



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN**

**TEMA: DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO DIDÁCTICO
DE UN SISTEMA DE AUTOMATIZACIÓN DE LLENADO Y
ENVASADO DE SÓLIDOS, UTILIZANDO SENSORES
FOTOELÉCTRICOS, ULTRASÓNICOS, CAPACITIVOS, ENCODERS,
GALGAS EXTENSIOMÉTRICAS Y BRAZOS ROBÓTICOS, PARA EL
LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE
PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS
ESPE EXTENSIÓN LATACUNGA**

**AUTORES: SEBASTIÁN FRANCISCO PANCHI OLIVO
WASHINGTON PAÚL HERRERA SINCHIGUANO**

**DIRECTOR: ING. PRUNA, EDWIN.
CODIRECTOR: ING. ÁVILA, GALO.**

LATACUNGA

2015

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

CERTIFICADO

Ing. Edwin Pruna (DIRECTOR)

Ing. Galo Ávila (CODIRECTOR)

CERTIFICAN

Que el trabajo titulado “DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO DIDÁCTICO DE UN SISTEMA DE AUTOMATIZACIÓN DE LLENADO Y ENVASADO DE SÓLIDOS, UTILIZANDO SENSOES FOTOELÉCTRICOS, ULTRASÓNICOS, CAPACITIVOS, ENCODERS, GALGAS EXTENSIOMÉTRICAS Y BRAZOS ROBÓTICOS, PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA” realizado por Sebastián Francisco Panchi Olivo y Washington Paúl Herrera Sinchiguano, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPE.

Debido a que constituye un trabajo de alto contenido de investigación y que ayuda a la formación profesional y aplicación de conocimientos, si se recomienda su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Sebastián Francisco Panchi Olivo y Washington Paúl Herrera Sinchiguano que lo entregue al Ing. Franklin Silva, en su calidad de Director de la Carrera.

Latacunga, Octubre del 2015.

Ing. Edwin Pruna P.
DIRECTOR

Ing. Galo Ávila R.
CODIRECTOR

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

DECLARACIÓN DE RESPONSABILIDAD

*Nosotros, SEBASTIÁN FRANCISCO PANCHI OLIVO y
WASHINGTON PAÚL HERRERA SINCHIGUANO*

DECLARAMOS QUE:

El proyecto de grado denominado “DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO DIDÁCTICO DE UN SISTEMA DE AUTOMATIZACIÓN DE LLENADO Y ENVASADO DE SÓLIDOS, UTILIZANDO SENSORES FOTOELÉCTRICOS, ULTRASÓNICOS, CAPACITIVOS, ENCODERS, GALGAS EXTENSIOMÉTRICAS Y BRAZOS ROBÓTICOS, PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las citas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Octubre del 2015

Sebastián Francisco Panchi Olivo
C.C.: 0503140675

Washington Paúl Herrera Sinchiguano
C.C.: 0503073041

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

AUTORIZACIÓN

Nosotros, *SEBASTIÁN FRANCISCO PANCHI OLIVO* y
WASHINGTON PAÚL HERRERA SINCHIGUANO.

Autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO DIDÁCTICO DE UN SISTEMA DE AUTOMATIZACIÓN DE LLENADO Y ENVASADO DE SÓLIDOS, UTILIZANDO SENSORES FOTOELÉCTRICOS, ULTRASÓNICOS, CAPACITIVOS, ENCODERS, GALGAS EXTENSIOMÉTRICAS Y BRAZOS ROBÓTICOS, PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, Octubre del 2015.

Sebastián Francisco Panchi Olivo
C.C.: 0503140675

Washington Paúl Herrera Sinchiguano
C.C.: 0503073041

DEDICATORIA

Dedico este trabajo a Dios por brindarme sabiduría en cada paso de mi vida, por darme fortaleza en los momentos muy duros de mi vida y por haber puesto en mi camino a aquellas personas que han sido el soporte para alcanzar esta meta.

A mi madre Rosa, por darme la vida, por ser padre y madre a la vez, por motivarme día a día aún en los momentos difíciles y sobre todo confiar en que lograría terminar mi carrera, gracias mamá.

A mi tío Milton, aún sin ser mi padre me motivaste, me apoyaste, acogí tus buenos consejos y sobre todo nunca perdiste la esperanza en que culminaría mi carrera, gracias tío.

A mis tíos Gerardo y Matilde, por darme fuerza y aliento, carácter a enfrentar los problemas y sobre todo el amor de mi familia, los quiero mucho.

A mis amigos Gladys, Mónica, Sebastián y Andrés por ser las personas que me levantaron en los peores momentos y sobre todo fueron como los hermanos que nunca tuve, gracias de todo corazón.

A todos mis amigos, Verónica, Ana, Evelyn, Jenny, Gabriela, Alex, Javier, Luis, Jorge, Patricio por compartir los buenos y malos momentos.

WASHINGTON

AGRADECIMIENTO

A mi madre, por brindarme su apoyo incondicional y enseñarme a perseverar a través de su amor y sus sabios consejos.

A mi familia, por nunca perder la fe y sobre todo animarme en terminar mi carrera.

Al Ing. Edwin Pruna, Director de tesis y al Ing. Galo Ávila, Codirector de la tesis, por sus valiosas guías y asesoramientos para la realización de este proyecto de grado.

Al Ing. Armando Álvarez y a la Ing. Amparo Meythaler docentes de la carrera por sus sabios consejos y enseñarme que las segundas oportunidades son muy valiosas para formar un cambio total en uno mismo.

A todas las personas que ayudaron directa e indirectamente en la realización de la misma.

WASHINGTON

DEDICATORIA

El presente proyecto va dedicado en primer lugar a Dios mi padre celestial que tan misericordioso me supo guiar en la trayectoria de mi carrera universitaria, mediante el ejemplo del señor Jesucristo impregnado en mis padres los cuales son el pilar y templo de enseñanza en mi diario caminar.

SEBASTIÁN

AGRADECIMIENTO

Agradezco a mis padres Patricio y Jakeline que, con sus sabios consejos, su gran esfuerzo y trabajo, me supieron encaminar por los caminos del bien para culminar con éxito mi carrera universitaria.

A mis hermanas Karen y Belén, mi hermano Eduardo y mi tía Leonor que, con la convivencia diaria durante estos años, aprendí y sigo aprendiendo que la vida tiene tropiezos, pero hay que saber levantarse seguir adelante y luchar por los sueños.

Al resto de mi familia y todos esos amigos y amigas que con sus consejos y alagos me incentivaron a esforzarme y ser mejor cada día.

A mi tío Pepe que, gracias a su colaboración, humildad y su gran ayuda en momentos difíciles fue y es de gran apoyo en mi vida.

A mi tío Trajanito y mis abuelitos paternos que, gracias a su apoyo moral, me permitieron fomentar y cubrir, imprevistos que tiene la vida del estudiante universitario.

Al Ing. Edwin Pruna y al Ing. Galo Ávila, por su comprensión, ayuda y enseñanza para la realización de este proyecto de grado.

SEBASTIÁN

ÍNDICE DE CONTENIDOS

PORTADA	i
CERTIFICADO	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
DEDICATORIA	vii
AGRADECIMIENTO	viii
ÍNDICE DE CONTENIDOS	ix
ÍNDICE DE FIGURAS	xvi
ÍNDICE DE TABLAS	xxii
RESUMEN	xxiii
ABSTRACT	xxiv
CAPÍTULO I	1
1. FUNDAMENTACIÓN TEÓRICA	1
1.1. Antecedentes	1
1.2. Automatización y control	2
1.2.1. Automatización de un sistema	2
a. Parte operativa	2
b. Parte de mando	3
1.2.2. Objetivos de la automatización	4
1.2.3. Sistema de control	5
1.2.4. Definiciones básicas de un sistema de control	5
a. Sistema	5
b. Proceso	6
c. Planta	6
d. Actuador	6
e. Transductor	6
f. Variable manipulada	7
g. Variable controlada	7
h. Punto de consigna	7

i.	Error	7
j.	Perturbación	7
k.	Sistema de control en lazo abierto	7
l.	Sistema de control en lazo cerrado	8
m.	HMI.....	9
1.2.5.	Campo de aplicación de la automatización	9
a.	Automatización parcial	10
b.	Automatización total	10
1.2.6.	Concepto de la automatización y la robótica	10
1.3.	Robot industrial	11
1.3.1.	Definición de un robot industrial	11
1.3.2.	Objetivo del robot industrial	13
1.3.3.	Estructura de un robot industrial	14
1.3.4.	Similitud entre el brazo humano y un brazo robótico	16
1.3.5.	Funcionamiento de un brazo robótico	17
1.3.6.	Aplicación de los brazos robóticos	17
1.3.7.	Programación del brazo robótico	18
1.4.	Sensor.....	19
1.5.	Tipos de sensores	20
1.5.1.	Sensor capacitivo	20
1.5.2.	Sensor fotoeléctrico.....	21
a.	Captador por barrera.....	22
b.	Captador por reflexión.....	22
c.	Captador por auto-reflex	22
1.6.	Sensor ultrasónico.....	23
1.7.	Encoders	24
1.8.	Galgas extensiométricas	24
1.9.	Elementos mecánicos	25
1.9.1.	Tolvas.....	25
1.9.2.	Banda transportadora.....	26
1.9.3.	Mesa giratoria	27
1.10.	Servo-motores.....	28
1.10.1.	Funcionamiento de un servomotor	29

1.11.	Motor DC.....	29
1.12.	Arduino.....	31
1.13.	Modbus	33
1.13.1.	Modos de transmisión	33
a.	Modbus RTU	34
b.	Modbus ASCII	34
1.14.	Filtros	35
CAPÍTULO II.....		36
2.	DISEÑO, CONSTRUCCIÓN Y AUTOMATIZACIÓN DE LA PLANTA.....	36
2.1.	Descripción del módulo didáctico	36
2.1.1.	Etapa 1	37
2.1.2.	Etapa 2.....	38
2.1.3.	Etapa 3.....	39
2.1.4.	Etapa 4.....	40
2.1.5.	Etapa 5.....	41
2.1.6.	Monitoreo de la planta.....	42
2.2.	Diseño de accesorios mecánicos.....	42
2.2.1.	Estructura base del dispensador de recipientes.....	43
2.2.2.	Mesa giratoria	44
2.2.3.	Tolva	44
2.2.4.	Estructura base de la envasadora.....	45
2.2.5.	Contenedor de cajas	45
2.2.6.	Estructura base de la banda transportadora	46
2.2.7.	Estructura base del cajón.....	46
2.2.8.	Estructura frontal del panel de control.....	47
2.2.9.	Caja.....	48
2.2.10.	Estructura base del módulo didáctico.....	48
2.3.	Construcción del módulo didáctico.....	49
2.3.1.	Construcción del dispensador de recipientes.....	49
a.	Sensor capacitivo amico LJC30A3-H-Z/AY	49
b.	Sensor láser QS30LDL	52

c.	Pistón eléctrico.....	54
d.	Driver L298N	54
e.	Motor de alto torque	55
f.	Tarjeta controladora de motores DC	56
g.	Tarjeta arduino MEGA 2560.....	57
h.	Encoder incremental TRD-S360BD.....	58
2.3.2.	Construcción del dosificador de sólidos	60
a.	Motor vibratorio	60
b.	Sensor láser QS30LDL	62
c.	Sensor ultrasónico U-GAGE T30UIPAQ	62
d.	DBQ5 portable demo box.....	64
e.	Tarjeta arduino uno	64
f.	Balanza electrónica.....	65
g.	Módulo HX711	66
h.	Driver L298N	67
2.3.3.	Construcción de la envasadora.....	68
a.	Sensor capacitivo AMICO LJC30A3-H-Z/AY.....	68
b.	Bomba de vacío	69
c.	Servomotor tower pro mg 995.....	70
d.	Pistón eléctrico.....	71
e.	Ventosa o copa de succión	71
f.	Sensor fotoeléctrico banner mini beam SME312D.....	72
2.3.4.	Construcción de la etapa de organización	74
a.	Brazo robótico banshi R680	74
b.	Sensor banner SMU315D	75
c.	Sensor láser QS30LDL	75
d.	Tarjeta arduino MEGA 2560.....	76
e.	Motor de alto torque	76
f.	Sensor fotoeléctrico G30-3A70NA	77
2.3.5.	Construcción de la etapa 5 o de inspección.....	78
a.	Sensor fotoeléctrico QS18VP6R	78
b.	Brazo robótico Saintsmart.....	79
c.	Balanza	80

d.	Tarjeta Arduino Uno	80
2.4.	Automatización de la planta	81
2.4.1.	Arduino software (IDE)	82
a.	Archivo	83
b.	Editar.....	84
c.	Sketch	85
d.	Tools	85
2.4.2.	Referencia del lenguaje arduino.....	86
2.4.3.	Librerías	89
a.	Librería Adafruit Motor Shield V2.0.h	89
b.	Librería EFLM MASTER.h.....	90
c.	Librería Servo.h.....	93
d.	Librería Wire.h.....	94
e.	Librería Encoder.h.....	97
f.	Librería HX711.h	98
2.4.4.	Diseño del controlador difuso de peso	99
a.	Identificación de las variables del controlador difuso	101
b.	Fuzzificación de la variable de entrada de la Etapa 2	101
c.	Defuzzificación de la variable de salida de la Etapa 2	102
d.	Base de conocimientos	103
2.4.5.	Diagrama de flujo de módulo didáctico de llenado y envasado de sólidos	104
2.4.6.	Programación de la tarjeta Arduino Mega R3 2560 de la Etapa 1	106
a.	Diagrama de flujo de la Etapa 1	107
2.4.7.	Programación de la tarjeta Arduino Uno R3 de la Etapa 2.....	108
a.	Diagrama de flujo de la Etapa 2	109
2.4.8.	Programación de la tarjeta Arduino Uno R3 de la Etapa 3.....	110
a.	Diagrama de flujo de la Etapa 3	111
2.4.9.	Programación de la tarjeta Arduino Mega R3 2560 de la Etapa 4	112
a.	Diagrama de flujo de la Etapa 4	113
2.4.10.	Programación de la tarjeta Arduino Uno R3 de la Etapa 5.....	114

a.	Diagrama de flujo de la Etapa 5	115
2.4.11.	Diseño del hmi del módulo didáctico	116
a.	HMI del módulo didáctico	117
CAPÍTULO III.....		124
3.	PRUEBAS Y RESULTADOS.....	124
3.1.	Generalidades	124
3.2.	Funcionamiento y ajuste de la mesa giratoria	124
3.3.	Pruebas de ajuste del sensor capacitivo LJC30A3-H-Z/AY	125
3.4.	Ajuste sensor láser banner QS30LDL	126
3.5.	Pruebas de ajuste del sensor fotoeléctrico banner SME312D	127
3.6.	Pruebas de ajuste del sensor fotoeléctrico banner mini-beam SMU315D.....	128
3.7.	Pruebas de ajuste del sensor QS18VP6R (RX) y el sensor QS18SE (TX)	128
3.8.	Pruebas de ajuste sensor fotoeléctrico G30-3A70NA	129
3.9.	Pruebas de ajuste del sensor ultrasónico U-GAGE U30	130
3.10.	Calibración de las balanzas en la Etapa 2 y Etapa 5	131
3.11.	Pruebas de accionamiento de alarmas	134
3.12.	Pruebas de funcionamiento del control de peso aplicado en la Etapa 2.....	134
3.13.	Pruebas de calidad de la Etapa 5	140
3.14.	Pruebas de comunicación	144
3.15.	Diagrama de tiempos del módulo didáctico de llenado y envasado de sólidos.....	146
3.16.	Pruebas y funcionamiento del HMI.....	148
CAPÍTULO IV		150
4.	CONCLUSIONES Y RECOMENDACIONES.....	150
4.1.	Conclusiones.....	150
4.2.	Recomendaciones.....	152

REFERENCIAS BIBLIOGRÁFICAS	153
ANEXOS.....	157
ANEXO A: HOJAS DE DATOS	
ANEXO B: CIRCUITOS ESQUEMÁTICOS DE LAS TARJETAS ARDUINO	
ANEXO C: CIRCUITOS IMPRESOS	
ANEXO D: PROGRAMACIÓN DE LAS TARJETAS ARDUINO	

ÍNDICE DE FIGURAS

Figura 1.1: Robot de Soldadura como elemento operativo	3
Figura 1.2: Autómata Programable en el centro del sistema	4
Figura 1.3: Sistema de control en Lazo Abierto	8
Figura 1.4: Sistema de control de Lazo Cerrado.....	8
Figura 1.5: Pirámide de la Automatización	11
Figura 1.6: Robot Industrial.....	14
Figura 1.7: Partes de un manipulador robótico	14
Figura 1.8: Articulaciones de un robot: a) lineal, b) rotacional	15
Figura 1.9: Elementos terminales: a) Pinzas, b) Herramienta.....	16
Figura 1.10: Similitud entre el brazo humano y el robótico	16
Figura 1.11: Brazo robótico transportando objetos	18
Figura 1.12: Estructura básica de un sensor	19
Figura 1.13: Sensor capacitivo.....	21
Figura 1.14: Sensor fotoeléctrico	21
Figura 1.15: Sensor captador por barrera.....	22
Figura 1.16: Captador por reflexión	22
Figura 1.17: Captador por auto-reflex	23
Figura 1.18: Sensor ultrasónico	23
Figura 1.19: Encoder	24
Figura 1.20: Galga extensiométrica	25
Figura 1.21: Descarga por: a) Gravedad. b) Forzada.	26
Figura 1.22: Banda transportadora	27
Figura 1.23: Mesa giratoria	28
Figura 1.24: Partes de un servo-motor	28
Figura 1.25: Componentes de un motor DC	30
Figura 1.26: Tarjeta Arduino	31
Figura 1.27: Interfaz de programación en Arduino.....	32
Figura 2.1: Diagrama de bloques del módulo didáctico	36
Figura 2.2: Diagrama de bloques Etapa 1	37
Figura 2.3: Diagrama de bloques Etapa 2	38
Figura 2.4: Diagrama de bloques Etapa 3	39
Figura 2.5: Diagrama de bloques Etapa 4	40

Figura 2.35: Conexión del encoder TRD - S360BD	59
Figura 2.36: Posición del encoder en la mesa giratoria	59
Figura 2.37: Etapa 1	60
Figura 2.38: Motor vibratorio	61
Figura 2.39: Ubicación del motor vibrador	61
Figura 2.40: Sensor Láser QS30LDL. a) detectando el recipiente. b) Sin detección	62
Figura 2.41: Sensor U-GAGE T30UIPAQ	63
Figura 2.42: Distancia de sensado del sensor T30UIPAQ	63
Figura 2.43: Ubicación del sensor ultrasónico U-GAGE T30UIPAQ	63
Figura 2.44: Módulo DBQ5 Portable Demo Box	64
Figura 2.45: Tarjeta Arduino/Genuino Uno	65
Figura 2.46: Balanza electrónica	65
Figura 2.47: Celda de carga de 200 gramos	66
Figura 2.48: Ubicación de la balanza	66
Figura 2.49: Módulo HX711	67
Figura 2.50: Etapa 2	68
Figura 2.51: Ubicación del sensor capacitivo AMICO LJC30A3-H-Z/AY	68
Figura 2.52: Bomba de vacío AIRPRO D2026B	69
Figura 2.53: Ubicación en la envasadora de la bomba de vacío	69
Figura 2.54: Servomotor de engranes metálicos MG995	70
Figura 2.55: Ubicación del pistón eléctrico	71
Figura 2.56: Copa de succión o ventosa	72
Figura 2.57: Sensor fotoeléctrico banner mini beam SME312D	72
Figura 2.58: Ubicación del sensor banner mini beam SME312D	73
Figura 2.59: Etapa 3	73
Figura 2.60: Brazo robótico banshi R680	75
Figura 2.61: Sensor fotoeléctrico banner SMU315D	75
Figura 2.62: Sensor QS30LDL a) sin detección de caja. b) con detección de caja	76
Figura 2.63: Motor de alto torque Etapa 4	76
Figura 2.64: Motor DC acoplado a la banda transportadora	77

Figura 2.91: Configuración de la comunicación	119
Figura 2.92: Creación de grupos en cada etapa	119
Figura 2.93: Creación de las variables del HMI	120
Figura 2.94: Variables de las etapas de módulo diáctico	120
Figura 2.95: Nuevo formulario para el diseño del HMI	121
Figura 2.96: Formularios del HMI.....	121
Figura 2.97: Formulario PRINCIPAL.....	122
Figura 2.98: Formulario PROCESO.....	122
Figura 2.99: Formulario TENDENCIAS_FUZZY	123
Figura 3.1: Prueba del posicionamiento inicial de la mesa giratoria	124
Figura 3.2: Ajuste del sensor capacitivo de la Etapa 1 con la existencia de material en el contenedor	125
Figura 3.3: Ajuste del sensor capacitivo con la no existencia de material en el contenedor.....	126
Figura 3.4: Procedimiento de ajuste del sensor laser QS30LDL para una ventana de dos puntos	127
Figura 3.5: Posición en línea de vista del sensor QS18VP6R (RX) y el sensor QS18SE (TX).....	129
Figura 3.6: Ajuste del sensor fotoeléctrico en una ventana de un solo punto.....	129
Figura 3.7: U-GAGE T30 pulsadores de programación del sensor e indicadores	130
Figura 3.8: Pesos patrón.....	131
Figura 3.9: Ubicación de pesos patrón en la balanza de la Etapa 2	132
Figura 3.10: Ubicación de pesos patrón en la balanza de la Etapa 5	132
Figura 3.11: Número de calibración obtenido para la balanza de la Etapa 2	133
Figura 3.12: Número de calibración obtenido para la balanza de la Etapa 5	133
Figura 3.13: Accionamiento de alarmas del módulo didáctico de llenado y envasado de sólidos	134
Figura 3.14: Gráfica de funcionamiento del dispensador de sólidos con un set point de 5 gramos	137

Figura 3.15: Gráfica de funcionamiento del dispensador de sólidos con un set point de 5 gramos para 3 muestras continuas	137
Figura 3.16: Gráfica de funcionamiento del dispensador de sólidos con un set point de 11 gramos	138
Figura 3.17: Gráfica de funcionamiento del dispensador de sólidos con un set point de 11 gramos para 3 muestras continuas	138
Figura 3.18: Gráfica de funcionamiento del dispensador de sólidos con un set point de 18 gramos	139
Figura 3.19: Gráfica de funcionamiento del dispensador de sólidos con un set point de 18 gramos para 3 muestras continuas	139
Figura 3.20: Aprobación por parte de la etapa de inspección del producto final	143
Figura 3.21: Rechazo por parte de la etapa de inspección del producto final	144
Figura 3.22: Asignación de puertos COM para las tarjetas Arduino	144
Figura 3.23: Estado de la conexión de la tarjeta Arduino en Acimut Monitoriza for Arduino.....	145
Figura 3.24: Comunicación errónea.....	146
Figura 3.25: Diagrama de tiempos del funcionamiento del Módulo didáctico de llenado y envasado de sólidos.....	147
Figura 3.26: UpdarateRate por defecto.....	148
Figura 3.27: UpdateRate modificado a 1ms.....	149

ÍNDICE DE TABLAS

Tabla 1.1: Diferencias entre Modbus RTU Y Modbus ASCII	34
Tabla 2.1: Constantes dieléctricas de algunos materiales	52
Tabla 2.2: Especificaciones del motor de alto torque 12VDC	56
Tabla 2.3: Especificaciones de la tarjeta controladora de motores DC	57
Tabla 2.4: Especificaciones del motor vibratorio	61
Tabla 2.5: Botones de la barra de herramientas Arduino	82
Tabla 2.6: Estructura, valores y funciones utilizadas en Arduino	87
Tabla 2.7: Ubicación de pines TWI en distintas tarjetas Arduino	95
Tabla 2.8: Pines de interrupción tarjetas Arduino	97
Tabla 2.9: Conjunto de reglas de inferencia	104
Tabla 3.1: Comparativa de valores calculados y valores medidos en el dispensador de sólidos de la Etapa 2	135
Tabla 3.2: Taras o pesos de los recipientes empleados en el producto final	140
Tabla 3.3: Valores medidos del producto final del módulo didáctico de llenado y envasado de sólidos	142

RESUMEN

El módulo didáctico de un sistema de llenado y envasado de sólidos se encuentra conformado principalmente por cinco etapas controladas mediante tarjetas Arduino. La primera etapa del módulo didáctico dispensa recipientes plásticos y los distribuye en una mesa giratoria dividida en 8 partes iguales. La segunda etapa del módulo didáctico se encarga del dispensado del material sólido contenido en la tolva sobre el recipiente plástico. La tercera etapa del módulo didáctico realiza el envasado de los recipientes anteriormente llenados, denominada también como etapa de envasado. La cuarta etapa se encarga de organizar los recipientes envasados con la ayuda del movimiento de un brazo robótico en cajas con 4 compartimentos movilizadas a través de una banda transportadora. La quinta etapa o etapa de inspección se encarga de verificar y clasificar el producto final en la respectiva ruta final de transporte, donde el operador tiene que retirar el producto final para ser enviado a su destino o rechazar los productos que no hayan pasado dicha inspección (rechazo). Las etapas anteriormente mencionadas son monitoreadas mediante un HMI (Human Machine Interface) desarrollado en un software libre (Monitoriza for Arduino-Scada Acimut), para observar y verificar el correcto funcionamiento de cada una de las etapas del módulo didáctico.

PALABRAS CLAVE:

- **AUTOMATIZACIÓN-ENVASADO**
- **SOFTWARE LIBRE-ARDUINO**
- **CONTROL DIFUSO**
- **MÓDULO DIDÁCTICO**

ABSTRACT

The didactic module of filling and packaging of solids system is mainly made up by five stages controlled by Arduino cards. The first stage of the didactic module dispenses and distributes plastic containers on a turntable divided into 8 equal parts. The second stage of the didactic module is responsible for dispensing the hopper's contents over the plastic container of solid material. The third stage of the didactic module performs the packaging of the containers previously filled, named also as stage of packaging. The fourth stage is responsible for organizing containers packed in boxes with 4 subdivisions mobilized through a conveyor belt with the help of a robotic arm. The fifth stage is the inspection; is responsible for verifying and classifying the final product in the right route of final transport where the operator has to remove the finished product to be sent to its destination or reject it if it has not passed the inspection (rejection). The stages previously mentioned are monitored by means of a HMI (Human Machine Interface) developed in a free software (Monitoriza for Arduino-Scada Acimut), to observe and verify the proper operation of each one of the stages of the didactic module.

KEYWORDS:

- **AUTOMATION-PACKING**
- **FREE SOFTWARE-ARDUINO**
- **FUZZY CONTROL**
- **DIDACTIC MODULE**

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. Antecedentes

Hoy en día la tendencia de las industrias se ha enfocado en la automatización y control de procesos para el mejoramiento de la calidad de su producción, debido a la variedad de sensores eléctricos o mecánicos que existen actualmente en el mercado que permiten convertir magnitudes físicas o químicas en valores medibles de dicha magnitud, suministrando información de algún proceso existente en una planta.

El impacto de automatizar una planta de procesos permite manejar con mayor facilidad cada área o bloque en donde se realizan las manufacturas para el producto final, además recorta el consumo de energía, reduce costos y su mantenimiento es mucho más sencillo, lo que es fundamental para la operación eficiente y segura de una planta de procesos.

El laboratorio de Redes Digitales y Control de Procesos de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga cuenta con una gran variedad de sensores capacitivos, fotoeléctricos, ultrasónicos y encoders a nivel industrial, además la utilización de brazos robóticos y galgas extensiométricas para el aprendizaje de los estudiantes.

En consecuencia, se aplica dichos sensores en el diseño y construcción de un módulo didáctico de un sistema de automatización de llenado y envasado de sólidos con los sensores anteriormente mencionados para el desarrollo de conocimientos en los alumnos.

1.2. Automatización y control

La automatización y control es una rama de la ingeniería que integra tecnologías de vanguardia que se utiliza en el campo de la automatización y el control automático industrial, complementando áreas disciplinarias como los sistemas de control y supervisión de datos.

La instrumentación industrial y las redes de comunicación industrial generando proyectos de procesos los cuales maximizan los estándares de productividad y preservando la integridad del talento humano quienes realizan las operaciones necesarias para su funcionamiento. Existen dos conceptos fundamentales que implican la automatización y control de un sistema.

1.2.1. Automatización de un sistema

Se lo define como un sistema capaz de ejecutar acciones previamente establecidas en un espacio y tiempo determinado sin la necesidad de la intervención humana frente a ambientes agresivos y hostiles, mejorando la cadencia y control de la producción.

La automatización se hace posible mediante los Sistemas de Control, mejorando el rendimiento en los procesos repetitivos (clasificación de objetos, semáforos, apertura y cierre de puertas, etc.), realizando tareas que implican desgaste físico importante en el ser humano y controlando procesos difícilmente controlables de forma manual. Un sistema automatizado consta de dos partes principales [1]:

- Parte Operativa.
- Parte de Mando.

a. Parte operativa

La parte operativa es la parte que actúa directamente sobre la máquina.

Son los elementos que permiten que la máquina se mueva y realice la operación deseada. Los elementos que forman la parte operativa son los accionadores de las máquinas como motores, cilindros de accionamiento, etc., y los captadores como fotodiodos, finales de carrera, entre otros (Figura 1.1).

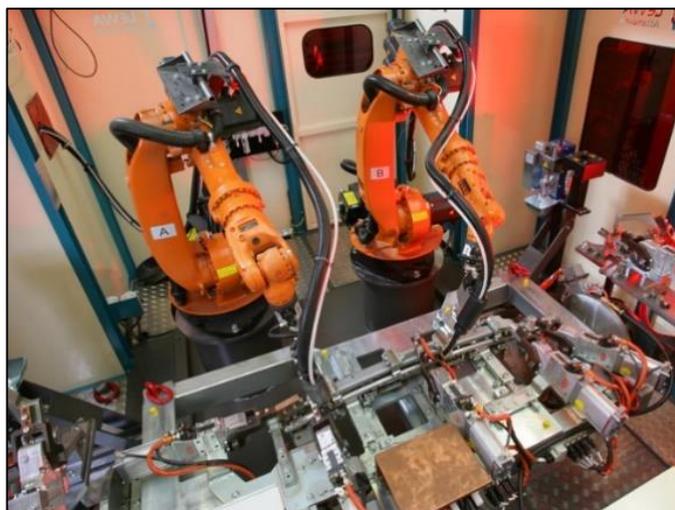


Figura 1.1: Robot de Soldadura como elemento operativo
Fuente: [2]

b. Parte de mando

En su mayoría suele ser un autómata programable (tecnología programada), aunque hasta hace poco se utilizaban relés electromagnéticos, tarjetas electrónicas o módulos lógicos (tecnología cableada). En un sistema de fabricación automatizado el autómata programable está en el centro del sistema (Figura 1.2), capaz de comunicarse con todos los constituyentes del sistema realizando operaciones específicas de forma secuencial.

