():
    #Instrucciones
    #Instrucciones

unir()

#####Hilo de programacion
def accept():
    #Instrucciones
    #Instrucciones

Conectar= threading.Thread(name="hilo_1", target=accept)
Conectar.start()

```

**Figura 31-3.** Funciones.

Fuente: Asqui Byron, 2020.

En la Figura 31-3 identifica dos ejemplos de código: el primer caso representa la forma de crear una función y como llamarlo, mientras que el segundo es un hilo de programación que permite ejecutar simultáneamente varias operaciones en el mismo espacio necesariamente acudiendo a un módulo “threading”.

El objeto “Thread” es el argumento “target” empleado para establecer el nombre de la función, una vez que el hilo se ha creado, inicia con el método “start”; por ende, la programación gana velocidad en el procesamiento de códigos.

### Conexión con Remote It

Un sistema de control y monitoreo requiere establecer un bloque de código que permita escuchar peticiones de conexión antes de emitir respuestas, bajo esta disposición se crea comunicación entre dos equipos a través de la red, cabe aclarar que si la Raspberry Pi posee acceso a internet se conectará automáticamente a Remote debido a su registro previo en la plataforma.

```

socket_s = socket.socket()
host = ''
port = 9999
backlog = 1
socket_s.bind ((host,port))
socket_s.listen(backlog)
print ("ESPERANDO UNA CONEXION ... :|")
socket_s, (host,port) = socket_s.accept()
print ("CONEXION ESTABLECIDA ..... :)")

```

**Figura 32-3.** Conexión Cliente/Servidor.

Fuente: Asqui Byron, 2020.

La Figura 32-3 presenta una forma sencilla de realizar conexiones cliente/servidor mediante el módulo “socket” de Python para realizar conexiones TCP/IP.

Las líneas de código de la Figura 33-3 emplea los métodos “send” y “recv” para el envío y recepción de datos respectivamente, “close” es primordial utilizarlo al final de la programación ya que garantiza el cierre correcto de la comunicación evitando errores en futuras conexiones.

```
socket_s.send(cadena)
client = socket_s.recv(1)
socket_s.close()
```

**Figura 33-3.** Metodos de Socket.  
Fuente: Asqui Byron, 2020.

### Control del módulo PCA

Los actuadores del sistema son controlados por el Módulo PC9685, para esta acción se muestra la Figura 34-3 con dos secciones de código: la primera tiene dos instrucciones al llamado del módulo PCA y establece una frecuencia de operación, mientras que la segunda resulta ser una función que interpreta los valores obtenidos del widget Escala a otra escala de valores para generar señales PWM a través del método “set\_pwm”.

```
pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(60)

def servol(posiciones1):
    global salida
    salida=posiciones1
    salida=interp(salida, [0,160], [150,600])
    salida=int(salida)
    pwm.set_pwm(0, 0, salida)
```

**Figura 34-3.** Control de PCA.  
Fuente: Asqui Byron, 2020.

### Conexión con Ubidots

Un código base para el envío de datos a Ubidots no es necesario, ya que la plataforma lo proporciona. ¿Cuál es el código?, La programación empleada es el registro del dispositivo y se encuentra en la dirección: <https://bit.ly/2xKzZzj>.

La programación mencionada debe ser copiada y pegada en el código principal añadiendo las variables necesarias para observar los datos en la plataforma además de considerar los códigos que inician y finalizan la comunicación con Ubidots.

```

if __name__ == '__main__':
    while (True):
        main()
        time.sleep(1)

w=0
def ubidots_web():
    global w
    while (w==0):
        main()
        time.sleep(1)

```

**Figura 35-3.** Reemplazo de Códigos.

Fuente: Asqui Byron, 2020.

La Figura 35-3 manifiesta dos bloques: el primero simboliza programación para eliminar, el segundo es el código que se ingresa como reemplazo del bloque eliminado.

La Figura 36-3 cuenta con dos líneas de código para aludir a la función “ubidots\_web” con un solo hilo de programación sin ralentizar la aplicación.

```

ubidotss = threading.Thread(name="hilo_3", target=ubidots_web)
ubidotss.start()

```

**Figura 36-3.** Llamado al hilo “ubidots\_web”.

Fuente: Asqui Byron, 2020.

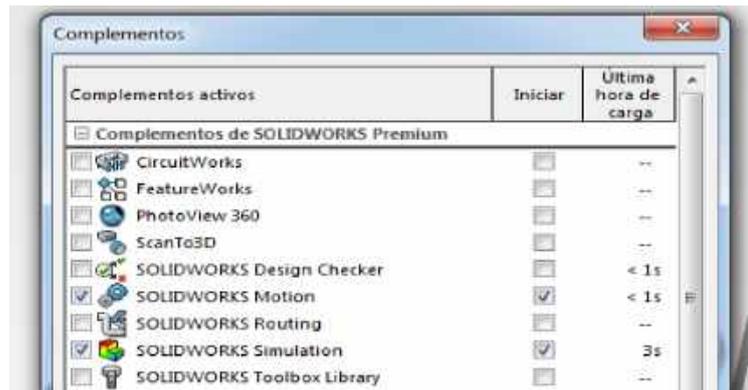
### 3.7.7. Enlace LabVIEW – SolidWorks

El enlace LabVIEW- SolidWorks en la investigación se formuló de acuerdo a la conexión en Hardware y Software con:

- LabVIEW 2014 (32 bits)
- LabVIEW NI Soft Motion Module Standard or Premium
- SolidWorks 2017
- LabVIEW Real Time Module 2014
- Procesador Intel (R) Core (TM) i3 M350 @ 2.27 GHz 2.27 GHz
- Memoria Ram de 8.0 GB

### Activación del módulo SOFT-MOTION

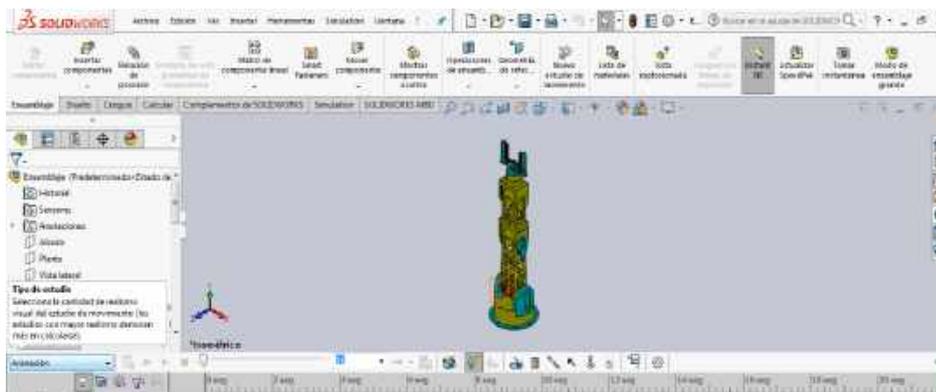
La activación del módulo se generó en SolidWorks bajo el procedimiento: dirigirse a la interfaz del software en la parte superior hasta “opciones”, desplegar la lista de configuraciones, seleccionar “complementos” y marcar en la ventana las opciones “SolidWorks Motion” y “SolidWorks Simulation”.



**Figura 37-3.** Complementos de Solidworks .  
Fuente: Asqui Byron, 2020.

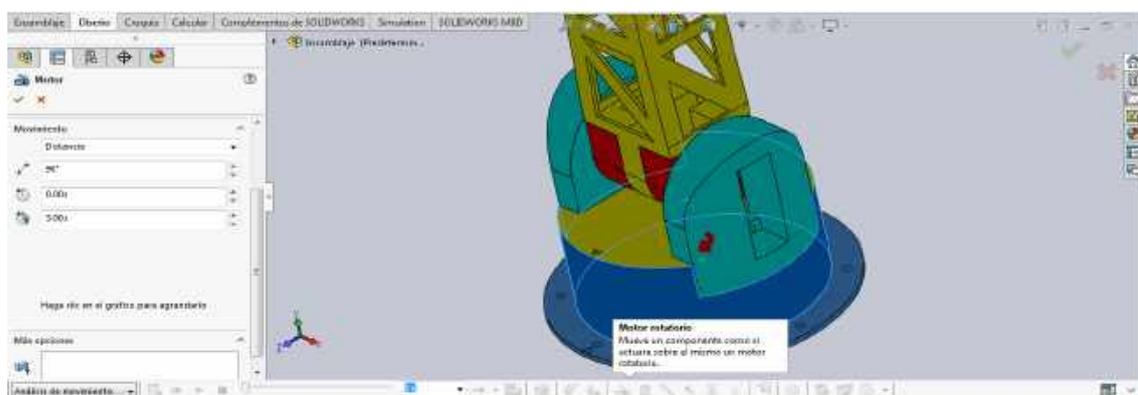
## Análisis de Movimiento

1. *Seleccionar el tipo de estudio:* Se llevó a cabo por seleccionar la pestaña “Estudio de movimiento 1” y asegurar que el menú desplegable este configurado en Análisis de movimiento.



**Figura 38-3.** Ensamblaje del Brazo.  
Fuente: Asqui Byron, 2020.

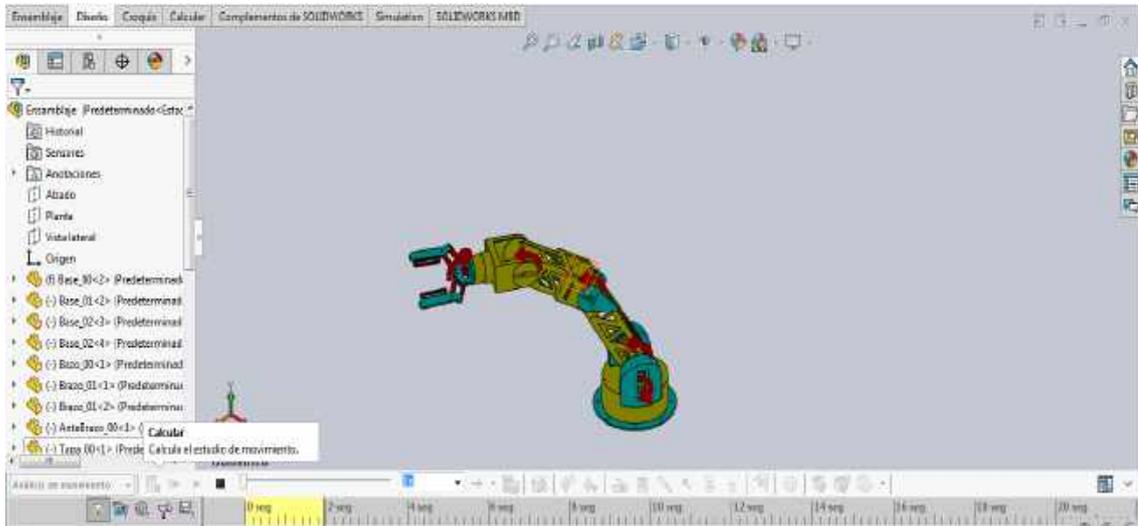
2. *Adicionar motores:* Dar clic izquierdo sobre el icono “motor rotatorio”, elegir la ubicación del motor, dirección de giro y componente de referencia, seleccionar “Movimiento”, aparecerá una lista de alternativas, marcar la opción “Distancia” (requisito para la simulación).



**Figura 39-3.** Inserción de motores.  
Fuente: Asqui Byron, 2020.

Nota: Este proceso se realiza para cada eslabón que posea movimiento.

3. *Simular*: Presionar “calcular” para iniciar el estudio de movimiento (encargado de verificar el funcionamiento de los eslabones que son parte del brazo).



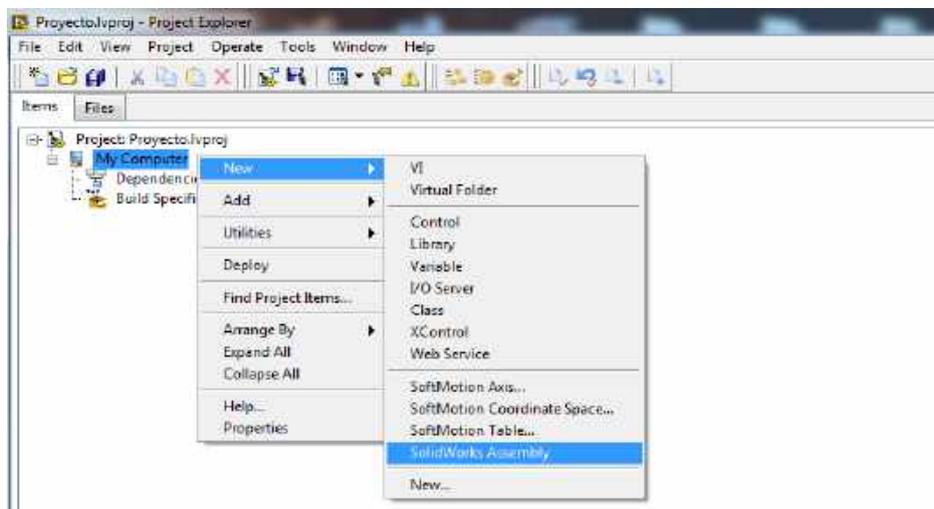
**Figura 40-3.** Simulación

Fuente: Asqui Byron, 2020.

## Vinculación de programas

Enlazar LabVIEW con SolidWorks y configurar los parámetros de sincronización.

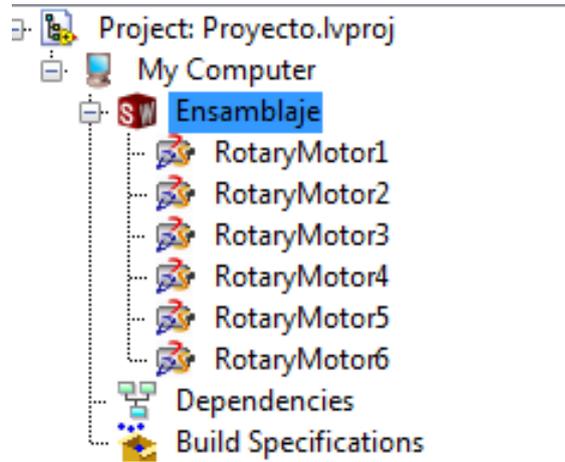
1. *Enlazar los programas*: Abrir un nuevo proyecto de LabVIEW, efectuar clic derecho sobre “My Computer”, ubicarse en “New”, seleccionar la opción “SolidWorks Assembly”, en la nueva ventana si el proyecto de SolidWorks está abierto lo reconocerá automáticamente, dar clic en la opción “OK”, por el contrario, si no reconoce el programa, explorar e importar el ensamblaje con la opción “Browse”.



**Figura 41-3.** Adición del ensamblaje

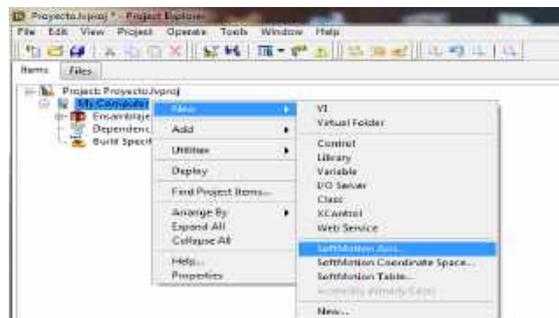
Fuente: Asqui Byron, 2020.

Al realizar la importación del archivo ensamblaje se observará los Motores rotatorios ingresados en el análisis de movimientos como lo muestra la Figura 42-3.



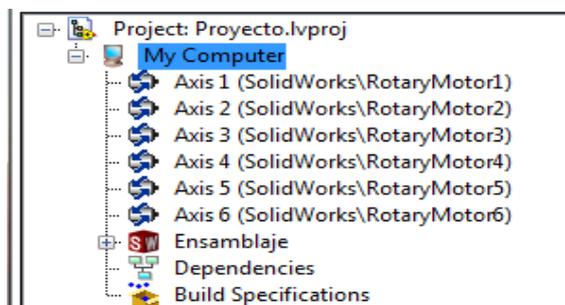
**Figura 42-3.** Motores rotatorios en LabVIEW  
Fuente: Asqui Byron, 2020.

2. *Colocar axis:* Dar clic derecho sobre “My Computer”, colocarse en “New”, seleccionar la opción “SoftMotion Axis”, aparecerá una pantalla para adicionar los Axis en el proyecto, presionar el icono “Add New Axis” según el número de motores a controlar. Una vez realizada la acción, seleccionar “OK” para ingresar los Axis al sistema. Figura 43-3.



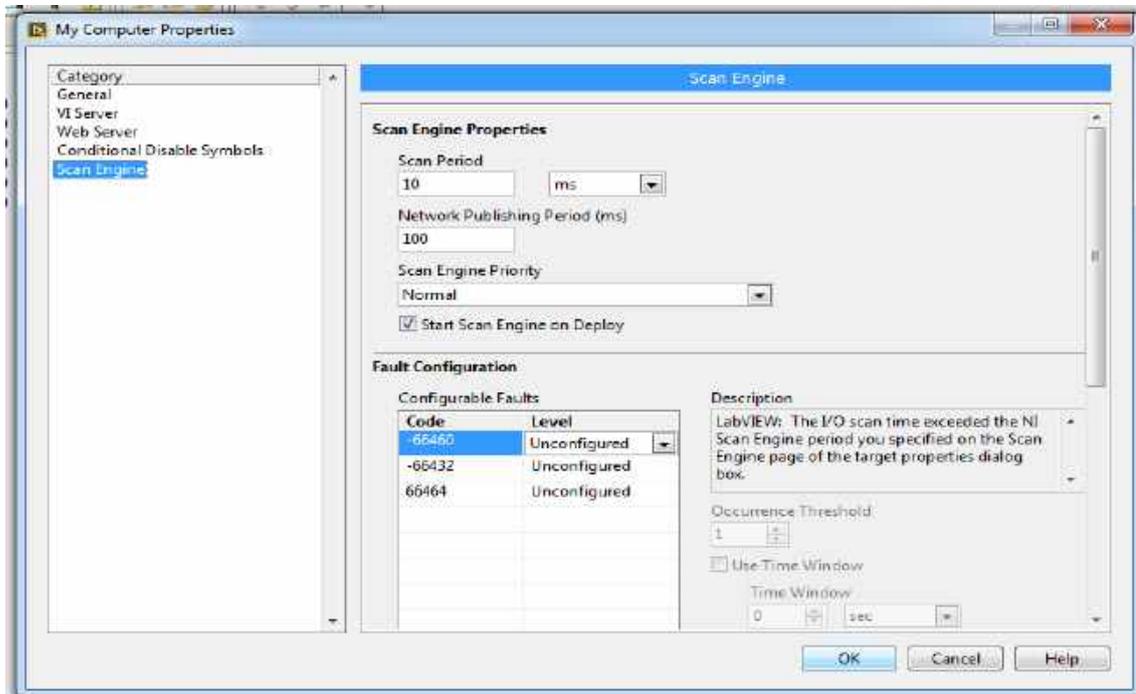
**Figura 43-3.** Colocación de axis  
Fuente: Asqui Byron, 2020.

La Figura 44-3 pone en evidencia los axis del sistema.



**Figura 44-3.** Axis en LabVIEW  
Fuente: Asqui Byron, 2020.

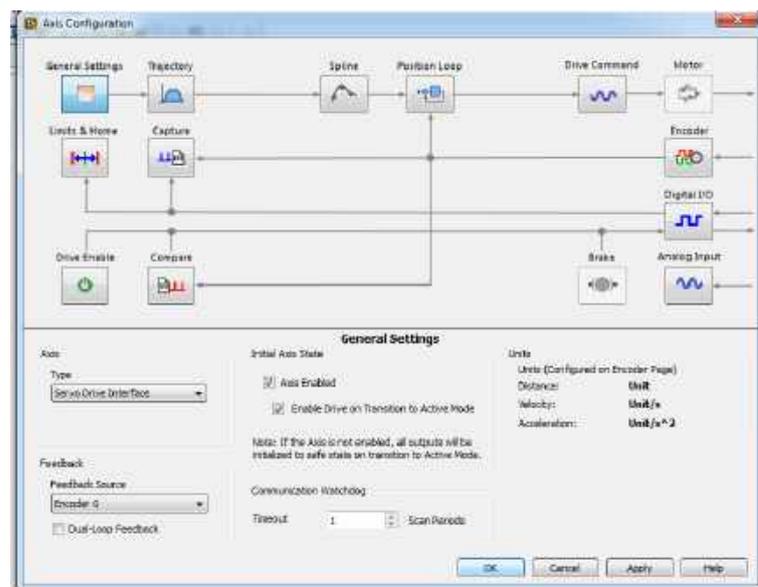
3. *Configurar la propiedad de escaneo:* Pulsar clic derecho en “My Computer”, seleccionar “properties”, en la nueva ventana, elegir la opción “Scan Engine”, en la sección de propiedades activar la casilla “Star Scan Engine on Deploy”.



**Figura 45-3.** Propiedades en My Computer

Fuente: Asqui Byron, 2020.