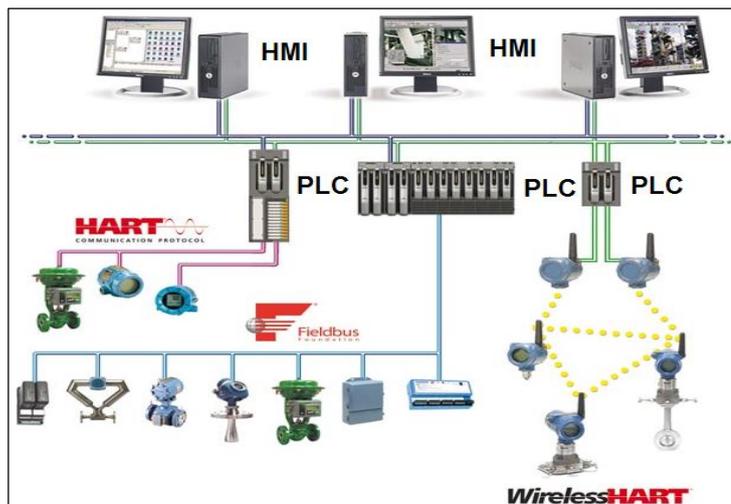


Figura 1.2: Autómata Programable en el centro del sistema
Fuente: [3]

1.2.2. Objetivos de la automatización

La automatización es un sistema que transfiere tareas de control y producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos y esto genera objetivos específicos que se detalla a continuación [4] :

- Mejora la productividad de la empresa, reduciendo los costos de la producción y mejorando la calidad de la misma.
- Mejora las condiciones de trabajo del personal, suprimiendo los trabajos penosos e incrementando la seguridad.
- Realiza las operaciones imposibles de controlar, intelectual o manualmente.
- Mejora la disponibilidad de los productos, pudiendo proveer las cantidades necesarias en el momento preciso.
- Simplifica el mantenimiento de forma que el operario no requiera grandes conocimientos para la manipulación del proceso productivo.
- Integra la gestión y la producción.

1.2.3. Sistema de control

Un sistema de control es un tipo de sistema que se caracteriza por una serie de elementos que influyen en el funcionamiento de un proceso. La finalidad de un sistema de control es conseguir, mediante la manipulación de las variables de control, un dominio sobre las variables de salida, de este modo alcanzan los valores prefijados o puntos de consigna.

Un sistema de control debe ser capaz de conseguir su objetivo cumpliendo con los siguientes puntos:

- Garantizar la estabilidad, ser robusto frente a perturbaciones y errores en los procesos.
- Ser eficiente de la mejor manera posible, según un criterio establecido, normalmente este criterio consiste en que la acción de control sobre las variables de entrada sea realizable, evitando comportamientos bruscos e irreales.
- Ser fácilmente implementable y cómodo de operar en tiempo real con la ayuda de un ordenador capaz de operar de manera efectiva.

El principio de todo sistema de control automático aplica el concepto de realimentación o feedback (medición tomada desde el proceso que entrega toda la información del estado actual de la variable que se desea controlar) cuya característica principal es la de mantener al controlador central informado del estado de las variables para generar todas las acciones correctivas necesarias en el proceso cuando así sea necesario [5].

1.2.4. Definiciones básicas de un sistema de control

a. Sistema

Es la combinación de componentes que actúan conjuntamente y cumplen un determinado objetivo.

b. Proceso

Es el desarrollo natural de un acontecimiento, caracterizado por una serie de eventos o cambios graduales, progresivamente continuos y que entienden a un resultado final.

c. Planta

Conjunto de piezas de una maquinaria que tienen por objetivo realizar cierta actividad en conjunto. En sistemas de control, por planta se entiende el sistema que se quiere controlar.

d. Actuador

Un 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). Dependiendo del origen de la fuerza el actuador se denomina “neumático”, “hidráulico” o “eléctrico”.

e. Transductor

Los transductores son elementos que transforman una magnitud física en una señal eléctrica, se pueden clasificar en dos grupos: Activos y pasivos. Los transductores *activos* son aquellos que generan corriente eléctrica o voltaje en respuesta directa a una estimulación del ambiente (magnitud física), ejemplo: Las termocuplas producen un voltaje relacionado a la temperatura de dos metales; Los transductores *pasivos* producen un cambio en magnitud de alguna propiedad eléctrica pasiva, como la capacitancia, resistencia o inductancia; como resultado de una estimulación. Estos generalmente requieren de energía eléctrica adicional para excitación, ejemplo: La galga extensiométrica varía su resistencia relacionado a una fuerza mecánica [6].

f. Variable manipulada

Es una variable del sistema tal que una modificación de su magnitud o condición puede alterar el estado del sistema (afecta directamente a la *variable controlada*), es decir, a través del cual se deben corregir las perturbaciones del proceso, colocando un actuador para estabilizar el sistema.

g. Variable controlada

Es una variable del sistema cuya magnitud o condición se mide y controla, debiéndose mantenerse estable (sin cambios), pues su variación alteraría las condiciones requeridas en el sistema [7].

h. Punto de consigna

Conocido como Set Point o punto de referencia, es la señal de entrada que especifica al sistema de control el valor deseado de la variable controlada.

i. Error

El error es la diferencia entre el punto de consigna y la variable controlada.

j. Perturbación

Una perturbación es algún suceso que afecta adversamente el desarrollo de algún proceso. Si la perturbación se genera dentro del sistema, se lo denomina *perturbación interna*, caso contrario la perturbación es *externa*.

k. Sistema de control en lazo abierto

Un sistema de control en lazo abierto son aquellos en los que la variable de salida del sistema (variable controlada) no tiene efecto sobre la acción de control (variable de control) como se muestra en la Figura 1.3.

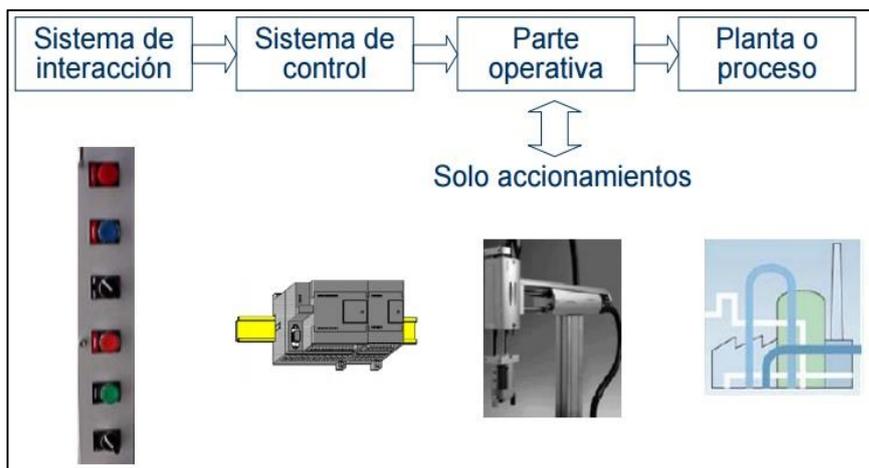


Figura 1.3: Sistema de control en Lazo Abierto
Fuente: [8]

I. Sistema de control en lazo cerrado

Un sistema de control en lazo cerrado (Figura 1.4) es aquel en donde la señal de salida del sistema (variable controlada) tiene efecto directo sobre la acción de control (variable de control).

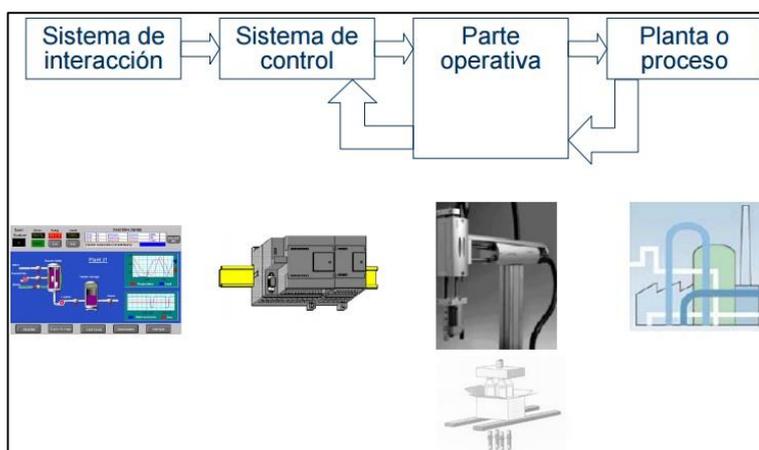


Figura 1.4: Sistema de control de Lazo Cerrado
Fuente: [8]

m. HMI

Un HMI (Human-Machine Interface por sus siglas en inglés) es una Interfaz Humano-Máquina, un panel de control diseñado para conseguir una comunicación interactiva entre operador y proceso/máquina, con la función de transmitir órdenes, visualizar gráficamente los resultados y obtener una situación del proceso/máquina en tiempo real. Tradicionalmente estos sistemas consistían en paneles compuestos por indicadores digitales y análogos, registradores, pulsadores, selectores y otros que se interconectaban con la máquina o proceso. En la actualidad, dado que las máquinas y procesos en general están implementadas con controladores y otros dispositivos electrónicos que dejan disponibles puertos de comunicación, es posible contar con sistemas de HMI más eficaces, además de permitir una conexión más sencilla y económica con el proceso o máquinas [9].

1.2.5. Campo de aplicación de la automatización

Se utiliza a gran medida en el campo de la producción industrial, desde los inicios de la era industrial hasta la actualidad, la automatización ha pasado a ser una herramienta de trabajo deseable a una herramienta indispensable para competir en el mundo globalizado debido a los dispositivos programables que permiten un control prácticamente absoluto en un proceso.

Ningún empresario puede omitir la automatización de sus procesos para aumentar la calidad de sus productos, reducir los tiempos de producción, realizar tareas complejas, reducir los desperdicios de las piezas mal fabricados y especialmente aumentar la rentabilidad. Para ello el empresario definirá si su automatización es parcial o total.

a. Automatización parcial

La maquinaria realiza varias operaciones en secuencia y de forma autónoma, pero necesita de la intervención humana para poner y retirar las piezas o productos.

b. Automatización total

La maquinaria es totalmente autónoma, no necesita intervención humana. El operador realiza tareas de supervisión y mantenimiento preventivo.

1.2.6. Concepto de la automatización y la robótica

Desde el punto de vista de las empresas, la automatización y la robótica es un área en pleno desarrollo en todos los sectores industriales, ya que permite mejorar la productividad de las empresas, la calidad de sus productos y la seguridad laboral, realizando operaciones de forma rápida y precisa, simplificando el mantenimiento de la instalación y controlando los procesos en tiempo real.

El problema básico del control automático de procesos es decidir qué actuaciones se deben realizar para conseguir que se cumplan los requisitos prefijados en base a realimentar el estado de los procesos (concepto de realimentación o “feedback”) [10]. En la actualidad la automatización puede describirse bajo el esquema piramidal multi-nivel como se puede apreciar en la Figura 1.5.

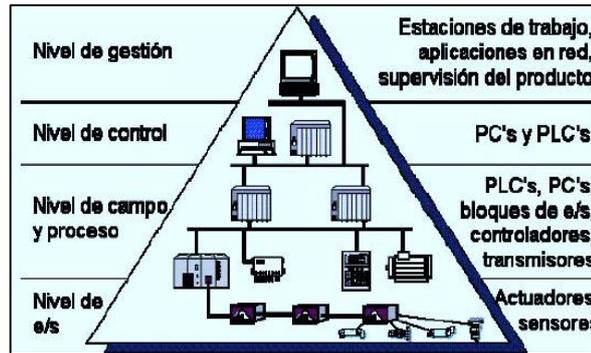


Figura 1.5: Pirámide de la Automatización
Fuente: [10]

Cuanto más alto se sitúa el nivel de la pirámide de automatización más se acerca a operaciones que, hasta hace poco tiempo, sólo se podían imaginar cómo tareas confiadas a operadores humanos, y por lo tanto reclaman la ayuda de Inteligencia Artificial. Por otra parte, una característica distintiva de los sistemas automáticos es la necesidad de ofrecer prestaciones en tiempo real, y por lo tanto de situarse en el campo de la informática avanzada y la microelectrónica.

1.3. Robot industrial

1.3.1. Definición de un robot industrial

La definición más comúnmente aceptada posiblemente sea la de la Asociación de Industrias de Robótica (RIA, *Robotic Industry Association*), según la cual:

“Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas”.

Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO) que define al robot industrial como:

“Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas”.

Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad. Una definición más completa es la establecida por la Asociación Francesa de Normalización (AFNOR), que define primero el manipulador y, luego basándose en dicha definición, el robot:

“Manipulador es el mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante un dispositivo lógico”.

“Robot es un manipulador automático servo-controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, adaptándose a otra sin cambios permanentes en su material”.

Por último, la Federación Internacional de Robótica (IFR, *International Federation of Robotics*) distingue entre robot industrial de manipulación y otros robots como:

“Por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento” [11].

1.3.2. Objetivo del robot industrial

El robot industrial que se conoce y utiliza en nuestros días nace con el objetivo de aumentar la productividad, mejorar la calidad de las piezas o productos fabricados y evitar la realización de trabajos tediosos o peligrosos para el hombre.

George Devol es considerado el padre de la Robótica Industrial, y a él se debe la patente de un manipulador programable en 1956, que fue el propulsor del robot industrial. A partir de ahí se fueron instalando en las fábricas cada vez más robots dedicados a diversas funciones, gracias entre otras cosas a los progresos en Electrónica e Informática, que instauraron el microprocesador para el manejo de estas máquinas de un modo potente, flexible y económico. El número de robots industriales existentes actualmente en el mundo supera con creces el millón de unidades.

Un robot industrial se suele caracterizar por sus grandes dimensiones y su posicionamiento fijo (Figura 1.6). Se distingue en él un manipulador o brazo mecánico al que se acoplan las diversas herramientas según la tarea, y es controlado por un sistema basado en un procesador. Los más sofisticados poseen sensores para detectar parámetros del entorno y alterar su funcionamiento en función de la información recibida, aunque éste no es un factor determinante. Trabajos que hasta hace poco era inconcebible que los realizase una máquina, hoy los efectúa un robot con rapidez y precisión. La industria que más robots aglutina es la del automóvil [12].



Figura 1.6: Robot Industrial
Fuente: [12]

1.3.3. Estructura de un robot industrial

Un manipulador robótico consta de una serie o secuencia de elementos estructurales rígidos, denominados enlaces o eslabones, conectados entre sí mediante juntas o articulaciones, que permiten el movimiento relativo de cada dos eslabones consecutivos (Figura 1.7).



Figura 1.7: Partes de un manipulador robótico
Fuente: [13]

Una articulación puede ser:

- Lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior
- Rotacional, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior (Figura 1.8).

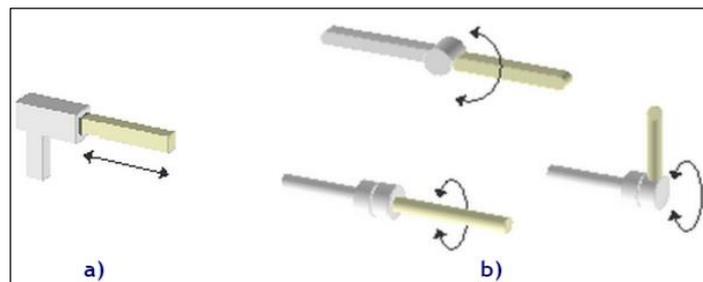


Figura 1.8: Articulaciones de un robot: a) lineal, b) rotacional

Fuente: [13]

La unión o conjunto de eslabones y articulaciones se denomina cadena cinemática, debido a que cada eslabón se conecta mediante articulaciones exclusivamente al anterior y al siguiente, exceptuando el primero, que se suele fijar al soporte, y el último, cuyo extremo final queda libre. Finalmente se une el *elemento terminal* o *actuador final*, definiéndose como la herramienta especial que permite al robot de uso general realizar una aplicación en particular.

Los elementos terminales se dividen en dos categorías (Figura 1.9):

- Pinzas, se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot.
- Herramienta, es el actuador final en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza de trabajo. Estas aplicaciones aplican soldadura por puntos, la soldadura por arco, etc. En cada caso, la herramienta particular está unida a la muñeca del robot para realizar la operación.

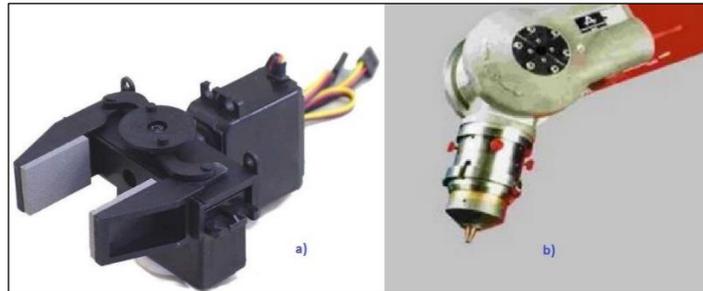


Figura 1.9: Elementos terminales: a) Pincas, b) Herramienta
Fuente: [13]

1.3.4. Similitud entre el brazo humano y un brazo robótico

El brazo robótico ha sido diseñado en base a la gran versatilidad del brazo humano, en un principio los diseños fueron rudimentarios pero con el tiempo se logró igualar, incluso, superar la complejidad de los movimientos (Figura 1.10).

El brazo robótico es de gran importancia en la industria no solo por su eficacia al realizar trabajos, sino porque sustituye al hombre en labores de producción que no corresponden a un humano, es decir, mejorar la calidad de vida. En otras palabras, el brazo robótico reemplaza la actividad.

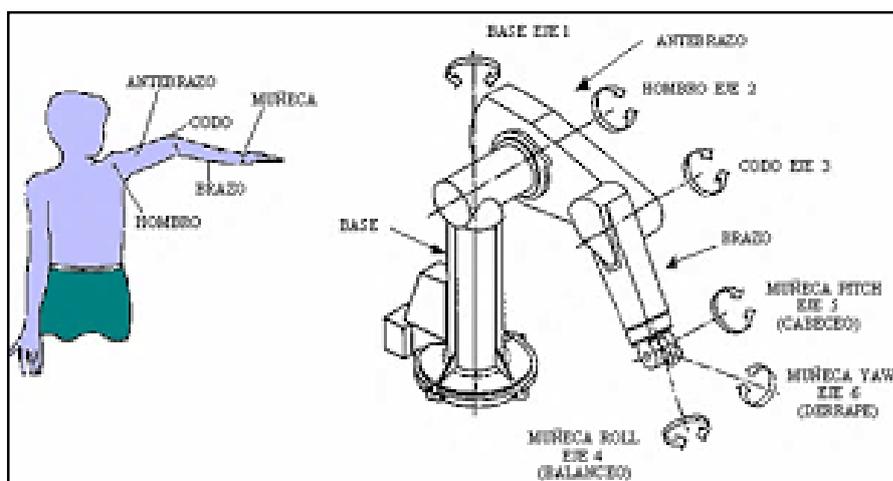


Figura 1.10: Similitud entre el brazo humano y el robótico

1.3.5. Funcionamiento de un brazo robótico

El funcionamiento básico de los brazos robóticos se realiza a través de sensores de presión que están integrados, indicando en el ordenador la presión ejercida a un objeto determinado, impidiendo al robot que deje caer o rompa lo que está transportando de un lugar hacia otro.

Todo está manejado mediante un entorno controlado con respecto al brazo robótico. Por ejemplo, en las cadenas de producción se usan para realizar una tarea concreta cuando un producto llega a ese punto de la cadena, como puede ser poner una tapa a un envase o atornillar un objeto.

Para enseñarle a realizar su trabajo, un programador debe guiar el brazo a través de los movimientos. El robot almacena la secuencia exacta de movimiento en su memoria, y lo hace una y otra vez cuando es necesario [14].

1.3.6. Aplicación de los brazos robóticos

En la mayoría de los robots industriales, son utilizados tanto en líneas de ensamblaje, transporte de químicos peligrosos, transporte de objetos (Figura 1.11), soldadura por puntos, transporte de carga, montaje de aparatos electrónicos, con el propósito de que su trabajo sea de una forma más eficiente en comparación de los humanos, con una gran exactitud y una gran fiabilidad en las áreas programadas al robot.

Además, este uso extensivo de los robots se ha visto propiciado por la buena adaptación del robot industrial a las tareas repetitivas en entornos estructurados, su rápida adaptación a series cortas, sus buenas características de precisión y rapidez, y por su posible reutilización con costos inferiores a otros sistemas permite que el robot sea competitivo frente a otras soluciones de la automatización.



Figura 1.11: Brazo robótico transportando objetos

1.3.7. Programación del brazo robótico

Al momento de programar un manipulador lo que se necesita usualmente es situar su punto terminal en una localización determinada del espacio, además que la dirección de aproximación a la misma esté también definida. Pero no sólo es importante que el manipulador alcance determinados puntos del espacio, sino que lo realice en el momento adecuado. Así se puede distinguir entre: *Camino*, sucesión de puntos del espacio, mientras que *trayectoria* es el camino sujeto a condiciones temporales [15].

Generalmente, es necesario diseñar y programar trayectorias, rectas o curvas arbitrarias en el plano cartesiano, en donde es importante conocer la relación entre la trayectoria cartesiana del punto terminal y la de las articulaciones. En el diseño de trayectorias se toma en cuenta los siguientes puntos:

- La prevención de posibles colisiones con objetos del entorno.
- El mantenimiento de una orientación fija del elemento terminal (transporte de sólidos, líquidos, soldadura, etc.).
- La utilización de aceleraciones moderadas, de modo que se eviten fuerzas de inercia elevadas que resulten inadecuadas para motores.

Existen determinadas circunstancias que los lenguajes de programación de robots resulten relativamente diferentes al resto:

- El entorno en el que actúa el robot no puede describirse normalmente en términos puramente cuantitativos.
- Se necesita incluir las condiciones no usuales, como la comentada prevención de colisiones.
- Las acciones del robot están sujetas a imprecisiones que pueden dar lugar a incidentes que el programa debe ser capaz de procesar.
- Ciertas informaciones de los sensores del sistema.

1.4. Sensor

La medida de un fenómeno físico, como la temperatura de un proceso, la intensidad de una fuente de luz o la fuerza aplicada a ciertos objetos, comienza con un sensor (Figura 1.12). Un sensor, también llamado un transductor, transforma un fenómeno físico en una señal eléctrica que se puede medir.

Dependiendo del tipo de sensor a utilizar, su salida eléctrica puede ser un voltaje, corriente, resistencia u otro atributo eléctrico que varía con el tiempo. Algunos sensores pueden requerir componentes adicionales y circuitos para producir correctamente una señal que puede ser leída con una alta precisión.

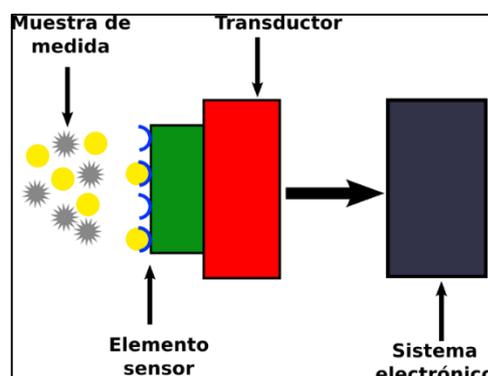


Figura 1.12: Estructura básica de un sensor

1.5. Tipos de sensores

Existe una gran variedad de sensores utilizados para captar las variables físicas de los diferentes procesos, los cuales permiten manejar de mejor manera perturbaciones o eventos que pueden ocurrir en el proceso y por ende afectar a la planta. Los sensores más utilizados a nivel industrial son los siguientes:

1.5.1. Sensor capacitivo

Están diseñados para detectar materiales aislantes tales como el plástico, el papel, la madera, entre otros, no obstante, también cuentan con la capacidad de detectar metales. Es importante tener en cuenta que los sensores capacitivos funcionan de manera inversa a los inductivos (Figura 1.13), es decir que a medida que el objetivo se va acercando al sensor las oscilaciones del mismo aumentan hasta que llega a un límite que activa el circuito que dispara las alarmas. En un principio éstos constan de una sonda que se encuentra situada en la cara posterior en donde se encuentra colocada una placa condensadora, y al aplicar una corriente al sensor por más mínima que sea, se produce una especie de campo electroestático cuya reacción se produce frente a los cambios de la capacitancia provocados por la presencia de un objeto cualquiera. En el caso de que el objeto se encuentre fuera del campo electroestático entonces el oscilador de los sensores capacitivos se encontrará inactivo, pero a medida que el objeto se va acercando al sensor, este se activa.

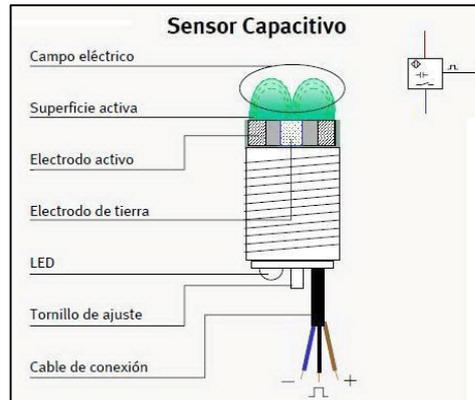


Figura 1.13: Sensor capacitivo

1.5.2. Sensor fotoeléctrico

Es un dispositivo electrónico que responde al cambio de luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor (Figura 1.14). El principio de funcionamiento de los detectores fotoeléctricos está basado en la generación de un haz luminoso por parte de un foto-emisor, proyectándose sobre un foto-receptor, o bien sobre un dispositivo reflectante. La interrupción o reflexión del haz por parte del objeto a detectar, provoca el cambio de estado de la fotocélula.

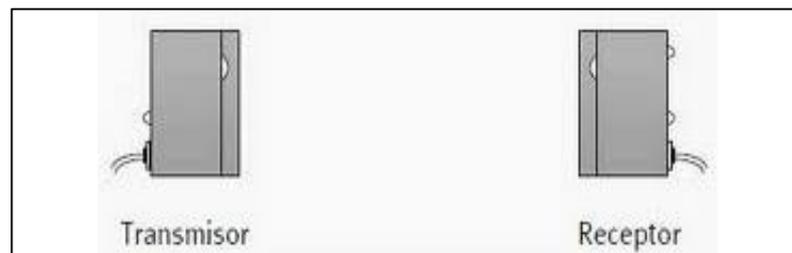


Figura 1.14: Sensor fotoeléctrico

Según la forma en que se produzca esta emisión y detección de luz divide este tipo de captadores en: captadores por barrera, por reflexión y auto-reflexión.

a. Captador por barrera

Un captador de barrera funciona cuando el emisor y el receptor se encuentran en diferentes contenedores y es necesario alinearlos con precisión; cuando el objeto atraviesa el haz de luz es cuando se activa el sensor (Figura 1.15). Se utiliza para grandes distancias (hasta 60m).

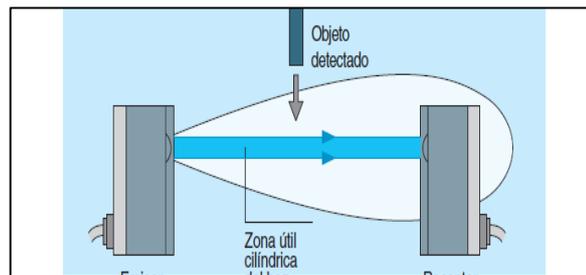


Figura 1.15: Sensor captador por barrera

b. Captador por réflex

Tiene el emisor y el receptor alojado en el mismo contenedor, el cual es necesario alinear con un espejo reflector (Figura 1.16). Se utilizan para distancias medias (hasta 15m). Dispone de una menor distancia de detección que el sistema de barrera, teniendo en cuenta que el trayecto que recorre el haz es doble.

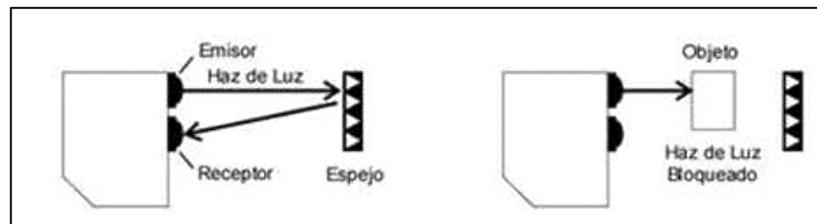


Figura 1.16: Captador por réflex

c. Captador por auto-réflex

Posee el mismo funcionamiento al captador de tipo réflex, no siendo necesario el espejo reflector (Figura 1.17). El propio objeto a detectar es el

encargado de reflejar el haz luminoso. Se utilizan para cortas distancias (entre 1 y 10 cm).

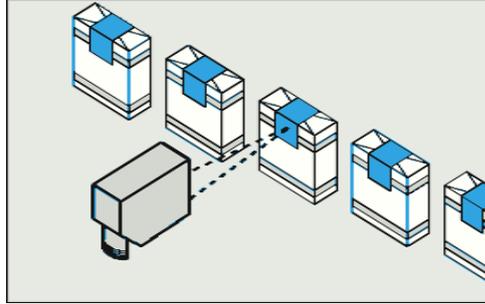


Figura 1.17: Captador por auto-réflex

1.6. Sensor ultrasónico

Su función es la de modificación de la distancia de objetos mediante la detección de ecos de ultrasonidos. Las ondas ultrasónicas tienen la capacidad que cuando viajan por un medio cualquiera son reflejadas, si encuentran en su camino una discontinuidad o algún elemento extraño (Figura 1.18). La reflexión de la onda es debido a la diferencia de impedancias acústicas entre el medio y el objeto. El tiempo de espera entre el envío de la onda ultrasónica hasta su recepción se denomina tiempo de eco, y es utilizado para determinar la distancia al objeto.



Figura 1.18: Sensor ultrasónico

1.7. Encoders

Constan de un disco transparente con una serie de marcas opacas colocadas radialmente y equidistantes entre sí, de un sistema de iluminación y de un elemento foto-receptor (Figura 1.19). El eje cuya posición se quiere medir va acoplado al disco, a medida que el eje gira se van generando pulsos en el receptor cada vez que la luz atraviese las marcas, llevando una cuenta de estos pulsos es posible conocer la posición del eje. La resolución depende del número de marcas que se pueden poner físicamente en el disco.

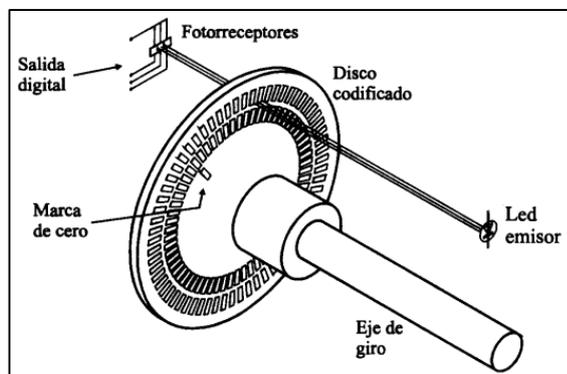


Figura 1.19: Encoder

1.8. Galgas extensiométricas

Conocido también como un extensómetro o “strain gage” (sus siglas en inglés) y es un dispositivo de medición electrónica de las magnitudes mecánicas como la carga, presión, deformación, posición, etc. Las galgas extensiométricas es un sensor cuya resistencia varía mediante una fuerza externa aplicada, produciéndose una tensión y estrés sobre la galga (Figura 1.20). El estrés se define como las fuerzas internas de resistencia del objeto, y la tensión se define como el desplazamiento y la deformación que se producen.

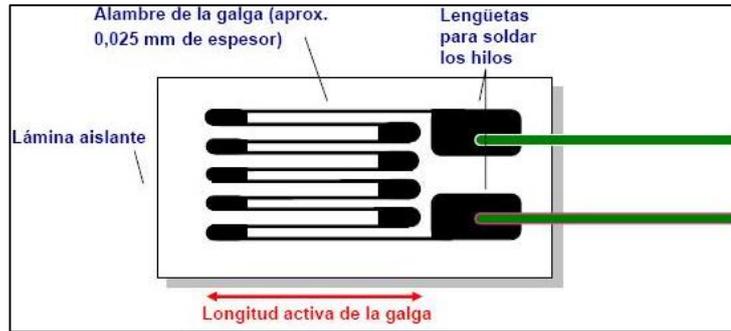


Figura 1.20: Galga extensiométrica
Fuente: [16]

La galga extensiométrica es una herramienta importante en la técnica aplicada de medición eléctrica de magnitudes mecánicas. Como su nombre indica, se utiliza para la medición de tensiones. "Tensión" como término técnico consiste en la deformación por tracción y compresión, que se distingue por un signo positivo o negativo. Por lo tanto, las galgas extensiométricas se pueden utilizar para medir la expansión y la contracción [16].

1.9. Elementos mecánicos

1.9.1. Tolvas

Recipientes fijos usados para contener material a granel o piezas sueltas de poco tamaño. Son diseñadas especialmente para ser vaciadas por el fondo mediante válvulas de construcción especial o alimentadores de diversos tipos.

Generalmente son de sección cuadrada, rectangular, o circular con el fondo en forma de pirámide o cono truncado invertido. Se pueden encontrar abiertos o cerrados en el tope, siendo la mayor aplicación aquellos abiertos que pueden contener material (piedra, granzón, material de construcción, etc.) que no es deteriorado ni puede sufrir alteraciones ni cambios en sus características por efectos de las condiciones ambientales. Muy utilizados en la industria de la construcción.

El almacenamiento en silos, es decir la carga de los mismos, se hace normalmente por el centro, mediante un sistema de elevación del producto, compuesto por un transportador inyector, otro transportador que sube el producto de forma vertical y el transportador repartidor. La descarga de los silos y tolvas de almacenamiento, se efectuara por gravedad o forzada las cuales utilizan tajaderas herméticas neumáticas o motorizadas (Figura 1.21).



Figura 1.21: Descarga por: a) Gravedad. b) Forzada.

1.9.2. Banda transportadora

Las bandas y rodillos transportadores son elementos auxiliares de las instalaciones con el objetivo de recibir un producto de forma más o menos continua y conducirlo a otro punto con un funcionamiento independiente e intercalado en las líneas de los procesos donde no requieren generalmente ningún operario que manipule directamente sobre ellos de forma continua (Figura 1.22).

Las bandas y rodillos transportadores son uno de los sistemas más eficientes en el transporte de materiales, minerales, materias primas y diversidad de productos por su sencillez de funcionamiento y una vez instalados en condiciones normales suelen dar pocos problemas mecánicos y de mantenimiento reducido.



Figura 1.22: Banda transportadora

1.9.3. Mesa giratoria

Las mesas giratorias se utilizan para cambiar las direcciones o girar mercancías en el transportador. Su construcción modular permite ajustarlas de forma óptima a diversas aplicaciones.

Los transportadores de rodillos o cadenas con o sin desplazamiento actúan como unidades de carga. Las mesas giratorias permiten 2, 3 ó 4 puntos de giro para ofrecer la máxima versatilidad (Figura 1.23). Consiguen velocidades de hasta 0,45 m/s (88 pies/min) y el tiempo de giro para 90 grados es de 4,5 segundos. No es necesario instalar cercas adicionales debido al alojamiento integrado.



Figura 1.23: Mesa giratoria

1.10. Servo-motores

En su definición más básica, es un motor que puede ser controlado en su velocidad de funcionamiento y en la posición dentro de un rango de operación (0 a 360 grados) para ejecutar la actividad requerida.

El control es realizado por el dispositivo llamado encoder, que, mediante una señal electrónicamente codificada, indica las acciones de movimiento y velocidad a ejecutar. El servomotor es instalado en un equipo o máquina (en el mayor de los casos en su eje de accionamiento), para permitir que este tenga control de la posición, dirección y velocidad de una carga (Figura 1.24).

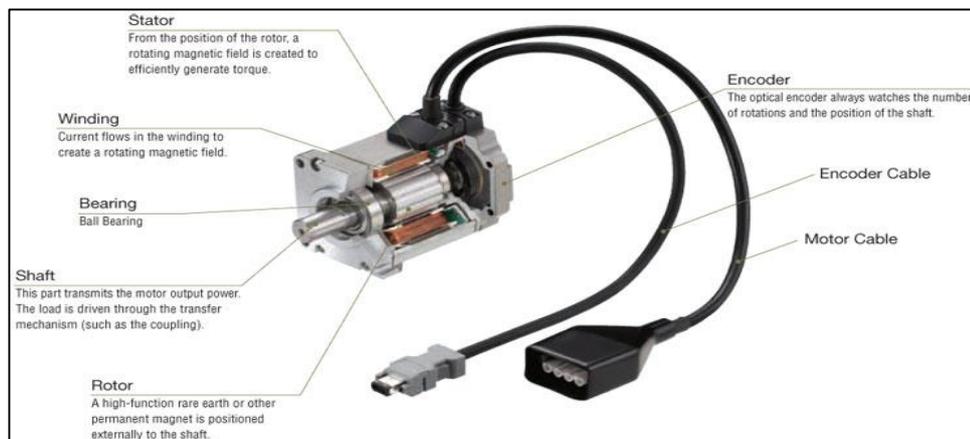


Figura 1.24: Partes de un servo-motor

1.10.1. Funcionamiento de un servomotor

El sistema servo se comunica mediante pulsos eléctricos a través de un circuito de control para determinar el ángulo de posición del motor, el servo espera recibir un pulso cada 20 milisegundos (0.02 segundos). La longitud del pulso determinará los giros de motor; un pulso de 1.5 ms.

Por ejemplo, hará que el motor vaya a una posición de 90 grados (posición neutra). Si el pulso es menor de 1.5 ms., entonces el motor se acercará a los 0 grados. Si el pulso es mayor de 1.5ms, el eje se moverá acercándose a los 180 grados.

Luego de esto, al interior del controlador de movimiento o posicionador está un programa guardado en un microcontrolador que tiene la capacidad de completar la tarea de una aplicación específica; el cual monitorea la posición del motor y comunica al accionamiento servo-controlado la necesidad de mover el servomotor hacia la posición deseada o comandada.

Dicho accionamiento aplica la cantidad de potencia necesaria sobre el motor para de esa forma mover la carga. En caso que el funcionamiento del motor no sea adecuado, en cuanto a velocidad, el dispositivo de retroalimentación alerta al control de la situación, que genera y ejerce más potencia sobre el motor hasta obtener la velocidad ideal para la acción realizada; si la velocidad es muy alta al principio, ocurrirá lo inverso [17].

1.11. Motor DC

Son los más usados en la actualidad por su gran facilidad de control, en la (Figura 1.25) se muestra los componentes de un motor DC, el cual constituye de dos devanados internos, el inductor e inducido, que se alimentan con corriente continua.

El inductor, también denominado devanado de excitación está situado en el

estator y crea un campo magnético de dirección fija, denominado de excitación.

El inducido, situado en el rotor, hace girar al mismo debido a la fuerza de Lorentz que aparece como la combinación de la corriente circulante por él y el campo magnético de excitación. Recibe la corriente del exterior a través del colector, en el que se apoyan unas escobillas de grafito.

La conversión que realiza un motor de DC de energía eléctrica a energía mecánica de forma continua es necesario que los campos magnéticos del estator y del rotor permanezcan estáticos entre sí [18].

Esta transformación es máxima cuando ambos campos se encuentran en cuadratura, es decir, cuando los ejes de los bobinados del rotor y del estator forman un ángulo de 90 grados. El conmutador de delgas es un conmutador sincronizado con el rotor encargado de que se mantenga el ángulo relativo entre el campo del estator y el creado por las corrientes rotóricas.

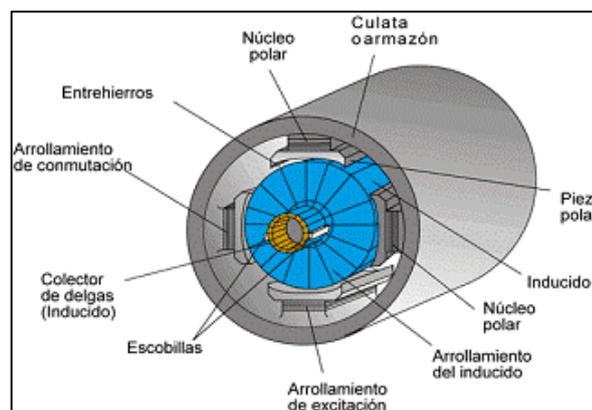


Figura 1.25: Componentes de un motor DC
Fuente: [19]

De esta forma se consigue transformar automáticamente, en función de la máquina la corriente continua que alimenta al motor en corriente alterna de frecuencia variable en el inducido. Este tipo de funcionamiento se conoce con el nombre de autopilotado.

1.12. Arduino

Arduino es una placa de hardware libre (Figura 1.26) que incorpora un microcontrolador reprogramable constituido de una serie de pines-hembra tanto analógicos como digitales (unidos internamente a las patillas de entrada /salida del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores.

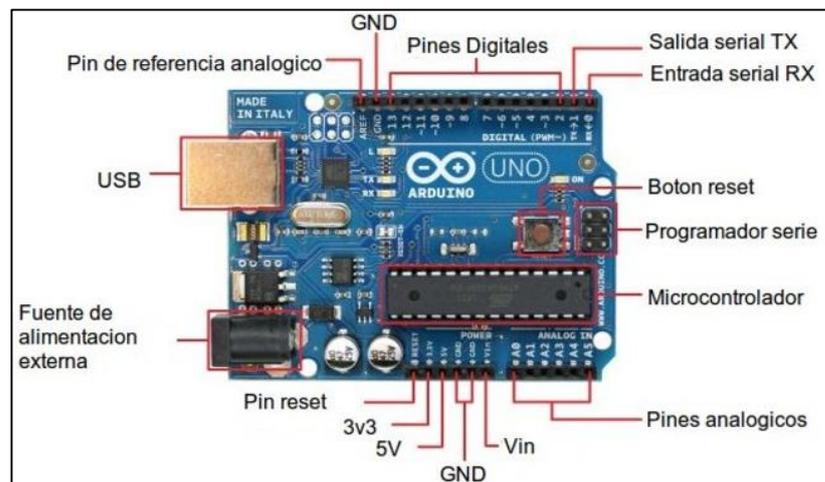


Figura 1.26: Tarjeta Arduino
Fuente: [20]

El hardware consiste en una placa con un microcontrolador Atmel AVR. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega 8 por su sencillez y bajo costo que permiten el desarrollo de múltiples diseños.

Por otro lado, el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader) que corre en la placa (Figura 1.27).

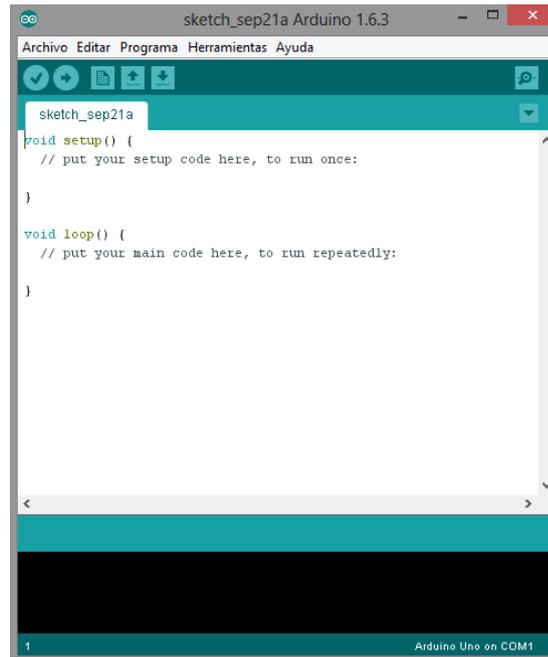


Figura 1.27: Interfaz de programación en Arduino

Al ser open-hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Según la Free Software Foundation (FSF), organización encargada de fomentar el uso y desarrollo del software libre a nivel mundial, un software para ser considerado libre a de ofrecer a cualquier persona u organización cuatro libertades básicas e imprescindibles [21].

- Libertad 0: la libertad de usar el programa con cualquier propósito y en cualquier sistema informático.
- Libertad 1: la libertad de estudiar cómo funciona internamente el programa, y adaptarlo a las necesidades particulares. El acceso al código fuente es un requisito previo para esto.
- Libertad 2: la libertad de distribuir copias.
- Libertad 3: la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

1.13. Modbus

Es un protocolo de comunicación de comunicaciones industriales de capa 7 del modelo OSI, basado en la arquitectura maestro-esclavo, diseñada en 1970 por MODICON para una gama de PLC. Convertido en un protocolo estándar dentro de la industria que tiene mayor disponibilidad para la conexión de dispositivos industriales.

El sistema maestro/esclavo tiene un nodo maestro que es encargado de enviar los comandos explícitos a cada uno de los nodos esclavos los cuales procesarán la respuesta requerida. Una característica de este tipo de bus de campo es que los nodos no transmiten información sin una petición del nodo maestro y además no se comunica con los demás nodos esclavos dentro de la red.

En el protocolo de comunicación MODBUS existen dos formas de intercambiar la información con los dispositivos de la red estas son:

- Modbus RTU
- Modbus ASCII

En cualquiera de los dos tipos de comunicación el usuario define el tipo de comunicación, así también como de los parámetros de comunicación serial como la velocidad, paridad, modo, etc., durante la configuración de cada controlador. Hay que tener en cuenta que el nodo y todos los parámetros seriales deben ser los mismos para todos los dispositivos de una red MODBUS.

1.13.1. Modos de transmisión

La selección del modo de comunicación RTU o ASCII solo se refiere a las redes estándar de MODBUS; esta selección define como será empaquetada la información dentro de las tramas MODBUS y a la vez su decodificación.

a. Modbus RTU

La Unidad de Terminal Remota (Remote Terminal Unit por sus siglas en inglés) realiza la comunicación entre dispositivos enviando cada mensaje de un byte (8 bits) conteniendo dos caracteres hexadecimales de 4 bits, y el mensaje se transmite en un flujo continuo. La ventaja de este modo es que entre mayor sea la densidad de caracteres permite un mejor rendimiento que los datos ASCII para la misma velocidad de transmisión.

b. Modbus ASCII

La comunicación mediante MODBUS ASCII (American Standard Code for Information Interchange por sus siglas en inglés) en donde cada byte de caracteres en un mensaje envía como dos caracteres ASCII. Este modo permite intervalos de tiempo de hasta un segundo durante la transmisión sin generar errores. A continuación, se presenta las diferencias entre los tipos de comunicación Modbus RTU y Modbus ASCII:

Tabla 1.1: Diferencias entre Modbus RTU Y Modbus ASCII

Elemento	MODO ASCII	MODO RTU
Caracteres	ASCII '0'.... '9', 'A'... 'F'	Binario 0 ... 255
Comprob. Error	LRC: Longitudinal Redundancy check	CRC: Cyclic Redundancy Check
Inicio de Trama	Carácter ':'	3.5 veces t de carácter
Final de trama	Character CR/CL	3.5 veces t de carácter
Distancia máx. entre caracteres	1 seg.	1.5 veces t de carácter
Bit de inicio	1	1
Bit de datos	7	8
Paridad	Par/Impar/Ninguna	Par/Impar/Ninguna
Bits de parada	1 si hay paridad 2 si ninguna	1 si hay paridad 2 si ninguna

1.14. Filtros

Cualquier combinación de elementos pasivos (R, L, C) y/o activos (transistores o amplificadores operacionales) diseñada para seleccionar o rechazar una banda de frecuencias se llama filtro.

En sistemas de comunicación se utilizan los filtros para dejar pasar aquellas frecuencias que contienen la información deseada, y rechazar las frecuencias restantes. En sistemas estereofónicos, los filtros pueden aislar bandas de frecuencias particulares para un énfasis incrementado o reducido por el sistema acústico de salida (amplificador, bocina, etcétera).

Se utilizan filtros para impedir el paso a cualesquier frecuencias indeseables, comúnmente llamadas ruido, debido a las características no lineales de algunos dispositivos electrónicos o señales captadas del medio circundante. Por lo general se especifican dos clasificaciones de filtros [22].

- **Filtros pasivos** son aquellos filtros compuestos de combinaciones en serie o en paralelo de elementos R, L y C.
- **Filtros activos** son los que emplean dispositivos activos como transistores, y amplificadores operacionales, combinados con elementos R, L y C.

CAPÍTULO II

2. DISEÑO, CONSTRUCCIÓN Y AUTOMATIZACIÓN DE LA PLANTA

2.1. Descripción del módulo didáctico

Para asimilar de manera mucho más fácil la estructura y los elementos que componen el módulo didáctico se lo ha dividido en etapas, permitiendo una mejor visualización del mismo. En la Figura 2.1 se indica el diagrama de bloques del módulo didáctico.

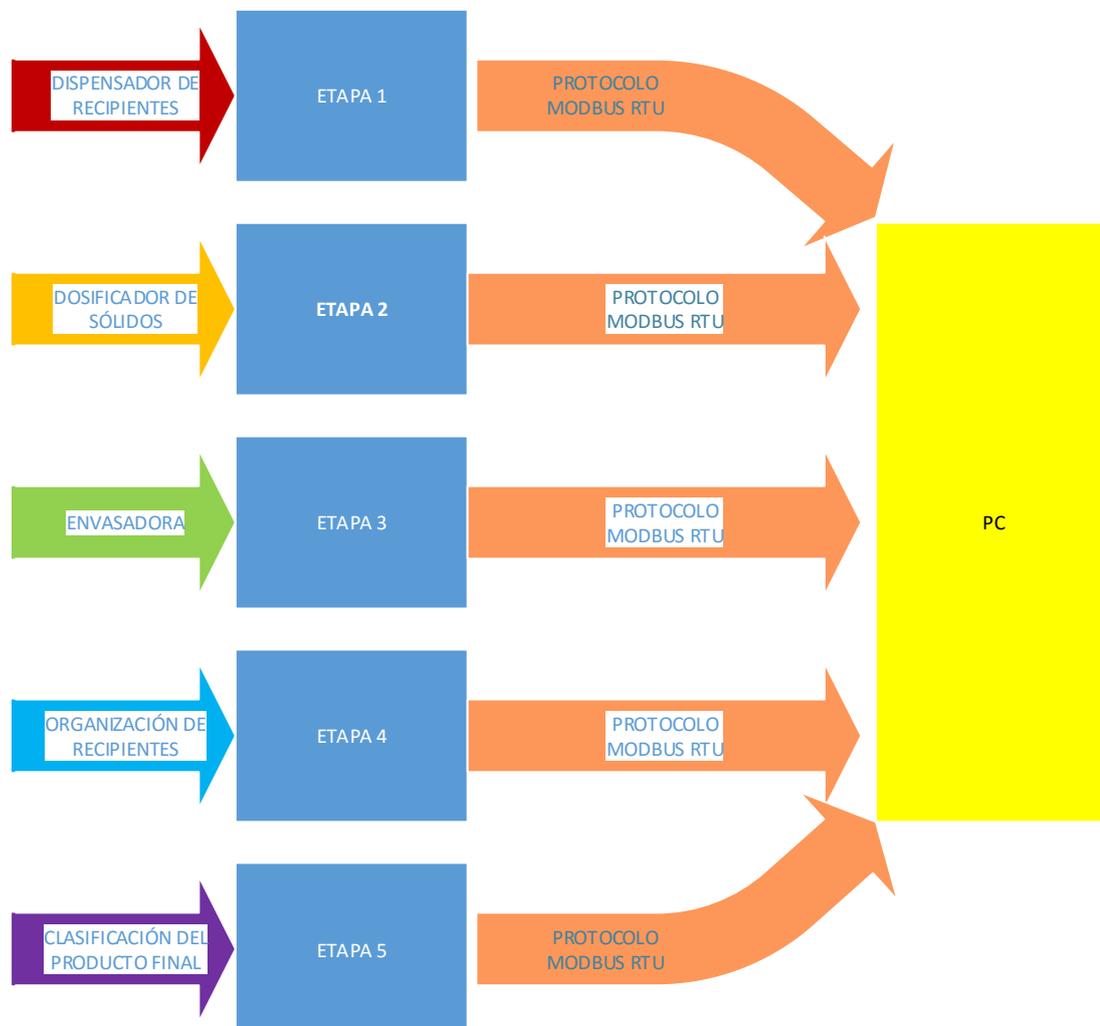


Figura 2.1: Diagrama de bloques del módulo didáctico

2.1.1. Etapa 1

La primera etapa del módulo didáctico consta de una máquina dispensadora de recipientes (vasos), cuya función es la de proporcionar recipientes y distribuirlos en una mesa giratoria dividida en 8 partes iguales, la cual dispone de un encoder empotrado en el eje central; el ajuste inicial de la mesa giratoria se realiza mediante un sensor fotoeléctrico el cual proporciona una señal después de haber completado una revolución (360 grados), de manera que los recipientes se colocan correctamente. La máquina dispensadora dispone de un sensor capacitivo para verificar la existencia de recipientes en el contenedor caso contrario genera una señal de alerta al usuario deteniendo temporalmente el proceso hasta que el operador adicione recipientes en el contenedor y continuar con el proceso, el diagrama de bloques de la presente etapa se indica en la Figura 2.2.

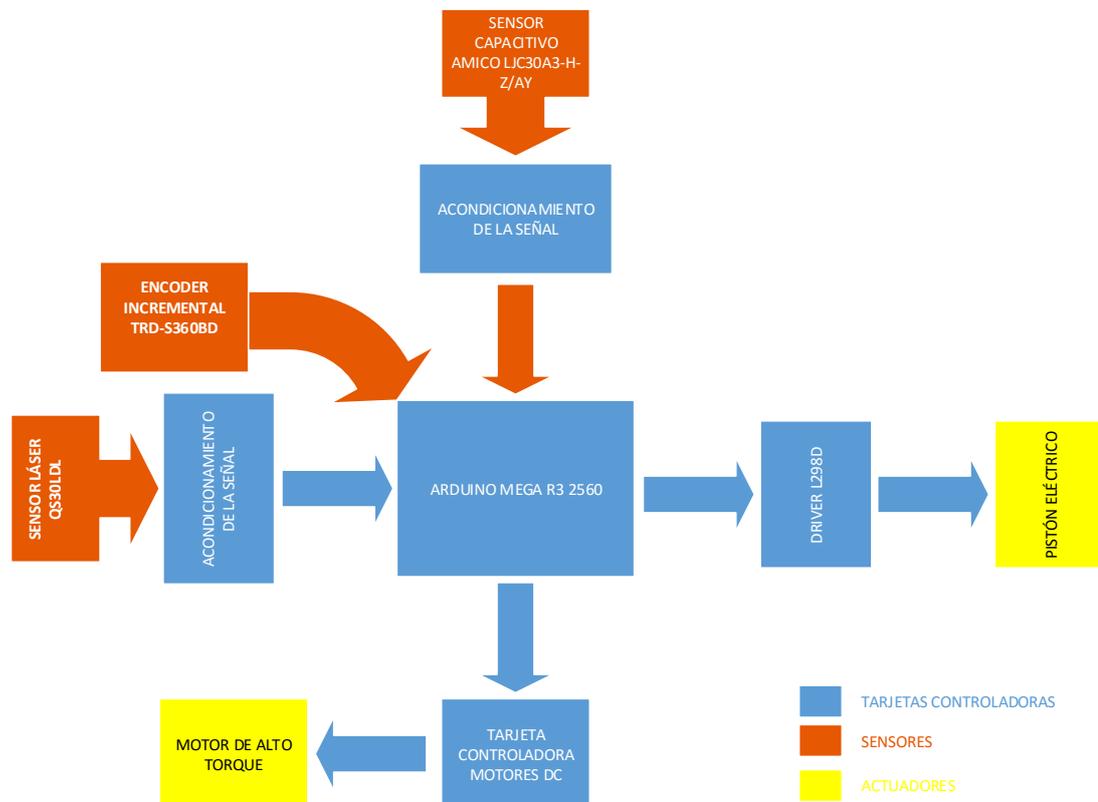


Figura 2.2: Diagrama de bloques Etapa 1

2.1.2. Etapa 2

La segunda etapa del módulo didáctico dispone de un sensor fotoeléctrico que detecta el recipiente colocado en una de las ranuras de la mesa giratoria, posteriormente mediante la acción de un motor vibratorio ubicado en un pedestal, dispensa un material sólido en el recipiente (vaso), dicho material es controlado y pesado, adicionalmente dispone de un sensor ultrasónico para verificar el nivel del material contenido en una tolva. El diagrama de bloques de la presente etapa se indica en la Figura 2.3.

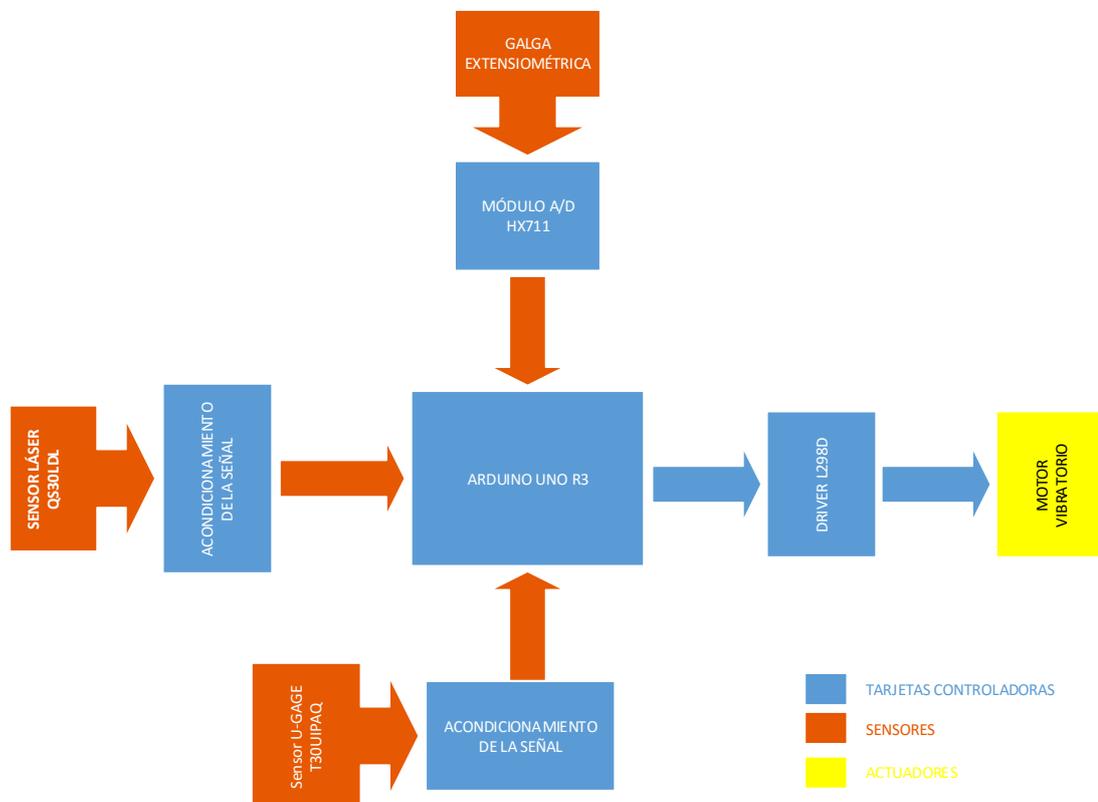


Figura 2.3: Diagrama de bloques Etapa 2

2.1.3. Etapa 3

La tercera etapa del módulo didáctico realiza el envasado de los recipientes, incluye un sensor fotoeléctrico que detecta el recipiente a envasar, mediante la acción de un servo-motor, posiciona en un ángulo de 180 grados a un pistón, para extraer la cubierta (tapa) mediante una bomba de vacío en conjunto con una ventosa, posteriormente el servo-motor posiciona el pistón en un ángulo de 0 grados ejecutando una acción de sellado en el recipiente. Dispone de un sensor capacitivo que detecta la existencia de cubiertas para envasar los recipientes, caso contrario genera una señal de alerta al usuario deteniendo temporalmente el proceso hasta que el operador adicione cubiertas en el contenedor y continuar con el proceso. El diagrama de bloques de la presente etapa se indica en la en la Figura 2.4.

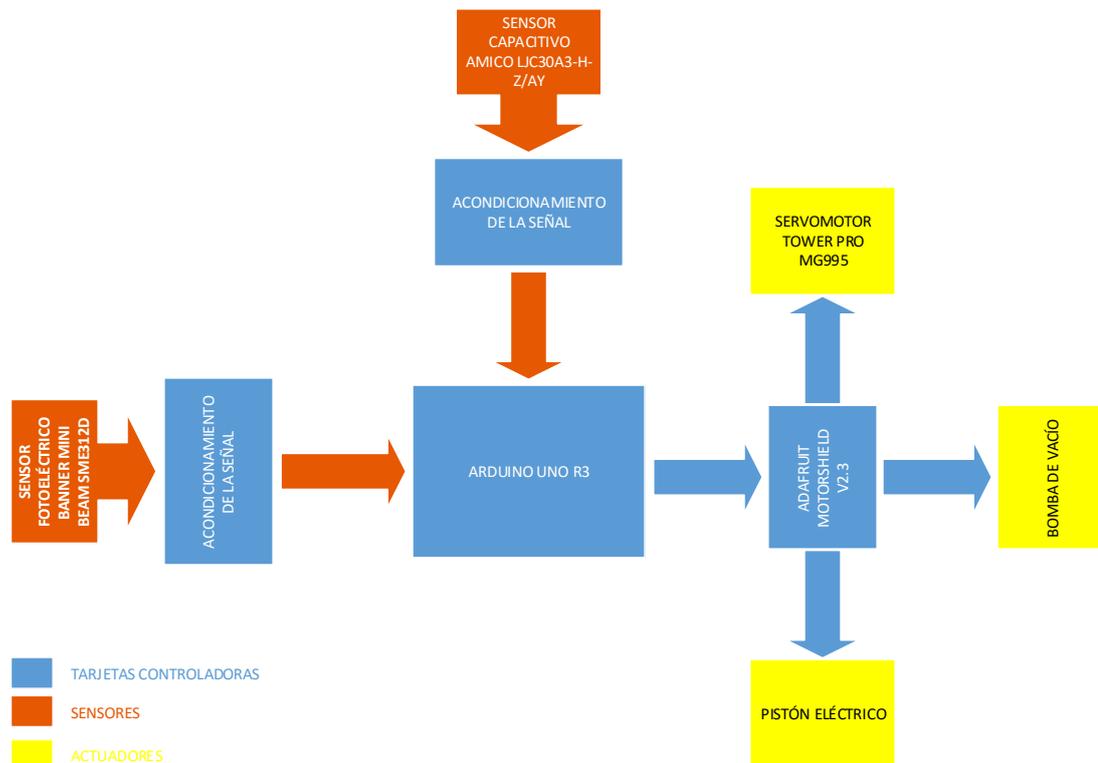


Figura 2.4: Diagrama de bloques Etapa 3

2.1.4. Etapa 4

La cuarta etapa del módulo didáctico organiza los recipientes envasados en cajas, consta de tres sensores fotoeléctricos, uno de ellos detecta los recipientes envasados para trasportarlos mediante la acción de un brazo robótico (BRAZO 1) colocandolos en una caja con 4 divisiones, la cual se extrae desde un contenedor de cajas mediante una banda transportadora y es posicionada gracias a otro sensor fotoeléctrico. Un sensor fotoeléctrico detecta la presencia de cajas en el contenedor, caso contrario genera una señal de alerta al usuario deteniendo temporalmente el proceso hasta que el operador adicione cajas en el contenedor y continuar con el proceso. El diagrama de bloques de la presente etapa se indica en la Figura 2.5.