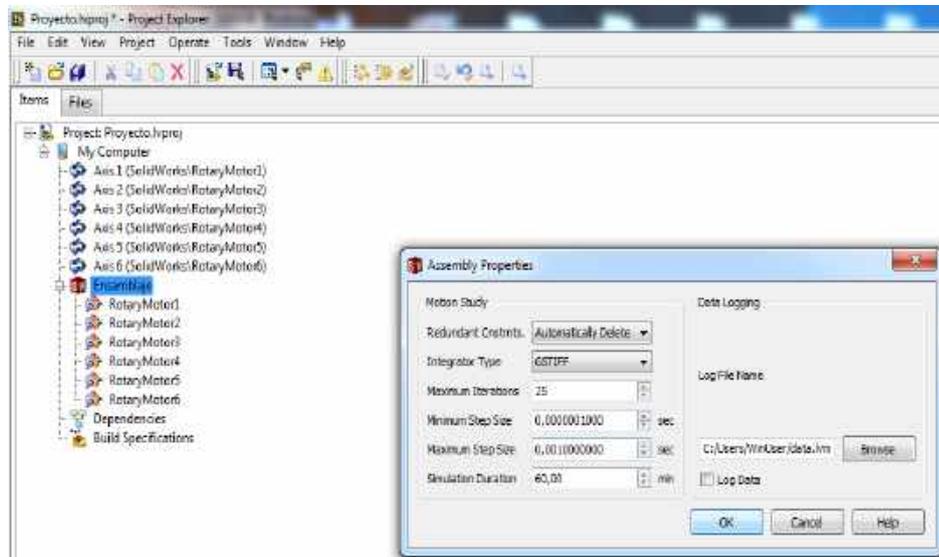
4. *Configurar las propiedades de los axis:* Se realiza con cada axis del proyecto, ir a propiedades, en la ventana de configuración habilitar la casilla “Enable Drive on Transition to Active Mode”, activar automáticamente los ejes cuando el “NI Scan Engine” cambia al modo Activo.



**Figura 46-3.** Configuración de axis

Fuente: Asqui Byron, 2020.

5. *Configurar el ensamblaje:* Efectuar clic derecho en “Ensamblaje”, ingresar en propiedades, modificar la opción “maximum Step Size” de 0,01 a 0,001 con valores óptimos para que el software calcule las lecturas de animación.



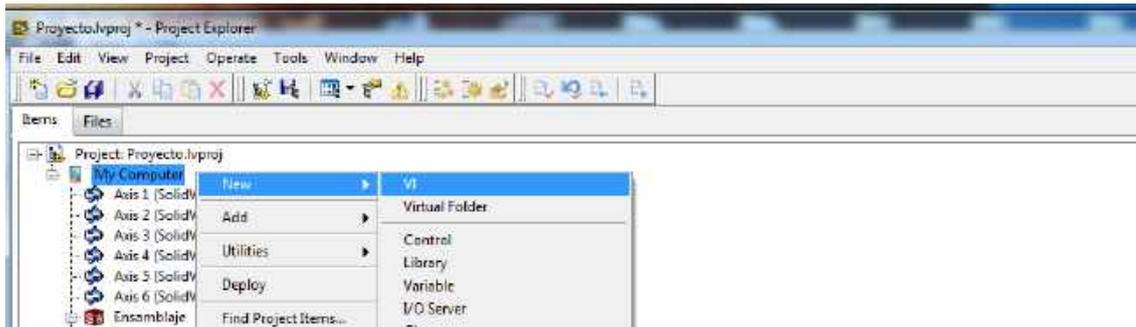
**Figura 47-3.** Propiedades de Ensamblaje  
Fuente: Asqui Byron, 2020.

### 3.7.8. Prueba de conexión LabVIEW – SolidWorks

La conexión entre los dos softwares inició con el siguiente procedimiento:

- Crear un VI en LabVIEW
- Diseñar un interfaz de mando en LabVIEW
- Configurar el mando numérico
- Adicionar un ciclo repetitivo
- Ingresar una función para controlar motores de SolidWorks
- Realizar las conexiones para el control de un motor
- Considerar los pasos 5 y 6 para cada Axis
- Poner en marcha, verificar y detener la simulación

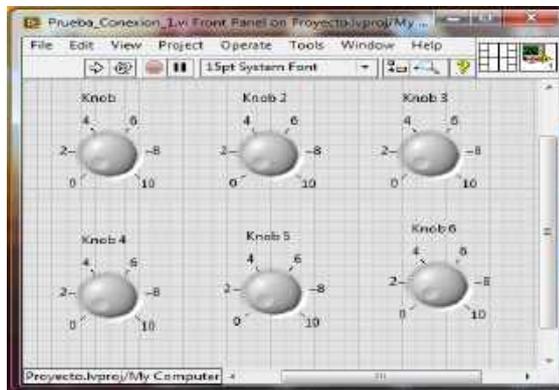
1. *Crear un VI en LabVIEW:* Dar clic derecho sobre “My Computer”, seleccionar “New” y elegir “VI”. Figura 47-3. En las dos ventanas siguientes se diseña y programa acorde a la necesidad.



**Figura 48-3.** Creación de VI en LabVIEW

Fuente: Asqui Byron, 2020.

2. *Diseñar una interfaz de mando en LabVIEW:* Adicionar 6 mandos numéricos (servirán para manipular cada articulación del brazo en el entorno SolidWorks), pulsar clic derecho sobre el panel frontal, en la nueva ventana seleccionar “Numeric” y elegir la opción “Knob”.



**Figura 49-3.** Interfaz de mando en LabVIEW

Fuente: Asqui Byron, 2020.

3. *Configurar el mando numérico:* Ingresar a las propiedades de mando, configurar una escala de 0 a 180 grados para mejorar el aspecto de los movimientos. Figura 50-3.



**Figura 50-3.** Propiedades del mando numérico

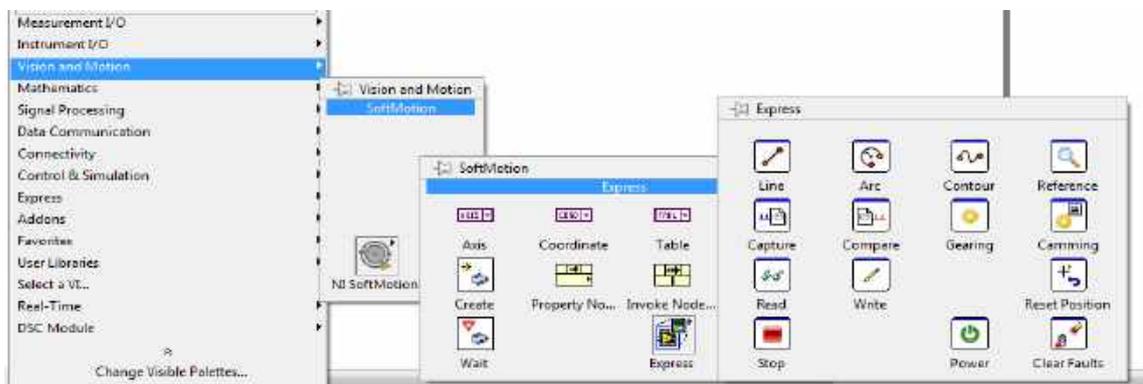
Fuente: Asqui Byron, 2020.

4. *Adicionar un ciclo repetitivo:* Dar clic derecho en el diagrama de bloques, en la nueva ventana seleccionar “Structures”, ingresar el bucle “while loop”, adicionar un control para detener el ciclo repetitivo, pulsar clic derecho sobre el punto rojo y escoger “Create Control”.



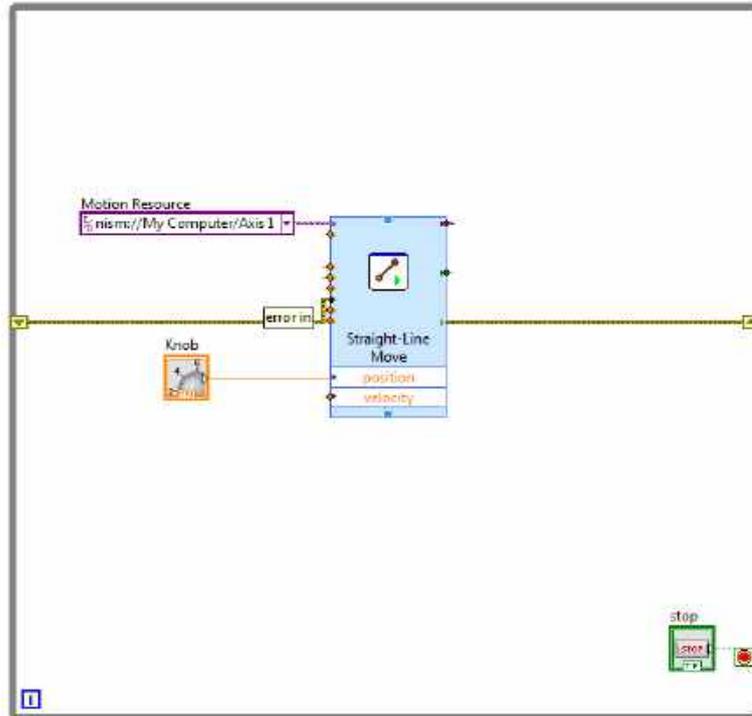
**Figura 51-3.** Bucle while loop  
**Fuente:** Asqui Byron, 2020.

5. *Ingresar una función para controlar motores de SolidWorks:* Pulsar clic derecho sobre el diagrama de bloques, en “Vision and motion” optar por “NI SoftMotion” y luego “Express” para añadir la función “Line”, cambiar el método relativo a absoluto. Figura 52-3.



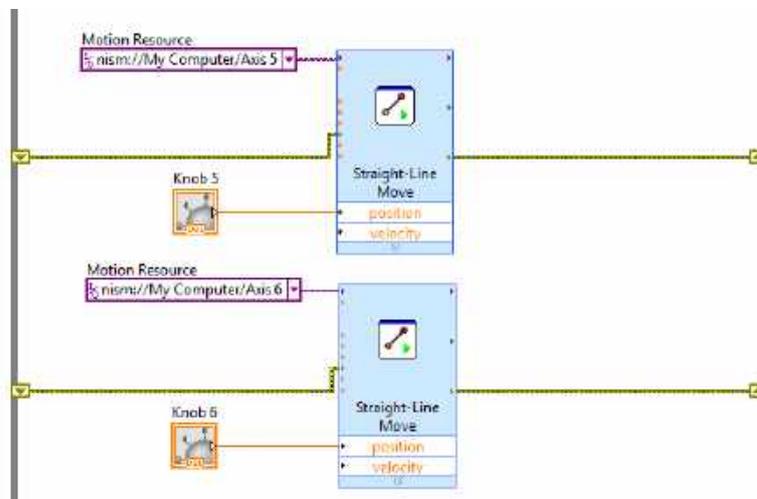
**Figura 52-3.** Funciones de Express en LabVIEW  
**Fuente:** Asqui Byron, 2020.

6. *Ejecutar las conexiones necesarias para el control de un motor:* abrir la ventana “Project Explore”, arrastrar el “Axis” al diagrama de bloque para conectar con “resource” de la función line, conectar “Knob” con “position” para manipular los motores de Solidwork, relacionar “error out” con el bucle “while loop”, dar clic derecho sobre el túnel, seleccionar “Replace with Shift Register” para transferir la información de error a la siguiente iteración, finalizar con la conexión “error in” con “shift register” creado al lado izquierdo del bucle.



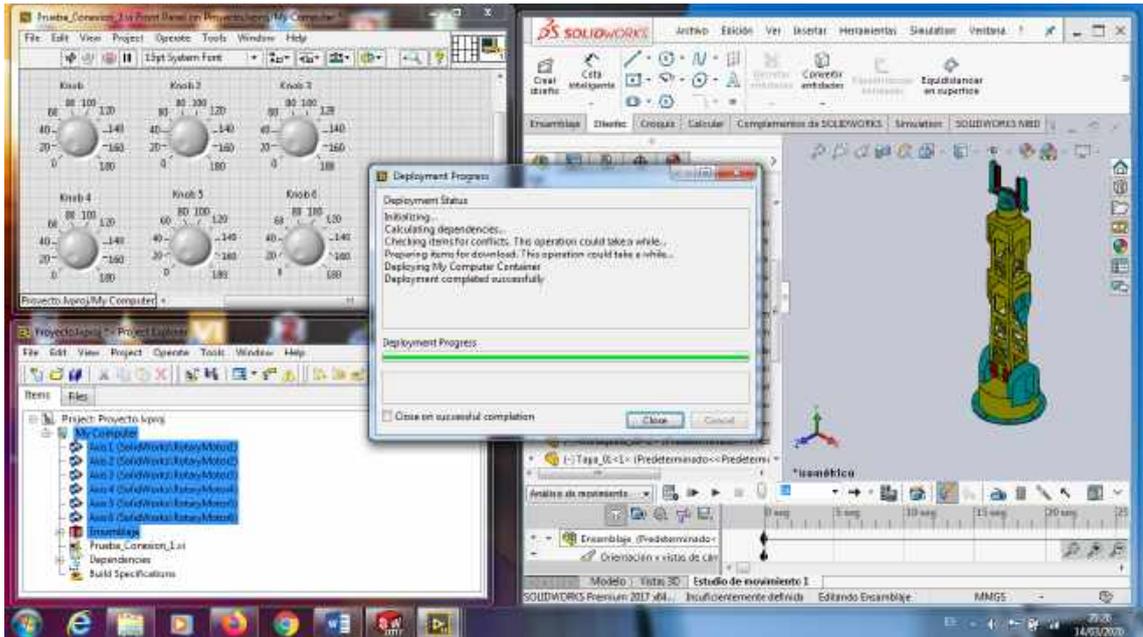
**Figura 53-3.** Conexiones con la función Line  
Fuente: Asqui Byron, 2020.

7. *Repetir los pasos 5 y 6 para cada Axis:* Esta acción es necesaria para comprobar el movimiento de cada eslabón.



**Figura 54-3.** Otras conexiones.  
Fuente: Asqui Byron, 2020.

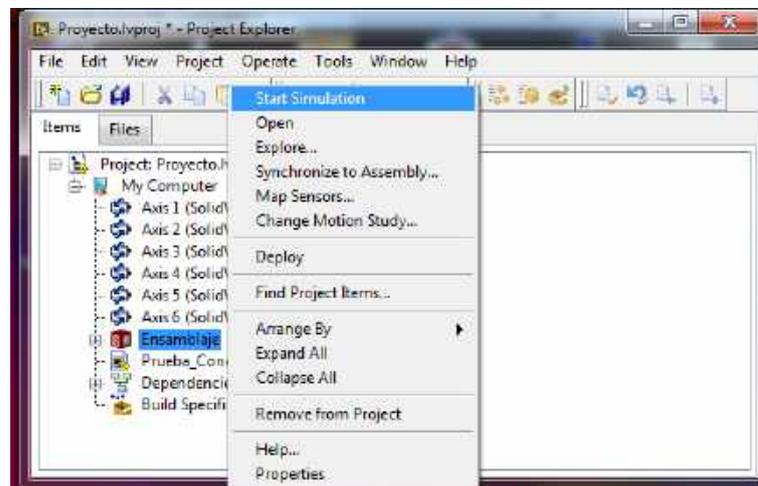
8. *Poner en marcha la simulación:* ubicarse en la ventana “Project Explore”, escoger My Computer, los Axis y el ensamblaje de SolidWorks, hacer clic derecho seleccionando inmediatamente “Deploy” para que cargue la comunicación, si se presenta una resolución de conflictos dar clic en “Apply” caso contrario pulsar “close”. Figura 55-3.



**Figura 55-3.** Verificación de conflictos

Fuente: Asqui Byron, 2020.

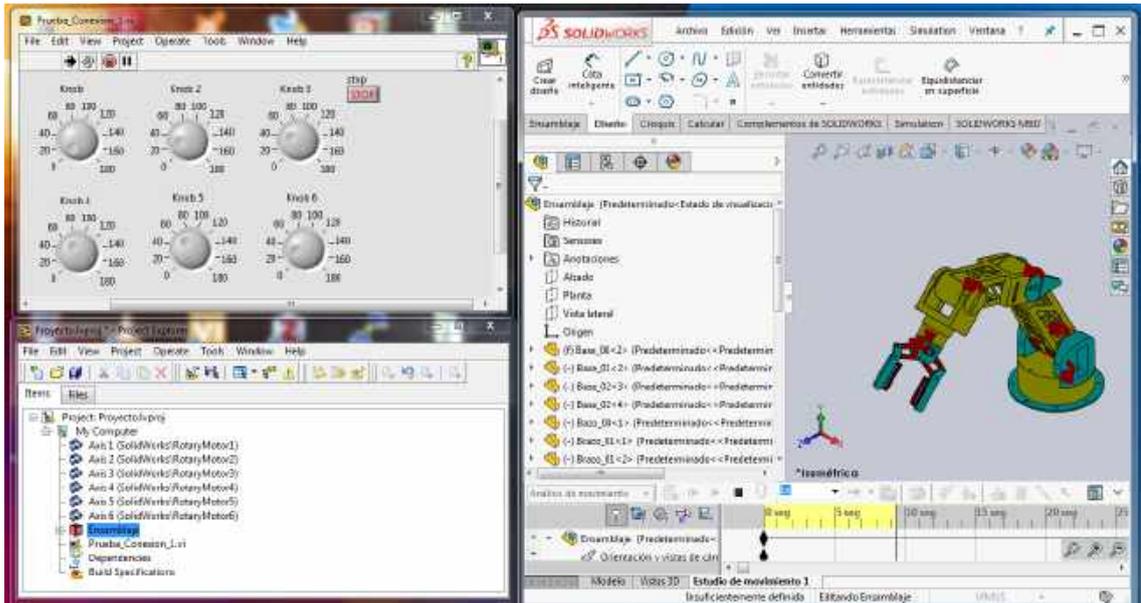
La Figura 56-3 se genera a partir de hacer clic derecho sobre el ensamblaje de SolidWorks, elegir la opción “Start Simulation”, observar las flechas del cambio en la línea de tiempo para el entorno SolidWorks (es una clara señal que ha comenzado la espera de información para los movimientos).



**Figura 56-3.** Simulación en marcha.

Fuente: Asqui Byron, 2020.

9. *Verificar la simulación:* Presionar el icono “Run” del panel frontal para cargar la programación del diagrama de bloques, dar posiciones con “Knob” a cada eslabón del prototipo verificando los desplazamientos realizados en el entorno SolidWorks, comprobar la adecuada comunicación entre los programas.

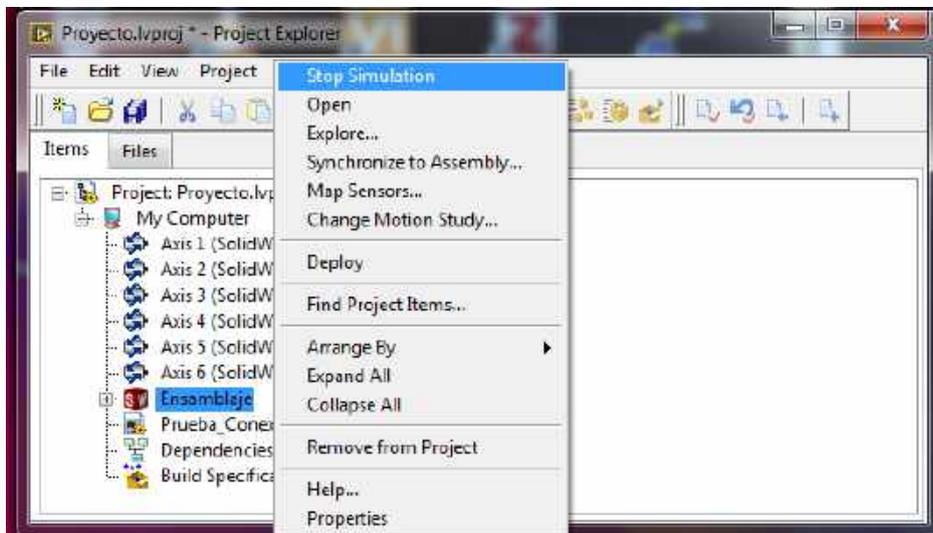


**Figura 57-3.** Verificación de movimientos.

Fuente: Asqui Byron, 2020.

Nota: Si el prototipo no responde, se guarda y cierra todo el proyecto, esperando un cierto lapso para retomar la simulación.

10. *Detener la simulación:* Presionar el botón “Stop” o “Abort Execution” que se hallan en el panel frontal, dar clic derecho sobre “Ensamblaje” y seleccionar “Stop Simulation”.



**Figura 58-3.** Parada de la Simulación.

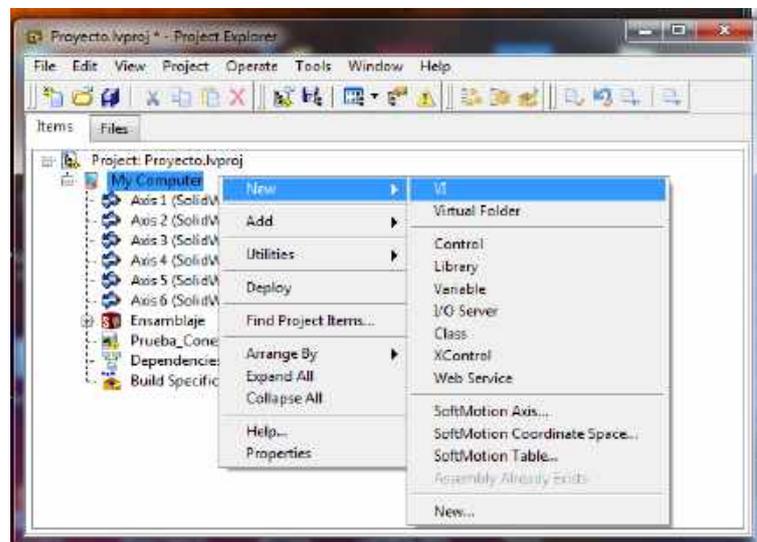
Fuente: Asqui Byron, 2020.

### 3.7.9. Prueba de conexión LabVIEW – Remote

La conexión entre el software Labview y Remote it se dio por:

- Crear de un nuevo VI
- Diseñar una interfaz gráfica
- Adicionar un ciclo repetitivo
- Ingresar y conectar funciones para interactuar con Remote.