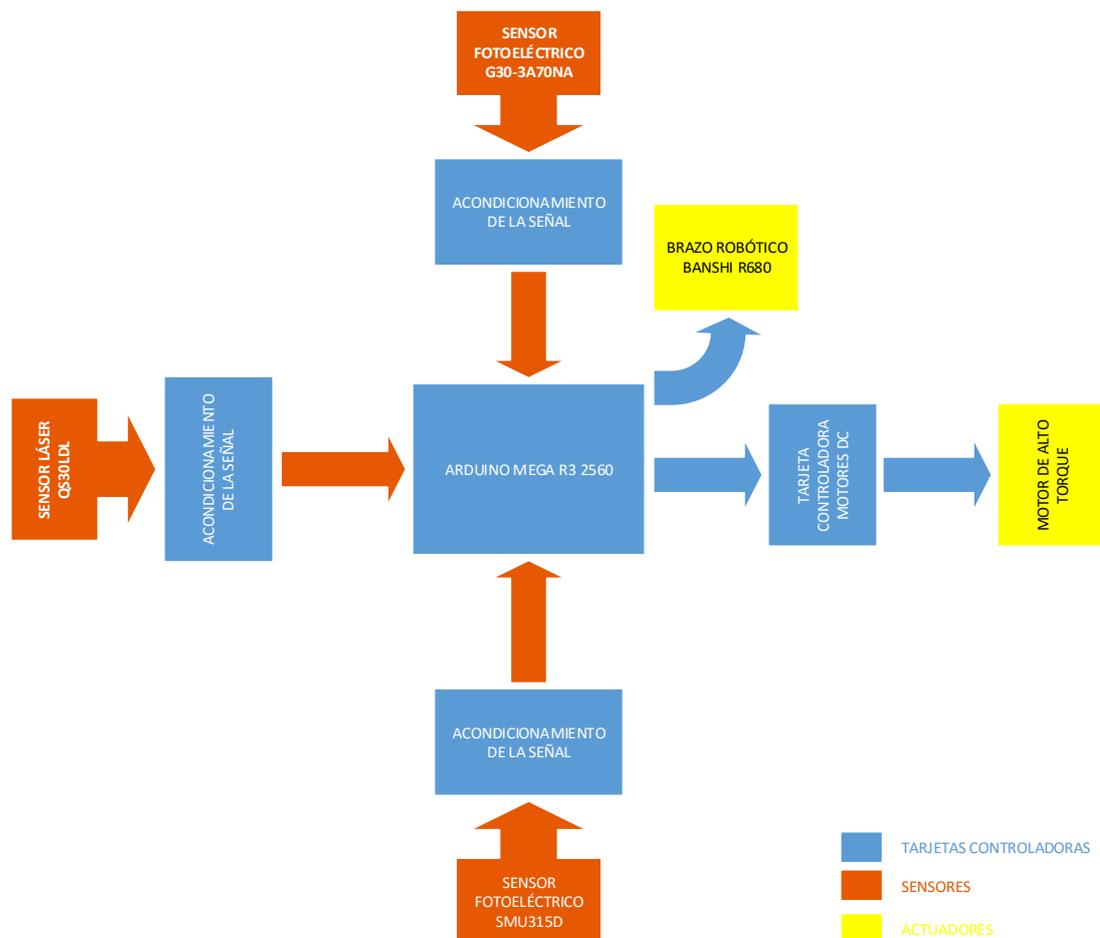


Figura 2.5: Diagrama de bloques Etapa 4

2.1.5. Etapa 5

La quinta etapa o etapa de inspección se encarga de clasificar el producto final, mediante un sensor fotoeléctrico detecta la presencia o ausencia de la caja al final de la banda transportadora, para posteriormente comprobar el peso en bruto del producto final con una galga extensiométrica y decidir si el producto final aprueba o reprueba la inspección, este a su vez se traslada con la ayuda de un brazo robótico (BRAZO 2) colocando el producto final en la respectiva ruta de transporte, donde el operario tendrá que retirar el producto final o el rechazo. El diagrama de bloques de la presente etapa se indica en la Figura 2.6.

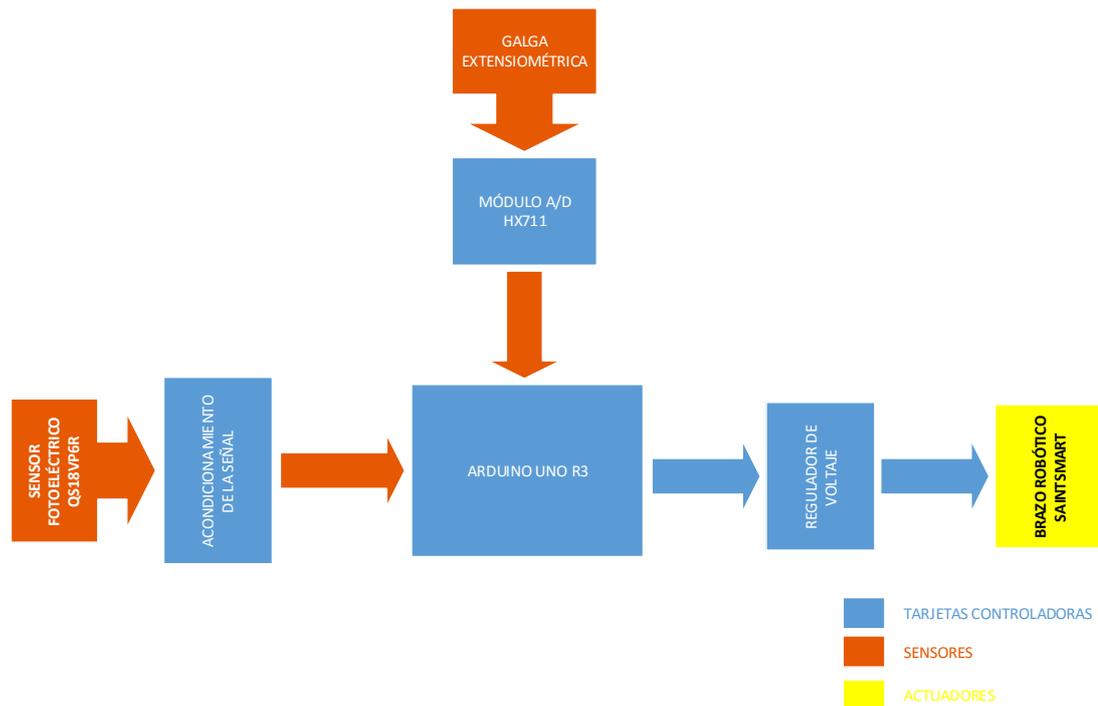


Figura 2.6: Diagrama de bloques Etapa 5

2.1.6. Monitoreo de la planta

Las etapas que forman parte del módulo didáctico son monitoreadas mediante un HMI (Human Machine Interface) desarrollado en el software Monitoriza for Arduino-Scada Acimut, para observar y verificar el correcto funcionamiento de cada una de las etapas del proceso.

2.2. Diseño de accesorios mecánicos

Mediante el programa SolidWorks se diseña cada una de los accesorios mecánicos del módulo didáctico.

SolidWorks es un programa de diseño mecánico CAD (Diseño Asistido por Computadora). Junto con las herramientas de diseño de pieza, ensamblajes y dibujo, SolidWorks incluye herramientas de productividad, de gestión de proyectos, de presentación, de análisis y simulación que lo hacen uno de los estándares de diseño mecánico más competitivo del mercado.

Las principales características que hace que SolidWorks sea una herramienta versátil y precisa es su capacidad de ser asociativo, variacional y paramétrico de forma bidireccional con todas sus aplicaciones. Además, utiliza el gestor de diseño (FeatureManager) que facilita enormemente la modificación rápida de operaciones tridimensionales y de croquis de operación sin tener que rehacer los diseños ya plasmados en el resto de documentos asociados.

Como se puede apreciar en la Figura 2.7 la plataforma de diseño gráfico incluye módulos como son: Pieza, Ensamblaje, Dibujo, DWGeditor, Simulación/movimiento, Análisis de interferencia, 2D a 3D, Traductores IDF y CAD, Lista de Materiales, Chapa metálica, Superficies, Estructuras, Soldadura, Moldes, Análisis de Elementos Finitos con COSMOSXpress, MoldFlowXpress, Configuraciones y SmartMates (Tecnología Inteligente).

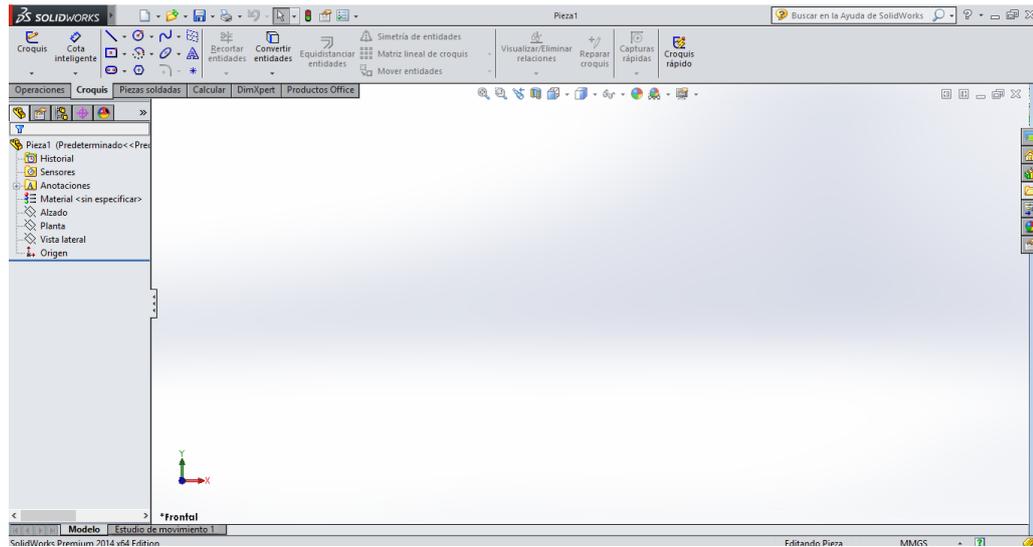


Figura 2.7: Plataforma de diseño en SolidWorks

2.2.1. Estructura base del dispensador de recipientes

La base del dispensador de recipientes (Figura 2.8) es fabricada en acero de 1.5 mm, aloja al dispensador de recipientes, que dispone de una capacidad de 30 recipientes individuales de 8 onzas con un diámetro standard de 75 mm.

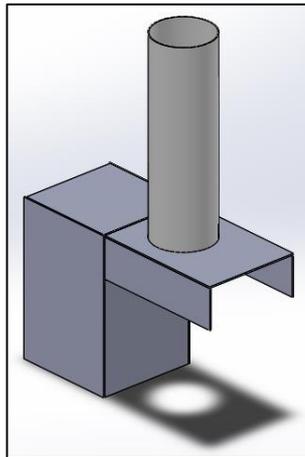


Figura 2.8: Soporte del dispensador de vasos

2.2.2. Mesa giratoria

La mesa giratoria (Figura 2.9) es parte fundamental de la estructura física de la planta, ya que contiene a los recipientes en las distintas etapas que conforman el módulo didáctico, es fabricada en aluminio de 2 mm, dispone de 8 divisiones circulares con un diámetro de 77 mm, el diámetro total de la circunferencia mayor es de 400 mm, además está empotrada en la parte central del cajón.

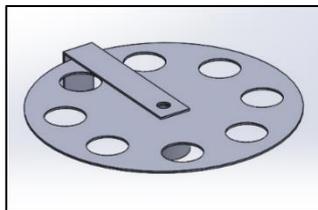


Figura 2.9: Mesa giratoria

2.2.3. Tolva

La tolva (Figura 2.10) contiene el material a dosificar (cereales, granos, semillas, etc.), está construida de acero de 1 mm de espesor, con una altura de 600 mm y 250 mm de lado cuadrado, con una abertura en la parte superior para insertar granos o cereales, además esta fusionada con un pedestal que facilita la descarga del material en un recipiente.

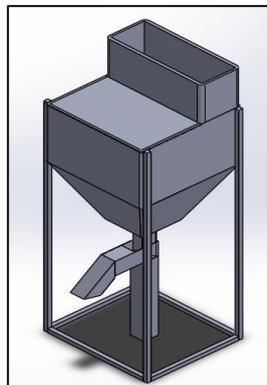


Figura 2.10: Tolva

2.2.4. Estructura base de la envasadora

La estructura base de la envasadora (Figura 2.11) está constituida por un contenedor de cubiertas con un diámetro standard de 82 mm, adiciona una caja rectangular de 450 mm de altura por 100 mm, con un compartimento ubicado en la parte media para alojar un servomotor.

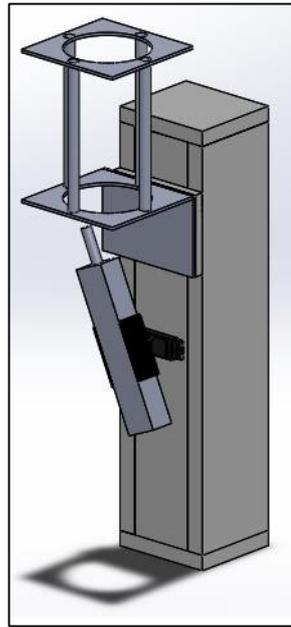


Figura 2.11: Estructura base de la envasadora

2.2.5. Contenedor de cajas

El contenedor de cajas se encuentra diseñado en madera de 3 mm de espesor con un marco metálico de aluminio con medidas de 175 mm en sus 4 lados, a una altura de 570 mm y con una capacidad de 6 cajas en su interior, como se indica en la Figura 2.12.

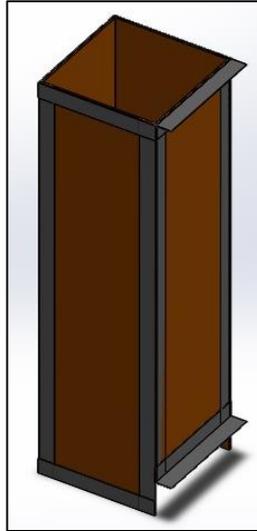


Figura 2.12: Contenedor de cajas

2.2.6. Estructura base de la banda transportadora

La banda transportadora está constituida de un armazón metálico de tubos de acero de 40 mm por 20 mm (Figura 2.13), con un juego de 4 chumaceras acopladas a dos ejes de aluminio, en los cuales se halla empotrada una cinta transportadora con tensiones iguales en los rodillos.

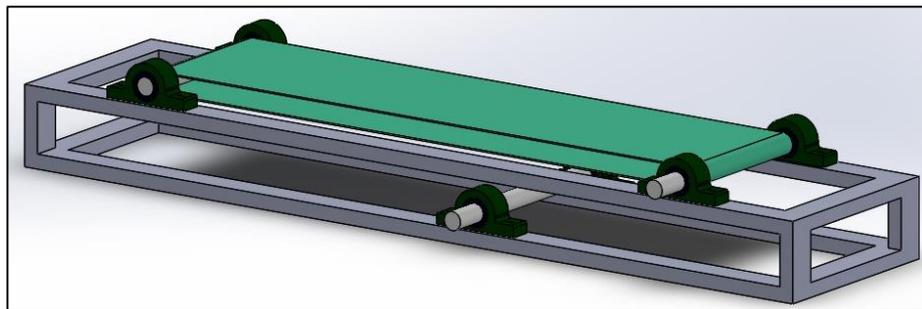


Figura 2.13: Estructura de la banda transportadora

2.2.7. Estructura base del cajón

La estructura base del cajón donde cada una de las etapas se encuentran estacionadas para realizar su respectiva función, se encuentra construido en madera MDP de 18 mm de espesor con enchapado blanco, diseñado con

medidas: 1200 mm de ancho y 200 mm de alto, además posee una superficie hueca en donde está alojado la banda transportadora de 1164 mm de largo por 330 mm de ancho y 182 mm de alto y finalmente contiene una rampa donde son depositadas cajas, como se indica en la Figura 2.14.

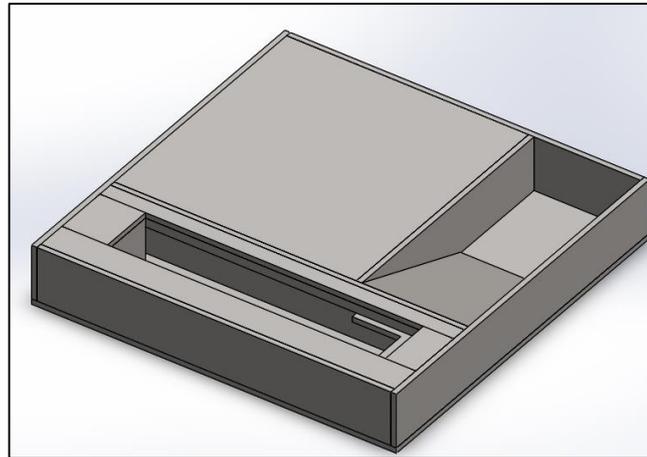


Figura 2.14: Base del módulo didáctico

2.2.8. Estructura frontal del panel de control

El panel de control contiene los elementos de mando y luces indicadoras de cada una de las etapas como son: indicadores luminosos e indicadores gráficos, salida de señales analógicas y digitales, botones de inicio y paro general de la planta, y finalmente el encendido y apagado de toda la Red. Las dimensiones del panel de control son de 1180 mm de largo por 207 mm de ancho, como se indica en la Figura 2.15.

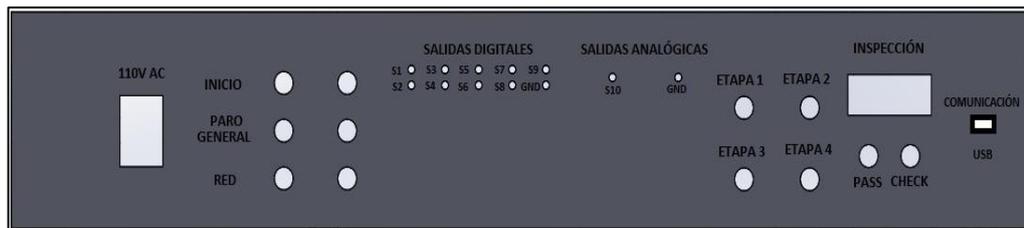


Figura 2.15: Panel de control

2.2.9. Caja

Las cajas son diseñadas en un material llamado syntra el cual es ligero y se adapta a las condiciones que se necesitan en el proceso. Las medidas utilizadas van acorde al contenedor de cajas, el diseño del prototipo de la caja se indica en la Figura 2.16.

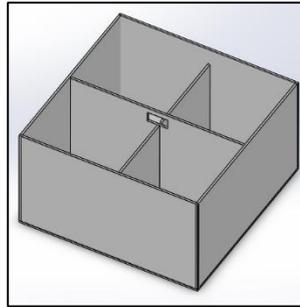


Figura 2.16: Caja

2.2.10. Estructura base del módulo didáctico

Una vez indicado los prototipos de los elementos que conforman el módulo didáctico el esquema total se indica en la Figura 2.17.

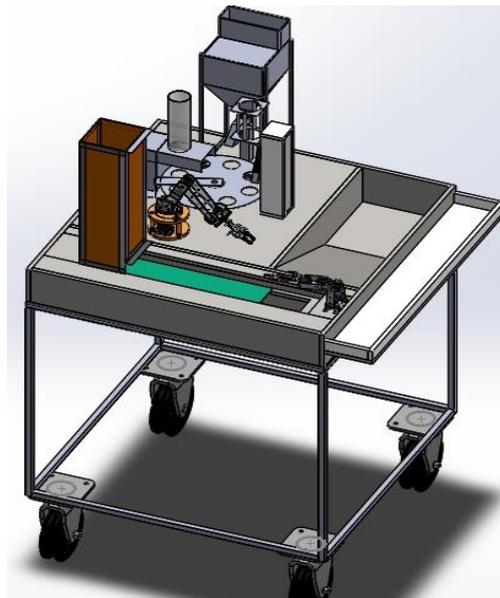


Figura 2.17: Prototipo del módulo didáctico de llenado y envasado de sólidos

2.3. Construcción del módulo didáctico

Una vez diseñada la parte mecánica del módulo didáctico de un sistema de automatización de llenado y envasado de sólidos, se construye cada una de las etapas, implementando tanto la parte electrónica como la parte de control, a su vez se añade todos los elementos como actuadores y sensores.

2.3.1. Construcción del dispensador de recipientes

El dispensador de recipientes, consta de elementos que son detallados a continuación:

a. Sensor capacitivo amico LJC30A3-H-Z/AY

El sensor capacitivo AMICO LJC30A3-H-Z/AY (Figura 2.18) se encuentra instalado en la parte inferior del contenedor de recipientes, tiene una salida tipo transistor PNP (verifique las características del sensor en el Anexo A-9), con una distancia máxima de detección de 1 a 25 mm dependiendo del tipo de material (mediante la constante dieléctrica).



Figura 2.18: Sensor AMICO LJC30A3-H-Z/AY

A continuación se indica en la Figura 2.19 la conexión que se debe realizar para el funcionamiento del sensor capacitivo.

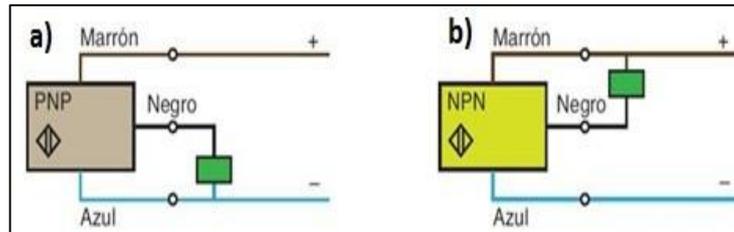


Figura 2.19: Conexión de la configuración: a) PNP. b) NPN

Por lo tanto el sensor capacitivo AMICO LJC30A3-H-z/AY genera a su salida un 0 lógico cuando detecte el recipiente y un 1 lógico cuando no lo detecte como se indica en la Figura 2.20.



Figura 2.20: Funcionamiento del sensor AMICO LJC30A3-H-z/AY en: a) 1 lógico. b) 0 lógico

Finalmente el sensor capacitivo AMICO LJC30A3-H-Z/AY posee un potenciómetro lineal (Figura 2.21), para realizar el ajuste para la detección del recipiente (vaso) desde 1 a 25mm de distancia.



Figura 2.21: Potenciómetro de ajuste del sensor capacitivo

Los objetivos estándar son especificados para cada sensor capacitivo. El objetivo estándar se define normalmente como un metal o agua.

Los sensores capacitivos dependen de la constante dieléctrica del objetivo. Mientras más grande sea la constante dieléctrica de un material es más fácil detectar.

La Figura 2.22 indica la relación de las constantes dieléctricas de un objetivo y habilidad del sensor de detectar el material basado en la distancia nominal de sensado (Sr).

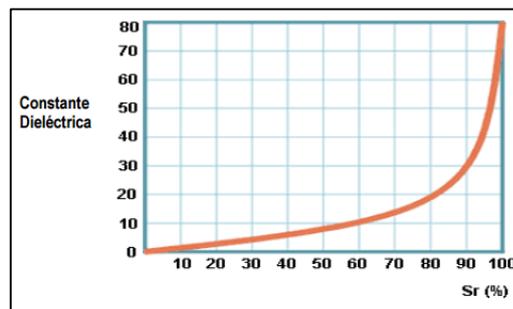


Figura 2.22: Relación de las constantes dieléctricas

Por ejemplo si un sensor capacitivo tiene una distancia nominal de sensado de 10 mm y el objetivo es alcohol, la distancia efectiva de sensado es aproximadamente el 85 % de la distancia nominal, o sea 8.5 mm, como se indica en la Figura 2.23.

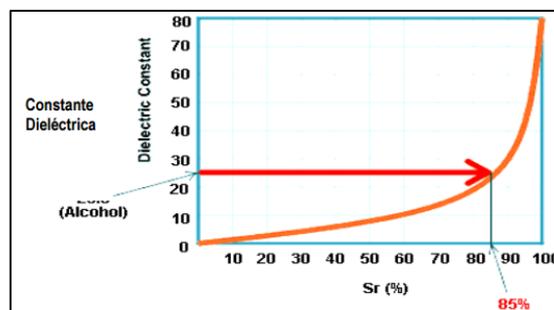


Figura 2.23: Distancia efectiva en relación a la constante dieléctrica

La Tabla 2.1 identifica las constantes dieléctricas (abreviadas como k) de varios materiales.

Tabla 2.1: Constantes dieléctricas de algunos materiales

Material	k	Material	k
Acetona	19.5	Plexiglás	3.2 - 3.5
Resina Acrílica	2.7-4.5	Petróleo	2.0 - 2.2
Aire	1.000264	Resina de Fenol	4 - 12
Alcohol	25.8	Poliacetato	3.6 - 3.7
Amoníaco	15-25	Poliamidas	5
Anilina	6.9	Resina de Poliéster	2.8 - 8.1
Soluciones Acuosas	50-80	Polietileno	2.3
Baquelita	3.6	Polipropileno	2.0 - 2.3
Bencina	2.3	Poliestireno	3
Dióxido de carbono	1.000985	Porcelana	4.4 - 7
Celuloide	3	Leche en polvo	3.5 - 4
Cemento en polvo	4	Papel de diario	2.5

b. Sensor láser QS30LDL

El sensor QS30LDL es un sensor láser de alto rendimiento y fácil de usar, cuyas opciones de configuración son adecuadas para aplicaciones exigentes (Figura 2.24). Proporciona alto rendimiento de detección en aplicaciones de bajo contraste en gamas relativamente largas.

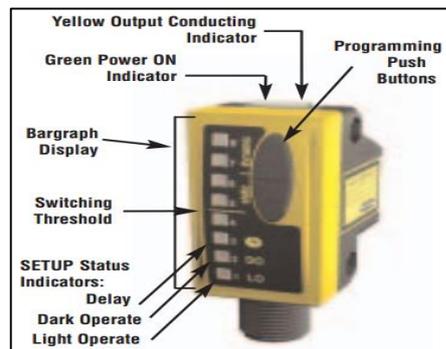


Figura 2.24: Sensor Láser QS30LDL

El sensor láser QS30LDL cuenta con las siguientes características especificadas en el Anexo A-1.

El sensor láser QS30LDL (Figura 2.25), está ubicado por debajo del dispensador de recipientes, la acción que ejecuta es calibrar la mesa giratoria al momento de terminar una revolución para su correcto posicionamiento, lo realiza gracias a un encoder ubicado en la mesa giratoria.



Figura 2.25: Sensor Láser QS30LDL ubicado en la Etapa 1

Este sensor genera un 1 lógico el momento que no detecta el material férreo (cejillo), es decir, detecta la pared negra en su frente y un 0 lógico el momento que detecta el material férreo incrustado en la mesa giratoria (Figura 2.26). Este sensor está denominado S1.

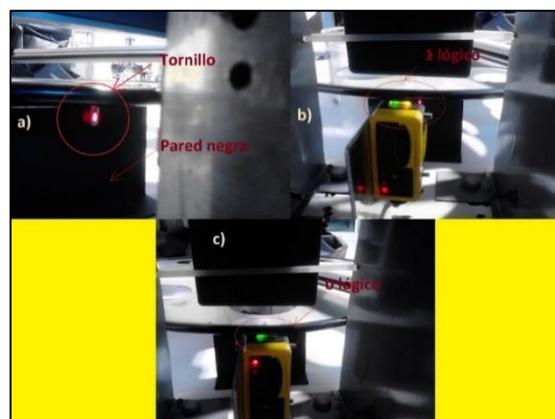


Figura 2.26: a) Materiales a detectar. b) Sensor en 1 lógico. c) Sensor en 0 lógico

c. Pistón eléctrico

EL pistón eléctrico (Figura 2.27) del dispensador de recipientes se encarga de accionar un elemento mecánico para extraer del contenedor los recipientes que se ubican en cada división de la mesa giratoria.



Figura 2.27: Pistón eléctrico

El pistón está alojado en un compartimento unido a un mecanismo que realiza el dispensado y retención de los recipientes como se muestra en la Figura 2.28.

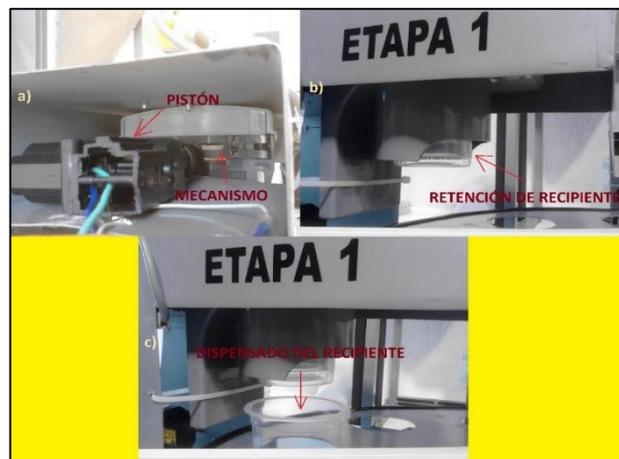


Figura 2.28: a) Mecanismo y pistón. b) Recipiente retenido. c) Recipiente dispensado

d. Driver L298N

El driver L298N tarjeta controladora de motores DC (Figura 2.29) es una tarjeta que cuenta con el circuito L298, el cual permite manejar motores DC con un suministro de corriente máximo de 2 amperios de acuerdo a la hoja de datos de dicho dispositivo.

El driver L298N es el encargado de activar y desactivar al pistón para dispensar o retener los recipientes (vasos) del contenedor.