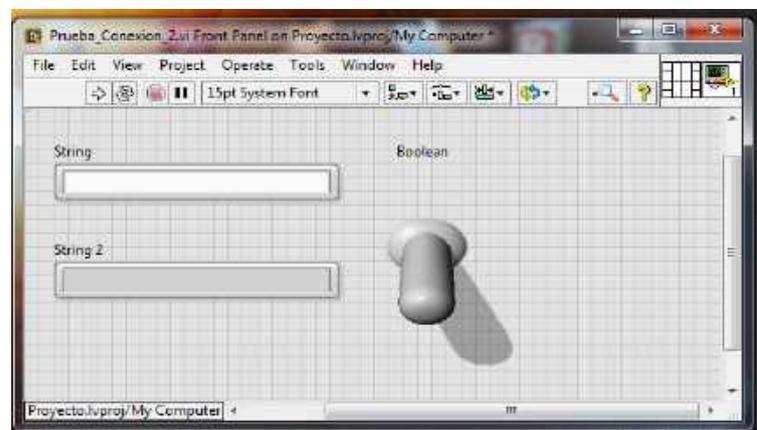
1. *Crear un Nuevo VI:* Dar clic derecho sobre “My Computer”, seleccionar “New”, presionar “VI”, en las dos ventanas se diseña una interfaz y programa acorde la necesidad.



**Figura 59-3.** Crear un VI.

Fuente: Asqui Byron, 2020.

2. *Diseñar una interfaz gráfica:* Agregar controles e indicadores para observar y controlar el flujo de datos presentes en las opciones de “Silver” y “Boolean”, presionar clic derecho en el panel frontal.



**Figura 60-3.** Creacion de la interfaz.

Fuente: Asqui Byron, 2020.

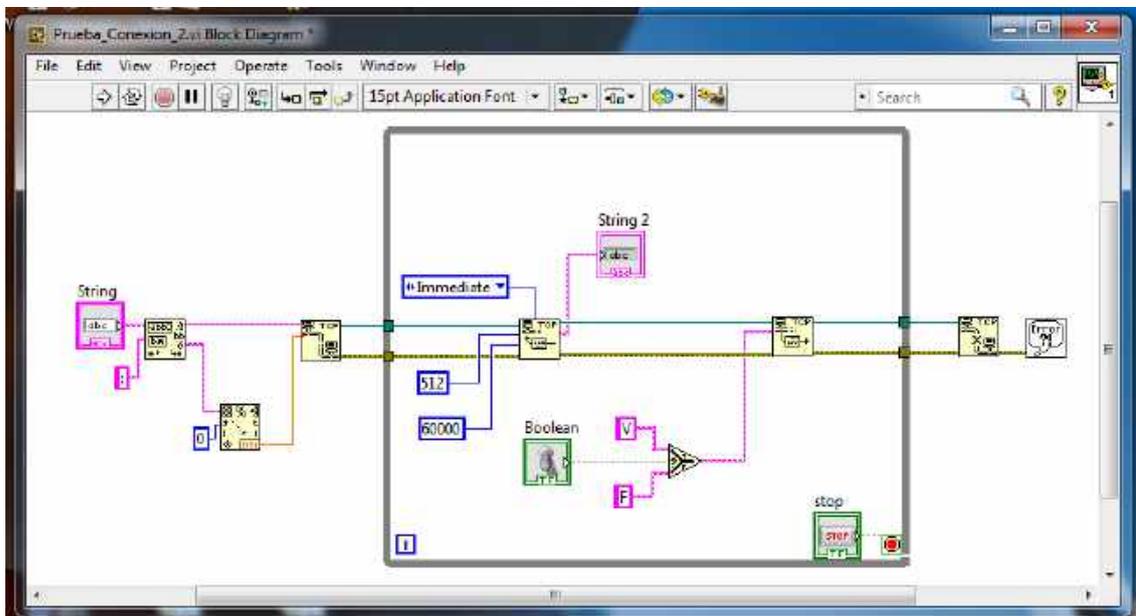
3. *Adicionar un ciclo repetitivo:* Dar clic derecho en el diagrama de bloques, en la nueva ventana seleccionar “Structures”, ingresar el bucle “while loop”, adicionar un control para detener el ciclo repetitivo, pulsar clic derecho sobre el punto rojo y elegir “Create Control”.



**Figura 61-3.** Bucle while loop.  
Fuente: Asqui Byron, 2020.

4. *Ingresar y conectar funciones que permitan interactuar con Remote:* Realizar clic derecho sobre el diagrama de bloques, en las opciones de “Programming” adicionar las siguientes funciones:

- String: “Scan From String” y “Math Pattern”
- Data Communication: “TCP Open Connection”, “TCP Read”, “TCP Write” y “TCP Close Connection”
- Dialog & User Interface: “Simple Error Handler.vi”
- Comparison: “Select”



**Figura 62-3.** Conexiones en el diagrama de bloques.  
Fuente: Asqui Byron, 2020.

En la Figura 62-3 se observa la integración de los elementos en la estructura “While loop” que permanecerá activa mientras no pulsemos el botón “Stop” o surja un error detectado por el componente “Simple Error Handler.vi”. El controlador “String” registra la dirección pública de

Remote, separando en secciones de IP y Puerto mediante las funciones “Math Pattern” y “Scan From String”. La comunicación TCP/IP se genera por el bloque “TCP Open Connection”.

La lectura y escritura de datos se inicia de acuerdo a las funciones “TCP Read”, “TCP Write”. Ahora bien, la información receptada se observa a través de “String 2”, la comunicación correcta se verifica por la plataforma Remote. Mientras que, la función “Select” envía caracteres o valores específicos según el accionar del botón “Boolean”.

5. *Verificar la conexión de Remote con LabVIEW:* Cargar el archivo ejecutable de Python, en la interfaz gráfica ingresar la contraseña programada, pulsar “Conectar”, en la nueva ventana para continuar la conexión presionar el botón “Si”, cambiar las posiciones de cada eslabón mostrados en pantalla para su validación. Figura 63-3.



**Figura 63-3.** Interfaz gráfica de prototipo.

Fuente: Asqui Byron, 2020.

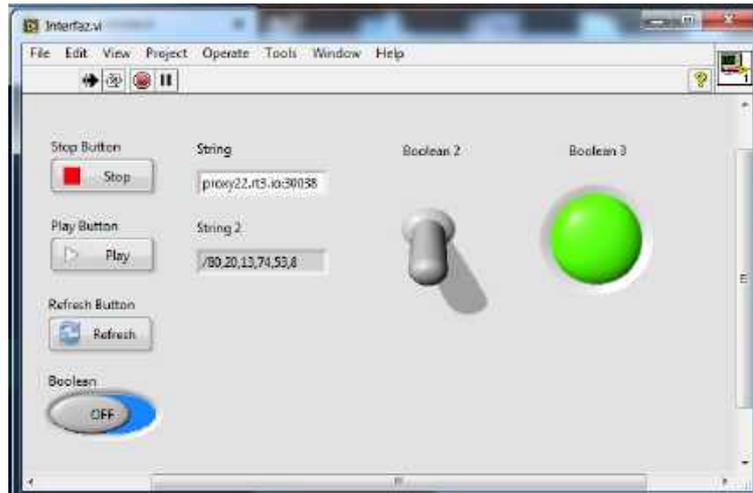
Terminado el anterior proceso dirigirse a la plataforma de Remote, seleccionar el dispositivo y el servicio registrado, luego de la acción anterior obtener la dirección pública de la cuenta para enlazar los dispositivos. Figura 64-3.



**Figura 64-3.** Dirección pública en Remote

Fuente: Asqui Byron, 2020.

En el entorno de LabVIEW digitar la dirección obtenida en el cuadro de texto llamado “String”, pulsar el botón “Run” para realizar la carga de la programación con los datos emitidos por el dispositivo “Raspberry pi” y determinar la buena comunicación con la plataforma Remote.



**Figura 65-3.** Verificación.

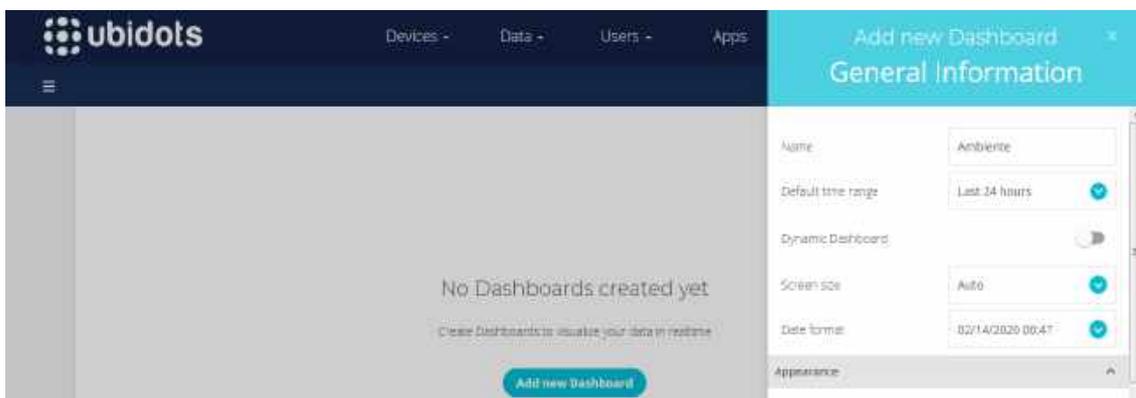
Fuente: Asqui Byron, 2020.

### 3.7.10. Creación de tableros y eventos en Ubidots

La creación de tableros y eventos en Ubidots integró las siguientes acciones:

- Adicionar un nuevo tablero
- Añadir un nuevo widget
- Verificar el nuevo widget
- Agregar un nuevo Evento para configurar las acciones del evento
- Comprobar el evento

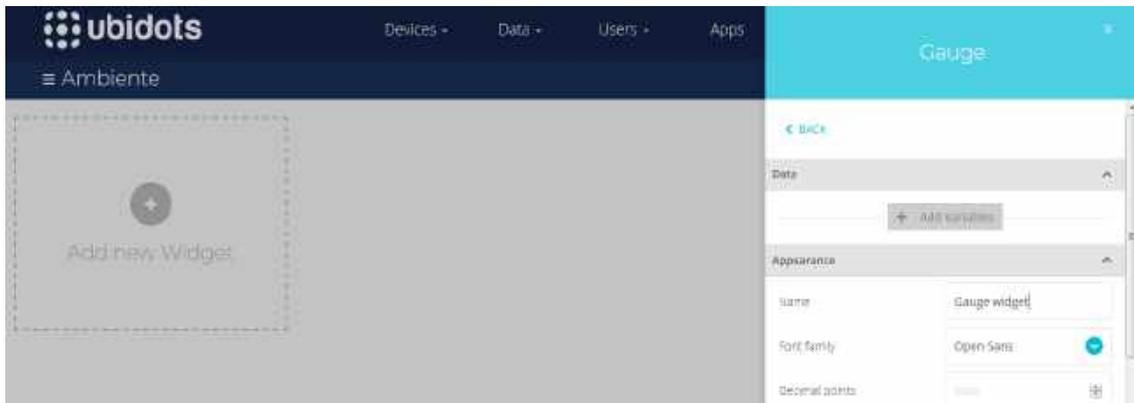
1. *Adicionar un nuevo tablero:* Ingresar a la plataforma “Ubidots”, elegir Data, dar clic en la opción “Dashboards”, en la nueva sección de configuraciones pulsar clic sobre “Add New Dashboards” e ingresar un tablero de visualización.



**Figura 66-3.** Tablero en Ubidots.

Fuente: Asqui Byron, 2020.

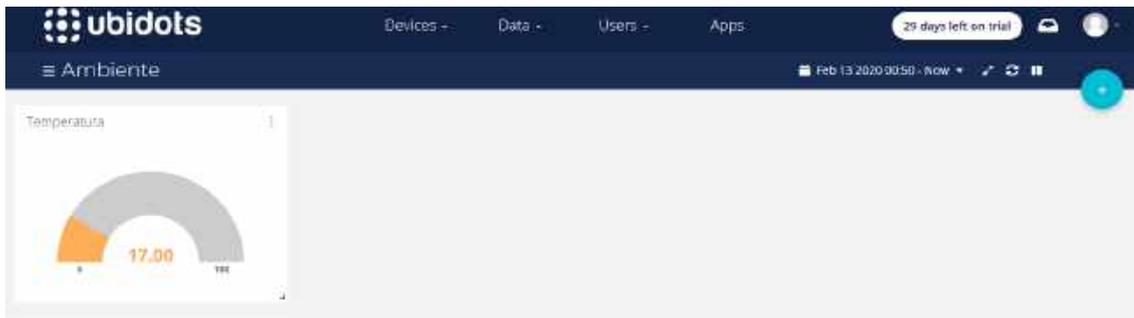
2. *Adicionar un nuevo widget:* Dar clic sobre “Add New Widget”, en la nueva sección de configuraciones seleccionar un tipo de widget, en el nuevo apartado ingresar la variable a monitorear.



**Figura 67-3.** Widget en Ubidots

Fuente: Asqui Byron, 2020.

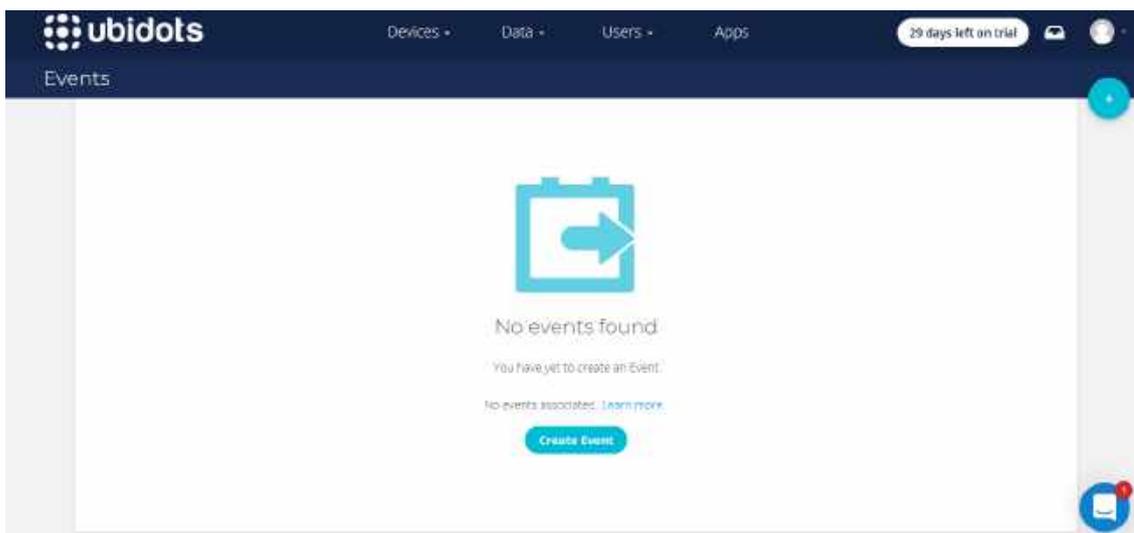
3. *Verificar el nuevo widget:* Observar la variación de datos mostrados en pantalla.



**Figura 68-3.** Verificación de Widget

Fuente: Asqui Byron, 2020.

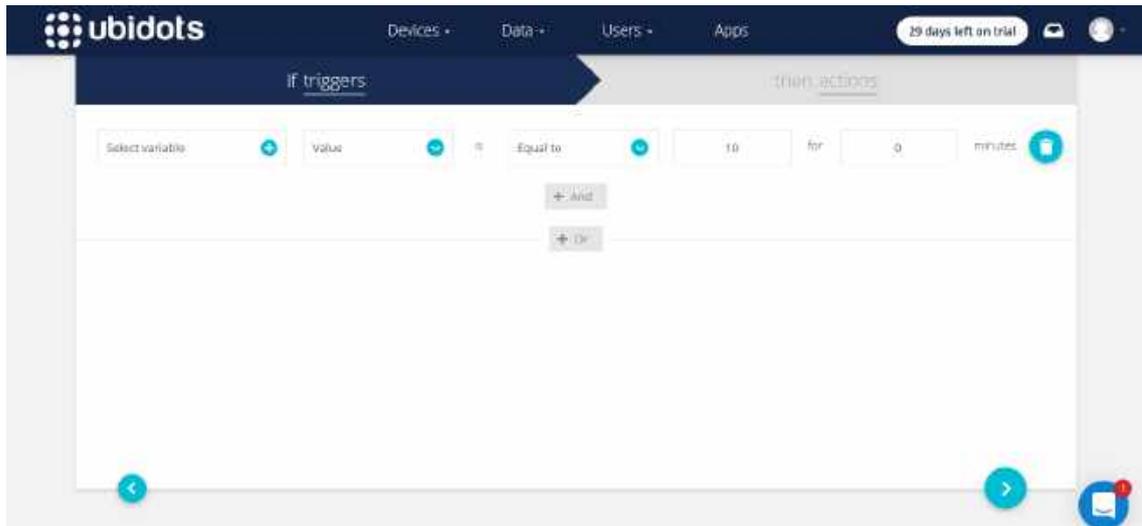
4. *Adicionar un nuevo Evento:* Seleccionar data, presionar clic en la opción “Events”, en la nueva pantalla elegir “Create Event”.



**Figura 69-3.** Eventos en Ubidots

Fuente: Asqui Byron, 2020.

5. *Configurar las acciones del evento:* Establecer la variable y el valor a reaccionar, seleccionar la acción con la frecuencia del reporte.

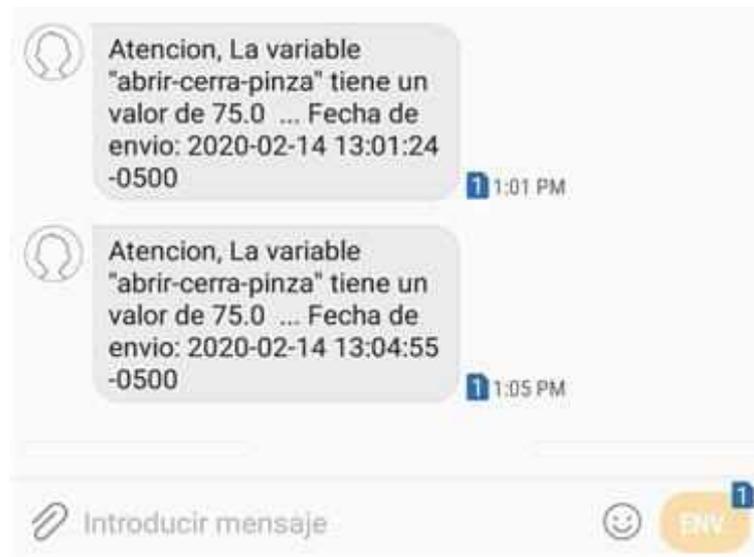


**Figura 70-3.** Configuración de Eventos

Fuente: Asqui Byron, 2020.

Nota: Para el desarrollo de la investigación es importante señalar que la acción del evento fue realizada con un dispositivo móvil debido a la cotidianidad de su uso; sin embargo, existen otros medios electrónicos que acogen la acción, todo depende de la frecuencia de uso.

6. *Verificar el evento:* Efectuar un valor de salto desde el dispositivo que será leído por la plataforma IOT, la acción del evento programado llegará mediante un mensaje de texto al dispositivo móvil enlazado. Figura 71-3.



**Figura 71-3.** Reporte del Evento

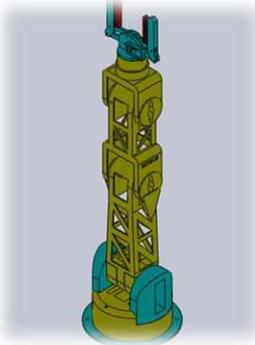
Fuente: Asqui Byron, 2020.

### 3.8. Resultados

#### 3.8.1. Diseño CAD

El rediseño CAD fue realizado bajo el esquema exhibido en el portal Thingiverse. Com, debido a la complejidad en los detalles se adaptó al Software SolidWorks Simulation, precisamente para encajar en el análisis de movimientos. Los resultados encontrados en la remodelación fueron correctos y eficientes, se resalta además que los detalles para la simulación no afectan el sentido funcional del prototipo. Tabla 8-3.

**Tabla 8-3:** Ensamblajes

ENSAMBLAJES	
Modelo Original	Modelo Actual
 <p>Prototipo (Impresión)</p>	 <p>Prototipo (Monitoreo)</p>

Realizado por: Asqui Byron, 2020.

#### 3.8.2. Análisis Estructural

El análisis estructural se manejó por dos acciones a través del Software SolidWorks, la primera fue el modelado, mientras que la segunda se acompañó por el ensamblaje de eslabones y componentes del brazo prototipo. La comprobación del diseño se encargó a la opción Simulación bajo la magnitud física Von Misses generada para considerar si el material aplicado cede o se fractura a una determinada carga, el criterio al fallo se efectuó con una fuerza de 100 N verificando que el material PLA es apto y no ceda fácilmente a la fractura.

La fuerza analizada fue suficiente para soportar todo el peso del sistema incluido los componentes internos del brazo; sin embargo, para mayor confiabilidad se aplicó un factor de seguridad de 1,2, asegurando el valor calculado de la capacidad máxima del sistema y el valor del requerimiento esperado real.

El posterior análisis se aplicó un material de propiedades mecánicas semejantes al PLA, esto debido a la biblioteca existente en el Software.

### 3.8.2.1. Análisis estático – Base

A continuación, se muestran las propiedades del modelo y material utilizadas para el análisis estático, así como las sujeciones o elementos de apoyo y cargas que va soportar la Base.

**Tabla 9-3:** Propiedades del Modelo - Base

Sólidos			Sólidos		
Nombre de documento y referencia	Tratado como	Propiedades volumétricas	Nombre de documento y referencia	Tratado como	Propiedades volumétricas
 Cortar-Extruir4	Sólido	Masa:0.0536953 kg Volumen:5.264 25e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.526214 N	 Cortar-Extruir6	Sólido	Masa:0.10123 kg Volumen:9.92462 e-005 m <sup>3</sup> Densidad:1019.99 kg/m <sup>3</sup> Peso:0.992053 N
 Redondeo1	Sólido	Masa:0.335822 kg Volumen:0.000 329238 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:3.29106 N	 Cortar-Extruir6	Sólido	Masa:0.10123 kg Volumen:9.92462 e-005 m <sup>3</sup> Densidad:1019.99 kg/m <sup>3</sup> Peso:0.992053 N

Realizado por: Asqui Byron, 2020.

**Tabla 10-3:** Propiedades del Material - Base

Referencia de modelo	Propiedades
	Nombre: ABS Tipo de modelo: Isotrópico elástico lineal Criterio de error predeterminado: Desconocido Límite de tracción: 3e+007 N/m <sup>2</sup> Módulo elástico: 2e+009 N/m <sup>2</sup> Coeficiente de Poisson: 0.394 Densidad: 1020 kg/m <sup>3</sup> Módulo cortante: 3.189e+008 N/m <sup>2</sup>

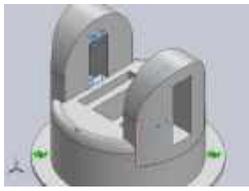
Realizado por: Asqui Byron, 2020.

**Tabla 11-3:** Sujeciones - Base

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción
Fijo-1		Entidades: 9 cara(s) Tipo: Geometría fija

Realizado por: Asqui Byron, 2020.

**Tabla 12-3:** Cargas - Base

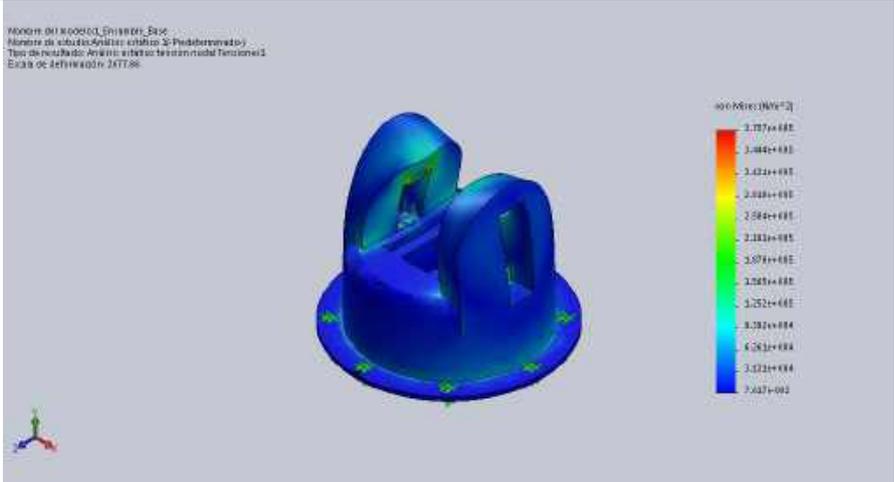
Nombre de carga	Cargar imagen	Detalles de carga
Fuerza-1		Entidades: 8 cara(s) Referencia: Arista < 1 > Tipo: Aplicar fuerza Valores: ---, ---, 100 N

Realizado por: Asqui Byron, 2020.

Una vez ubicados los parámetros de la base en SolidWorks se efectuó el análisis estático, logrando obtener las siguientes tablas como resultado del estudio.

**Tabla 13-3:** Análisis estático tensión nodal - Base

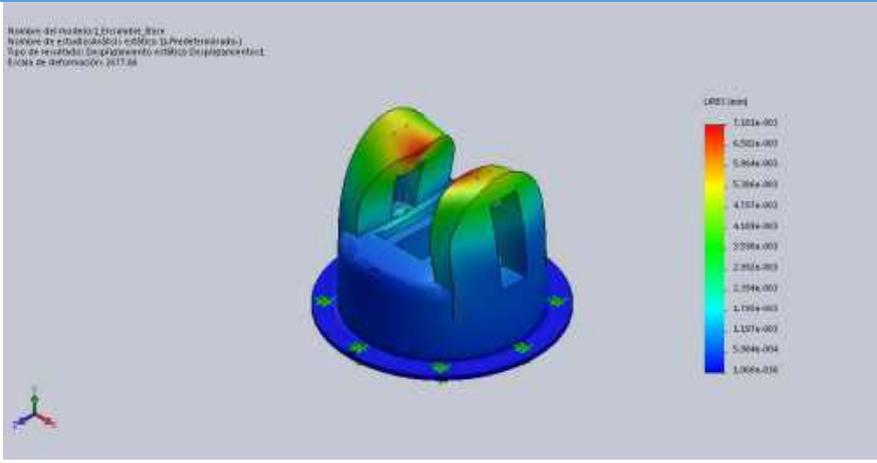
Nombre	Tipo	Mín.	Máx.
Tensiones1	VON: Tensión de von Mises	7.617e-002N/m <sup>2</sup> Nodo: 2861	3.757e+005N/m <sup>2</sup> Nodo: 14828



1\_Ensamble\_Base-Análisis estático 1-Tensiones-Tensiones 1

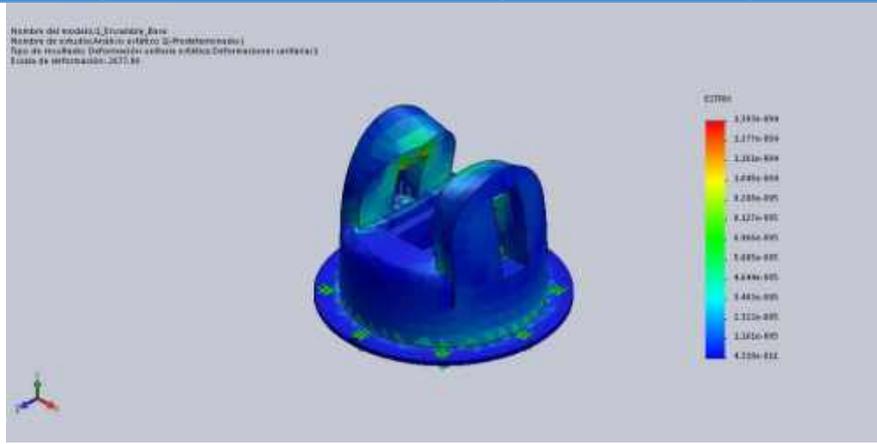
Realizado por: Asqui Byron, 2020.

**Tabla 14-3:** Desplazamiento estático - Base

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0.000e+000mm Nodo: 56	7.181e-003mm Nodo: 12345
 <p>1_Ensamble_Base-Análisis estático 1-Desplazamientos-Desplazamientos1</p>			

Realizado por: Asqui Byron, 2020.

**Tabla 15-3:** Deformaciones unitarias estática - Base

Nombre	Tipo	Mín.	Máx.
Deformaciones unitarias1	ESTRN: Deformación unitaria equivalente	4.319e-011 Elemento: 440	1.393e-004 Elemento: 10488
 <p>1_Ensamble_Base-Análisis estático 1-Deformaciones unitarias-Deformaciones unitarias1</p>			

Realizado por: Asqui Byron, 2020.

Mediante la utilización de un material que posee un límite a la tracción de 30MPa en el análisis estático, demuestra que la base adquiere una deformación de 0,00013 lo cual es imperceptible al aplicar una fuerza de 100 N, por ende, se acepta para la construcción de los elementos.

### 3.8.2.2. Análisis estático – Brazo

A continuación, se muestran las propiedades del modelo y material utilizadas para el análisis estático, así como las sujeciones o elementos de apoyo y cargas que va soportar la Brazo.

**Tabla 16-3:** Propiedades del Modelo - Brazo

 <p>Nombre del modelo: 2_Ensamble_Brazo Configuración actual: Predeterminado</p>		
Sólidos		
Nombre de documento y referencia	Tratado como	Propiedades volumétricas
Cortar-Extruir12 	Sólido	Masa:0.0806118 kg Volumen:7.90312e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.789996 N
Cortar-Extruir3 	Sólido	Masa:0.0266018 kg Volumen:2.60802e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.260698 N
Cortar-Extruir3 	Sólido	Masa:0.0266018 kg Volumen:2.60802e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.260698 N

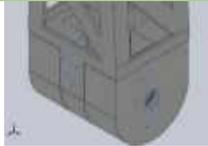
Realizado por: Asqui Byron, 2020.

**Tabla 17-3:** Propiedades del Material - Brazo

Referencia de modelo	Propiedades
	Nombre: ABS Tipo de modelo: Isotrópico elástico lineal Criterio de error predeterminado: Desconocido Límite de tracción: 3e+007 N/m <sup>2</sup> Módulo elástico: 2e+009 N/m <sup>2</sup> Coeficiente de Poisson: 0.394 Densidad: 1020 kg/m <sup>3</sup> Módulo cortante: 3.189e+008 N/m <sup>2</sup>

Realizado por: Asqui Byron, 2020.

**Tabla 18-3:** Sujeciones - Brazo

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción
Fijo-1		Entidades: 16 cara(s) Tipo: Geometría fija

Realizado por: Asqui Byron, 2020.

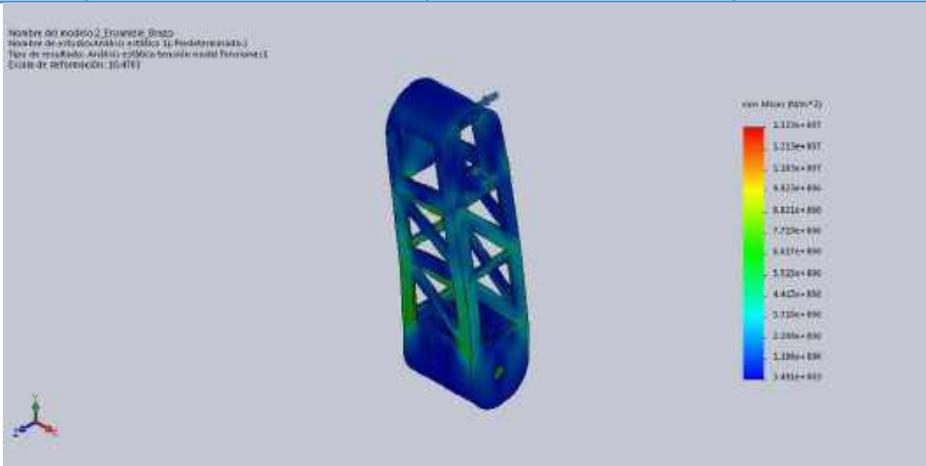
**Tabla 19-3:** Cargas - Brazo

Nombre de carga	Cargar imagen	Detalles de carga
Fuerza-1		Entidades: 6 cara(s) Referencia: Arista < 1 > Tipo: Aplicar fuerza Valores: ---, ---, 100 N

Realizado por: Asqui Byron, 2020.

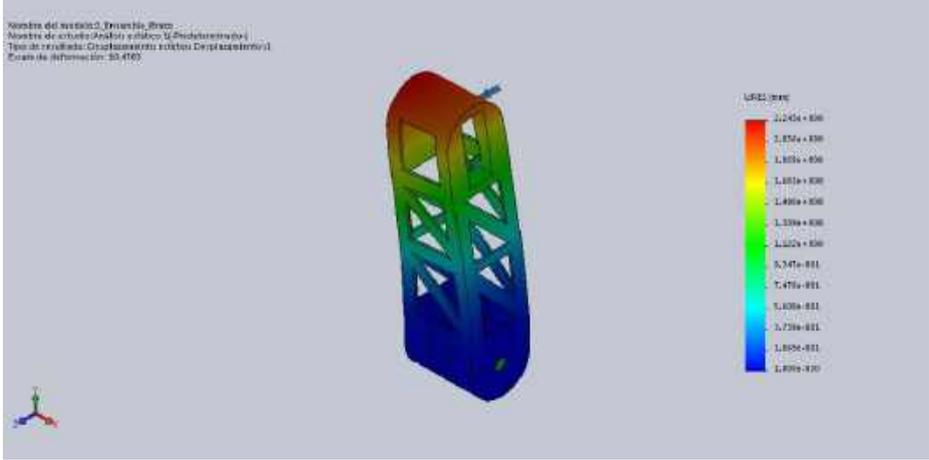
Una vez ubicados los parámetros del brazo en SolidWorks se efectuó el análisis estático, logrando obtener las siguientes tablas como resultado del estudio.

**Tabla 20-3:** Análisis estático tensión nodal - Brazo

Nombre	Tipo	Mín.	Máx.
Tensiones1	VON: Tensión de von Mises	3.491e+003N/m <sup>2</sup> Nodo: 12873	1.323e+007N/m <sup>2</sup> Nodo: 6361
 <p>2_Ensamble_Brazo-Análisis estático 1-Tensiones-Tensiones1</p>			

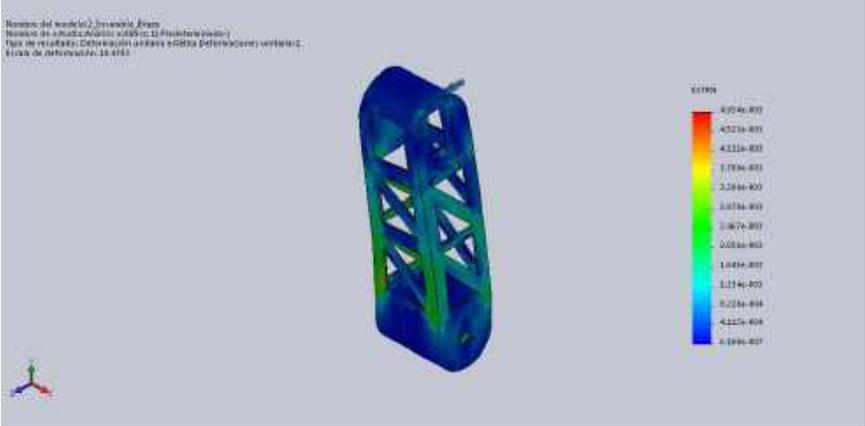
Realizado por: Asqui Byron, 2020.

**Tabla 21-3:** Desplazamiento estático - Brazo

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0.000e+000mm Nodo: 56	2.243e+000mm Nodo: 322
 <p>2_Ensamble_Brazo-Análisis estático 1-Desplazamientos-Desplazamientos1</p>			

Realizado por: Asqui Byron, 2020.

**Tabla 22-3:** Deformaciones unitarias estática - Brazo

Nombre	Tipo	Mín.	Máx.
Deformaciones unitarias1	ESTRN: Deformación unitaria equivalente	6.168e-007 Elemento: 488	4.934e-003 Elemento: 5333
 <p>2_Ensamble_Brazo-Análisis estático 1-Deformaciones unitarias-Deformaciones unitarias1</p>			

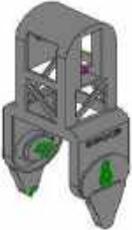
Realizado por: Asqui Byron, 2020.

Al utilizar un material que posee un límite a la tracción de 30MPa para el análisis estático, se demuestra que el brazo adquiere una deformación de 0,0049 lo cual es imperceptible al aplicar una fuerza de 100 N, por ende, se acepta para la construcción de los elementos.

### 3.8.2.3. Análisis estático – Antebrazo

A continuación, se muestran las propiedades del modelo y material utilizadas para el análisis estático, así como las sujeciones o elementos de apoyo y cargas que va soportar la Antebrazo.

**Tabla 23-3:** Propiedades del Modelo - Antebrazo

 Nombre del modelo: 3_Ensamble_Antebrazo Configuración actual: Predeterminado					
Sólidos					
Nombre de documento y referencia		Tratado como	Propiedades volumétricas		
Redondeo1 		Sólido	Masa:0.0587425 kg Volumen:5.75907e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.575676 N		
Sólidos			Sólidos		
Nombre de documento y referencia	Tratado como	Propiedades volumétricas	Nombre de documento y referencia	Tratado como	Propiedades volumétricas
Cortar-Extruir2 	Sólido	Masa:0.00497033 kg Volumen:4.87288 e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0487093 N	Cortar-Extruir1 	Sólido	Masa:0.00952569 kg Volumen:9.33891 e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0933518 N
Cortar-Extruir2 	Sólido	Masa:0.00497033 kg Volumen:4.87288 e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0487093 N	Cortar-Extruir3 	Sólido	Masa:0.00812616 kg Volumen:7.96683 e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0796364 N

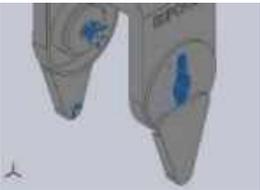
Realizado por: Asqui Byron, 2020.

**Tabla 24-3:** Propiedades del Material - Antebrazo

Referencia de modelo	Propiedades
	Nombre: ABS Tipo de modelo: Isotrópico elástico lineal Criterio de error predeterminado: Desconocido Límite de tracción: $3e+007 \text{ N/m}^2$ Módulo elástico: $2e+009 \text{ N/m}^2$ Coeficiente de Poisson: 0.394 Densidad: $1020 \text{ kg/m}^3$ Módulo cortante: $3.189e+008 \text{ N/m}^2$

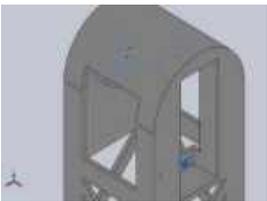
Realizado por: Asqui Byron, 2020.

**Tabla 25-3:** Sujeciones - Antebrazo

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción
Fijo-1		Entidades: 11 cara(s) Tipo: Geometría fija

Realizado por: Asqui Byron, 2020.

**Tabla 26-3:** Cargas - Antebrazo

Nombre de carga	Cargar imagen	Detalles de carga
Fuerza-1		Entidades: 6 cara(s) Referencia: Arista < 1 > Tipo: Aplicar fuerza Valores: ---, ---, 100 N

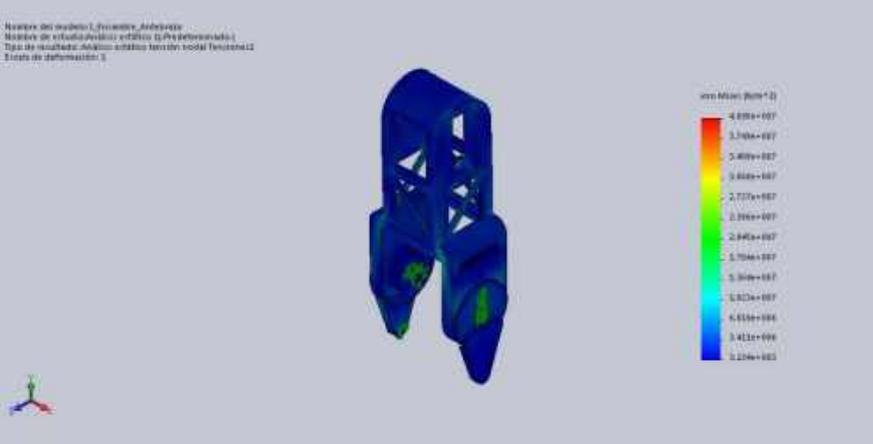
Realizado por: Asqui Byron, 2020.

Una vez ubicados los parámetros del antebrazo en SolidWorks se efectuó el análisis estático, logrando obtener las siguientes tablas como resultado del estudio.

**Tabla 27-3:** Análisis estático tensión nodal - Antebrazo

Nombre	Tipo	Mín.	Máx.
Tensiones1	VON: Tensión de von Mises	3.134e+003N/m <sup>2</sup> Nodo: 56297	4.090e+007N/m <sup>2</sup> Nodo: 54481

Nombre del problema: 3\_Ensamble\_Antebrazo  
 Nombre de estudio: Análisis estático 1 (Predefinido)  
 Tipo de resultado: Análisis estático tensión nodal Tensiones1  
 Estado de deformación: 1



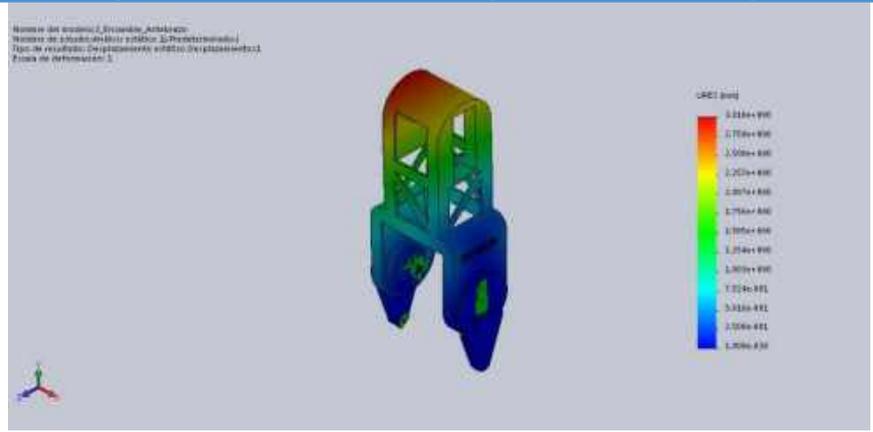
3\_Ensamble\_Antebrazo-Análisis estático 1-Tensiones-Tensiones1

Realizado por: Asqui Byron, 2020.

**Tabla 28-3:** Desplazamiento estático - Antebrazo

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0.000e+000mm Nodo: 51285	3.010e+000mm Nodo: 1166

Nombre del problema: 3\_Ensamble\_Antebrazo  
 Nombre de estudio: Análisis estático 1 (Predefinido)  
 Tipo de resultado: Desplazamiento estático (Desplazamientos1)  
 Estado de deformación: 1



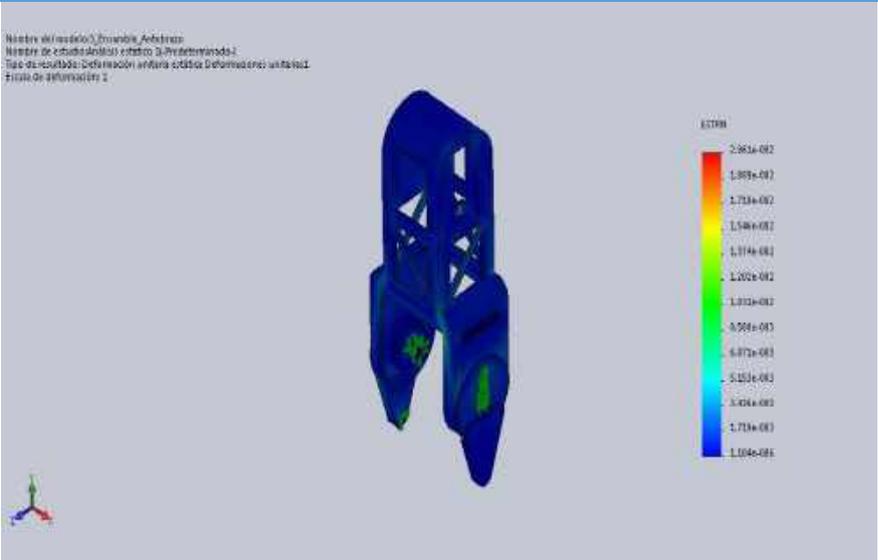
3\_Ensamble\_Antebrazo-Análisis estático 1-Desplazamientos-Desplazamientos1

Realizado por: Asqui Byron, 2020.