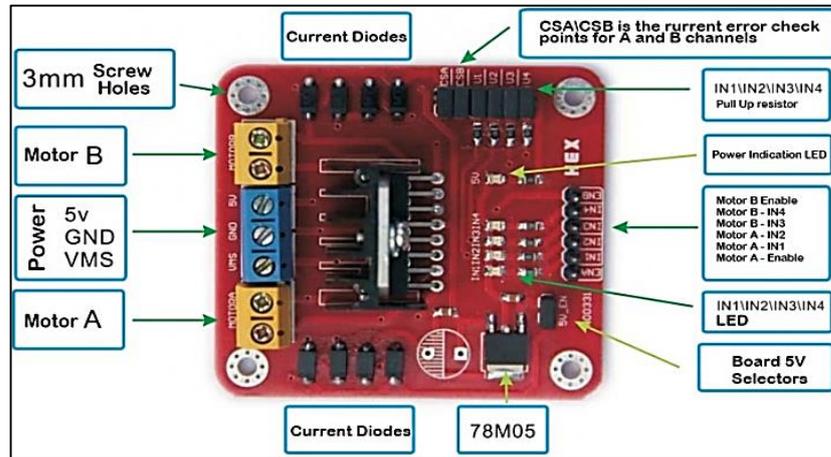


Figura 2.29: Driver L298N

El driver L298N tiene las siguientes especificaciones:

- Usa el chip 298.
- Capaz de controlar 2 (3-30VDC) Motores
- Soporta control de 3.3V y 5V.
- Capaz de controlar la velocidad y dirección de 2 motores de DC.
- Capaz de controlar la velocidad y dirección de un motor a pasos.
- Corriente máxima 2A por puente H.
- Niveles de voltaje lógicos 4.5 a 5.5V (Alto) / 0V (Bajo).
- Potencia de consumo máxima: 20W.

e. Motor de alto torque

El motor de 12VDC es adecuado para el movimiento de la mesa giratoria por su alto torque (Figura 2.30).



Figura 2.30: Motor de alto torque 12VDC

La Tabla 2.2 contiene las especificaciones del motor de alto torque de 12VDC.

Tabla 2.2: Especificaciones del motor de alto torque 12VDC

MODELO	ZD1830
Régimen de voltaje	12VDC
Energía del grado	80W
Corriente sin carga	2.0/2.5A
Velocidad sin carga	45 ± 5 RPM
Corriente de carga	10/15 A
Velocidad de carga	30/45 ± 5RPM
Stall actual	43 A
Par máximo	48nm (min)
Circuito de protección	EMC o sobrecarga de la protección
Clase de protección	IP 33
Peso	2.9 Kg

f. Tarjeta controladora de motores DC

La tarjeta controladora de motores DC (Figura 2.31) está conectada al motor 12VDC de alto torque que será la encargada de recibir la señal PWM desde el Arduino para el control de velocidad de la mesa giratoria, así realizando un control sobre el mismo.

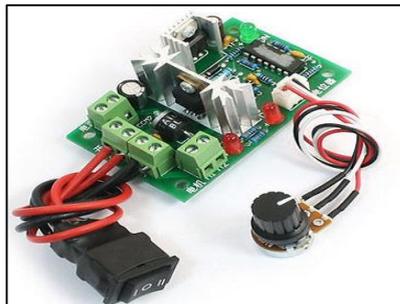


Figura 2.31: DC motor control

Las especificaciones técnicas se detallan en la Tabla 2.3.

Tabla 2.3: Especificaciones de la tarjeta controladora de motores DC

Nombre del producto	DC regulador de velocidad de motor
Modelo	J809
Voltaje de funcionamiento	DC 5V – 30V
Potencia de salida (máx.)	80 W
Corriente (máx.)	2.5 A
Rango PWM	10% - 95%
Circuito sobrecarga Protección	3A
Peso	70 Gr.

g. Tarjeta arduino MEGA 2560

El Arduino / Genuino Mega 2560 es una placa electrónica basada en el Atmega2560 (Figura 2.32). Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio.

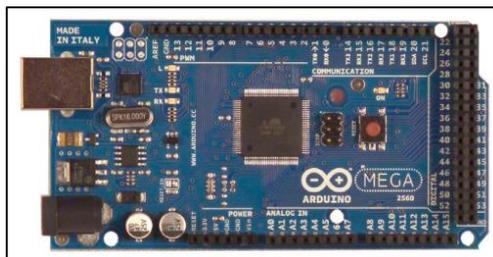


Figura 2.32: Arduino MEGA 2560

Contiene todo lo necesario para la programación del microcontrolador; simplemente se conecta al ordenador con un cable USB o un adaptador de CA o la batería de CC. La Arduino MEGA 2560 es compatible con la mayoría de Shields diseñados para Arduino/ Genuino Uno y las antiguas tarjetas Duemilanove o Diecimila.

h. Encoder incremental TRD-S360BD

El encoder incremental TRD - S360BD (Figura 2.33) es un transductor rotativo que provee un número específico de pulsos equitativamente espaciados por revolución (PPR) o por pulgada o milímetro de movimiento lineal (para mas detalles lea el Anexo A-8 sobre este sensor). Se utiliza un solo canal de salida para aplicaciones donde el sentido de la dirección de movimiento no es importante (unidireccional).



Figura 2.33: Encoder incremental TRD - S360BD

Cuando se requiere más resolución, es posible para el contador computar los márgenes de dirección y rastreo de la serie de pulsos de un canal, el cual duplica el número de pulsos contados para una rotación o pulgada de movimiento. Al contar ambos márgenes de dirección y de rastreo de ambos canales darán una resolución x4 (Figura 2.34).

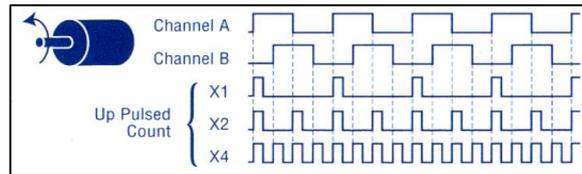


Figura 2.34: Resolución de un encoder incremental

En la Figura 2.35 se indica las conexiones para el funcionamiento del encoder incremental:

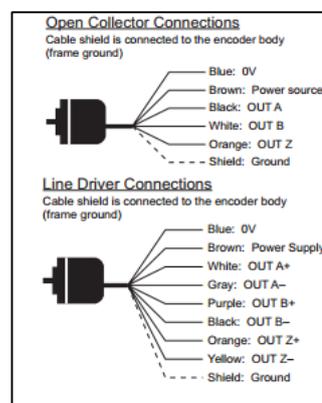


Figura 2.35: Conexión del encoder TRD - S360BD

El encoder TRD - S360BD está acoplado a la mesa giratoria con el fin de calibrar cada revolución que realice la mesa mediante el control de la tarjeta Arduino MEGA 2560, como se indica en la Figura 2.36.

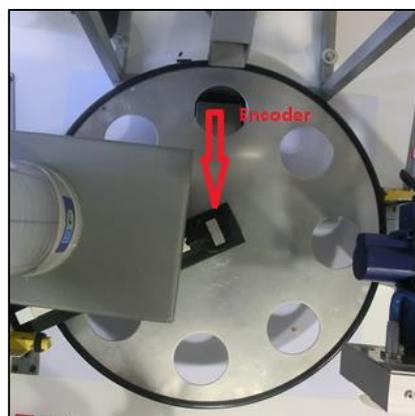


Figura 2.36: Posición del encoder en la mesa giratoria

Una vez especificados los elementos que conforman la Etapa 1 se procede a unificarlos en un solo bloque de tal manera que cumpla con los requerimientos establecidos. En la Figura 2.37 se indica la forma final de la Etapa 1.

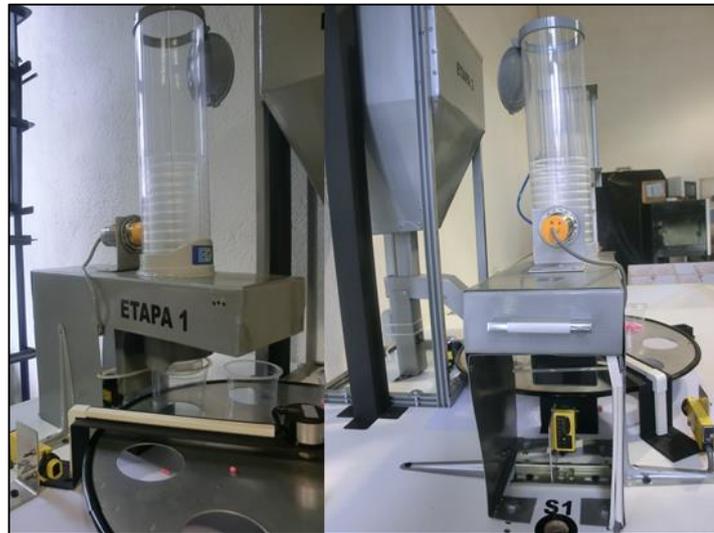


Figura 2.37: Etapa 1

2.3.2. Construcción del dosificador de sólidos

El dosificador de sólidos será denominado como Etapa 2 el cual se encuentra conformado por los siguientes elementos:

a. Motor vibratorio

El motor vibratorio (Figura 2.38) produce una fuerza de excitación el cual combina potencia y fuente de vibración, está conformado por un eje a desnivel ubicado en el eje central del motor lo que permite que el motor genere vibraciones.

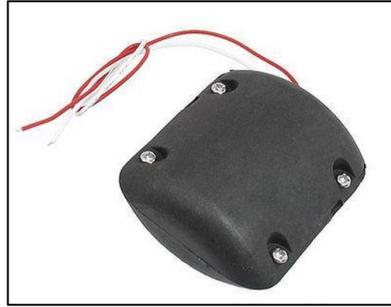


Figura 2.38: Motor vibratorio

En la Tabla 2.4 contiene las especificaciones del motor vibratorio.

Tabla 2.4: Especificaciones del motor vibratorio

Nombre del producto	Motor de vibración
Tensión nominal	DC 8V
Velocidad sin carga	4200 RPM
Color principal	Negro/Blanco
Peso	72 gr.

En la Figura 2.39 se muestra el motor vibratorio, está instalado en el centro del pedestal. Permite que se distribuya de manera uniforme las vibraciones emitidas por el motor para dispensar el material sólido.

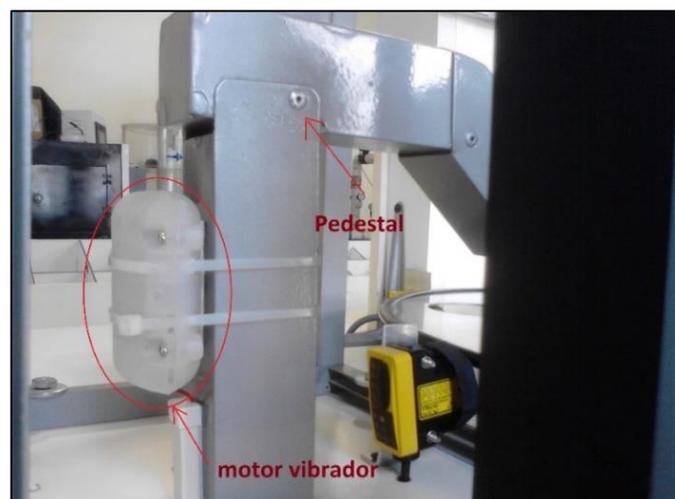


Figura 2.39: Ubicación del motor vibrador

b. Sensor láser QS30LDL

El sensor Láser QS30LDL está instalado para detectar los recipientes (vasos) posicionados en la mesa giratoria, pero a diferencia del sensor Láser QS30LDL perteneciente a la Etapa 1 su ajuste es distinto ya que el material a detectar es plástico, como se indica en la Figura 2.40.



Figura 2.40: Sensor Láser QS30LDL. a) detectando el recipiente. b) Sin detección

c. Sensor ultrasónico U-GAGE T30UIPAQ

El sensor ultrasónico U-GAGE T30UIPAQ (Figura 2.41) es ideal para entornos exigentes, su programación de simple pulsador proporciona flexibilidad para una gran variedad de aplicaciones.

Excelente para aplicaciones de medición, tales como la detección de niveles de líquido de un tanque o materiales sólidos, por ejemplo, la determinación de alturas de caja con fines de clasificación.

Cada sensor incluye tanto una salida analógica y una salida discreta, que puede ser programado independientemente con diferentes límites de la ventana o junto con límites idénticos. Cada salida tiene la opción de ser fijado con un punto de detección a distancia de un Set Point centrado a 10 mm de la ventana.



Figura 2.41: Sensor U-GAGE T30UIPAQ

El sensor ultrasónico U-GAGE T30UIPAQ mide desde 150 mm de distancia (zona muerta desde 0-150 mm) hasta 1 m máximo de distancia (Figura 2.42).

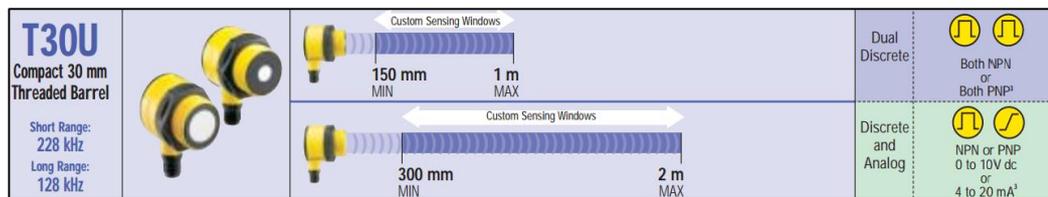


Figura 2.42: Distancia de sensado del sensor T30UIPAQ

Para más información sobre el sensor ultrasónico T30UIPAQ lea el Anexo A-5. En la Figura 2.43 se indica la ubicación del sensor ultrasónico en el dosificador de sólidos para medir el nivel del material que contiene en su interior.



Figura 2.43: Ubicación del sensor ultrasónico U-GAGE T30UIPAQ

d. DBQ5 portable demo box

Como se indica en la Figura 2.44 el módulo portable DBQ5 es un complemento del sensor ultrasónico U-GAGE T30UIPAQ, este dispositivo suministra una señal de corriente y una señal discreta (información más detallada en el Anexo A-6).

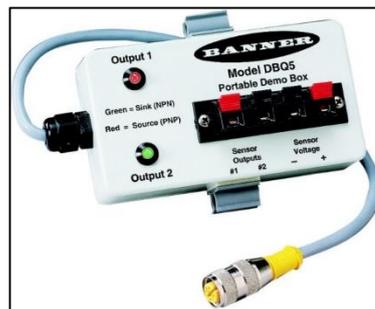


Figura 2.44: Módulo DBQ5 Portable Demo Box

e. Tarjeta arduino uno

Arduino/Genuino Uno es una placa electrónica basada en el ATmega328P (Figura 2.45). Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; simplemente conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC (Revise el circuito esquemático de la tarjeta Arduino Uno en el Anexo B-1).

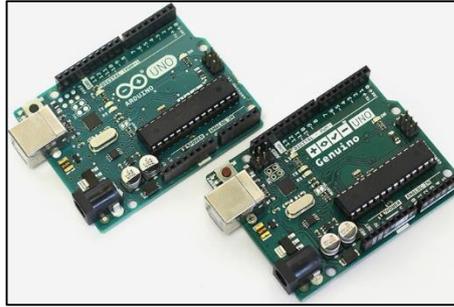


Figura 2.45: Tarjeta Arduino/Genuino Uno

Esta tarjeta Arduino Uno al igual que la tarjeta MEGA 2560 esta encargada tanto de recibir y enviar las señales analógicas como digitales necesarias para activar o desactivar actuadores, indicadores y dispositivos conectados en la tarjeta. Está ubicado dentro de la caja de control conectado a la Etapa 2.

f. Balanza electrónica

La balanza se la describe como un aparato creado por el hombre para calcular el peso de un elemento (Figura 2.46). Este procedimiento se realiza a partir de que se coloca tal elemento u objeto a pesar en una superficie y la misma calcula, a través de diferentes métodos, su peso casi de manera instantánea.



Figura 2.46: Balanza electrónica

Esta balanza contiene una celda de carga (conformada de una galga extensiométrica y un dispositivo férrico) que dependiendo del límite de peso máximo a medir (máxima deformación de la galga extensiométrica) es

acondicionada a una tarjeta interna para su respectiva medición (gramos, kilogramos, etc.) como se indica en la Figura 2.47.

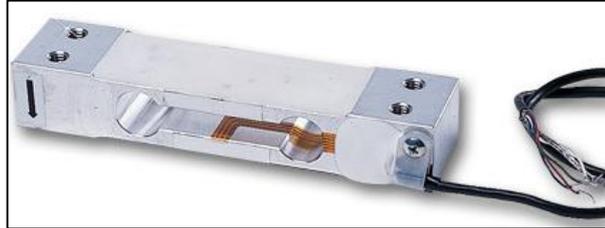


Figura 2.47: Celda de carga de 200 gramos

En la Figura 2.48 se observa la ubicación de la balanza en la Etapa 2.

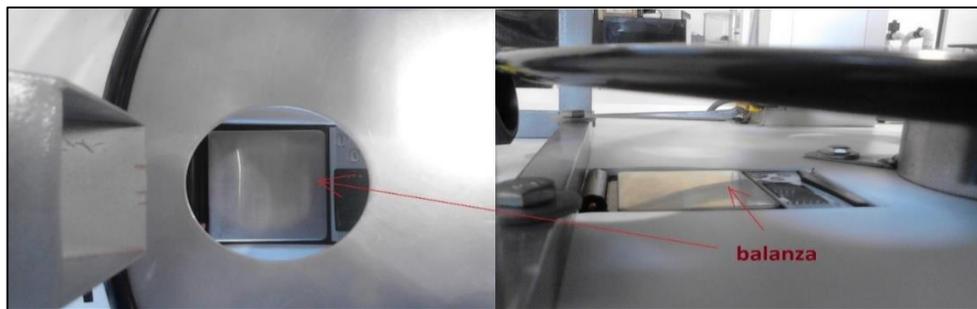


Figura 2.48: Ubicación de la balanza

g. Módulo HX711

Basada en la tecnología patentada por Avia Semiconductor's, el módulo HX711 (Figura 2.49) es un conversor ADC (Analog Digital Converter por sus siglas en inglés) de 24 bits, diseñado para escalas de peso y aplicaciones de control industrial para interconectarse directamente con un puente weathstone.

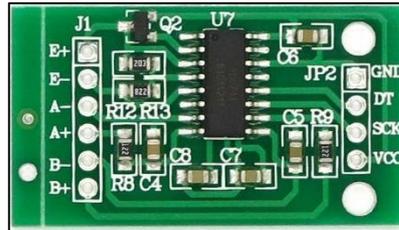


Figura 2.49: Módulo HX711

Sus características generales son:

- Modelo: HX711
- Voltaje operacional: 5V
- Voltaje de entrada diferencial: $\pm 40\text{mV}$
- Corriente de operación: $< 10\text{mA}$
- Conversión: 24 Bits
- Frecuencia de operación: 80Hz
- Interfaz: TTL232 serial

h. Driver L298N

El driver L298N descrito anteriormente en la construcción de la Etapa 1, se encuentra conectado al motor vibratorio para poder dispensar el material contenido en la tolva de manera exacta gracias a la ayuda de la balanza electrónica descrita anteriormente y un control difuso.

Una vez especificados los elementos que conforman la Etapa 2 se procede a unificarlos en un solo bloque de tal manera que cumpla con los requerimientos establecidos. En la Figura 2.50 se indica la forma final de la Etapa 2.



Figura 2.50: Etapa 2

2.3.3. Construcción de la envasadora

La envasadora será denominada como Etapa 3 la cual se encuentra conformada por los siguientes elementos:

a. Sensor capacitivo AMICO LJC30A3-H-Z/AY

El sensor capacitivo AMICO LJC30A3-H-z/AY descrito en la construcción de la Etapa 1 se encuentra ubicado en la parte inferior del contenedor de cubiertas (Figura 2.51) de tal manera que detecte la presencia o ausencia de cubiertas generando una señal digital la cual es procesada por una tarjeta de control Arduino Uno.



Figura 2.51: Ubicación del sensor capacitivo AMICO LJC30A3-H-Z/AY

b. Bomba de vacío

La bomba de vacío (Figura 2.52) cumple un papel importante, ya que proporciona una fuerza de succión gracias a la ayuda de un motor DC de 12 voltios que absorbe aire, acoplado a una ventosa para extraer tapas del contenedor.



Figura 2.52: Bomba de vacío AIRPRO D2026B

La bomba de vacío se encuentra ubicada en la parte superior de la envasadora, como se indica en la Figura 2.53.



Figura 2.53: Ubicación en la envasadora de la bomba de vacío

c. Servomotor tower pro mg 995

La envasadora está conformada por un servomotor que es el mecanismo que posiciona un pistón eléctrico el cual está acoplado a una ventosa que extrae la cubierta mediante la acción de succión de la bomba de vacío, para ello se posiciona el servomotor a 180 grados y 0 grados para extraer y depositar la cubierta sobre el recipiente colocado en la mesa giratoria.

El servomotor utilizado cuenta con engranes metálicos (Figura 2.54) para ayudar a la fijación del pistón debido al peso del mismo, con ello se evita que tenga un juego en el sellado.



Figura 2.54: Servomotor de engranes metálicos MG995

El servomotor dispone de las siguientes características:

- Fabricante: Tower Pro
- Voltaje de operación: 4.8 V a 7.2 V
- Velocidad de operación: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
- Torque detenido: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Con doble cojinete
- Ángulo de rotación: 120° aprox.
- Banda muerta: 5 μ s
- Peso: 55 g
- Dimensiones: Largo 40.7 mm, ancho 19.7 mm, altura 42.9 mm aprox.

- Largo del cable: 31 cm aprox.
- Con piñonera metálica
- Incluye 2 brazos o cuernos (horns), 1 disco, tornillo de sujeción, 4 tornillos para montaje del servo, 4 gomas de suspensión, 4 ribetes metálicos para las gomas, y cable de conexión con conector universal tipo "S" compatible con la mayoría de receptores incluyendo Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum y Hitec, entre otros.

d. Pistón eléctrico

Al igual que en la Etapa 1 el pistón eléctrico en la Etapa 3 forma parte de la envasadora cumpliendo un papel importante, ya que este va empotrado junto al servomotor en la parte media de la estructura base de la envasadora y realiza la acción de extraer la tapa y sellar el recipiente en conjunto con la bomba de vacío y una ventosa, como se indica en la Figura 2.55.

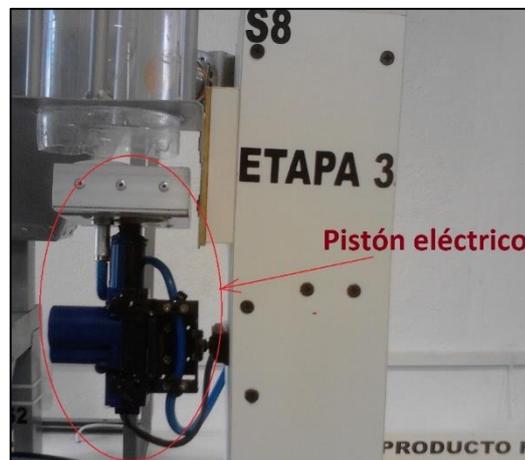


Figura 2.55: Ubicación del pistón eléctrico

e. Ventosa o copa de succión

Una copa de succión (Figura 2.56) hecha de silicona, que forma parte de dispositivos de vacío, tiene características de resistencia al aceite, al combustible, al agua, al ozono y a la intemperie. Esta ventosa de silicona es más fácil de construir, más conveniente, y tiene un aspecto fuerte. La ventosa

consta de las siguientes características:

- Nombre del producto: Vacío ventosa; Material: silicona, metal.
- Color: Blanco claro, tono de plata.
- Tubo de aire Conector Diámetro: 4,5 mm / 0,2".
- Diámetro de succión: 40 mm / 1,6 ", Altura total: 77 mm / 3".
- Peso neto: 33g.



Figura 2.56: Copa de succión o ventosa

f. Sensor fotoeléctrico banner mini beam SME312D

El sensor fotoeléctrico banner mini beam SME312D (Figura 2.57) es el encargado de generar una señal digital el momento de detectar el recipiente lleno para poder envasarlo. Al igual que el sensor láser de la Etapa 1 y 2 tiene un modo de programación de tipo botón lo que permite realizar una programación fácil para la detección del material requerido.



Figura 2.57: Sensor fotoeléctrico banner mini beam SME312D

En el Anexo A-3 se indica las especificaciones del sensor fotoeléctrico banner mini beam SME312D.

El sensor fotoeléctrico banner mini beam SME312D se encuentra ubicado en la Etapa 3 como se indica en la Figura 2.58.

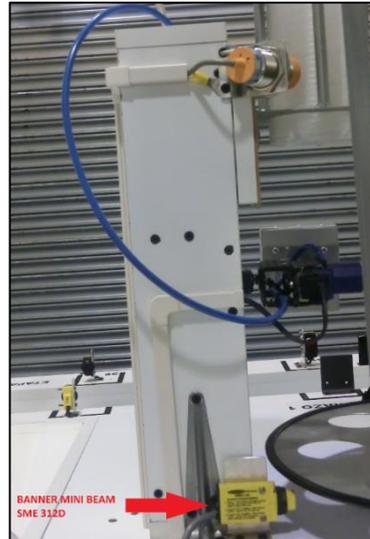


Figura 2.58: Ubicación del sensor banner mini beam SME312D

Una vez especificados los elementos que conforman la Etapa 3 se procede a unificarlos en un solo bloque de tal manera que cumpla con los requerimientos establecidos. En la Figura 2.59 se indica la forma final de la Etapa 3.

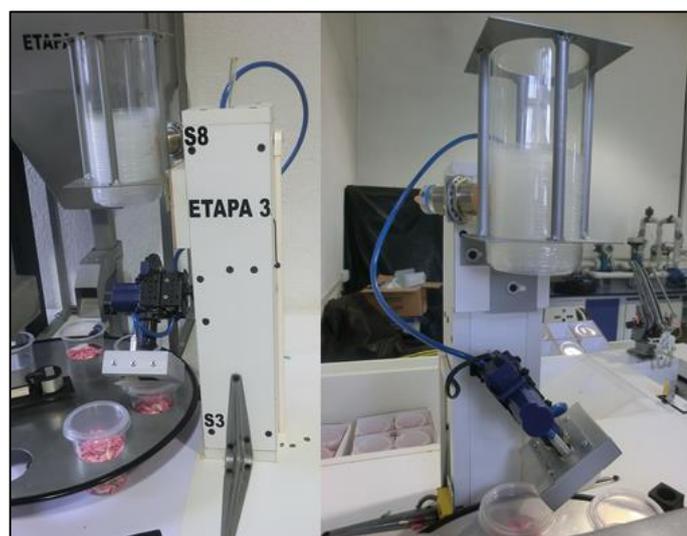


Figura 2.59: Etapa 3

2.3.4. Construcción de la etapa de organización

La etapa de organización o Etapa 4 está formada por dispositivos mecánicos y electrónicos que ayudan a la organización de los recipientes ya envasados en cajas proporcionadas desde un contenedor y extraídas por la banda transportadora gracias al alto coeficiente de rozamiento que tiene el material de la cinta transportadora, dichas cajas son posicionadas de manera que un brazo robótico coloque los recipientes en los 4 compartimientos de cada caja. Todos los elementos anteriormente mencionados se los detallan a continuación.

a. Brazo robótico banshi R680

Banshi es un robot asequible para el aficionado. Es ideal para aprender los fundamentos de la electrónica, la mecánica y la programación. El Banshi es controlado por un potente microcontrolador ATMEGA64 que se puede programar a través de herramientas de código abierto en C.

El robot viene con muchos programas de ejemplo ya escritos. Se los puede descargar fácilmente al robot mediante la interfaz USB suministrado y el software RobotLoader. O escribir sus propios programas personalizados mediante el software libre WinAVR de código abierto.

Se puede controlar el robot usando el teclado o el software RACS que puede grabar los movimientos del Banshi y reproducirlos. Las entradas y salidas, junto con el sistema de bus I2C flexibles, permite la adición de módulos para mejorar el desempeño del robot, en la Figura 2.60 se indica el brazo robótico banshi R680.

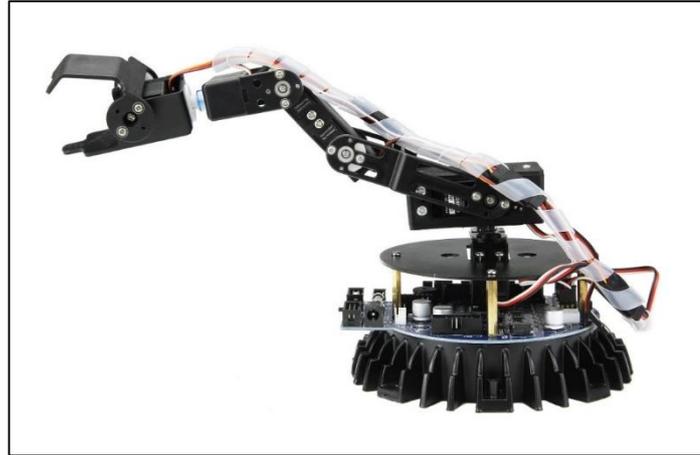


Figura 2.60: Brazo robótico banshi R680

b. Sensor banner SMU315D

El sensor banner SMU315D (Figura 2.61) es excelente para detectar objetos de gran tamaño y de reflectividad adecuada a corta distancia. Son útiles para detectar objetos pequeños y materiales translúcidos transparentes a corta distancia.



Figura 2.61: Sensor fotoeléctrico banner SMU315D

En el Anexo A-4 se indica las especificaciones del sensor fotoeléctrico banner con salida tipo relé SMU315D.

c. Sensor láser QS30LDL

El sensor Láser QS30LDL descrito en la Etapa 1 y 2, está instalado para posicionar las cajas extraídas del contenedor mediante la banda

transportadora, pero a diferencia del sensor Láser QS30LDL perteneciente a la Etapa 1 y 2 su ajuste es distinto, en este caso el material a detectar se denomina syntra tal como se indica en la Figura 2.62.

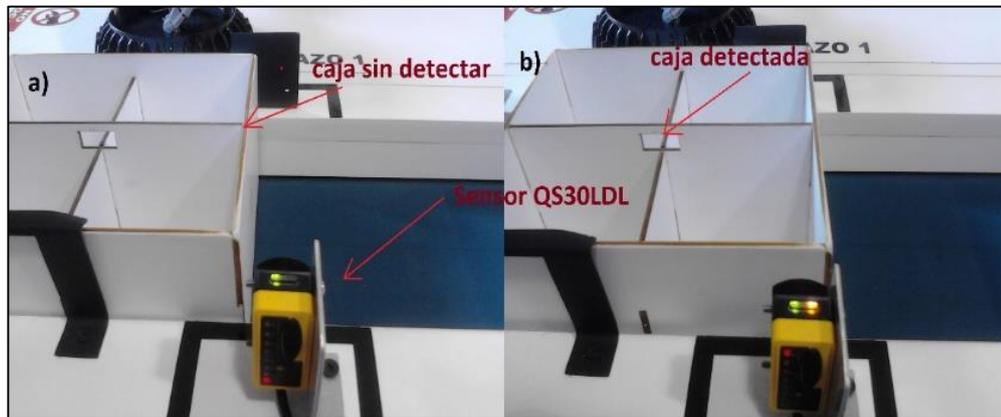


Figura 2.62: Sensor QS30LDL a) sin detección de caja. b) con detección de caja

d. Tarjeta arduino MEGA 2560

Al igual que la Etapa 1 la tarjeta Arduino MEGA 2560 se encarga de procesar las señales analógicas y digitales, también interrupciones que son necesarias para activar o desactivar actuadores e indicadores.

e. Motor de alto torque

El motor de 12VDC al igual que el motor de la Etapa 1, está acoplado para el movimiento de la banda transportadora por su alto torque (Figura 2.63).



Figura 2.63: Motor de alto torque Etapa 4

Este motor DC está acoplado a un eje de la banda transportadora que mediante una polea transmite la fuerza de movimiento de toda la banda equitativamente como se muestra en la Figura 2.64.

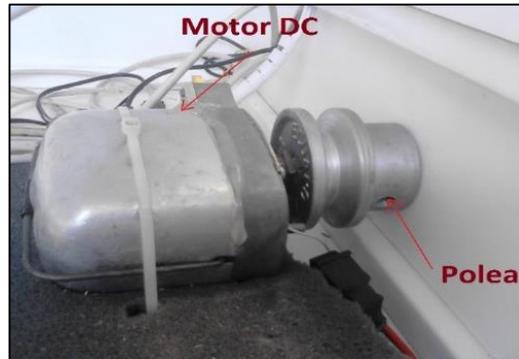


Figura 2.64: Motor DC acoplado a la banda transportadora

f. Sensor fotoeléctrico G30-3A70NA

Son sensores fotoeléctricos de la serie Gxx son de uso múltiple utilizados en sistemas y equipos de máquinas sin vigilancia. En el interior del sensor hay un emisor de señal de fotos y el receptor de la señal de retorno. Permiten orientar el ajuste de la distancia y se desarrollan en base a la transición PNP y NPN. Son de carcasa cilíndrica de metal resistente a altas vibraciones resistentes al agua y al aceite (Figura 2.65).



Figura 2.65: Sensor fotoeléctrico G30-3A70NA

En el Anexo A-2 se indica las especificaciones del sensor fotoeléctrico SASSIN G30-3A70NA.

Una vez especificados los elementos que conforman la Etapa 4 se procede a unificarlos en un solo bloque de tal manera que cumpla con los requerimientos establecidos. En la Figura 2.66 se indica la forma final de la Etapa 4.



Figura 2.66: Etapa 4

2.3.5. Construcción de la etapa 5 o de inspección

La etapa de inspección o Etapa 5 está conformada por dispositivos mecánicos y electrónicos que ayudan a la clasificación del producto final. Los elementos utilizados en su construcción se detallan a continuación.

a. Sensor fotoeléctrico QS18VP6R

La serie WORLD-BEAM QS18 de sensores fotoeléctricos ofrece un rendimiento óptico excepcional en una carcasa QS18 compacta y un diseño de carcasa de ahorro de espacio exclusivo que sustituye directamente los sensores de tipo rectangular/cilíndrico (Figura 2.67).

Los sensores tienen una detección fiable cuando hay un fondo fijo y el color del objeto o la forma pueden variar. La serie QS18 de sensores está disponible en los siguientes modos de detección: opuesto, retrorreflectivo polarizado o

sin polarizar, convergente, difuso de gran angular o normal, láser, ultrasónico, óptica de fibra de vidrio o plástico, campo fijo o ajustable.

En el Anexo A-7 se indica las especificaciones del sensor fotoeléctrico QS18VP6R.



Figura 2.67: Sensor fotoeléctrico QS18VP6R

b. Brazo robótico Saintsmart

Construido de material PVC, procesamiento CNC (conexión conjunta de apoyo activo). Adopta Servo de engranaje metálico de la serie MG995 de 55g de 3 pcs. La distancia máxima es de 300mm, usando tres MG995 de cobre 55g, estos servos trabajan de 0 a 180 grados (Figura 2.68).



Figura 2.68: Brazo robótico Saintsmart

c. Balanza

Al igual que en la Etapa 2, la balanza toma el peso del producto final para poder verificar si pasa o reprueba la inspección.

d. Tarjeta Arduino Uno

Esta tarjeta Arduino Uno al igual que en la Etapa 2 cumple la función de recibir y enviar las señales analógicas como digitales necesarias para activar o desactivar actuadores, indicadores y dispositivos conectados en la tarjeta.

Una vez especificados los elementos que conforman la Etapa 5 se procede a unificarlos en un solo bloque de tal manera que cumpla con los requerimientos establecidos. En la Figura 2.69 se indica la forma final de la Etapa 5.

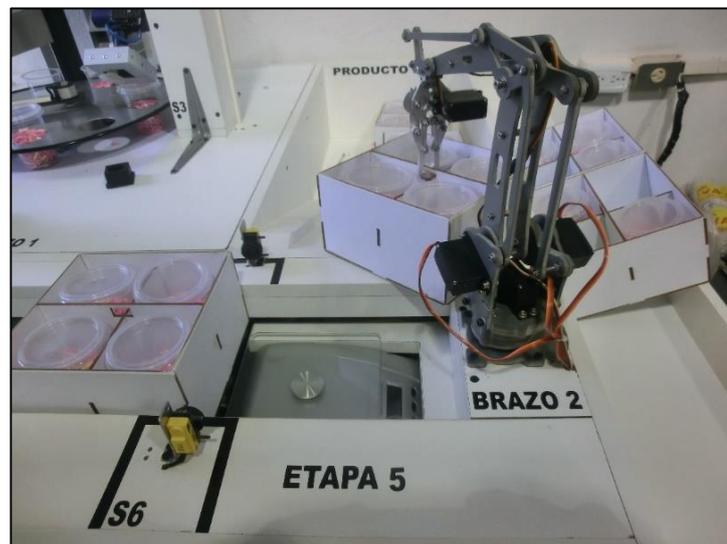


Figura 2.69: Etapa 5

La unificación de todas las etapas y componentes que conforman el módulo didáctico de llenado y envasado de sólidos se indica en la Figura 2.70.



Figura 2.70: Planta del módulo didáctico de llenado y envasado de sólidos

2.4. Automatización de la planta

En esta parte se enfoca plenamente a la parte de programación en la plataforma Arduino, luego de construir la planta y adicionar elementos de control y de medida. Cada una de las etapas a automatizar dispone de una tarjeta controladora lo que permite que el módulo didáctico funcione paralelamente, optimizando los recursos y reduciendo los tiempos de producción.

A continuación, se detallan aspectos de programación en las tarjetas controladoras, el manejo de los actuadores y adquisición de señales generadas por sensores, para realizar controles que permitan un correcto funcionamiento del módulo didáctico.

2.4.1. Arduino software (IDE)

El entorno de desarrollo integrado Arduino o Arduino software (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús.

Los programas escritos utilizando Arduino Software (IDE) se llaman bocetos (Sketches). Estos bocetos se escriben en el editor de texto y se guardan con la extensión de archivo. ino. El editor tiene funciones para cortar/pegar y para buscar/reemplazar texto. El área de mensajes proporciona información muestra el texto de salida para el entorno de Arduino, incluyendo mensajes de información y los posibles errores en la compilación. La esquina derecha inferior de la ventana muestra el dispositivo configurado y puerto serie. Los botones de la barra le permiten verificar y cargar programas, crear, abrir y guardar dibujos, y abrir el monitor de serie, como se indica en la Tabla 2.5.

Tabla 2.5: Botones de la barra de herramientas Arduino

	Verify Chequea los errores al compilar el programa.
	Upload Compila el programa y lo carga en el dispositivo configurado.
	New Crea un nuevo sketch (boceto de escritura).
	Open Presenta un menú de todos los programas de su "sketchbook". Un clic sobre uno de ellos lo abrirá en la ventana actual.
	Save Guarda el sketch.
	Serial Monitor Abre el monitor serie.

Además, se encuentran comandos adicionales en los cinco menús: Archivo, Editar, Sketch, Herramientas, Ayuda. Los menús son sensibles al contexto, lo que significa que sólo aquellos elementos pertinentes a la labor que está llevando a cabo están disponibles.

a. Archivo

- **New:** Crea una nueva instancia del editor, con la estructura mínima en su lugar.
- **Open:** Permite cargar archivos al boceto a través de la navegación en el disco duro del ordenador y las carpetas.
- **Open Recent:** Proporciona una breve lista de los bocetos (sketches) más recientes, listos para ser abiertos.
- **Sketchbook:** Muestra los bocetos (sketches) actuales dentro de la estructura de carpetas; al hacer clic en cualquier nombre se abre el programa correspondiente en una nueva instancia del editor.
- **Examples:** Cualquier ejemplo proporcionado por el software de Arduino (IDE) o biblioteca aparece en este elemento del menú. Todos los ejemplos se estructuran en un árbol que permite un fácil acceso por tema o biblioteca.
- **Close:** Cierra la instancia del software de Arduino de la que se hace clic.
- **Save:** Guarda el programa con el nombre actual. Si el archivo no ha sido nombrado antes, un nombre se proporcionará en una ventana "Guardar como ..".
- **Save as ... :** Permite guardar el dibujo actual con un nombre diferente.
- **Page Setup:** Se muestra la ventana de configuración de página para la impresión.
- **Print:** Envía el programa actual a la impresora de acuerdo con la configuración definida en Configurar página.
- **Preferences:** Abre la ventana de preferencias donde algunos ajustes del IDE se pueden personalizar, como el idioma de la interfaz IDE.
- **Quit:** Cierra todas las ventanas del IDE. Los mismos bocetos cuando Quit fue elegido son reabiertos automáticamente la próxima vez que se inicie.

b. Editar

- **Undo/Redo:** Vuelve de uno o más pasos que se realizó durante la edición; cuando vuelva, puede seguir adelante con Redo.
- **Cut:** Elimina el texto seleccionado desde el editor y lo coloca en el portapapeles.
- **Copy:** Duplica el texto seleccionado en el editor y lo coloca en el portapapeles.
- **Copy for Forum:** Copia el código de su boceto en el portapapeles en una forma adecuada para su publicación en el foro, con coloreado de sintaxis.
- **Copy as HTML:** Copia el código de su boceto en el portapapeles como HTML, adecuado para incrustar en páginas web.
- **Paste:** Coloca el contenido del portapapeles en la posición del cursor, en el editor.
- **Select All:** Selecciona y resalta todo el contenido del editor.
- **Comment/Uncomment:** Coloca o retira el marcador de comentario “//” al comienzo de cada línea seleccionada.
- **Increase/Decrease Indent:** Suma o resta un espacio al principio de cada línea seleccionada, moviendo el texto de un espacio en la derecha o la eliminación de un espacio al principio.
- **Find:** Abre la ventana “Find and Replace” donde se puede especificar el texto a buscar en el interior del sketch actual de acuerdo con varias opciones.
- **Find Next:** Destaca la siguiente aparición - en caso - de la cadena especificada como el elemento de búsqueda en la ventana Buscar, en relación con la posición del cursor.
- **Find Previous:** Destaca la aparición anterior - en caso - de la cadena especificada como el elemento de búsqueda en la ventana Buscar en relación con la posición del cursor.

c. Sketch

- **Verify/Compile:** Comprueba el sketch de los errores de compilación en él; se informará el uso de memoria para el código y las variables en el área de la consola.
- **Upload:** Compila y carga el archivo binario en el dispositivo configurado a través del puerto configurado.
- **Upload Using Programmer:** Esto sobrescribirá el gestor de arranque en el dispositivo; usted tendrá que usar Herramientas> Burn Bootloader para restaurarlo y poder subir al puerto serie USB. Sin embargo, se le permite utilizar toda la capacidad de la memoria Flash para su sketch.
- **Export Compiled Binary:** Guarda un archivo .hex que puede mantenerse como archivo o enviarse a la junta usando otras herramientas.
- **Show Sketch Folder:** Abre la carpeta de archivo actual.
- **Include Library:** Añade una biblioteca para su sketch mediante la inserción de declaraciones `#include` en el inicio de su código. Para más detalles, consulte las bibliotecas de abajo. Además, desde este punto de menú puede acceder al Administrador de Biblioteca e importar nuevas bibliotecas de archivos .zip.
- **Add File...:** Añade un archivo de origen en el sketch (que se copiará desde su ubicación actual). El nuevo archivo aparece en una nueva pestaña en la ventana del sketch. Los archivos pueden ser retirados del boceto usando el menú de la ficha clic accesible en el icono de triángulo pequeño debajo del monitor serie uno en el lado derecho o la barra de herramientas.

d. Tools

- **Auto Format:** Formatea el código muy bien: es decir, los guiones para que de apertura y cierre de llaves se alinean.
- **Archive Sketch:** Archiva una copia del sketch actual en formato .zip. El archivo se coloca en el mismo directorio que el boceto.
- **Fix Encoding & Reload:** Corrige posibles discrepancias entre la codificación editor de mapas.

- **Serial Monitor:** Abre la ventana del monitor de serie y se inicia el intercambio de datos con cualquier placa conectada en el puerto seleccionado. Esto por lo general restablece el tablero.
- **Board:** Seleccione la junta que está utilizando. Vea a continuación las descripciones de las diversas juntas.
- **Port:** Este menú contiene todos los dispositivos serie (reales o virtuales) en su máquina. Debe actualizar automáticamente cada vez que abra el menú de herramientas de primer nivel.
- **Programmer:** Para seleccionar un programador hardware al programar una junta o chip y no utilizar la conexión USB de serie a bordo. Normalmente no será necesario, pero si usted está quemando un gestor de arranque a un nuevo microcontrolador va a utilizar esto.
- **Burn Bootloader:** Los elementos de este menú le permiten grabar un gestor de arranque en el microcontrolador en una placa Arduino. Esto no es necesario para el uso normal de una placa Arduino o Genuino, pero es útil si usted compra un nuevo ATmega microcontrolador (que normalmente viene sin un gestor de arranque). Asegúrese de que ha seleccionado el consejo correcto de la Juntas de menú antes de grabar el gestor de arranque en el tablero.

2.4.2. Referencia del lenguaje arduino

Los programas de Arduino se pueden dividir en tres partes principales: la estructura, los valores (variables y constantes), y funciones como se indica en la Tabla 2.6.

Tabla 2.6: Estructura, valores y funciones utilizadas en Arduino

Estructuras	Variables	Funciones
setup() loop()	Constantes	Digital I/O
Estructuras de control	HIGH LOW INPUT OUTPUT INPU T_PULLUP LED_BUILTIN	pinMode() digitalWrite() digitalRead()
if if...else for switch case while do... while break continue return goto	true false integer constants floating point constants	Analog I/O
Sintaxis	Tipos de datos	analogReference() analogRead() analogWrite() - PWM Due & Zero only analogReadResoluti on() analogWriteResoluti on()
;(semicolon) { } (curly braces) // (single line comment) /* */ (multi-line comment) #define #include	void boolean char unsigned char byte int unsigned int word long unsigned long short float double string - char array String - object array	Advanced I/O
Operadores aritméticos	Conversion	tone() noTone() shiftOut() shiftIn() pulseIn()
= (assignment operator) + (addition) - (subtraction) * (multiplication) / (division) % (modulo)	char() byte() int() word() long() float()	Tiempo
Operadores de comparación	Variables de alcance y calificadores	millis() micros() delay() delayMicroseconds()
== (equal to) != (not equal to) < (less than) > (greater than) <= (less than or equal to)	variable scope static volatile const	Matemática
		min() max() abs() constrain() map() pow() sqrt()
		Trigonometría
		sin()

<p>>= (greater than or equal to)</p> <p>Operadores booleanos</p> <p>&& (and) (or) ! (not)</p> <p>Operadores de acceso de puntero</p> <p>* dereference operator & reference operator</p> <p>Operadores bit a bit</p> <p>& (bitwise and) (bitwise or) ^ (bitwise xor) ~ (bitwise not) << (bitshift left) >> (bitshift right)</p> <p>Operadores compuestos</p> <p>++ (increment) -- (decrement) += (compound addition) -= (compound subtraction) *= (compound multiplication) /= (compound division) &= (compound bitwise and) = (compound bitwise or)</p>	<p>Utilidades</p> <p>sizeof() PROGMEM</p>	<p>cos() tan()</p> <p>Números randómicos</p> <p>randomSeed() random()</p> <p>Bits y bytes</p> <p>lowByte() highByte() bitRead() bitWrite() bitSet() bitClear() bit()</p> <p>Interrupciones externas</p> <p>attachInterrupt() detachInterrupt()</p> <p>Interrupciones</p> <p>interrupts() noInterrupts()</p> <p>Comunicación</p> <p>Serial Stream</p>
--	---	--

2.4.3. Librerías

El entorno de Arduino se puede ampliar mediante el uso de las librerías, al igual que la mayoría de las plataformas de programación. Las librerías proporcionan funcionalidad adicional para el uso en sketches, por ejemplo, trabajar con el hardware o la manipulación de los datos. Un número de librerías viene instalado con el IDE, pero también se puede descargar o crear una propia.

Para utilizar una biblioteca en un sketch, selecciónelo de Sketch> Import Library.

Las librerías utilizadas en el proyecto se describen a continuación:

a. Librería Adafruit Motor Shield V2.0.h

Esta librería permite manejar motores DC, paso a paso y servomotores, mediante la tarjeta electrónica de Adafruit para Arduino y la comunicación I2C sin necesidad de la utilización de la librería Wire.h. Esta librería tiene las siguientes funciones:

- `Adafruit_MotorShield(uint8_t addr = 0x60)`: Añade una dirección por defecto (0x60), para la comunicación I2C con el dispositivo Arduino.
- `void begin(uint16_t freq = 1600)`: Inicializa el escudo (shield) con una frecuencia de 1.6 Khz por defecto, debe ser llamado en la función de configuración (setup).
- `Adafruit_DCMotor *getMotor(uint8_t n)`: Declara el canal de los 4 existentes para el manejo de motores DC.
- `void run(uint8_t)`: Indica el sentido de giro (BACKWARD o FORWARD) del motor DC y si se detiene (RELEASE).
- `void setSpeed(uint8_t)`: Establece un valor entre 0 y 255 para controlar la velocidad del motor DC.

b. Librería EFLL MASTER.h

Esta librería se encuentra escrita en C ++ / C, por lo que eFLL MASTER es una librería para cualquier sistema embebido. No tiene explícitas limitaciones en la cantidad de controles Fuzzy, reglas Fuzzy, entradas o salidas, en si está limitada por la capacidad de procesamiento y almacenamiento de cada microcontrolador.

La librería utiliza los métodos de máximos y mínimos y el mínimo de Mamdani para la inferencia y la composición y el centro de área para la defuzzificación en un universo continuo. Esta librería dispone de las siguientes funciones:

- Fuzzy object

Uso: (Fuzzy* fuzzy = new Fuzzy());. Este objeto incluye todo el sistema Fuzzy, a través de él, puede manipular los conjuntos difusos, reglas lingüísticas, entradas y salidas.

- FuzzyInput object

Uso: (FuzzyInput* peso = new FuzzyInput(1));. Este objeto agrupa todas las entradas de conjuntos difusos que pertenecen al mismo dominio.

- FuzzyOutput object.

Uso: (FuzzyOutput* vel = new FuzzyOutput(1));. Este objeto es similar a FuzzyInput, se utiliza para agrupar todas las salidas de los conjuntos difusos que pertenecen al mismo dominio.

- FuzzySet object.

Uso: (FuzzySet* pesm = new FuzzySet(30, 60, 60, 90));. Este es uno de

los principales objetos de la librería Fuzzy, con cada conjunto es posible modelar el sistema en cuestión. Actualmente la librería consta de funciones triangulares de pertenencia, trapezoidal y Singleton, que se ensamblan en puntos A, B, C y D, que son pasados por parámetro en su constructor FuzzySet (float a, float b, float c, float d) ejemplos:

La función de la pertenencia Triangular:

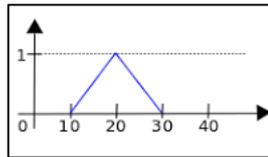


Figura 2.71: FuzzySet * fs = FuzzySet (10, 20, 20, 30);

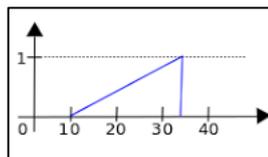


Figura 2.72: FuzzySet * fs = FuzzySet (10, 33, 33, 33);

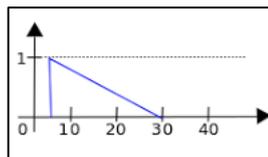


Figura 2.73: FuzzySet * fs = FuzzySet (5, 5, 5, 30);

Función de pertenencia trapezoidal:

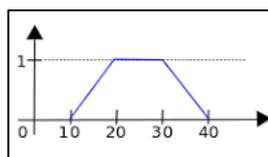


Figura 2.74: FuzzySet * fs = FuzzySet (10, 20, 30, 40);

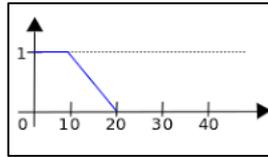


Figura 2.75: FuzzySet * fs = FuzzySet (0, 0, 10, 20);

Cualquier valor por debajo de 10 tendrá pertinencia = 1

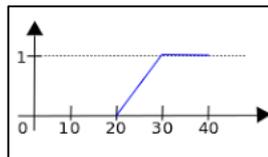


Figura 2.76: FuzzySet * fs = FuzzySet (20, 30, 40, 40);

Cualquier valor por encima de 30 tendrá pertinencia = 1

Función de la pertinencia Singleton:

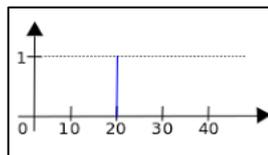


Figura 2.77: FuzzySet * fs = FuzzySet (20, 20, 20, 20);

- FuzzyRule object

Este objeto se utiliza para montar la regla de base del objeto Fuzzy. Instanciada con `FuzzyRule fr = new FuzzyRule (ID, antecedente, consecuente)`.

- FuzzyRuleAntecedent object.

Uso: `(FuzzyRuleAntecedent* ifPmuybajo = new FuzzyRuleAntecedent());`

Este objeto se utiliza para componer el objeto FuzzyRule, responsable de montar el antecedente de la expresión condicional de una regla Fuzzy.

- FuzzyRuleConsequente object

Uso: (FuzzyRuleConsequent* thenVbaja = new FuzzyRuleConsequent();). Este objeto se utiliza para representar el objeto FuzzyRule, responsable de montaje de la expresión de salida de una regla Fuzzy.

c. Librería Servo.h

Esta librería permite a una tarjeta Arduino controlar servomotores RC. Los servos estándar permiten al eje posicionarse en varios ángulos, por lo general de 0 a 180 grados. Los servos de rotación continua permiten al eje ajustarse a diversas velocidades. Esta librería dispone de las siguientes funciones:

- attach().

Uso: (*servo.attach(pin)* y *servo.attach(pin,min,max)*). Fija la variable servo a un pin, donde pin es el número donde está unido el servo a min (opcional): el ancho de pulso, en microsegundos, correspondiente al mínimo (0 grados) ángulo del servo (por defecto es 544). max (opcional): el ancho de pulso, en microsegundos, correspondiente al máximo (180 grados) ángulo del servo (por defecto es 2400).

- write().

Uso: (*servo.write(angle)*). Escribe un valor en el servo, controlando el eje en consecuencia. En un servo estándar, mueve el eje en una orientación escrita en la variable angle (en grados). En un servo de rotación continua, controla la velocidad del servo (siendo 0 la velocidad en una dirección y 180 la velocidad completa en la otra, con un valor cerca de 90 el servo queda sin movimiento).

- `writeMicroseconds()`

Uso: (`servo.writeMicroseconds(uS)`). Escribe un valor en microsegundos (uS) para el servo controlando el eje en consecuencia. En un servo estándar un valor de parámetro de 1000 es completamente hacia la izquierda, 2000 es totalmente a la derecha, y 1500 está en la mitad.

- `read()`

Uso: (`servo.read()`). Lee el ángulo actual del servo (el valor transferido a la última llamada de `write()`).

- `attached()`

Uso: (`servo.attached()`). Comprueba si la variable servo está unido a un pin.

- `detach()`

Uso: (`servo.detach()`). Desconecta la variable servo del pin. Si todas las variables servo son desconectadas, los pines 9 y 10 se puede utilizar para la salida PWM con la función `analogWrite()`.

d. Librería Wire.h

Esta librería permite comunicarse con dispositivos I2C/TWI. En las tarjetas Arduino con el diseño R3, la SDA (línea de datos) y SCL (línea de reloj) están en los conectoras macho cerca del pin AREF.

La Tabla 2.7 muestra la ubicación de los pines TWI en varias placas Arduino.

Tabla 2.7: Ubicación de pines TWI en distintas tarjetas Arduino

Tarjeta	Pines I2C/TWI
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 2 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

Esta librería dispone de las siguientes funciones:

- `begin()`

Uso: (`Wire.begin()` y `Wire.begin(address)`). Inicia la librería Wire y une al bus I2C como maestro o esclavo. Normalmente se llama una sola vez. Address: el address del esclavo (opcional); si no se especifica se une al bus como maestro.

- `requestFrom()`

Uso: (`Wire.requestFrom(address, quantity)` y `Wire.request(address, quantity, stop)`). Utilizado por el maestro para solicitar los bytes desde un dispositivo esclavo. Los bytes se pueden recuperar con las funciones `available()` y `read()`.

Si es verdadero, `requestFrom()` envía un mensaje de detención después de la respuesta, liberando al bus I2C.

Si es falso, `requestFrom()` envía un mensaje de reinicio después de la solicitud. El bus no podrá ser liberado, lo que impide que otro dispositivo maestro solicite los mensajes de respuesta. Esto permite que un dispositivo maestro envíe varias solicitudes mientras exista control de dicho dispositivo.

- `beginTransmission()`

Uso: (`Wire.beginTransmission(address)`). Inicia la transmisión del dispositivo esclavo I2C con la dirección indicada. Posteriormente, los bytes de datos para la transmisión se construyen con la función `write()` y son transmitidos llamando la función `endTransmission()`.

- `endTransmission()`

Uso: (`Wire.endTransmission()` y `Wire.endTransmission(stop)`). Finaliza la transmisión de un dispositivo esclavo que fue iniciado por `beginTransmission()` y transmite los bytes que se ha puesto en cola con la función `write()`.

- `write()`

Uso: (`Wire.write(value)`, `Wire.write(string)` y `Wire.write(data, length)`). Escribe los datos en un dispositivo esclavo en respuesta a una petición desde un maestro.

- `available()`

Uso: (`Wire.available()`). Devuelve el número de bytes para la recuperación con `read()`. Se debe llamar a un dispositivo maestro después de llamar a `requestFrom()` ó en un esclavo dentro de un controlador `onReceiver()`.

- `read()`

Uso: (`Wire.read()`). Lee un byte que fue transmitido desde un dispositivo esclavo a un maestro después de llamar a `requestFrom()` o fue transmitido desde un maestro a un esclavo.

- `onReceive()`

Uso: (`Wire.onReceive(handler)`). Registra una función que se llama cuando un

dispositivo esclavo recibe una transmisión de un maestro.

- `onRequest()`

Uso: (`Wire.onRequest(handler)`). Registra una función que se llama cuando un maestro solicita los datos desde un dispositivo esclavo.

e. Librería `Encoder.h`

La librería `Encoder` cuenta los pulsos de cuadratura de señales codificadas, los cuales son comúnmente disponibles de botones giratorios, sensores de motor o ejes y demás sensores de posición. Los encoders comúnmente deben ser conectados a dos pines. Hay tres opciones:

- Mejor rendimiento: Cuando ambas señales son conectadas a los pines de interrupción.
- Buen rendimiento: Cuando se conecta a un pin de interrupción, y el segundo pin a un pin que no tiene capacidad de interrupción.
- Bajo rendimiento: Cuando ambas señales se conectan a los pines que no tienen capacidad de interrupción.

La Tabla 2.8 muestra los pines de interrupciones de las diferentes tarjetas Arduino:

Tabla 2.8: Pines de interrupción tarjetas Arduino

Tarjeta	Pines de interrupción	Pin de LED (no usar)
Uno, Ethernet	2,3	13
Mega2560	2,3,18,19,20,21	13
Leonardo	0,1,2,3	13
Due	Todos los pines digitales	13

Los encoders de bajo costo sólo conectan sus pines a tierra. El encoder activa las resistencias pullup del chip. Si se conecta cables largos, se añaden resistencias pullup de 1K proporcionando una mejor señal.

- Encoder myEnc(pin1, pin2)

Uso: (Encoder myEnc(2, 3);). Crea un objeto tipo Encoder, por medio de dos pines. Se puede crear un encoder multiple, donde cada uno utiliza sus propios 2 pines. El primer pin activa las interrupciones.

- myEnc.read()

Uso: (myEnc.read();). Devuelve la posición en la que se encuentra el encoder actualmente de forma acumulada. Este número puede ser positivo o negativo.

- myEnc.write()

Uso: (myEnc.write());. Ajusta la posición acumulada del encoder a un nuevo número modificado por este comando.

f. Librería HX711.h

La librería Arduino de Avia Semiconductor HX711 24-Bit tiene un convertidor Análogo-Digital (ADC) para escalas de peso. Esta librería dispone de las siguientes funciones:

- tare()

Uso: (tare();). Colocar un recipiente vacío en la escala, llame a la función tare() para establecerlas lecturas a 0, llenar el recipiente y obtener el peso del contenido.

- `power_down()`

Uso: (`power_down()`). Escalas que funcionan con batería, por consiguiente, esta función permite aislar al chip del modo de baja potencia.

- `set_gain()`

Uso: (`set_gain()`). Establece el factor de ganancia y selecciona el canal, el valor por defecto es 128, lo que significa que “el canal A tiene un factor de ganancia de 128”, por lo que uno puede simplemente llamar `set_gain()`.

- `get_units()` y `get_value()`

Uso: (`get_units()`; / `get_value()`;). Pueden recibir un parámetro extra “times”, y devuelven el promedio de varias lecturas en lugar de una sola lectura.

Como calibrar la escala:

1. Llamar a `set_scale()` sin ningún parámetro.
2. Llamar a `tare()` sin ningún parámetro.
3. Coloque un peso conocido en la escala y llame la función `get_units(10)`.
4. Divida el resultado del paso 3 para su peso conocido. Obtener el parámetro que tu necesitas para pasar a la función `set_scale()`.
5. Ajustar el parámetro en el paso 4 hasta obtener una lectura precisa.