**Tabla 29-3:** Deformaciones unitarias estática - Antebrazo

Nombre	Tipo	Mín.	Máx.
Deformaciones unitarias1	ESTRN: Deformación unitaria equivalente	1.104e-006 Elemento: 28173	2.061e-002 Elemento: 26778

Nombre del modelo: 3\_Ensamble\_Antebrazo  
 Nombre de estudio/análisis: estático 3D-Predefinido-1  
 Tipo de resultado: Deformación unitaria estática Deformaciones unitarias1  
 Estado de deformación: 1



3\_Ensamble\_Antebrazo-Análisis estático 1-Deformaciones unitarias-Deformaciones unitarias1

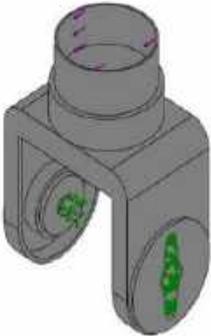
Realizado por: Asqui Byron, 2020.

Al aplicar un material que posee un límite a la tracción de 30MPa para el análisis estático, se demuestra que el antebrazo adquiere una deformación de 0,020 lo cual es imperceptible al emplear una fuerza de 100 N, por ende, se acepta para la construcción de los elementos.

#### 3.8.2.4. Análisis estático – Muñeca

A continuación, se muestran las propiedades del modelo y material utilizadas para el análisis estático, así como las sujeciones o elementos de apoyo y cargas que va soportar la Muñeca.

**Tabla 30-3:** Propiedades del Modelo - Muñeca

		
Nombre del modelo: 4_Ensamble_Muñeca Configuración actual: Predeterminado		
Sólidos		
Nombre de documento y referencia	Tratado como	Propiedades volumétricas
Redondeo1 	Sólido	Masa:0.0406584 kg Volumen:3.98612e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.398452 N
Cortar-Extruir1 	Sólido	Masa:0.00952569 kg Volumen:9.33891e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0933518 N
Cortar-Extruir3 	Sólido	Masa:0.00812616 kg Volumen:7.96683e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0796364 N

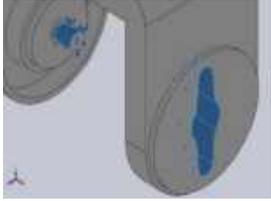
Realizado por: Asqui Byron, 2020.

**Tabla 31-3:** Propiedades del Material - Muñeca

Referencia de modelo	Propiedades
	Nombre: ABS Tipo de modelo: Isotrópico elástico lineal Criterio de error predeterminado: Desconocido Límite de tracción: 3e+007 N/m <sup>2</sup> Módulo elástico: 2e+009 N/m <sup>2</sup> Coeficiente de Poisson: 0.394 Densidad: 1020 kg/m <sup>3</sup> Módulo cortante: 3.189e+008 N/m <sup>2</sup>

Realizado por: Asqui Byron, 2020.

**Tabla 32-3:** Sujeciones – Muñeca

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción
Fijo-1		Entidades: 9 cara(s) Tipo: Geometría fija

Realizado por: Asqui Byron, 2020.

**Tabla 33-3:** Cargas – Muñeca

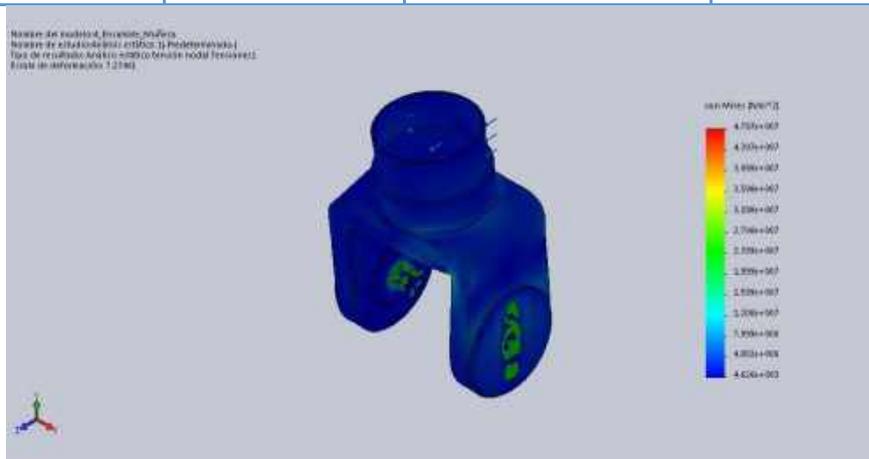
Nombre de carga	Cargar imagen	Detalles de carga
Fuerza-1		Entidades: 1 cara(s) Referencia: Arista < 1 > Tipo: Aplicar fuerza Valores: ---, ---, -100 N

Realizado por: Asqui Byron, 2020.

Una vez ubicados los parámetros de la muñeca en SolidWorks se efectuó el análisis estático, logrando obtener las siguientes tablas como resultado del estudio.

**Tabla 34-3:** Análisis estático tensión nodal - Muñeca

Nombre	Tipo	Mín.	Máx.
Tensiones1	VON: Tensión de von Mises	4.626e+003N/m <sup>2</sup> Nodo: 9331	4.797e+007N/m <sup>2</sup> Nodo: 14214



4\_Ensamble\_Muñeca-Análisis estático 1-Tensiones-Tensiones1

Realizado por: Asqui Byron, 2020.

**Tabla 35-3:** Desplazamiento estático - Muñeca

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0.000e+000mm Nodo: 12320	1.584e+000mm Nodo: 10309
<p>4_Ensamble_Muñeca-Análisis estático 1-Desplazamientos-Desplazamientos1</p>			

Realizado por: Asqui Byron, 2020.

**Tabla 36-3:** Deformaciones unitarias estática - Muñeca

Nombre	Tipo	Mín.	Máx.
Deformaciones unitarias1	ESTRN: Deformación unitaria equivalente	4.665e-006 Elemento: 8108	1.991e-002 Elemento: 7263
<p>4_Ensamble_Muñeca-Análisis estático 1-Deformaciones unitarias-Deformaciones unitarias1</p>			

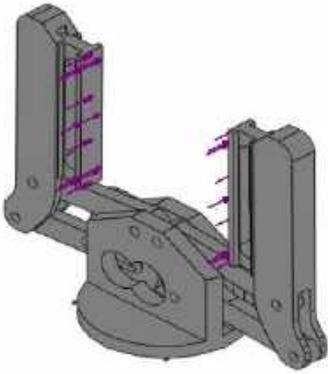
Realizado por: Asqui Byron, 2020.

El material utilizado para el análisis estático del prototipo posee un límite a la tracción de 30MPa, demostrando que la muñeca adquiere una deformación de 0,019 lo cual es imperceptible al aplicar una fuerza de 100 N, por ende, se acepta para la construcción de los elementos.

### 3.8.2.5. Análisis estático – Pinza

A continuación, se muestran las propiedades del modelo y material utilizadas para el análisis estático, así como las sujeciones o elementos de apoyo y cargas que va soportar la Pinza.

**Tabla 37-3:** Propiedades del Modelo - Pinza

 <p>Nombre del modelo: 5_Ensamble_Pinza Configuración actual: Predeterminado</p>					
<b>Sólidos</b>					
Nombre de documento y referencia		Tratado como	Propiedades volumétricas		
Cortar-Extruir6 		Sólido	Masa:0.0157893 kg Volumen:1.54797e-005 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.154735 N		
<b>Sólidos</b>			<b>Sólidos</b>		
Nombre de documento y referencia	Tratado como	Propiedades volumétricas	Nombre de documento y referencia	Tratado como	Propiedades volumétricas
Cortar-Extruir1 	Sólido	Masa:0.00185952 kg Volumen:1.82306e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0182233 N	Cortar-Extruir2 	Sólido	Masa:0.00683697 kg Volumen:6.70291e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0670023 N
Saliente-Extruir1 	Sólido	Masa:0.00202185 kg Volumen:1.9822e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0198141 N	Cortar-Extruir2 	Sólido	Masa:0.00683697 kg Volumen:6.70291e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0670023 N

 Saliente-Extruir1	Sólido	Masa:0.00133446 kg Volumen:1.30829 e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0130777 N	 Cortar-Extruir1	Sólido	Masa:0.00167168 kg Volumen:1.6389e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0163825 N
 Saliente-Extruir1	Sólido	Masa:0.00133446 kg Volumen:1.30829 e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0130777 N	 Cortar-Extruir1	Sólido	Masa:0.00167168 kg Volumen:1.6389e-006 m <sup>3</sup> Densidad:1020 kg/m <sup>3</sup> Peso:0.0163825 N

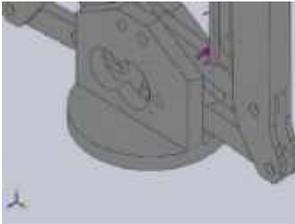
Realizado por: Asqui Byron, 2020.

**Tabla 38-3:** Propiedades del Material - Pinza

Referencia de modelo	Propiedades
	Nombre: ABS Tipo de modelo: Isotrópico elástico lineal Criterio de error predeterminado: Desconocido Límite de tracción: 3e+007 N/m <sup>2</sup> Módulo elástico: 2e+009 N/m <sup>2</sup> Coeficiente de Poisson: 0.394 Densidad: 1020 kg/m <sup>3</sup> Módulo cortante: 3.189e+008 N/m <sup>2</sup>

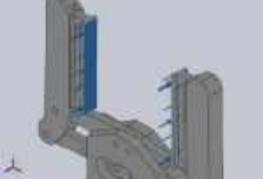
Realizado por: Asqui Byron, 2020.

**Tabla 39-3:** Sujeciones - Pinza

Nombre de sujeción	Imagen de sujeción	Detalles de sujeción
Fijo-1		Entidades: 1 cara(s) Tipo: Geometría fija

Realizado por: Asqui Byron, 2020.

**Tabla 40-3:** Cargas - Pinza

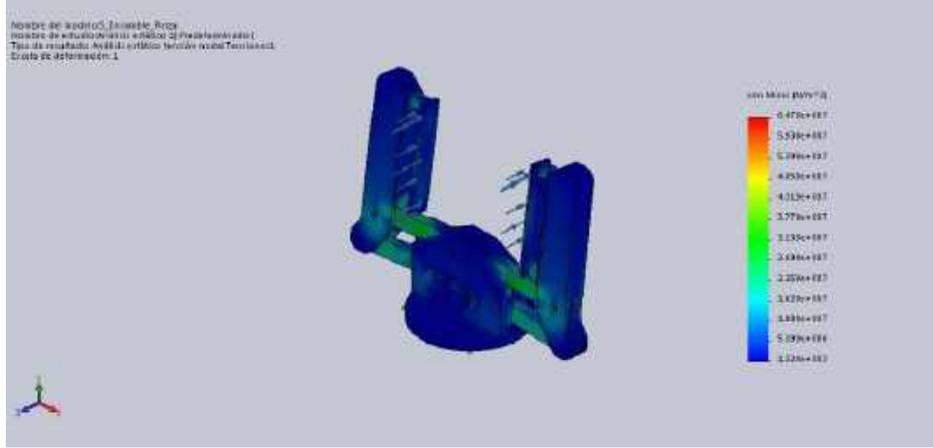
Nombre de carga	Cargar imagen	Detalles de carga
Fuerza-1		Entidades: 2 cara(s) Referencia: Arista< 1 > Tipo: Aplicar fuerza Valores: ---, ---, -100 N

Realizado por: Asqui Byron, 2020.

Una vez ubicados los parámetros de la pinza en SolidWorks se efectuó el análisis estático, logrando obtener las siguientes tablas como resultado del estudio.

**Tabla 41-3:** Análisis estático tensión nodal - Pinza

Nombre	Tipo	Mín.	Máx.
Tensiones1	VON: Tensión de von Mises	1.320e+003N/m <sup>2</sup> Nodo: 7989	6.478e+007N/m <sup>2</sup> Nodo: 1337

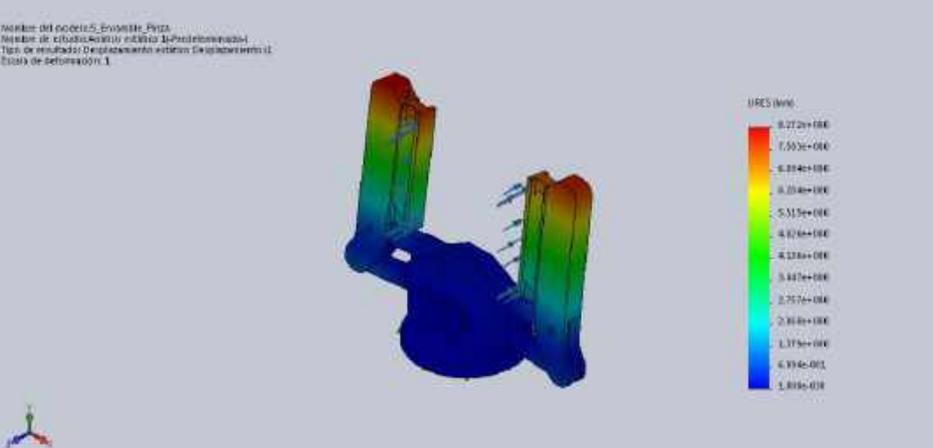


5\_Ensamble\_Pinza-Análisis estático 1-Tensiones-Tensiones1

Realizado por: Asqui Byron, 2020.

**Tabla 42-3:** Desplazamiento estático - Pinza

Nombre	Tipo	Mín.	Máx.
Desplazamientos1	URES: Desplazamientos resultantes	0.000e+000mm Nodo: 3076	8.272e+000mm Nodo: 12987

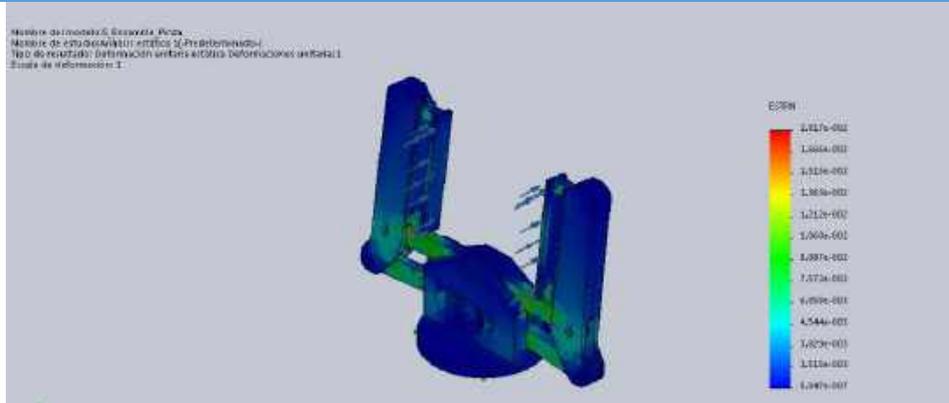


5\_Ensamble\_Pinza-Análisis estático 1-Desplazamientos-Desplazamientos1

Realizado por: Asqui Byron, 2020.

**Tabla 43-3:** Deformaciones unitarias estática - Pinza

Nombre	Tipo	Mín.	Máx.
Deformaciones unitarias1	ESTRN: Deformación unitaria equivalente	5.047e-007 Elemento: 5830	1.817e-002 Elemento: 370



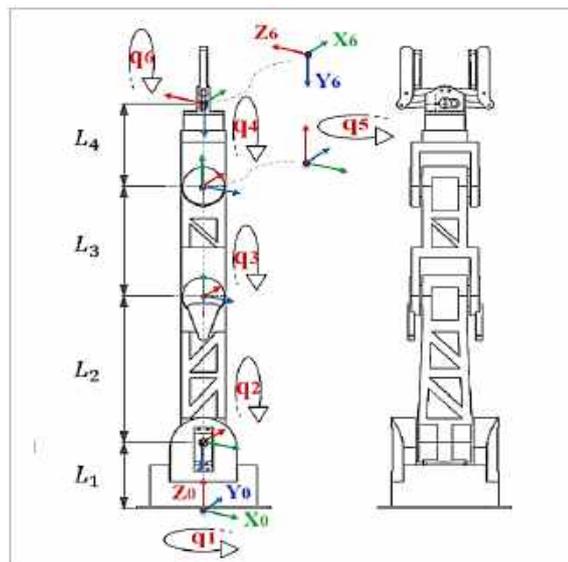
5\_Ensamble\_Pinza-Análisis estático 1-Deformaciones unitarias-Deformaciones unitarias1

Realizado por: Asqui Byron, 2020.

En conclusión, el material utilizado para el análisis estático del prototipo que posee un límite a la tracción de 30MPa, demuestra que la pinza adquiere una deformación de 0,018 lo cual es imperceptible al aplicar una fuerza de 100 N, por ende, se acepta para la construcción de los elementos.

### 3.8.3. Análisis Cinemático

El análisis cinemático se determinó por la posición del extremo del robot con respecto al sistema de referencia fija en la base, de acuerdo al algoritmo Denavit-Hartenberg (D-H). Figura 72-3.



**Figura 72-3.** Diagrama sistemas de referencia del brazo

Realizado por: Asqui Byron, 2020.

La Tabla 44-3 señala los parámetros correspondientes a la geometría y relación de las articulaciones en el prototipo que se obtiene por el algoritmo Denavit-Hartenberg.

**Tabla 44-3:** Parámetros D-H

	<b>Teta</b>	<b>D</b>	<b>a</b>	<b>Alfa</b>
1	$q_1$	$L_1$	0	-90
2	$q_2 - 90$	0	$L_2$	0
3	$q_3$	0	$L_3$	0
4	$q_4 + 90$	0	0	+90
5	$q_5 + 90$	$L_4$	0	-90
6	$q_6$	0	0	0

**Realizado por:** Asqui Byron, 2020.

Las matrices de transformación permiten llegar a la matriz "T" entre la base y el extremo del robot, como se muestra a continuación:

$$A_1 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \cos(q_2 - 90) & -\sin(q_2 - 90) & 0 & L_2 * \cos(q_2 - 90) \\ \sin(q_2 - 90) & \cos(q_2 - 90) & 0 & L_2 * \sin(q_2 - 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & L_3 * \cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & L_3 * \sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} \cos(q_4 + 90) & 0 & \sin(q_4 + 90) & 0 \\ \sin(q_4 + 90) & 0 & -\cos(q_4 + 90) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} \cos(q_5 + 90) & 0 & -\sin(q_5 + 90) & 0 \\ \sin(q_5 + 90) & 0 & \cos(q_5 + 90) & 0 \\ 0 & -1 & 0 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La comprobación de la matriz de transformación total que relaciona la base del brazo con su herramienta es la multiplicación de todas las matrices.

$$T = A_1 * A_2 * A_3 * A_4 * A_5 * A_6$$

El proceso de validación de resultados se efectuó en el Software MATLAB, formando la matriz resultante por una submatriz dedicada a la orientación (m, n, o) y un vector dedicado al posicionamiento (p) del ultimo grado de libertad.

$$T = \begin{bmatrix} m_x & n_x & o_x & p_x \\ m_y & n_y & o_y & p_y \\ m_z & n_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & L1 + L2 + L3 + L4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 3.8.4. Implementación del Brazo

La implementación del prototipo tomó en cuenta varios factores de diseño, funcionalidad y adaptabilidad de los diferentes dispositivos respecto a las características específicas de cada uno de ellos.

El análisis del torque fue necesario para adaptar los servomotores bajo especificaciones técnicas en función a las necesidades requeridas en el ensamblaje del brazo robótico. La Tabla 45-3 indica las características de configuración para el servo R.

**Tabla 45-3:** Especificaciones técnicas del servo MG996R

Características:	Descripción
Peso:	55g
Dimensiones:	40.7x19.7.x42,9mm
Torque:	9.4 kgf*cm (4.8 V), 11 kgf·cm (6 V)
Velocidad de operación:	0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
Voltaje de operación:	4.8 V a 7.2 V
Rango de temperatura:	0 °C –55 °C
Corriente:	500 mA- 900 mA (6V)
Número de servomotores:	5



Realizado por: Asqui Byron, 2020.

La descripción de la Tabla 45-3, hace alusión a las características de los servomotores que controlan la posición del eje en un momento dado para desplazarse ciertos grados y luego posicionarse fijo. Los parámetros de selección se enfocaron en fallos al sistema físico, por ende, se tomó en cuenta el peso del brazo robótico y los grados de libertad.

La Tabla 46-3 presenta las características de funcionamiento en los micro servos que requieren un espacio pequeño pero potente, ideales para pequeños mecanismos que exigen alta precisión. El ensamblaje del brazo robótico prototipo responde la precisión mencionada, por lo cual se empleó un microservo dentro de los eslabones del brazo para facilitar versatilidad al momento de ejecutar el ensamblaje de todo el conjunto de eslabones.