2.4.4. Diseño del controlador difuso de peso

El controlador difuso es una extensión de la tradicional lógica de Boole (valores 1 ó 0), utilizando variables lingüísticas, permite expresar valores lógicos intermedios entre falso y verdadero, que describe con mayor eficiencia el principio de incertidumbre en el mundo real. La lógica difusa permite tratar información imprecisa, como estatura media o temperatura baja, en términos de conjuntos borrosos que se combinan en reglas para definir acciones: si la

temperatura es alta entonces enfriar mucho. De esta manera, los sistemas de control basados en la lógica difusa combinan variables de entrada, definidas en términos de conjuntos difusos, por medio de grupos de reglas que producen uno o varios valores de salida.

Un controlador difuso trabaja de manera similar a un sistema convencional (PI o PID): acepta un valor de entrada, realiza cálculos, y genera un valor de salida. Esta técnica se denomina como proceso de inferencia difusa y emplea tres pasos:

- Fuzzificación, en donde una entrada real es trasladada a un valor difuso.
- Evaluación de reglas o inferencia, en donde los valores de verdad de salida difusos son calculados.
- Defuzzificación, el valor de salida difuso es trasladado a un valor de salida real.

La Figura 2.78 indica el diagrama de bloques de los elementos que compone el controlador difuso.

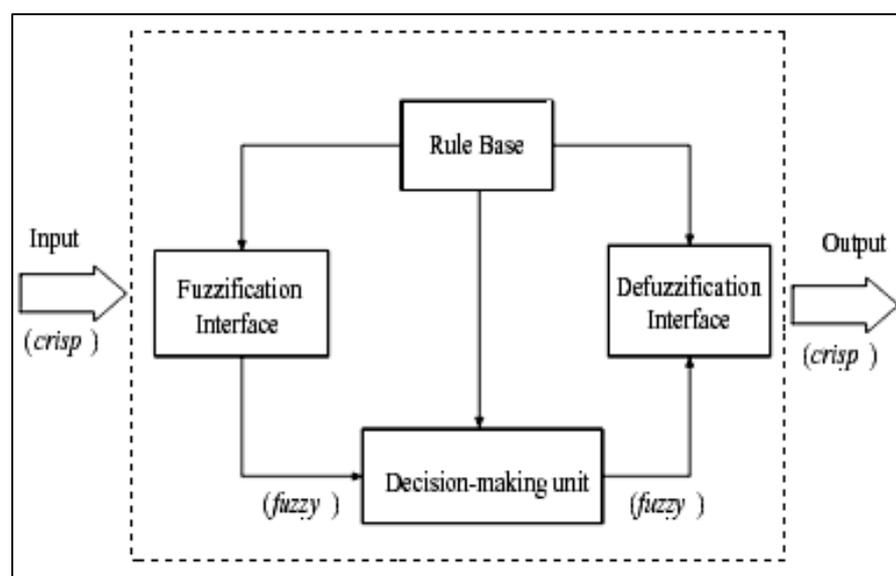


Figura 2.78: Configuración básica de un sistema de lógica difusa

a. Identificación de las variables del controlador difuso

Se identifica las siguientes variables:

- Variable de entrada: Error.
- Variable de salida: Velocidad.

b. Fuzzificación de la variable de entrada de la Etapa 2

Para la fuzzificación es necesario manipular la variable “error”.

En “error” se obtiene entre la diferencia del Set Point (establecida en el software) y el valor del peso medido por el sensor (Galga extensiométrica), este se define como la variable de entrada la cual ingresa al controlador difuso.

Se define los siguientes conjuntos de la variable lingüística “error”:

- Peso muy bajo (PMB), valores de (0, 0, 30).
- Peso Bajo (PB), valores de (20, 30, 35).
- Peso normal (PN), valores de (32, 35, 40).
- Peso alto (PA), valores de (37, 40, 50).
- Peso muy alto (PMA), valores de (45, 50, 50).

En la Figura 2.79 se indican los conjuntos definidos para la variable lingüística “error”.

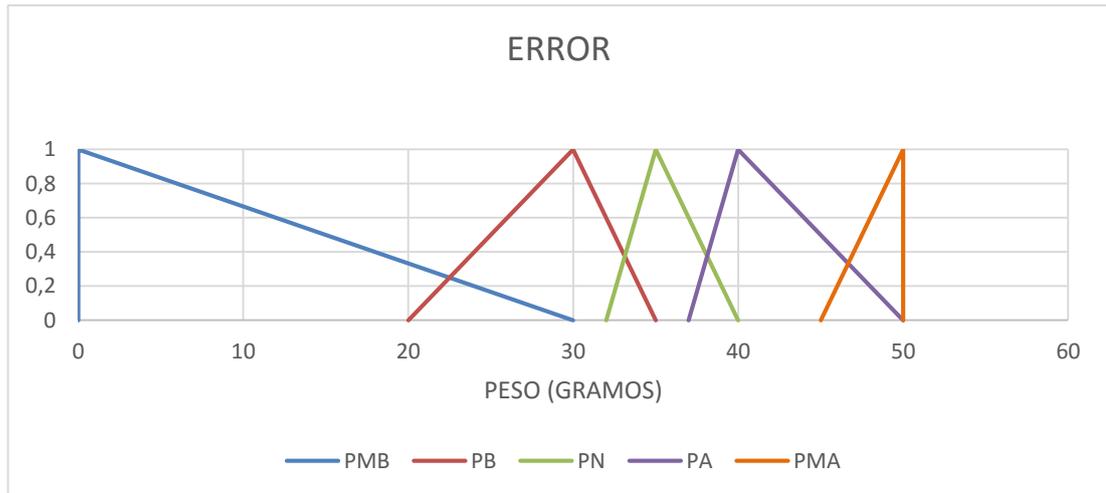


Figura 2.79: Conjuntos de la variable lingüística "error"

c. Defuzzificación de la variable de salida de la Etapa 2

Para la defuzzificación es necesario manipular la variable "velocidad".

La "velocidad" se define como la variable de salida, la cual actúa sobre un motor vibratorio permitiendo el control del llenado del recipiente con respecto al error que se genera en la entrada.

Se definen los siguientes conjuntos de la variable lingüística "velocidad" aplicada sobre el motor vibratorio:

- Velocidad muy baja (VMB), valores de (3765, 3765, 4518).
- Velocidad baja (VB), valores de (4267, 4767, 5270).
- Velocidad normal (VN), valores de (5019, 5270, 5772).
- Velocidad alta (VA), valores de (5521, 5772, 6400).
- Velocidad muy alta (VMA), valores de (6023, 6400, 6400).

En la Figura 2. 80 se indican los conjuntos definidos para la variable lingüística "velocidad".

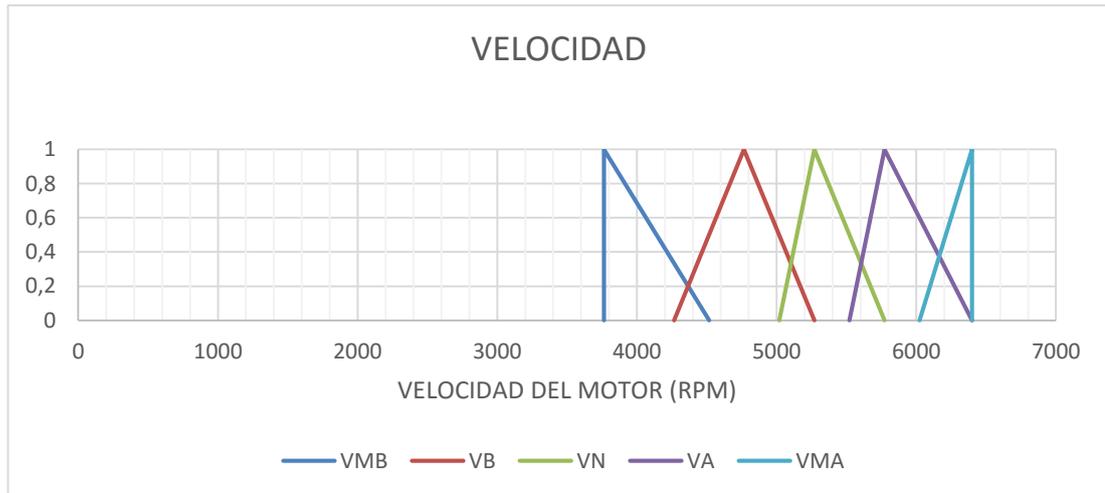


Figura 2. 80: Conjuntos de la variable lingüística "velocidad"

d. Base de conocimientos

La base de conocimientos es el punto de partida para la generación del conjunto de reglas sobre las que rige la inferencia. Se describen los términos usados y se muestra la base en forma de una matriz de asociación difusa. Se desarrolla una base de conocimiento, debido a que se tiene una sola salida a controlar.

Como la variable de salida es la variación de velocidad se consideran las siguientes reglas:

- Bajada muy grande (BMG).
- Bajada grande (BG).
- Mantener (M).
- Subida grande (SG).
- Subida muy grande (SMG)

Por consiguiente, se realiza la siguiente matriz como se muestra en la Tabla 2.9 .

Tabla 2.9: Conjunto de reglas de inferencia

ERROR GRAMOS	VELOCIDAD DEL MOTOR EN RPM'S				
	VMB	VB	VN	VA	VMA
PMB	SMG	SMG	SMG	SMG	SMG
PB	SG	SG	SG	SG	SG
PN	M	M	M	M	M
PA	BG	BG	BG	BG	BG
PMA	BMG	BMG	BMG	BMG	BMG

El siguiente paso es ingresar las reglas de inferencia que permiten relacionar los conjuntos de entrada y salida, y representar las reglas que definen el sistema. La entrada a este bloque son conjuntos difusos (grados de pertenencia) y las salidas también conjuntos difusos, asociados a la variable de salida, de la siguiente manera:

- IF (peso muy bajo AND velocidad muy grande) THEN (subida muy pequeña).
- IF (peso bajo AND velocidad baja) THEN (subida grande)
- IF (peso normal AND velocidad normal) THEN (mantener).
- IF (peso alto AND velocidad alta) THEN (bajada grande).
- IF (peso muy alto AND velocidad muy alta) THEN (bajada muy grande).

2.4.5. Diagrama de flujo de módulo didáctico de llenado y envasado de sólidos

Una vez investigado los parámetros y herramientas de programación para la funcionalidad del módulo didáctico se procede a generar un diagrama de flujo resumido de cual parte el diseño del software de dicho módulo, como se indica en la Figura 2.81.

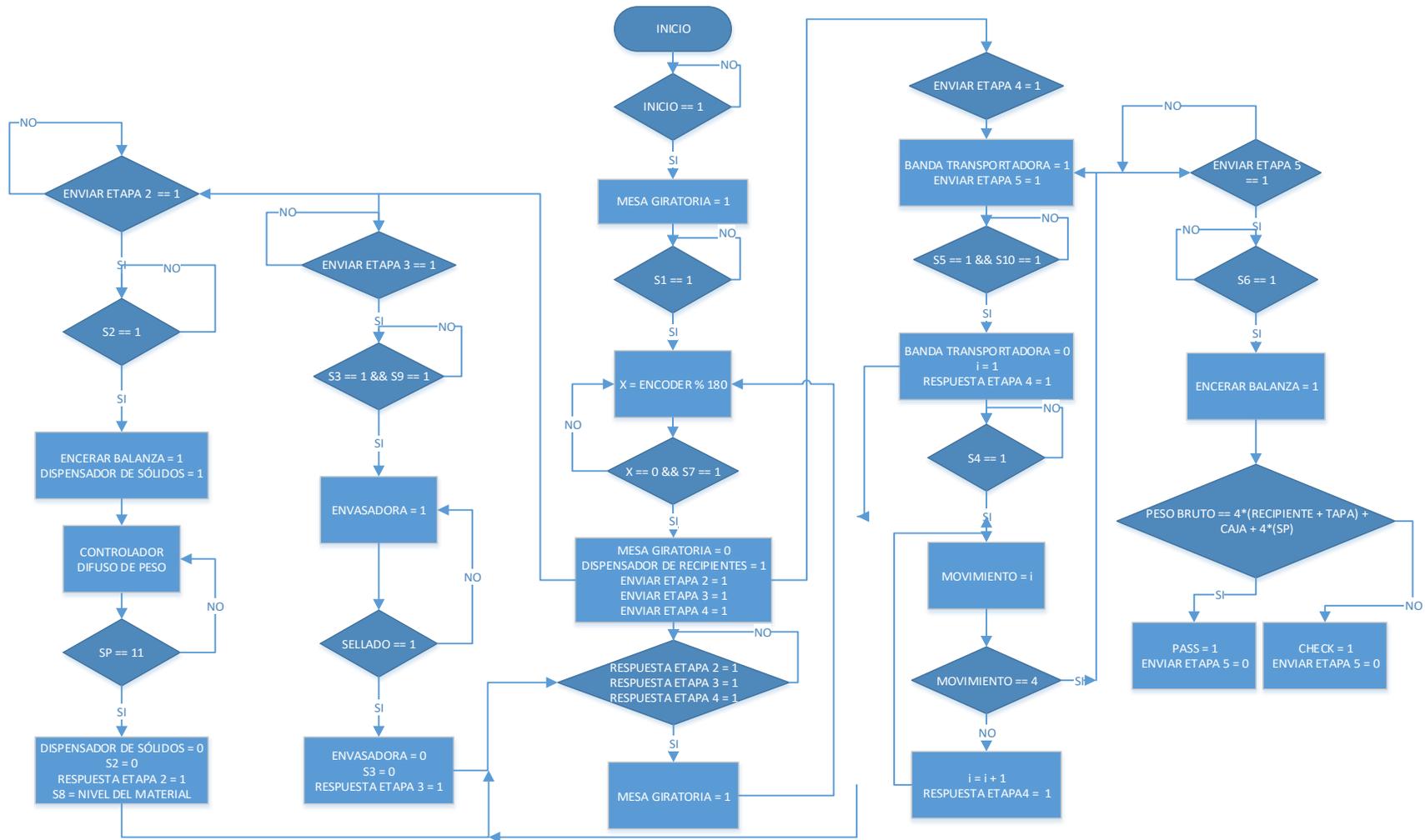


Figura 2.81: Diagrama de flujo del módulo didáctico de llenado y envasado de sólidos

2.4.6. Programación de la tarjeta Arduino Mega R3 2560 de la Etapa 1

El algoritmo programado en la tarjeta Arduino Mega R3 2560 correspondiente a la Etapa 1 realiza las siguientes tareas:

- Envía y recibe datos, desde y hacia la PC utilizando el protocolo de comunicación Modbus RTU.
- Lee la señal acondicionada del sensor capacitivo (S7) para emitir una alarma el momento en que el contenedor de recipientes se encuentre casi vacío, deteniendo el proceso temporalmente hasta colocar recipientes en el contenedor.
- Lee la señal acondicionada del sensor láser (S1) como interrupción externa para ajustar la posición inicial de la mesa giratoria y encerrar el encoder.
- Lee la posición de la mesa giratoria gracias al encoder empotrado en el eje central.
- Al medir 180 posiciones detiene el motor de alto torque para ubicar un recipiente en una ranura de la mesa giratoria.
- Envía señales digitales (flancos de bajada) hacia las etapas 2, 3 y 4 para preguntar el estado de cada una de ellas.
- Recibe mediante interrupciones externas las respuestas de las etapas 2, 3 y 4 para continuar con el proceso.
- Acciona un pistón eléctrico mediante señales digitales de control enviadas al driver L298d.
- Acciona un motor de alto torque mediante una señal PWM enviada a una tarjeta controladora de motores DC.
- Activa un módulo de relés por medio de señales digitales para encender o apagar las luces piloto de inicio, paro general y alarma 1.
- Lee el estado de los pulsadores de inicio y paro general, para iniciar o detener el proceso en caso de emergencia.

En el Anexo D-1 se muestra la codificación con sus instrucciones debidamente detalladas.

a. Diagrama de flujo de la Etapa 1

En la Figura 2.82 se indica el diagrama de flujo de la programación que se implementó en la tarjeta Arduino Mega R3 2560 de la Etapa 1.

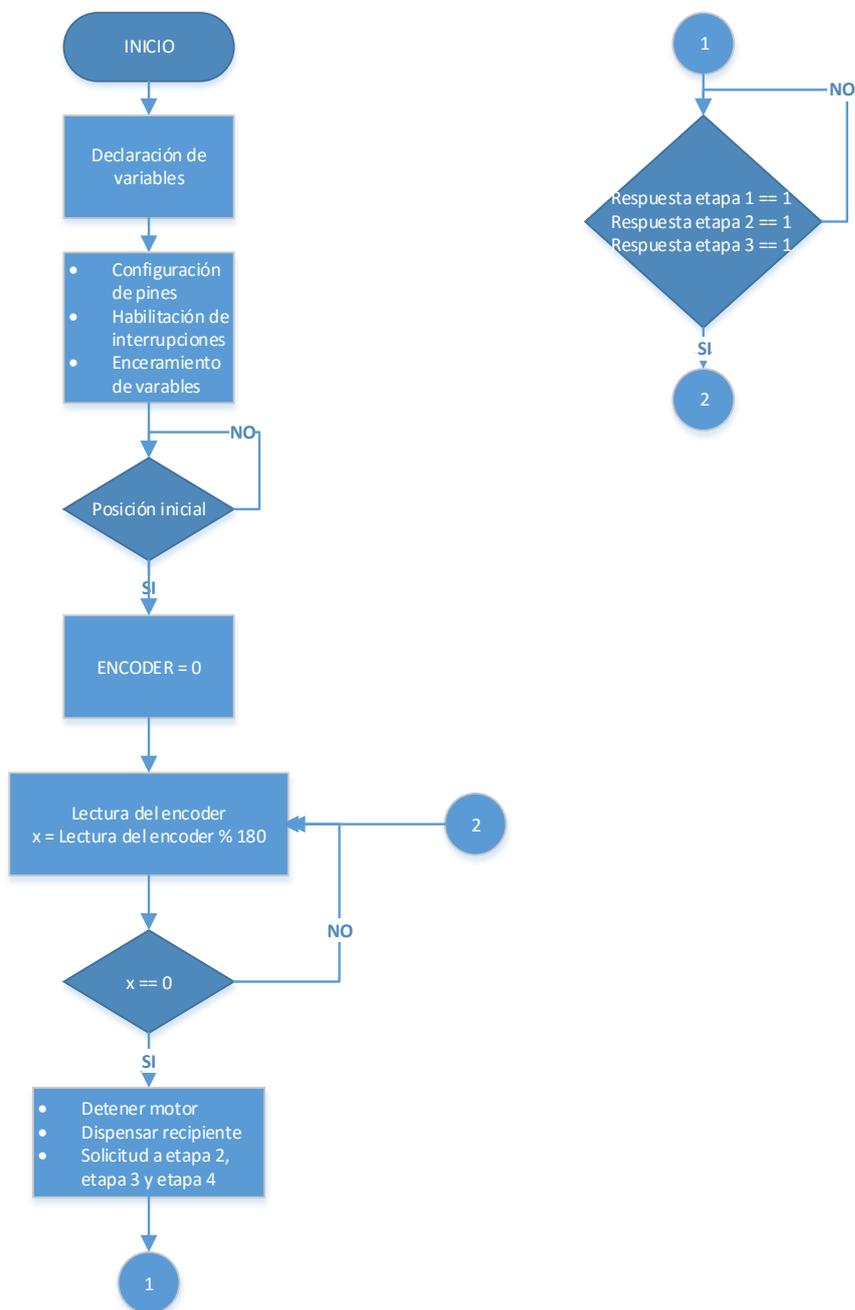


Figura 2.82: Diagrama de flujo de la tarjeta Arduino Mega R3 2560 de la Etapa 1

2.4.7. Programación de la tarjeta Arduino Uno R3 de la Etapa 2

El algoritmo que se programó en la tarjeta Arduino Uno R3 correspondiente a la Etapa 2 realiza las siguientes funciones:

- Envía y recibe datos, desde y hacia la PC utilizando el protocolo de comunicación Modbus RTU.
- Recibe una señal digital (flanco de bajada) como interrupción externa desde la Etapa 1 como señal de que la mesa giratoria se ha ubicado en una nueva posición.
- Lee la señal acondicionada del sensor láser QS30LDL (S2) como signo de que ha detectado un recipiente para llenarlo.
- Encera la balanza para poder verter el material sólido dentro del recipiente ya posicionado.
- Realiza un control difuso para verter el material sólido contenido en la tolva de manera que sea exacto y preciso.
- Acciona un motor vibratorio mediante señales digitales de control enviadas al driver L298d, el cual mediante el movimiento vibratorio del motor deja caer el material sólido en el recipiente de forma continua.
- Lee la señal analógica acondicionada del sensor ultrasónico U-GAGE T30UIPAQ (S8).
- Escala la señal analógica del sensor ultrasónico U-GAGE T30UIPAQ (S8) de acuerdo al contenido vertido en la tolva (el calculo es realizado de acuerdo a la densidad del material y a las dimensiones de la tolva).
- Activa un módulo de relés por medio de señales digitales para encender o apagar la luz piloto de la alarma 2, cuando el contenido de la tolva es menor a 20 gr.
- Envía una señal digital (flanco de bajada) como respuesta hacia la Etapa 1, dando a entender que ha terminado con el llenado.

En el Anexo D-2 se muestra la codificación con sus instrucciones debidamente detalladas.

a. Diagrama de flujo de la Etapa 2

En la Figura 2.83 se indica el diagrama de flujo de la programación que se implementó en la tarjeta Arduino Mega R3 2560 de la Etapa Dos.

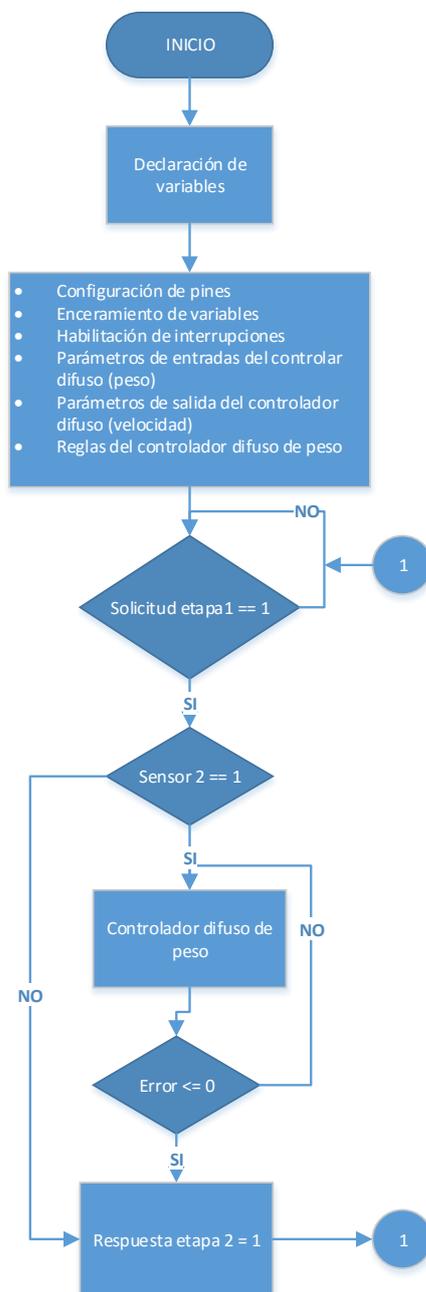


Figura 2.83: Diagrama de flujo de la tarjeta Arduino Uno R3 de la Etapa 2

2.4.8. Programación de la tarjeta Arduino Uno R3 de la Etapa 3

El algoritmo que se programó en la tarjeta Arduino Uno R3 correspondiente a la Etapa Tres realiza las siguientes funciones:

- Envía datos, hacia la PC utilizando el protocolo de comunicación Modbus RTU.
- Recibe una señal digital (flanco de bajada) como interrupción externa desde la Etapa 1 como señal de que la mesa giratoria se ha ubicado en una nueva posición.
- Lee la señal acondicionada del sensor capacitivo (S9) como signo de que ha detectado o no cubiertas (tapas) en el contenedor.
- Lee la señal acondicionada del sensor fotoeléctrico banner mini beam SME312D (S3) como señal de que ha detectado un recipiente lleno para envasarlo.
- Posiciona un pistón eléctrico con la ayuda de un servo motor a 180 grados para extraer cubiertas y a 0 grados para sellar el recipiente anteriormente llenado.
- Extrae las cubiertas mediante una ventosa y el accionamiento de una bomba de vacío controlada por la tarjeta controladora de motores DC Adafruit_MotorShield v2.3.
- Acciona un piston eléctrico mediante la tarjeta controladora de motores DC Adafruit_MotorShield v2.3, para realizar la acción (movimiento) de extracción de cubiertas y sellado de los recipientes anteriormente llenados.
- Activa un módulo de relés por medio de señales digitales para encender o apagar la luz piloto de la alarma 3, cuando el contenedor de cubiertas esta por vaciarse, deteniendo el proceso temporalmente hasta llenar el contenedor de cubiertas.
- Envía una señal digital (flanco de bajada) como respuesta hacia la Etapa 1, dando a entender que ha terminado con el envasado.

En el Anexo D-3 se muestra la codificación con sus instrucciones debidamente detalladas.

a. Diagrama de flujo de la Etapa 3

En la Figura 2.84 se indica el diagrama de flujo de la programación que se implementó en la tarjeta Arduino Uno R3 de la Etapa 3.

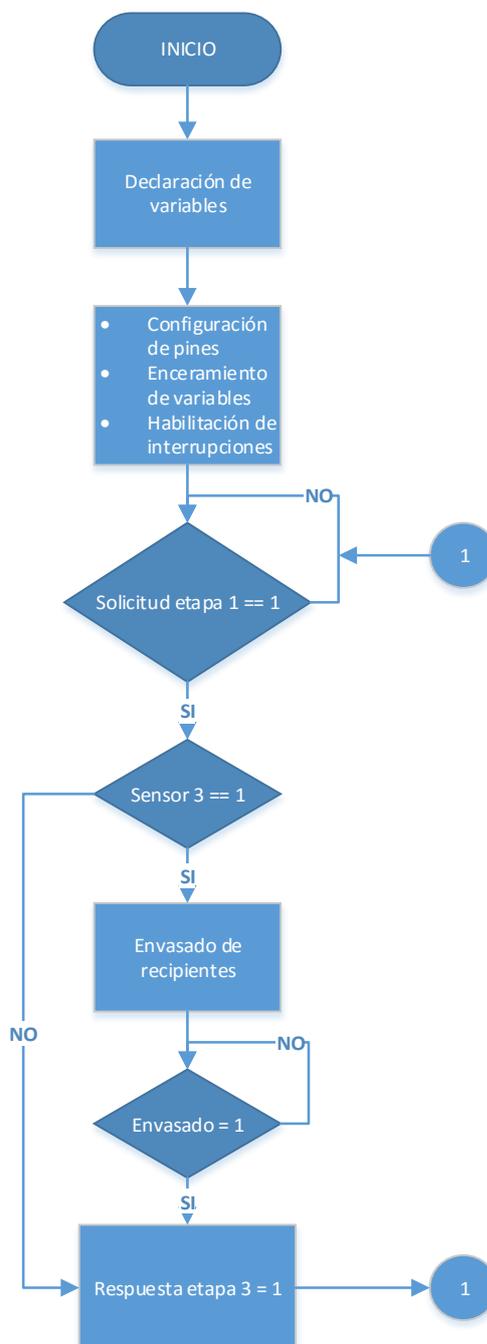


Figura 2.84: Diagrama de flujo de la tarjeta Arduino Uno R3 de la Etapa 3

2.4.9. Programación de la tarjeta Arduino Mega R3 2560 de la Etapa 4

El algoritmo que se programó en la tarjeta Arduino Mega R3 2560 correspondiente a la Etapa 4 realiza las siguientes funciones:

- Envía datos, hacia la PC utilizando el protocolo de comunicación Modbus RTU.
- Recibe una señal digital (flanco de bajada) como interrupción externa desde la Etapa 1 como señal de que la mesa giratoria se ha ubicado en una nueva posición.
- Lee la señal acondicionada del sensor fotoeléctrico (S10) como signo de que ha detectado o no cajas en el contenedor.
- Lee la señal acondicionada del sensor fotoeléctrico banner SMU315D (S4) como señal de que ha detectado un recipiente envasado para trasladarlo mediante un brazo robótico (Brazo 1).
- Lee la señal acondicionada del sensor láser (S5) como signo de que ha detectado una caja y detiene el movimiento de la banda transportadora posicionando la caja para que el brazo robótico (Brazo 1) realice el traslado de los recipientes envasados.
- Acciona un motor de alto torque mediante la tarjeta controladora de motores DC, para realizar la acción (movimiento) de la banda transportadora extrayendo una caja del contenedor de cajas.
- Realiza cuatro movimientos programados para dar movimiento al brazo robótico (Brazo 1), colocando los recipientes de manera ordenada.
- Activa un módulo de relés por medio de señales digitales para encender o apagar la luz piloto de la alarma 4, cuando el contenedor de cajas esta por vaciarse, deteniendo el proceso temporalmente hasta llenar el contenedor de cajas.
- Envía una señal digital (flanco de bajada) como respuesta hacia la Etapa 1, dando a entender que ha terminado con el envasado.

En el Anexo D – 4 se muestra la codificación con sus instrucciones debidamente detalladas.

a. Diagrama de flujo de la Etapa 4

En la Figura 2.85 se muestra el diagrama de flujo de la programación que se implementó en la tarjeta Arduino Mega R3 2560 de la Etapa 4.