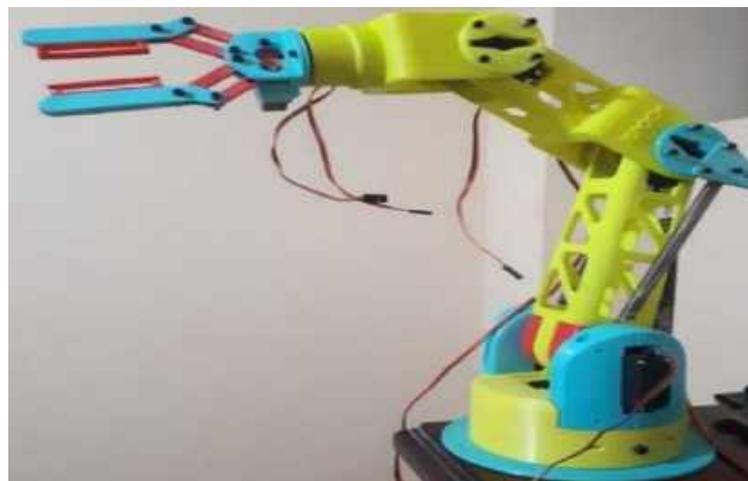
**Tabla 46-3:** Especificaciones técnicas del Microservo MG90S

Características:	Descripción
Peso:	13.4 g
Dimensiones:	22.5 x 12 x 35.5mm
Torque:	1.8 kgf·cm (4.8V) 2.2 kgf·cm (6 V)
Velocidad de operación:	0.1 s/60 degree (4.8 V) 0.08 s/60 degree (6 V)
Voltaje de operación:	4.8 V - 6.0 V
Rango de temperatura:	0 °C -55 °C
Número de servomotores:	2



Realizado por: Asqui Byron, 2020.

La Figura 73-3 es la impresión final del prototipo (brazo robótico) realizada bajo diversos criterios importantes como la ductilidad del material que tuvo distintas pruebas de impresión, de igual manera se integró factores en tiempo de impresión, velocidad de impresión y costo de impresión.



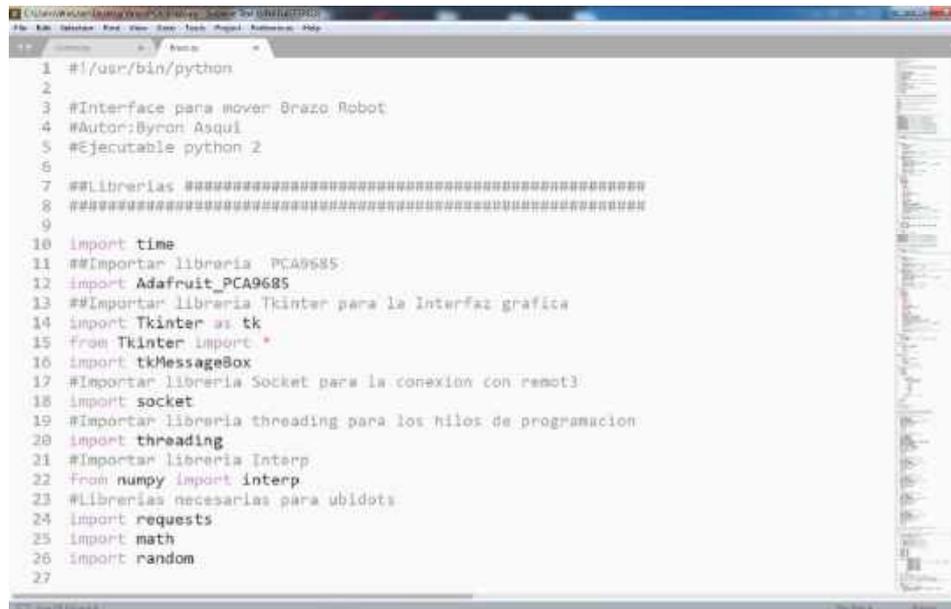
**Figura 73-3.** Brazo Robótico con 6 grados de libertad

Fuente: Asqui Byron, 2020.

Prototipo final de impresión 3D para un brazo robótico con 6 grados de libertad y ensamblaje de todos los componentes físicos bajo programación, integración y comunicación de programas a través de la tecnología IOT.

### 3.8.5. Programación Python

La programación en el lenguaje Python incluyó varias bibliotecas para el desarrollo del código, por mencionar las más importantes fueron: time, tkinter, tkMessageBox, socket, threading, requests, math, random, con funciones específicas en la programación del prototipo.



```
1 #!/usr/bin/python
2
3 #Interface para mover Brazo Robot
4 #Autor:Byron Asqui
5 #Ejecutable python 2
6
7 ##Librerias #####
8 #####
9
10 import time
11 ##Importar libreria PCA9685
12 import Adafruit_PCA9685
13 ##Importar libreria Tkinter para la Interfaz grafica
14 import Tkinter as tk
15 from Tkinter import *
16 import tkMessageBox
17 #Importar libreria Socket para la conexion con remot3
18 import socket
19 #Importar libreria threading para los hilos de programacion
20 import threading
21 #Importar libreria Interp
22 from numpy import interp
23 #Librerias necesarias para ubidots
24 import requests
25 import math
26 import random
27
```

Figura 74-3. Programación Python.

Fuente: Asqui Byron, 2020.

Nota: El código completo se muestra en los anexos.

### 3.8.6. Programación LabVIEW

La programación en el Software LabVIEW se realizó de acuerdo a:

1. *Crear un Nuevo VI*: Dar clic derecho sobre “My Computer”, seleccionar “New”, presionar sobre “VI”, en las dos ventanas diseñar una interfaz y programar. Figura 75-3.

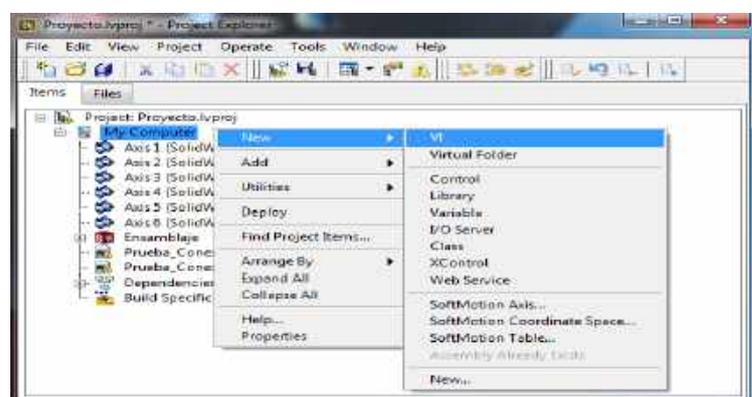
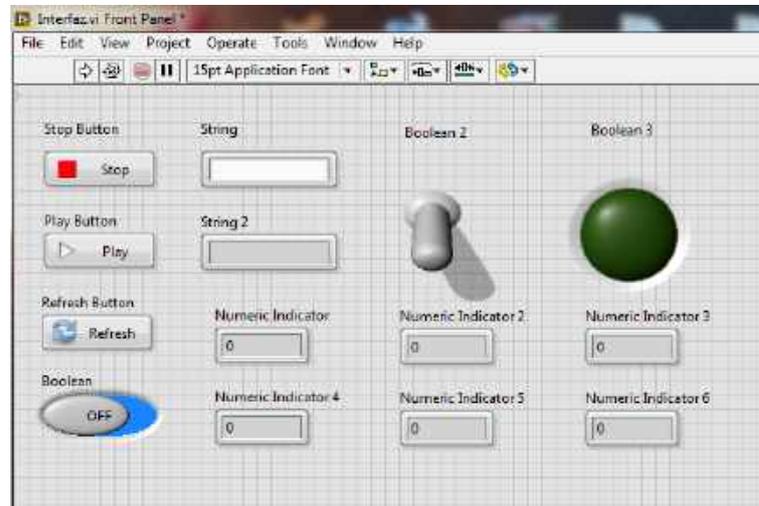


Figura 75-3. Primera sección Flat Sequence Structure.

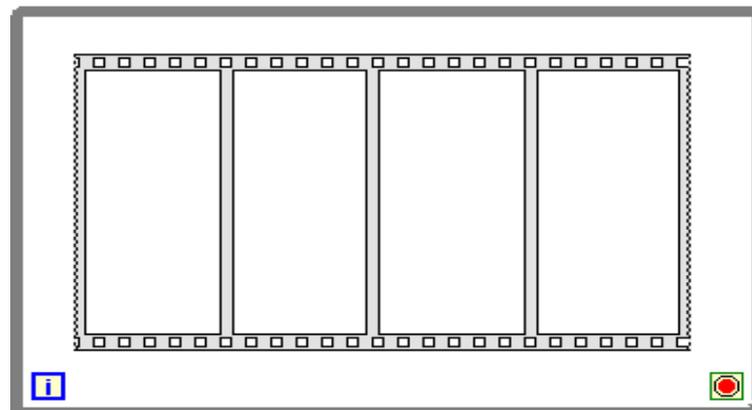
Fuente: Asqui Byron, 2020.

2. *Diseñar la interfaz gráfica:* Agregar controles e indicadores para el flujo de datos, que se encuentran en las opciones “silver” y “boolean” con un clic derecho sobre el panel frontal, además configurar la acción mecánica de los botones (stop, play y reset) de “Latch When Released” a “Switch until released”.



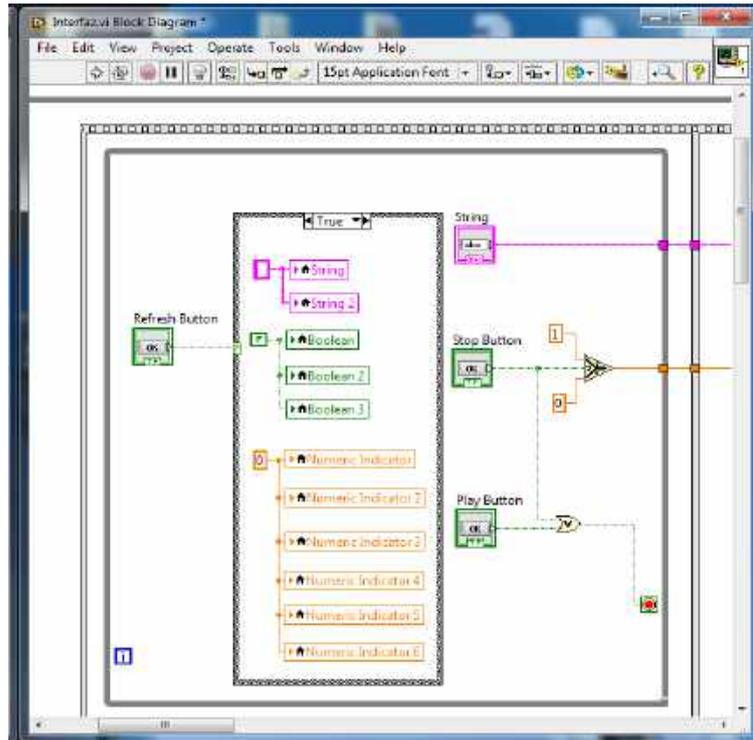
**Figura 76-3.** Diseño de la Interfaz  
Fuente: Asqui Byron, 2020.

3. *Ingresar ciclos de repetición:* Pulsar clic derecho en el diagrama de bloques, específicamente para las opciones “Structures” e ingresar los bucles “While Loop” y “Flat Sequence Structure”, luego presionar clic derecho sobre la estructura “Flat Sequence Structure” y seleccionar “add frame after”, hasta completar cuatro marcos de programación. Figura 77-3.

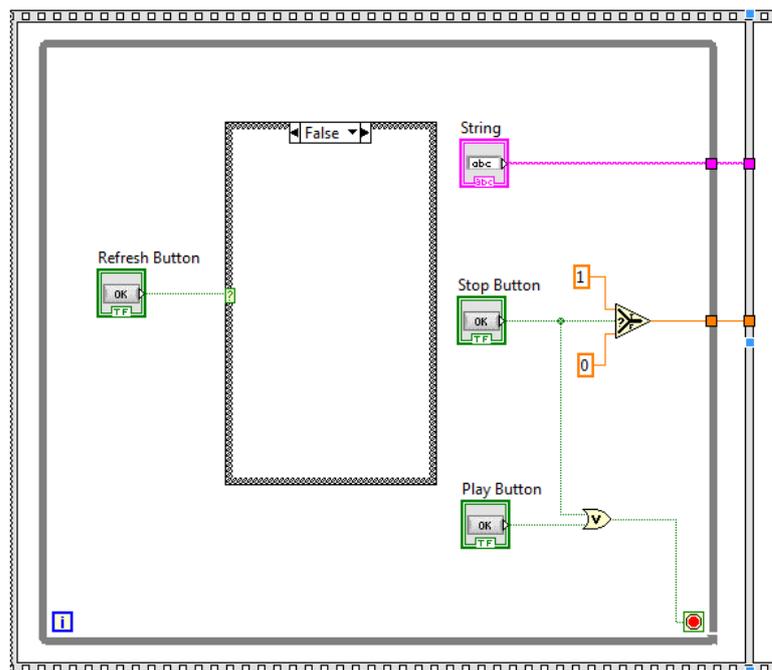


**Figura 77-3.** Bucle repetitivo  
Fuente: Asqui Byron, 2020.

4. *Programar la primera sección de Flat Sequence Structure:* Ingresar los botones de control, variables globales, estructura de caso, ciclo de repetición y una función de comparación para conectarlas de manera que cumplan tareas como reseteo, ejecución y paro del sistema. Figura 78-3 y 79-3.

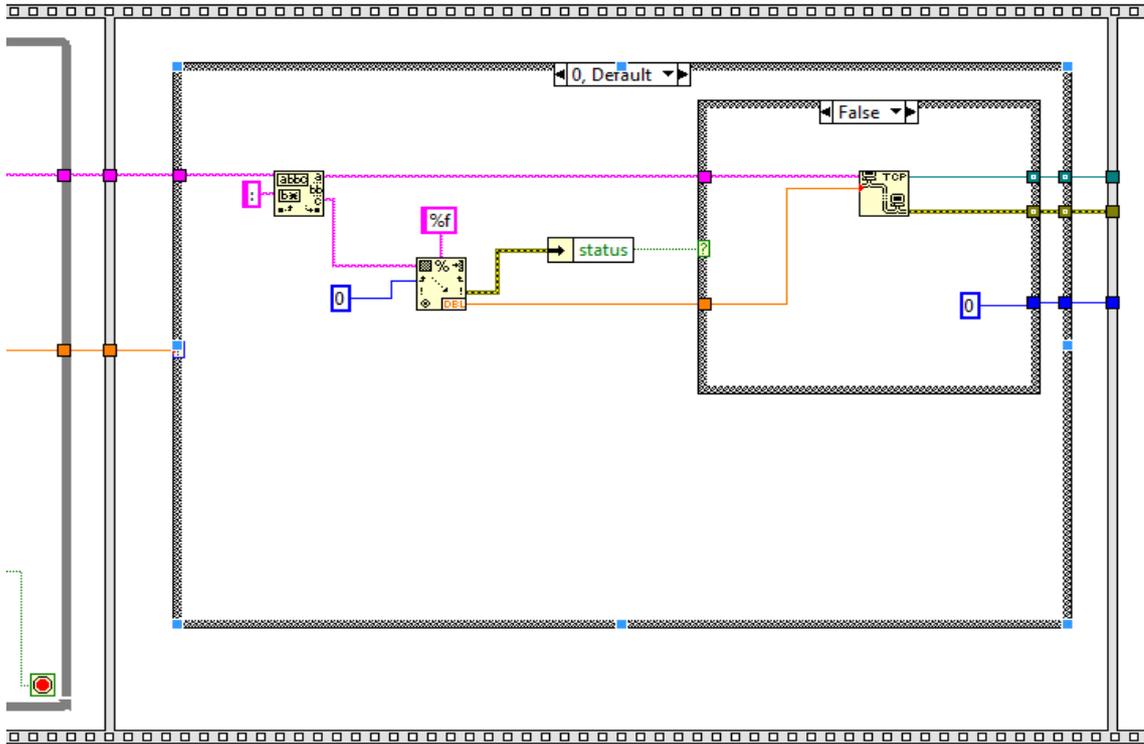


**Figura 78-3.** Primera sección Flat Sequence Structure, parte 1.  
Fuente: Asqui Byron, 2020.

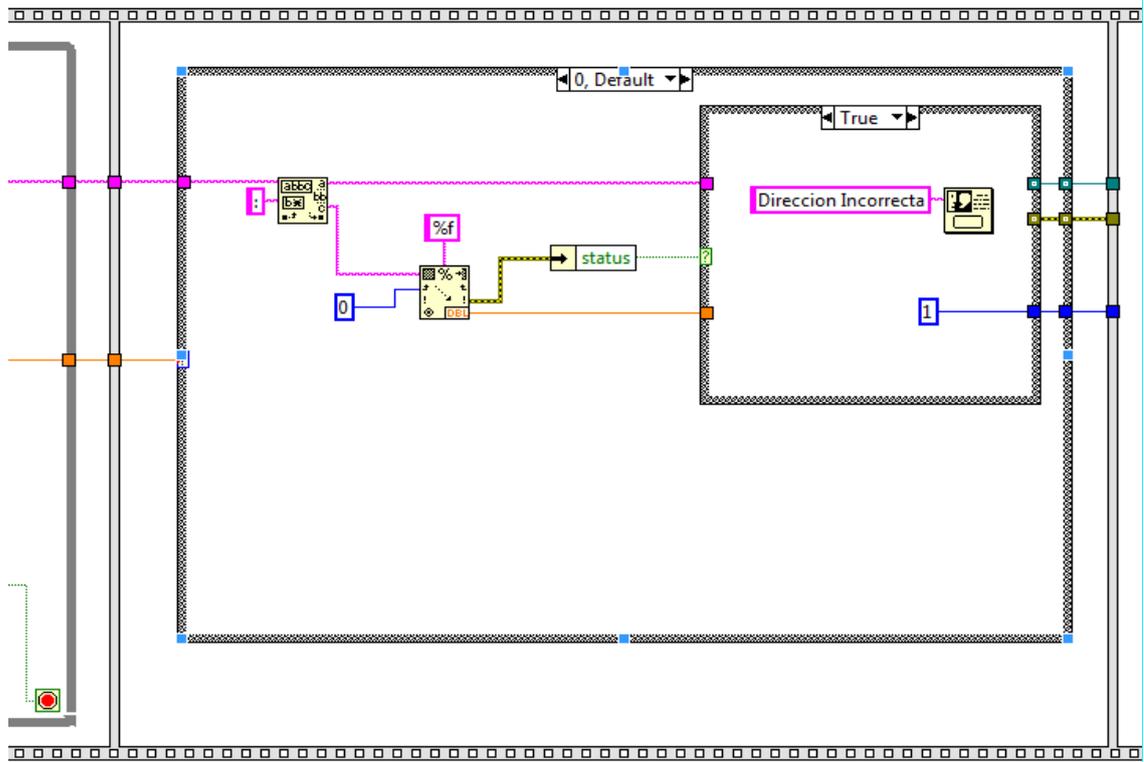


**Figura 79-3.** Primera sección Flat Sequence Structure, parte 2.  
Fuente: Asqui Byron, 2020.

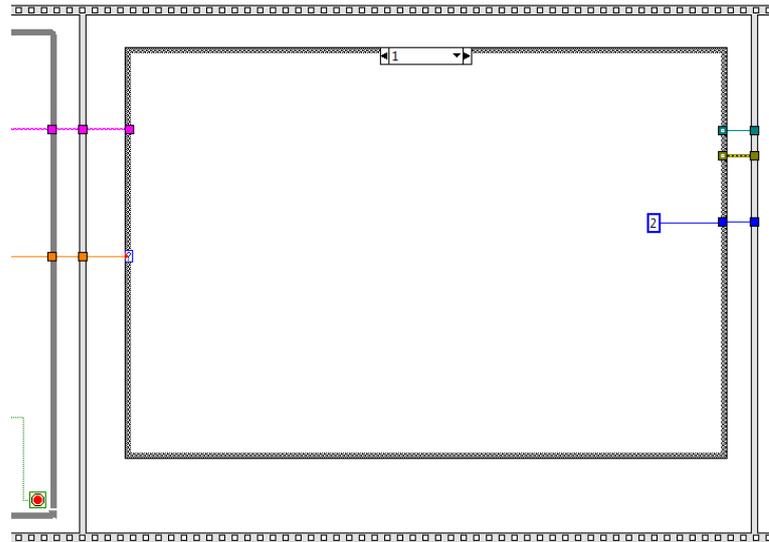
5. Programar la segunda sección de Flat Sequence Structure: Añadir las funciones de: “Math Pattern”, “Scan From String”, “Unbundle By Name”, “TCP Open Connection”, “One Button Dialog” y “Case Structure”, conectadas de manera que realicen un enlace con la plataforma remote it.



**Figura 80-3.** Segunda sección Flat Sequence Structure, parte 1.  
Fuente: Asqui Byron, 2020.

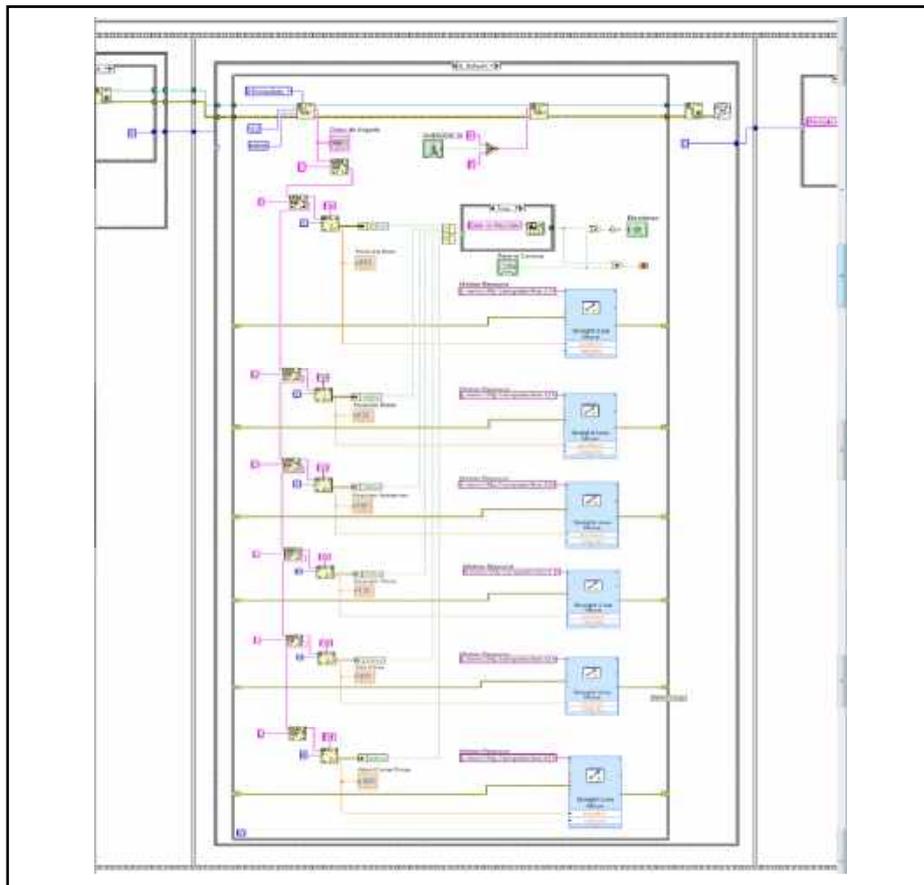


**Figura 81-3.** Segunda sección Flat Sequence Structure, parte 2.  
Fuente: Asqui Byron, 2020.

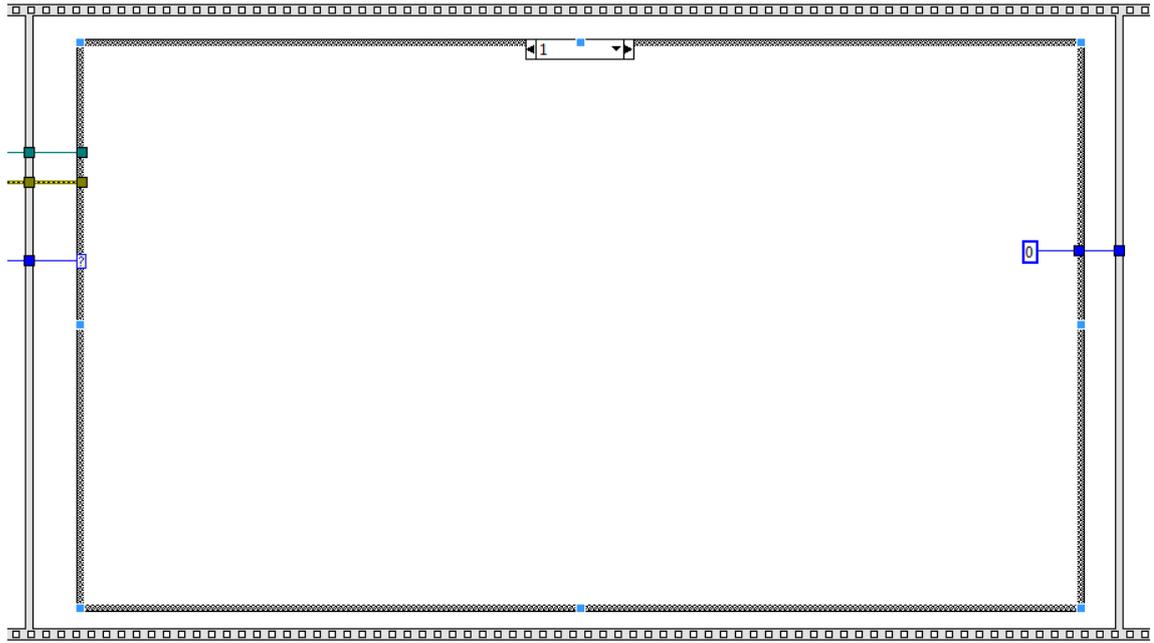


**Figura 82-3.** Segunda sección Flat Sequence Structure, parte 3.  
Fuente: Asqui Byron, 2020.

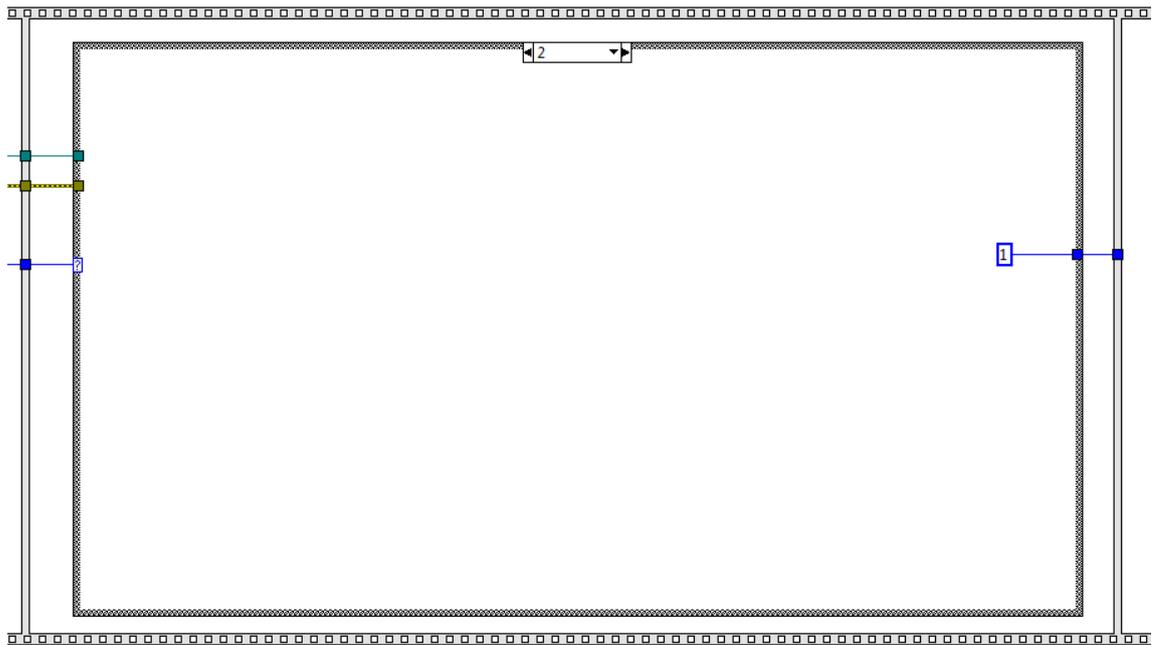
6. Programar la tercera sección de Flat Sequence Structure: Insertar las siguientes funciones: “Select”, “TCP Read”, “TCP Write”, “TCP Close Connection”, “Simple Error Handle.vi”, “While Loop” y “Case structure”, conectadas con los botones de control e indicadores para recibir y enviar datos a remote.



**Figura 83-3.** Tercera sección Flat Sequence Structure, parte 1  
Fuente: Asqui Byron, 2020.

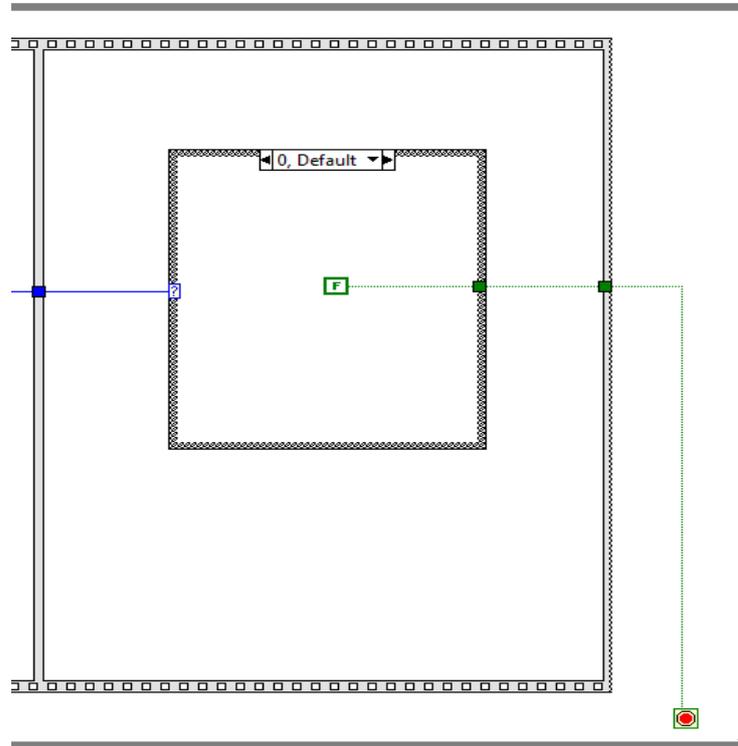


**Figura 84-3.** Tercera sección Flat Sequence Structure, parte 2.  
Fuente: Asqui Byron, 2020.

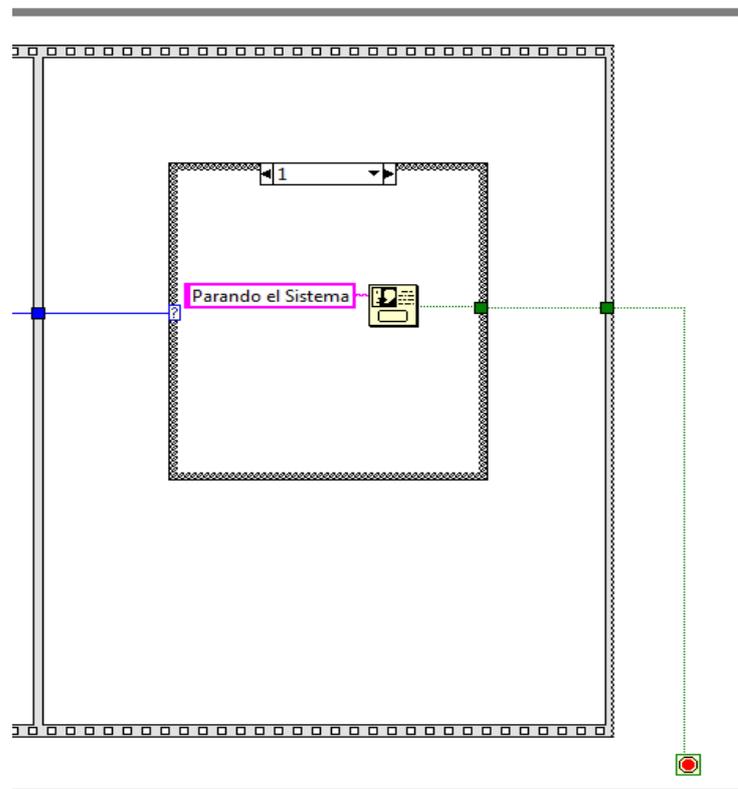


**Figura 85-3.** Tercera sección Flat Sequence Structure, parte 3.  
Fuente: Asqui Byron, 2020.

7. Programar la cuarta sección de Flat Sequence Structure: Añadir las funciones de “Case Structure” y “One Button Dialog” para un cierre adecuado del sistema.



**Figura 86-3.** Cuarta sección Flat Sequence Structure, parte 1.  
Fuente: Asqui Byron, 2020.



**Figura 87-3.** Cuarta sección Flat Sequence Structure, parte 2.  
Fuente: Asqui Byron, 2020.

### 3.8.7. Prueba de Funcionalidad

Las pruebas de funcionalidad se basaron en los movimientos de los grados de libertad del prototipo desde la interfaz gráfica ingresando a la plataforma de Remote para enlazar los dispositivos y el software LabVIEW verificando el flujo de datos al sistema.

#### 3.8.7.1. Ejecución del sistema

La ejecución del sistema incluyó el código del archivo Python comprobando si existen conflictos de resolución para enlazarla a Labview, con el siguiente proceso:

- Ingresar a la plataforma de Remote para enlazar los dispositivos
- Verificar el flujo de datos en la plataforma Ubidots, además de las acciones realizadas en la interfaz gráfica del prototipo en Labview, Solidworks y prototipado físico.

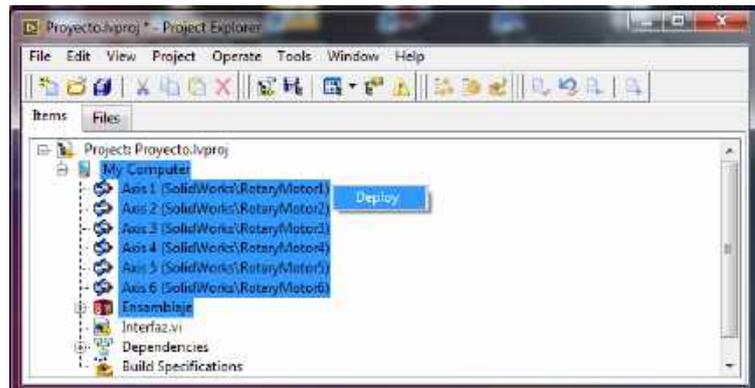
1. *Ejecutar el archivo Python*: mostrará la interfaz gráfica del prototipo para ingresar la contraseña previamente programada, presionar el botón conectar para realizar una conexión con la plataforma remote it. Figura 88-3.



**Figura 88-3.** Interfaz grafica del prototipo.

Fuente: Asqui Byron, 2020.

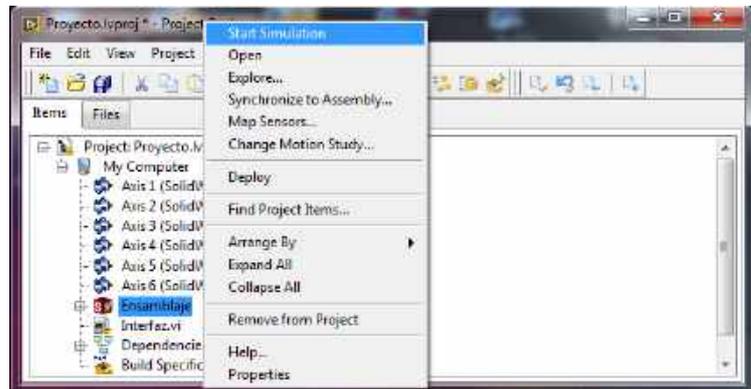
2. *Verificar si existe un conflicto de resolución:* en “My Computer”, axis y ensamblaje de solidworks mediante la opción deploy. Figura 89-3.



**Figura 89-3.** Verificación de conflictos.

Fuente: Asqui Byron, 2020.

3. *Ejecutar Star Simulation:* en el ensamblaje que se encuentra enlazado a Labview. Figura 90-3.



**Figura 90-3.** Ejecución de la simulación.

Fuente: Asqui Byron, 2020.

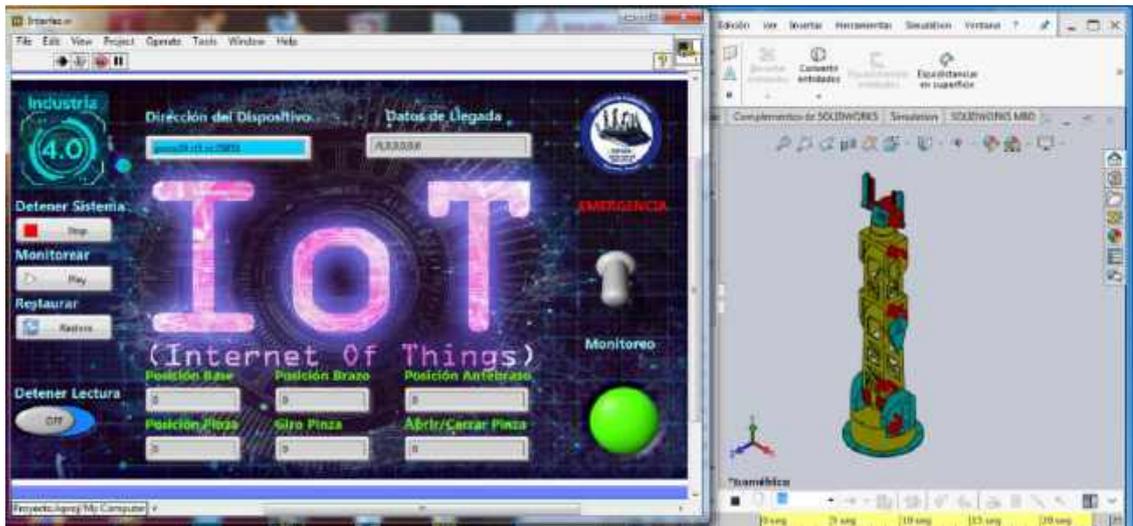
4. *Ingresar a la plataforma de Remote:* obtener la dirección pública para enlazar los dispositivos. Figura 91-3.



**Figura 91-3.** Dirección pública en Remote It.

Fuente: Asqui Byron, 2020.

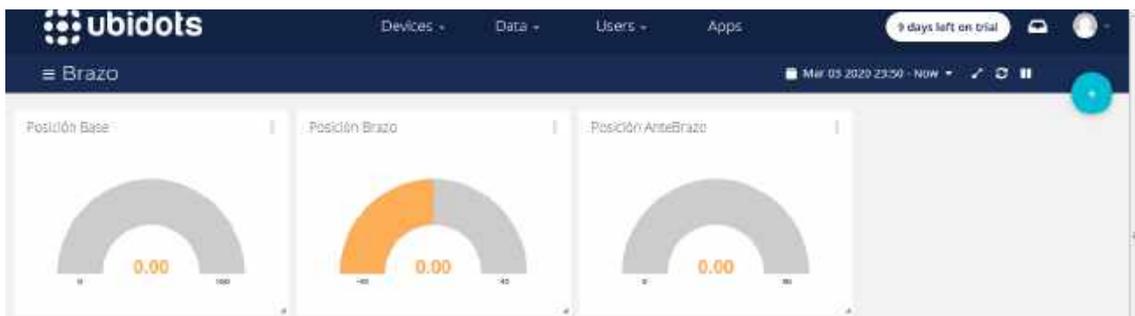
5. Cargar el programa de Labview: pegar la dirección pública en el cuadro de texto y presionar el botón “Play” para monitorear el sistema. Figura 92-3.



**Figura 92-3.** Interfaz grafica en LabVIEW.

Fuente: Asqui Byron, 2020.

6. Verificar el flujo de datos en la plataforma Ubidots.



**Figura 93-3.** Tableros en Ubidots

Fuente: Asqui Byron, 2020.

7. Cambiar los grados de libertad de los eslabones del brazo robótico prototipo.



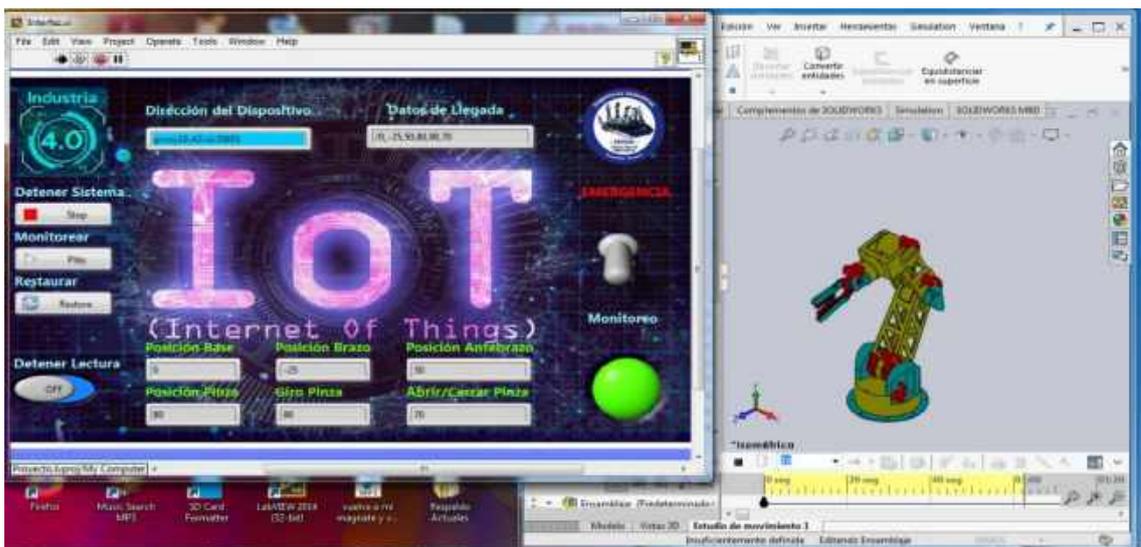
**Figura 94-3.** Interfaz grafica del prototipo

Fuente: Asqui Byron, 2020.

8. Comprobar las acciones realizadas en la interfaz gráfica del prototipo a través de la observación en: Ubidots, Labview, Solidworks y prototipado físico.



**Figura 95-3.** Brazo robótico prototipo.  
Fuente: Asqui Byron, 2020.



**Figura 96--3.** Interfaz grafica en LabVIEW.  
Fuente: Asqui Byron, 2020.



**Figura 97-3.** Tableros en Ubidots.  
Fuente: Asqui Byron, 2020.

La ejecución del sistema incluyó verificaciones en:

- La interfaz gráfica de LabVIEW
- El posicionamiento de los grados de libertad en los tableros de Ubidots
- Manual de usuario para la funcionalidad del dispositivo Raspberry Pi 3, LabVIEW, la interfaz gráfica del prototipo, monitoreo del sistema y procesamiento del flujo de datos.

### 3.8.7.2. Funcionamiento del Prototipo

Al realizar varias pruebas comparativas entre la interfaz gráfica con el prototipo físico, se verificó los desplazamientos de cada grado de libertad, llegando a concluir que existe un error de 1 a 2 grados. Tabla 47-3.

**Tabla 47-3:** Interfaz vs Prototipo

INTERFAZ VS PROTOTIPO				
Grado de libertad	Ejecución Real			
N° 1				
Movimiento	✓	✓	✗	✓
N° 2				
Movimiento	✓	✗	✗	✓
N° 3				
Movimiento	✓	✓	✗	✓
N° 4				
Movimiento	✓	✗	✓	✗
N° 5				
Movimiento	✓	✓	✓	✓
N° 6				
Movimiento	✓	✓	✓	✓

Realizado por: Asqui Byron, 2020.

### 3.8.7.3. Consumo de corriente

El consumo ideal de los servomotores acorde a las especificaciones es de 500-900mA, teniendo en cuenta lo mencionado se procedió a realizar la medición del consumo de corriente cuando el prototipo se encontraba inmóvil con ángulos mínimos y máximos, a su vez un control en los movimientos individuales, concluyendo que el consumo máximo de corriente es aproximadamente 1 A. Tabla 48-3.

**Tabla 48-3:** Consumo de Corriente

<b>CONSUMO DE CORRIENTE</b>			
<b>Amperio (máx.)</b>			
<b>Grado de libertad</b>	<b>Ángulos min. Estático</b>	<b>Ángulos en Movimiento</b>	<b>Ángulos máx. Estático</b>
N°1		0.40(A)	
N°2		0.90(A)	
N°3		0.38(A)	
N°4		0.32(A)	
N°5		0.38(A)	
N°6		0.32(A)	

Realizado por: Asqui Byron, 2020.

## 4. GESTIÓN DEL PROYECTO

La viabilidad del proyecto se define con uno de los apartados más importantes en cualquier implementación, se hace mención al cálculo de los costos directos e indirectos generados a la escala del prototipo, se incluye además componentes eléctricos, mecánicos y electrónicos.

### 4.1. Costos Directos

**Tabla 1-4:** Cálculo de costos directos

<b>COSTOS DIRECTOS</b>		
<b>DESCRIPCIÓN</b>	<b>CANTIDAD</b>	<b>COSTO (USD)</b>
Dispositivo Raspberry Pi 3V (+Accesorios)	1	120,00
Módulo PCA9685	1	10,00
Servo MG996R	5	55,00
Microservo MG90S	2	12,00
Rollos de PLA	4	120,00
Tornillos, arandelas, tuercas y llaves hexagonales (Diferentes longitudes)	90	15,00
Fuente de alimentación 5V	1	22,00
Base y soporte para la fuente	2	20,00
Cables para servos (Diferentes longitudes)	16	15,00
Rodamientos	2	12,00
Resortes	4	4,00
Canaleta en espiral	1	5,00
<b>COSTO TOTAL</b>		<b>410,00</b>

Realizado por: Asqui Byron, 2020.

La Tabla 1-4 hace referencia a todos los costos que afectan de forma inmediata a la cotización de los componentes físicos que intervinieron en el prototipo

### 4.2. Costos Indirectos

**Tabla 2-4:** Cálculo de costos indirectos

<b>COSTOS INDIRECTOS</b>	
<b>DESCRIPCIÓN</b>	<b>COSTOS (USD)</b>
Transporte	20,00
Imprevistos	30,00
<b>COSTO TOTAL</b>	<b>50,00</b>

Realizado por: Asqui Byron, 2020.

La Tabla 2-4 contiene los costos indirectos, que se generan a partir de actividades extras que no necesariamente pertenecen al proceso de fabricación.

### 4.3. Costos Totales:

**Tabla 3-4:** Cálculo de costos Totales

<b>COSTOS TOTALES</b>	
<b>DESCRIPCIÓN</b>	<b>COSTO</b>
Costos directos	410,00
Costos indirectos	50,00
<b>COSTO TOTAL</b>	<b>460,00</b>
<b>Fuente de Financiamiento</b>	<b>Autónomo</b>

Realizado por: Asqui Byron, 2020.

La Tabla 3-4 es la suma de todos los costos evidenciados a lo largo de la investigación destinada al proceso de diseño para un equipo de control y monitoreo.

## 5. CONCLUSIONES

- La integración del hardware, software y el IOT con enlaces a diversas plataformas digitales, permitieron la creación del proceso destinado a: envío, monitoreo y control de datos en tiempo real.
- El brazo prototipo en las pruebas realizadas demostró que las tareas a las que fue destinado a ejecutar tuvieron éxito total, por ende, su réplica en cualquier proceso productivo de la industria es cuestión de manejar únicamente escalas, además de resaltar que contribuye al crecimiento de las PYMES en el país.
- Las acciones de: supervisión, control local escalable, manipulación y configuración del brazo robótico se sustentaron en lenguajes de programación como Python además en ordenadores como Raspberry Pi y softwares de código abierto que dan acceso a la modificación del código acorde los requerimientos propios en sistemas embebidos.
- La simulación virtual remota del brazo robótico prototipo fue generada por el enlace entre el software LabVIEW y SolidWorks con total éxito, pues existieron diferentes etapas encargadas de validar acciones previas, el caso de esta simulación se comprobó en distintos softwares por mencionar la aplicación creada para visualización remota.
- El control del sistema global se destinó en el paro de emergencia creado en LabVIEW, no obstante, también el software se encargó del monitoreo de los datos visualizados en tiempo real, al igual que Ubidots. Las dos plataformas se interpretaron en SolidWorks para verificar el comportamiento del proceso.
- Algunas aplicaciones del brazo prototipo presentan ciertas restricciones a la hora de agregar una cantidad de dispositivos electrónicos, limitando el alcance; sin embargo, por tratarse de un prototipado para un dispositivo tecnológico el análisis estático, simulación y software son idóneos.
- El brazo robótico prototipo fue rediseñado para el monitoreo del sistema, debido a que los archivos originales se encuentran en formato “STL” (Estereolitografía) generando gran cantidad de elementos que ralentizaban el análisis de movimientos.
- El análisis de materiales del brazo prototipo se realizó a través del software Solidworks siendo la mejor alternativa el PLA (ácido poliláctico) presentando una deformación de de 0,00013 - 0,020 imperceptible a una fuerza de 100N
- La Raspberry Pi considerado un mini computador fue el dispositivo de control local del brazo robótico debido a su software residente se desarrolló una interfaz gráfica que emite señales de control al prototipo.

- Los ángulos presentados por la interfaz gráfica para realizar una acción por el brazo robótico poseen un margen de error de 1 a 2 grados debido a la conversión de datos en señales eléctricas en la programación.
- El consumo ideal en las corrientes y la verificación en cada uno de los actuadores fue aproximadamente de 1A garantizando que la fuente de alimentación implementada es la adecuada con una entrega de 6A.

## 6. RECOMENDACIONES

- Se recomienda no efectuar movimientos bruscos ni sobrecargar el brazo podrían generarse averías o bloqueo en los actuadores.
- El Brazo robótico prototipo se utilizó como fin demostrativo para simulación, en procesos de mayor precisión se debe rediseñar la estructura para usar motores a paso.
- La idea de conocer un entorno de programación puede dar una falsa interpretación del sistema, en nuevas investigaciones se puede mejorar la programación acorde de las necesidades optimizando sus líneas de código o a su vez reduciéndolas.
- Continuar aportando con investigaciones de carácter tecnológico bajo el concepto de mejorar los procesos de control y monitoreo a través de la incorporación de un sistema de video-vigilancia, entre otros.
- El prototipo para ser expuesto a la intemperie debe someterse a nuevas investigaciones relacionadas con la humedad y temperatura del ambiente que va a soportar.
- Se recomienda diseñar una carcasa que permita proteger contra golpes o caídas del sistema. La construcción debe realizarse bajo un estudio previo para validar dicho elemento.

## GLOSARIO

**Algoritmo:** una serie de instrucciones sencillas que se llevan a cabo para solventar un problema. (Fanjul, 2018)

**Actuador:** Es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). (Vildósola, 2008)

**Datos en tiempo real:** Datos habilitados temporalmente cuyo seguimiento y respuesta se produce en tiempo real. Utilizados por sistemas de respuesta de emergencia, seguimiento de flotas comerciales, detección de amenazas para la defensa. (Vester, 2019)

**Dúctil:** Es una propiedad que presentan algunos materiales, los cuales, bajo la acción de una fuerza, pueden deformarse plásticamente de manera sostenible sin romperse. (Fernandini, 2017)

**Factor de seguridad:** Es el cociente resultante entre el valor calculado de la capacidad máxima de un sistema y el valor del requerimiento esperado real al que se verá sometido. (Stemmer, 2018)

**Grados de libertad:** El número de coordenadas independientes necesarias para expresar la posición de todas sus partes. (Rodríguez, 2015)

**Hardware:** Es la parte física de un ordenador o sistema informático. Está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y luz, placas, memorias, discos duros, dispositivos periféricos y cualquier otro material en estado físico que sea necesario para hacer que el equipo funcione. (Vester, 2019)

**Nube:** Red enorme de servidores remotos de todo el mundo que están conectados para funcionar como un único ecosistema. Están diseñados para almacenar y administrar datos, ejecutar

aplicaciones o entregar contenido o servicios, como streaming de videos, correo web, software de ofimática o medios sociales. (Vester, 2019)

**Open Source:** es un modelo de desarrollo de software que **se basa en la colaboración abierta**, es decir, se permite que el programa en cuestión sea modificado de manera abierta, con libertad de que los usuarios expertos en el tema puedan elegir cómo funcionará el programa y cambiar ciertas características de éste. (Méndez, 2018)

**Protocolo de comunicación:** Consiste en cualquier mecanismo utilizado para comunicar dispositivos, desde lo más sencillo, hasta lo más complejo. (Vester, 2019)

**Software:** Programa o conjunto de programas interrelacionados con funciones tan diversas como operar y controlar el ordenador. (Ávila et al., 2011)

**Superposición:** Es el acto y el resultado de superponer. Es decir, ubicar algo sobre otra cosa o a hacer que dos elementos se encimen. (Pérez, 2016)

## BIBLIOGRAFÍA

**ALVEAR, Vanessa; et al.** “Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura”. *Enfoque UTE* [En línea], 2017, (Ecuador),8(1), p. 244-56. [Consulta: 12 octubre 2019]. Disponible en: <https://doi.org/10.29019/enfoqueute.v8n1.121>

**ÁVILA GONZÁLEZ, Barbarita Librada; et al.** *Software educativo la naturaleza y el hombre* [En línea]. México: Eumed, 2011. [Consulta: 19 agosto 2020]. Disponible en: <https://www.eumed.net/libros-gratis/2011b/953/anexo.pdf>

**BARRIO, Andrés Moisés.** *Internet de las Cosas* [En Línea]. 1ª ed. Madrid-España: Reus, 2018. [Consulta: 5 octubre 2019]. Disponible en: [https://www.editorialreus.es/static/pdf/primeraspaginas\\_9788429020380\\_internetdelascosas.pdf](https://www.editorialreus.es/static/pdf/primeraspaginas_9788429020380_internetdelascosas.pdf)

**BERCHON, Mathilde; & LUYT, Bertier.** *La impresión 3D: guía definitiva para makers, diseñadores, estudiantes, profesionales, artistas y manitas en general* [En Línea]. Barcelona-España: Gustavo Gili, 2016. [Consulta: 17 octubre 2019]. Disponible en: <http://ebookcentral.proquest.com/lib/epochsp/detail.action?docID=4536132>

**COQUE MORILLO, Edison Patricio.** Diseño e implementación de un sistema de control para el brazo robótico de cinco grados de libertad, utilizando LabVIEW e internet [En Línea]. (Trabajo de titulación). (Ingeniería) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales. Riobamba-Ecuador. 2013. [Consulta: 2019-10-10]. Disponible en: <http://dspace.espace.edu.ec/bitstream/123456789/3187/1/108T0088.pdf>

**DA COSTA, L; et al.** Equipos de control a distancia para actividades de clausura. *IOEA* [En Línea], 2015, (Austria),27(3), p. 35-39. [Consulta: 7 octubre 2019]. Disponible en: [https://www.iaea.org/sites/default/files/27405393539\\_es.pdf](https://www.iaea.org/sites/default/files/27405393539_es.pdf)

**EVANS, Dave.** *Internet de las cosas Cómo la próxima evolución de Internet lo cambia todo* [En Línea]. Cisco IBSG, 2011. [Consulta: 5 octubre 2019]. Disponible en: [https://www.cisco.com/c/dam/global/es\\_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf](https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf)

**FANJUL, Sergio.** *En realidad, ¿qué [...] es exactamente un algoritmo?* [blog]. Madrid: El País Retina, 24 marzo, 2018. [ Consulta: 19 agosto 2020]. Disponible en: [https://retina.elpais.com/retina/2018/03/22/tendencias/1521745909\\_941081.html](https://retina.elpais.com/retina/2018/03/22/tendencias/1521745909_941081.html)

**FERNANDINI, Javiera.** *Ventajas y desventajas de los biomateriales* [En Línea]. Buenos Aires: wordpress, 2017. [Consulta: 19 agosto 2020]. Disponible en: <https://materiainvestigacion.files.wordpress.com/2017/12/fernandini-plan.pdf>

**HERNÁNDEZ SÁNCHEZ, Carlos Andrés.** Integración de SolidWorks y LabVIEW para la elaboración de un prototipo virtual de un manipulador paralelo tipo delta de tres grados de libertad [En Línea]. (Trabajo de titulación). (Ingeniería) Universidad Tecnológica de Pereira, Facultad Ingeniería Mecánica. Pereira. 2016.[Consulta: 2019-10-20]. Disponible en: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/6958/004165H557.pdf?sequence=1&isAllowed=y>

**INTEREMPRESAS, R.** *La industria 4.0 y sus Orígenes* [blog]. 5 marzo, 2019. [ Consulta: 7 octubre 2019]. Disponible en: <https://aspromec.org/la-industria-4-0-y-sus-origenes/>

**JOHANSSON, Robert.** *Introduction to Scientific Computing in Python* [En Línea]. GitHub. 2016. [ Consulta: 15 octubre 2019]. Disponible en: <https://www.freetechbooks.com/introduction-to-scientific-computing-in-python-t1117.html>

**LAJARA, José Rafael; & PELEGRÍ, José.** *LabVIEW: Entorno gráfico de programación* [En Línea]. 2ª ed, Barcelona-España: Marcombo, 2011. [ Consulta: 16 octubre 2019]. Disponible en: [https://books.google.com.ec/books?hl=es&lr=&id=ZFQua3-eeQEC&oi=fnd&pg=PA21&dq=COMUNICACION+SERIAL+EN+LABVIEW&ots=qI1xU8SFhu&sig=gfJ14V93tVGnHRzA5eg1xa5zszI&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.ec/books?hl=es&lr=&id=ZFQua3-eeQEC&oi=fnd&pg=PA21&dq=COMUNICACION+SERIAL+EN+LABVIEW&ots=qI1xU8SFhu&sig=gfJ14V93tVGnHRzA5eg1xa5zszI&redir_esc=y#v=onepage&q&f=false)

**LIÑÁN COLINA, Antonio; et al.** *Internet de las Cosas* [En Línea]. 2015. [ Consulta: 13 octubre 2019]. Disponible en: <http://wireless.ictp.it/Papers/InternetdelasCosas.pdf>

**LLANEZA GONZÁLEZ, Paloma.** *Seguridad y responsabilidad en la Internet de las cosas (IoT)*. 1ª ed. España: Wolters Kluwer, 2018. p 57.

**LÓPEZ ALDEA, Eugenio.** *Raspberry pi fundamento y aplicaciones*. Madrid, España: Ra-Ma, 2017. ISBN:978-84-9964-719-7. p. 63

**LÓPEZ GARZÓN, Wilmar Ferney; & CÁRDENAS LÓPEZ, Juan Camilo.** “Tecnología internet of things (IoT) y el big data”. *Data. Mare Ingenii. Ingenierías* [En Línea], 2019, (Colombia), 1(1), p. 73-79. [Consulta: 14 octubre 2019]. Disponible en: <http://cipres.sanmateo.edu.co/index.php/mi>

**LÓPEZ I SEUBA, Manel.** *Internet de las cosas La transformación digital de la sociedad* [En Línea]. Madrid-España: Ra-Ma, 2019. [ Consulta: 12 octubre 2019] Disponible en: <http://www.tecno-libro.es/ficheros/descargas/9788499647999.pdf>

**MEJÍA FLORES, Heriberto Julio.** “Ventajas y desventajas de las impresoras 3D”. *Rev. Tecnológica* [En Línea]. 2016, 12(18). pp. 30-34. [Consulta: 2019-10-17]. ISSN 1729-7532. Disponible en: [http://www.revistasbolivianas.org.bo/scielo.php?script=sci\\_arttext&pid=S1729-75322016000100006&lng=es&nrm=iso](http://www.revistasbolivianas.org.bo/scielo.php?script=sci_arttext&pid=S1729-75322016000100006&lng=es&nrm=iso)

**MÉNDEZ, David.** *Open Source* [blog]. Economía Simple, 03 diciembre, 2018. [Consulta: 19 agosto 2020]. Disponible en: <https://www.economiasimple.net/glosario/open-source>

**NADAL S, Victoria M.** *Siete formas en que el Internet de las Cosas (IoT) cambia nuestro trabajo* [blog]. Madrid: El País Retina, 1 febrero, 2018. [ Consulta: 10 octubre 2019]. Disponible en: [https://retina.elpais.com/retina/2018/01/22/talento/1516639374\\_582456.html](https://retina.elpais.com/retina/2018/01/22/talento/1516639374_582456.html)

**PARRA, Jorge; et al.** “IOT: Una aproximación desde ciudad inteligente a universidad inteligente”. *Revista Ingenio UFPSO* [En línea], 2017, (Colombia),13(1), p. 9-20. [Consulta: 13 octubre 2019]. Disponible en: <https://doi.org/10.22463/2011642X.2128>

**PÉREZ PORTO, Julián; & GARDEY, Ana.** *Definición de superposición* [blog]. Definicion.de, 2016. [Consulta: 19 agosto 2020]. Disponible en: <https://definicion.de/superposicion/>

**RODRÍGUEZ, Manuel.** *¿Sabes en robótica la diferencia entre grados de libertad y grados de movilidad?* [blog] Revista Digital, 29 julio, 2015. [Consulta: 19 agosto 2020]. Disponible en:

<https://revistadigital.inesem.es/gestion-integrada/diferencia-robotica-grados-libertad-movilidad-3/>

**RODRÍGUEZ GONZÁLEZ, Dana.** “Arquitectura y Gestión de la IoT”. *Telemática Magazine* [En Línea], 2013, 12(3), p. 49-60. [Consulta: 14 octubre 2019]. ISSN 1729-3804. Disponible en: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/119/115>

**RODRÍGUEZ VIDAL, Carlos.** *Diseño mecánico con SolidWorks 2015* [En Línea]. Madrid – España: Ra-Ma, 2015. [ Consulta: 16 octubre 2019]. Disponible en: <http://ebookcentral.proquest.com/lib/espochsp/detail.action?docID=5758939>

**ROSE, Karen; et al.** *La internet de las cosas— una breve reseña* [En Línea]. Internet Society (ISOC), 2015. [ Consulta: 14 octubre 2019]. Disponible en: <https://www.internetsociety.org/es/resources/doc/2015/iot-overview>

**SEMICONDUCTORS, NXP.** *PCA9685 16-channel, 12-bit PWM Fm+ I2C-bus LED controller* [En Línea]. NXP B.V, 2015. [ Consulta: 16 octubre 2019]. Disponible en: <https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf>

**SERNA C., Liliana; et al.** “Ácido Poliláctico (PLA): Propiedades Y Aplicaciones”. *Ingeniería y Competitividad* [En Línea], 2003, 5(1), p. 16-26. [Consulta: 18 octubre 2019]. Disponible en: <https://doi.org/10.25100/iyc.v5i1.2301>

**STEMMER.** *Coeficiente de seguridad: toda una filosofía de diseño* [blog]. Infaimo ,11 mayo, 2018. [Consulta: 19 agosto 2020]. Disponible en: <https://blog.infaimon.com/coeficiente-de-seguridad-filosofia-diseno/>

**TORRES, Edgar; et al.** “Sistema de posicionamiento aplicado a la técnica de impresión 3D modelado por deposición fundida”. *Revista de Investigación, Desarrollo e Innovación*, vol. 3, n° 1 (2012), (Colombia) pp. 25-32.

**VESTER.** *Glosario* [blog]. Vester Business, 2019 [Consulta: 19 agosto 2020]. Disponible en: <https://vesterbusiness.com/glosario/>

**VERA LUZURIAGA, John Germán.** Control de trayectoria de la simulación de un brazo robot de 5 grados de libertad, controlado mediante la plataforma C2000 Piccolo LAUNCHXL-F28027F

[En Línea]. Universidad Politécnica de Valencia. Valencia-España. 2017. [Consulta: 2019-10-10]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/79512/VERA%20-%20Control%20de%20trayectoria%20de%20la%20simulaci%C3%B3n%20de%20un%20brazo%20robot%20de%205%20grados%20de%20libertad%2C%20control....pdf?sequence=1&isAllowed=y>

**VILDÓSOLA, Eugenio.** *Actuadores* [En línea]. Chile: Soltex, 2008. [Consulta: 19 agosto 2020]. Disponible en: <http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>



28-08-2020

0257-DBRAI-UPT-2020



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: 10 / 09 / 2020

<b>INFORMACIÓN DEL AUTOR/A (S)</b>
<b>Nombres – Apellidos:</b> BYRON IVAN ASQUI RAMIREZ
<b>INFORMACIÓN INSTITUCIONAL</b>
<b>Facultad:</b> MECÁNICA
<b>Carrera:</b> INGENIERÍA INDUSTRIAL
<b>Título a optar:</b> INGENIERO INDUSTRIAL
<b>f. Analista de Biblioteca responsable:</b> Lcdo. Holger Ramos, MSc.   0257-DBRAI-UPT-2020