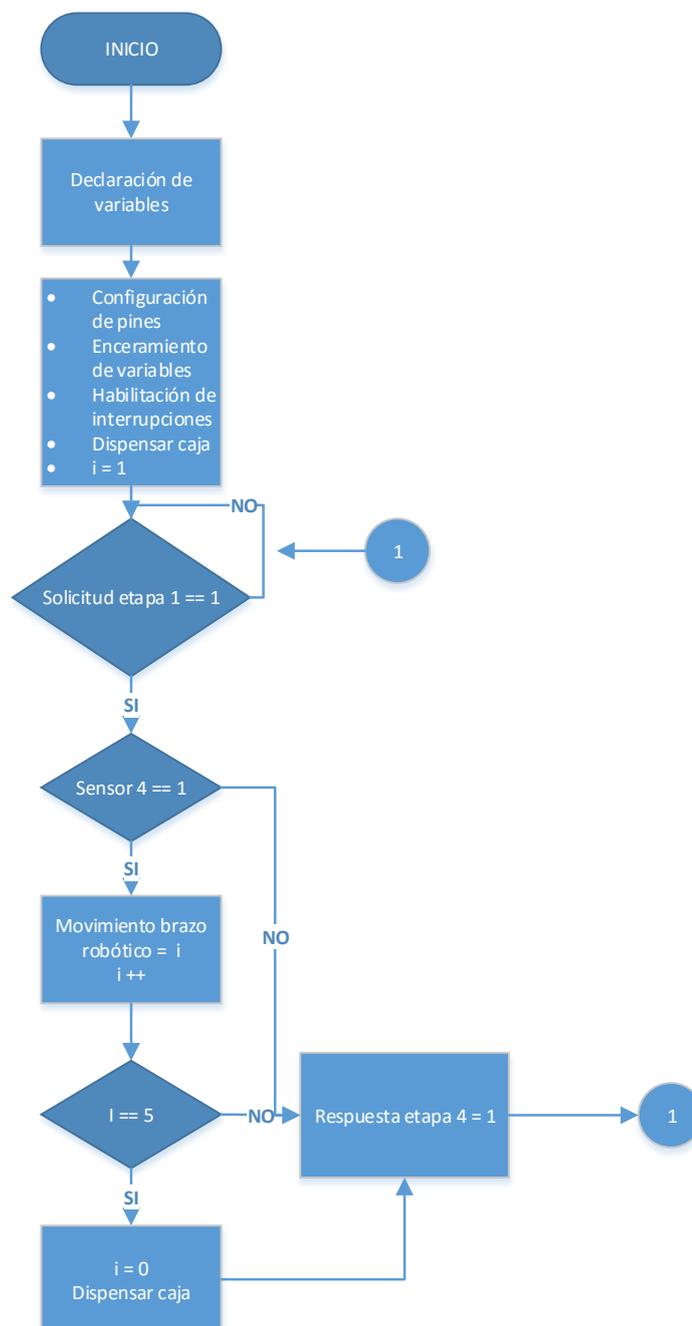


Figura 2.85: Diagrama de flujo de la tarjeta Arduino Mega R3 2560 de la Etapa 4

2.4.10. Programación de la tarjeta Arduino Uno R3 de la Etapa 5

El algoritmo que se programó en la tarjeta Arduino Uno R3 correspondiente a la Etapa Cinco realiza las siguientes funciones:

- Envía y recibe datos, desde y hacia la PC utilizando el protocolo de comunicación Modbus RTU.
- Lee la señal acondicionada del sensor fotoeléctrico (S6) como signo de que ha llegado el producto final a la etapa de inspección.
- Envía una señal digital (flanco de bajada) hacia la Etapa 4 para preguntar si la banda transportadora se encuentra en movimiento y leer el estado del sensor fotoeléctrico (S6).
- Recibe una señal digital (flanco de bajada) mediante una interrupción externa la respuesta de la Etapa 4 para poder ejecutar la acción de clasificación del producto final.
- Encera la balanza para poder iniciar la medida y verificar si pasa o no la inspección el producto final.
- Realiza un control de dos movimientos pre programados sobre el brazo robótico (Brazo 2) para clasificar el producto final por el camino correspondiente.
- Activa un módulo de relés por medio de señales digitales para encender o apagar las luces piloto de “Pass” y “Check” indicando si pasa o no la inspección el producto final.
- El proceso de la Etapa 5 se visualiza en un LCD 20X4 con comunicación I2C para que sea más interactiva la forma de verificar el producto final.

En el Anexo D – 5 se muestra la codificación con sus instrucciones debidamente detalladas.

a. Diagrama de flujo de la Etapa 5

En la Figura 2.86 se muestra el diagrama de flujo de la programación que se implementó en la tarjeta Arduino Uno R3 de la Etapa Cinco.

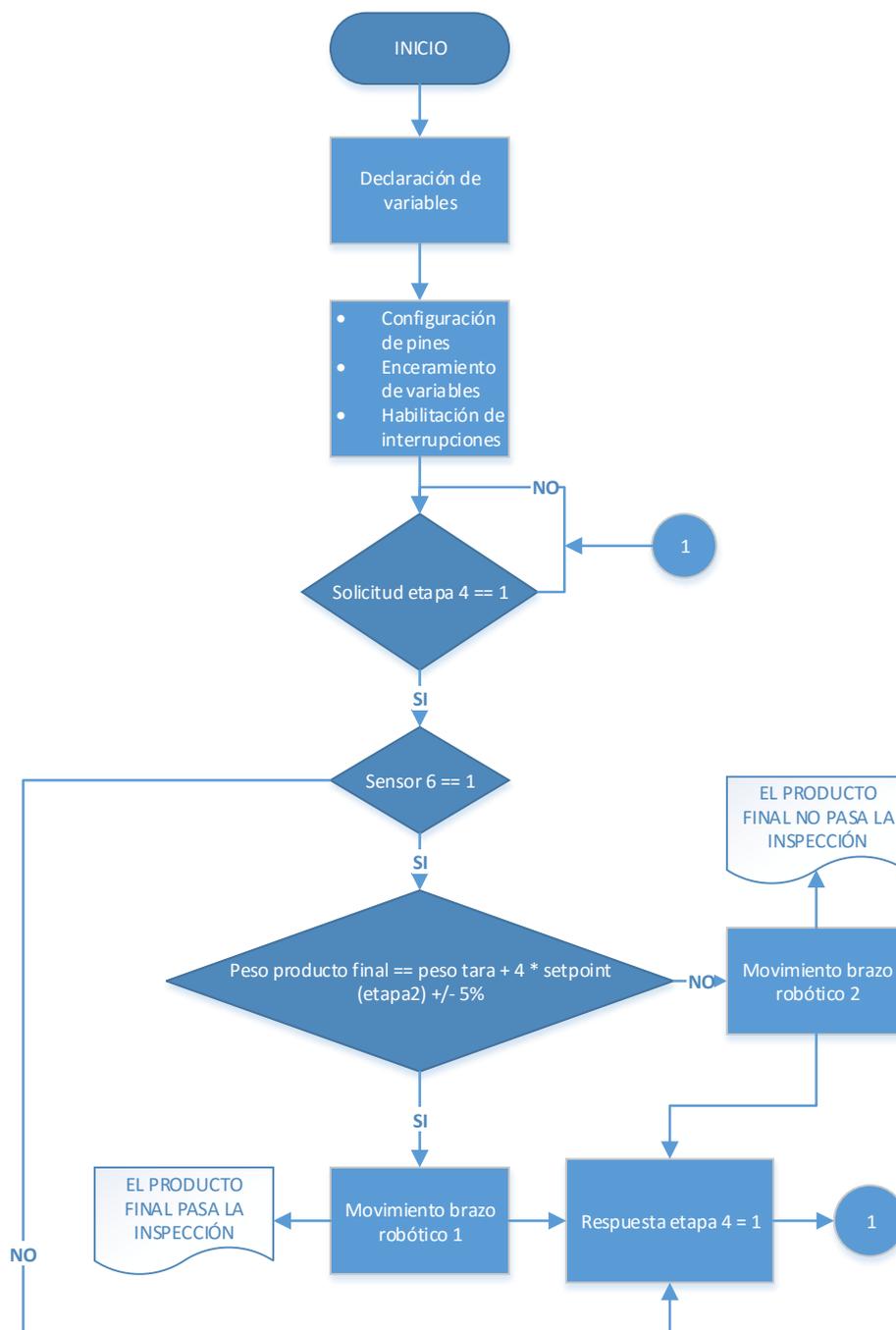


Figura 2.86: Diagrama de flujo de la tarjeta Arduino Uno R3 de la Etapa 5

2.4.11. Diseño del hmi del módulo didáctico

El diseño del HMI en el software Monitoriza For Arduino Scada-Acimut para el monitoreo del módulo didáctico realiza las siguientes funciones:

- Indica cada una de las etapas implementadas en el módulo didáctico.
- Visualiza el estado de cada uno de los sensores capacitivos, fotoeléctricos y láser.
- El Indicador muestra el peso del recipiente (vaso) llenado en la Etapa Dos y el peso en bruto del producto final de la Etapa de Inspección.
- Muestra las tendencias que se generan en la Etapa Dos como son; el Set Point, Control Value y Process Value del control Fuzzy, además contiene un teclado numérico para ingresar de forma manual el Set Point.
- Indica el estado de los botones tanto de Inicio como el de Paro General simulando las luces piloto de cada uno de ellos.
- Muestra el contenido de la tolva en donde se aloja el material a dispensar, además mediante un visualizador se observa el contenido en Gramos.
- El HMI contiene un contador de cajas aceptadas y rechazadas (denominada como "PASS" y "CHECK" respectivamente) luego de la inspección y transporte de la misma.

a. HMI del módulo didáctico

Ingresar en la página oficial:

- <http://www.acimut.com/monitoriza/monitorizaforarduino.html>

Muestra la siguiente ventana (Figura 2.87):



Figura 2.87: Página oficial de acimut monitoriza for arduino

Descargar Acimut Monitoriza for Arduino, tomar en cuenta que se necesita .NET Framework 3.5 SP1. Al ejecutar el setup.exe comprueba si está o no instalado, caso contrario lo descarga desde la web de Microsoft. Una vez instalados los componentes de Acimut Monitoriza for Arduino ejecutar el Editor Monitoriza (Figura 2.88) para acceder a la ventana principal y realizar el diseño y configuración del HMI.

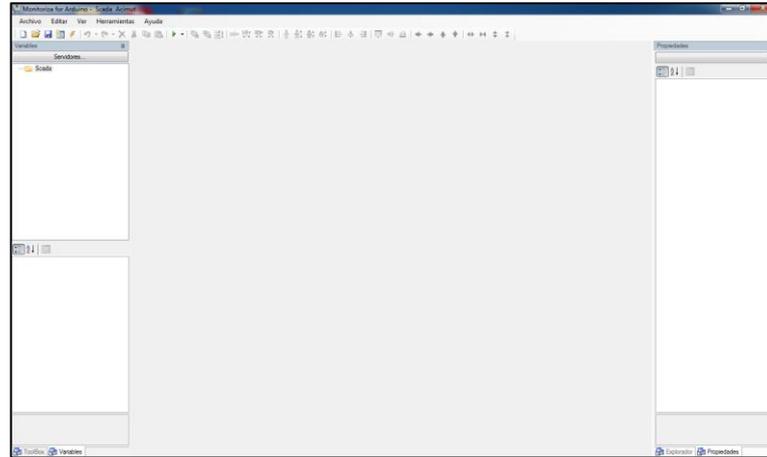


Figura 2.88: Ventana del Editor Monitoriza

Crear los servidores, en este caso servidores Modbus RTU Arduino como se indica en la Figura 2.89 :

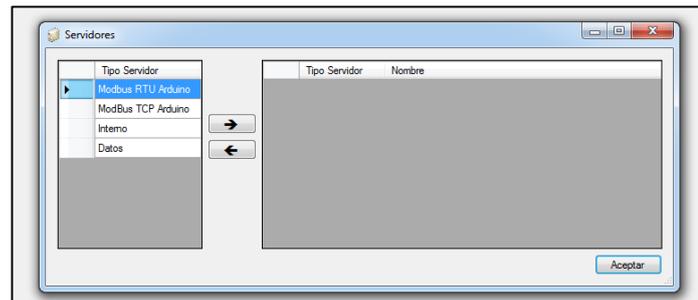


Figura 2.89: Creación de los servidores Modbus RTU Arduino

En la Figura 2.90 se puede observar los diferentes servidores Modbus RTU creados, correspondientes a cada etapa del módulo didáctico.

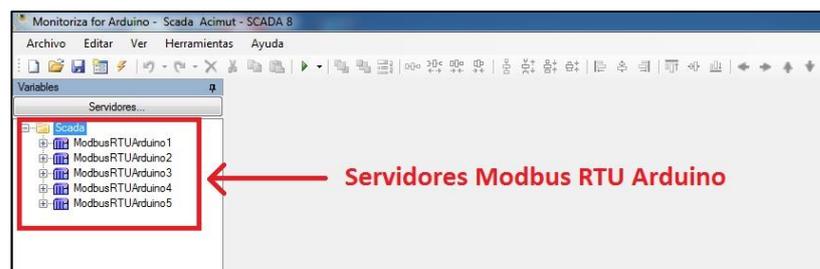


Figura 2.90: Servidores Modbus RTU

A continuación se configura el número del esclavo, el puerto serial con el que se comunica la tarjeta Arduino de cada etapa con el computador, los bits de paridad, los bits de datos y la velocidad con la transmite, tomar en cuenta que tanto la configuración de comunicación del HMI y del Arduino deben ser los mismos, caso contrario no transmitirá los datos que se desea visualizar en el HMI (Figura 2.91).

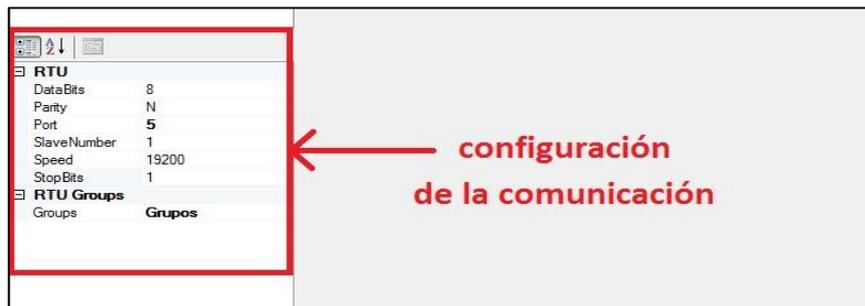


Figura 2.91: Configuración de la comunicación

A continuación crear los grupos correspondientes a cada servidor, donde se alojarán cada una de las variables que contienen la información del funcionamiento del módulo didáctico, como se indica en la Figura 2.92.

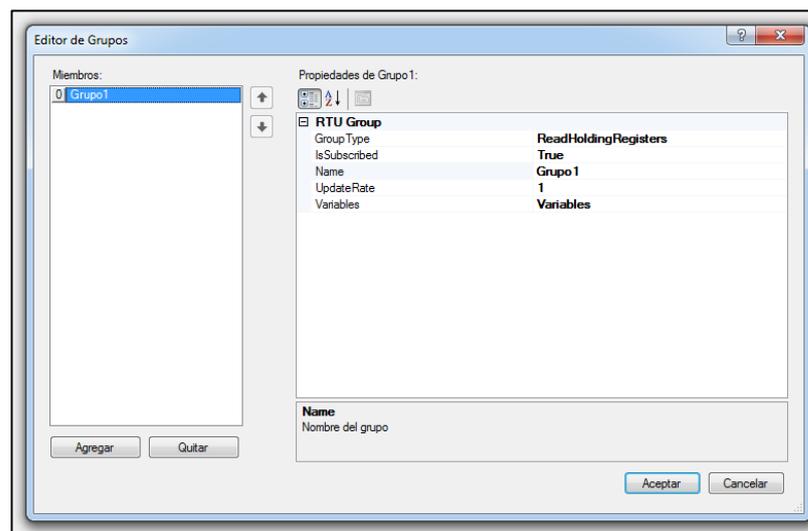


Figura 2.92: Creación de grupos en cada etapa

Después de crear los grupos para cada Servidor Modbus RTU Arduino, agregar las variables correspondientes a cada etapa del módulo didáctico, como se indica en la Figura 2.93.

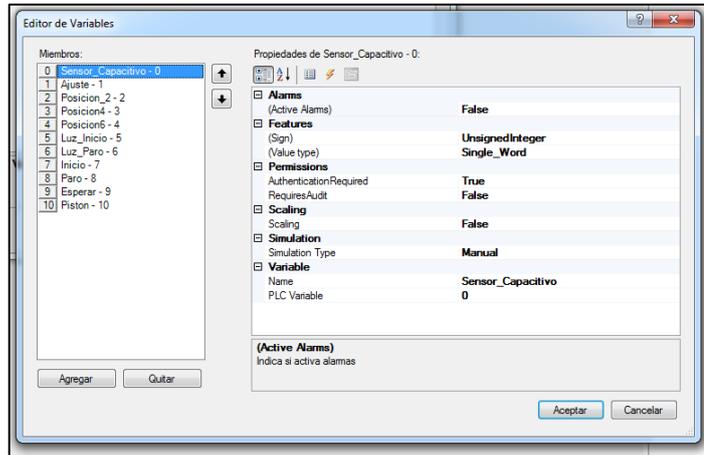


Figura 2.93: Creación de las variables del HMI

La Figura 2.94 muestra las variables creadas pertenecientes a cada servidor generado para la comunicación mediante el protocolo Modbus RTU con las tarjetas Arduino de cada etapa del módulo didáctico.

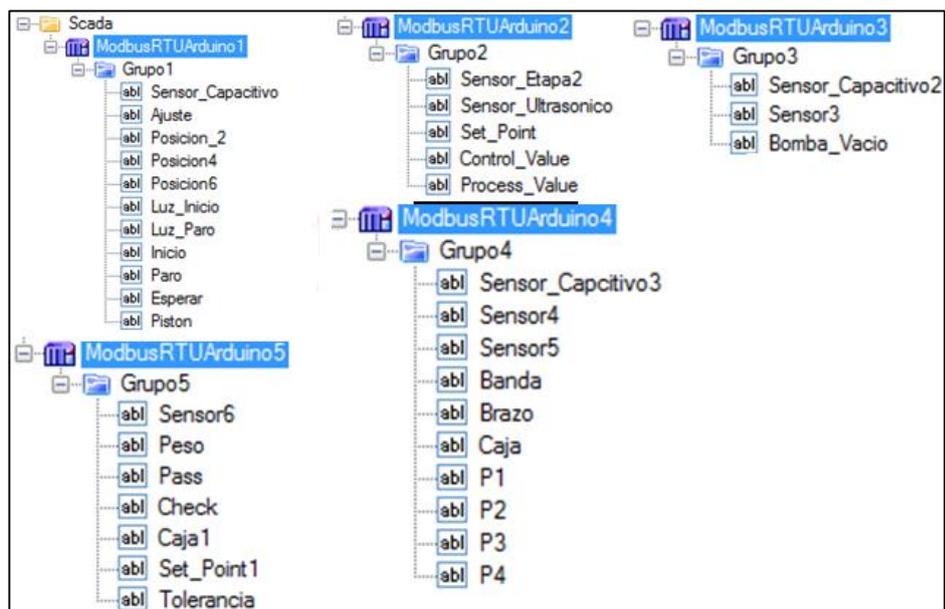


Figura 2.94: Variables de las etapas de módulo diáctico

Crear los formularios (Figura 2.95) donde se alojará una interfaz gráfica amigable con el usuario, la cual permita visualizar el proceso de llenado y envasado de sólidos.

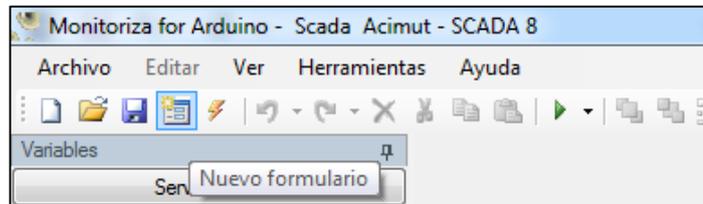


Figura 2.95: Nuevo formulario para el diseño del HMI

Se generan tres formularios como se indica en la Figura 2.96, los cuales están designados como: “PRINCIPAL”, “PROCESO” y “TENDENCIAS_FUZZY”.

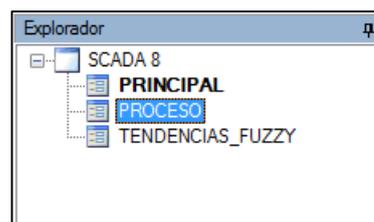


Figura 2.96: Formularios del HMI

El formulario denominado PRINCIPAL (Figura 2.97) contiene botones que permiten navegar entre los formularios PROCESO donde se encuentra diseñado el HMI del módulo didáctico, y el formulario TENDENCIAS donde se visualiza el comportamiento y las gráficas del controlador difuso.



Figura 2.97: Formulario PRINCIPAL

El formulario denominado PROCESO (Figura 2.98) contiene las variables asignadas en cada servidor Modbus RTU Arduino con la finalidad de mostrar en tiempo real el comportamiento de cada sensor, actuador y alarmas.

Contiene indicadores numéricos como son: Nivel del material de la tolva, el peso neto del recipiente llenado con el material dispensado en la Etapa 2 y el peso en bruto del producto final. Además, simula los movimientos del brazo robótico de la Etapa 4. Finalmente, el HMI contiene un contador para la clasificación del producto final.

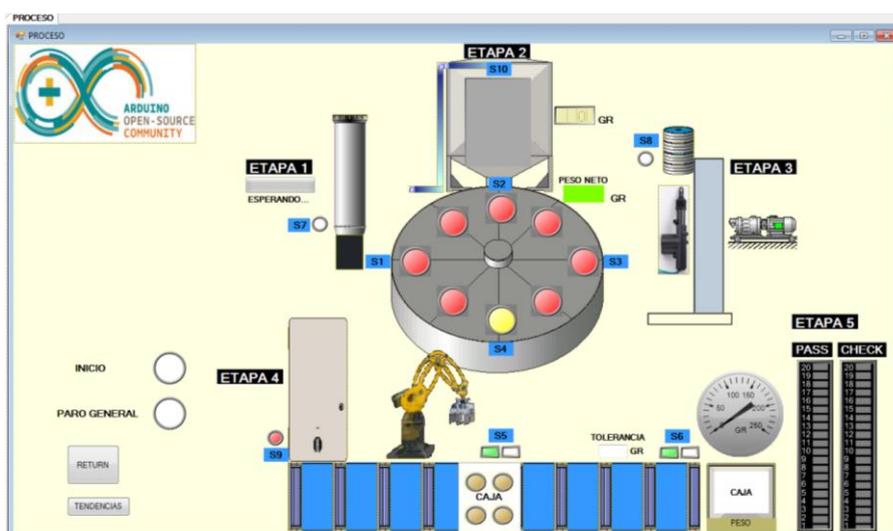


Figura 2.98: Formulario PROCESO

Y finalmente el formulario denominado TENDENCIAS_FUZZY contiene las curvas del Set Point, Control Value y Process Value del controlador difuso implementado en la Etapa Dos (Figura 2.99), además contiene un teclado numérico que permite cambiar el Set Point del proceso.

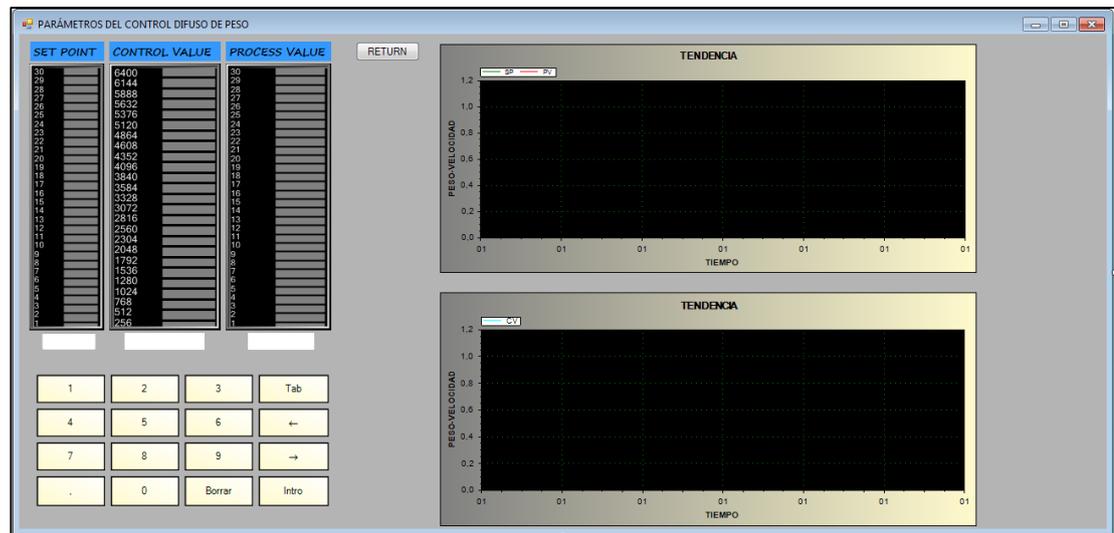


Figura 2.99: Formulario TENDENCIAS_FUZZY

CAPÍTULO III

3. PRUEBAS Y RESULTADOS

3.1. Generalidades

El correcto desempeño del módulo didáctico se verifica en la sincronización de los elementos como sensores, actuadores y tarjetas de control que conforman cada una de las etapas del módulo didáctico de automatización de llenado y envasado de sólidos, optimizando así los recursos y minimizando el tiempo de producción.

Para ello se determinó la importancia de realizar diferentes pruebas sobre los dispositivos de medida y actuadores, comprobando algunos de los conceptos frecuentemente utilizados en la instrumentación.

3.2. Funcionamiento y ajuste de la mesa giratoria

Esta prueba se realiza a partir del sensor laser de la Etapa 1 que detecta un cejillo ubicado en la mitad de una de las divisiones de la mesa giratoria, enviando una señal digital "FALLING" (Flanco de bajada), hacia la tarjeta de control de la Etapa 1, de tal manera que posiciona dicha mesa giratoria de forma que coincida con la ubicación de las etapas restantes, como se indica en la Figura 3.1.

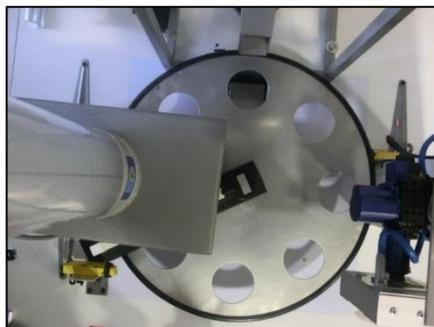


Figura 3.1: Prueba del posicionamiento inicial de la mesa giratoria

Es importante recalcar que la velocidad de ajuste del motor empotrado en la mesa giratoria debe ser la misma que la velocidad de trabajo el momento que se encuentra iniciado el proceso.

3.3. Pruebas de ajuste del sensor capacitivo LJC30A3-H-Z/AY

El sensor capacitivo fue sometido a pruebas de ajuste verificando que la distancia con respecto al material se encuentre en los límites establecidos en la hoja de datos del sensor, mediante el giro de un potenciómetro se ajusta la sensibilidad con respecto al material que se requiere detectar, hasta que la luz indicadora se apague, como se indica en la Figura 3.2.

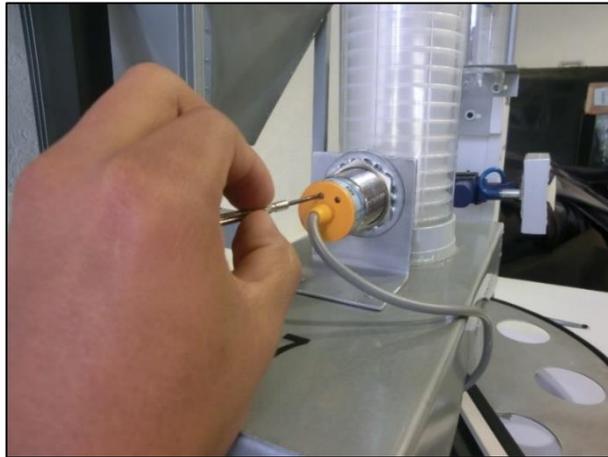


Figura 3.2: Ajuste del sensor capacitivo de la Etapa 1 con la existencia de material en el contenedor

Posteriormente se vacía el recipiente y se comprueba que la luz indicadora se encienda en el caso de que no lo haga se procede a girar el potenciómetro nuevamente y repetir el procedimiento anterior, como se indica en la Figura 3.3.

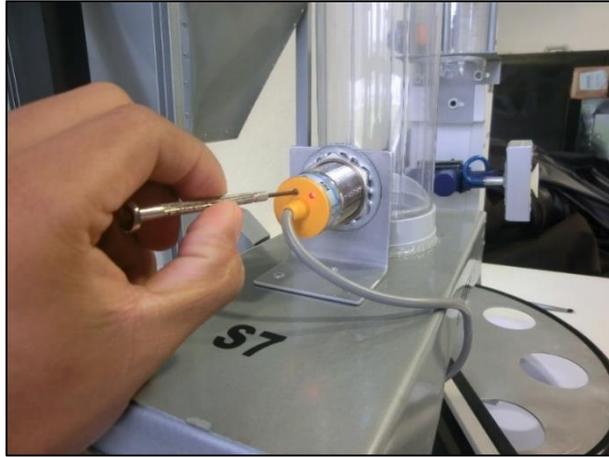


Figura 3.3: Ajuste del sensor capacitivo con la no existencia de material en el contenedor

3.4. Ajuste sensor láser banner QS30LDL

El sensor laser QS30LDL fue sometido a pruebas de ajuste, acorde a los procedimientos detallados en las hojas de datos de dicho sensor, de la siguiente manera:

- Presionar el botón (-) ubicado en el sensor laser durante 2 segundos aproximadamente.
- Verificar que las luces indicadoras 5 y 6 se enciendan de forma alternada.
- Presionar el botón (-) durante un segundo y fijar el primer límite de la ventana.
- Presionar el botón (-) durante un segundo y fijar el segundo límite de la ventana.
- El sensor se dirigirá automáticamente a modo Run y estará ajustado para detectar el material que va a interceptar dicha ventana.

El procedimiento detallado en imágenes se indica en la Figura 3.4.

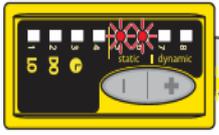
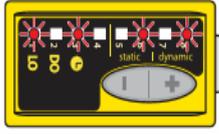
	Push Button	Remote Line	Display Shows
Access TEACH Mode	<ul style="list-style-type: none"> Press and Hold 	<ul style="list-style-type: none"> Present Output ON condition Single-pulse remote line 	Power LED: OFF Output LED: ON (Push Button) Output LED: OFF (Remote) Bargraph: #5 & 6 flashing 
Learn Set Point (Output ON) Condition	<ul style="list-style-type: none"> Present Output ON condition Double-click 	<ul style="list-style-type: none"> Double-pulse remote line 	Teach Accepted Power LED: ON Bargraph: #3 & 6 flash to show Single-Point TEACH Sensor returns to RUN mode with new settings 
			Teach Unacceptable Power LED: OFF Bargraph: #1, 3, 6, 8 flash to show fail, then #5 & 6 begin to flash Sensor waits for valid Teach condition 

Figura 3.4: Procedimiento de ajuste del sensor laser QS30LDL para una ventana de dos puntos

3.5. Pruebas de ajuste del sensor fotoeléctrico banner SME312D

El sensor fotoeléctrico SME312D fue sometido a pruebas de ajuste con respecto al material a detectar, acorde a los procedimientos detallados en las hojas de datos de dicho sensor, de la siguiente manera:

- Presione el botón “TEACH” ubicado en la parte posterior del sensor hasta que el LED verde empiece a parpadear en rojo (o se apague) y el LED amarillo se mantenga encendido.
- Colocar el objeto frente al sensor y a la distancia establecida por el usuario para poder detectar dicho objeto y presionar el botón “TEACH”. El LED amarillo se apagará.
- Retirar el objeto a detectar y presionar el botón “TEACH”. El LED rojo se tornará verde, aceptando la nueva configuración del sensor.

3.6. Pruebas de ajuste del sensor fotoeléctrico banner mini-beam SMU315D

El sensor fotoeléctrico SMU315D fue sometido a pruebas de ajuste con respecto al material a detectar, acorde a los procedimientos detallados en las hojas de datos de dicho sensor, de la siguiente manera:

- El sensor consta de un potenciómetro ubicado en la parte posterior mediante el cual ajusta la sensibilidad de dicho sensor.
- Colocar el objeto frente al sensor y a la distancia establecida por el usuario para poder detectar dicho objeto y verificar que el LED de estado se encienda.
- Retirar el objeto a detectar verificar que el LED de estado se apague.
- Además, consta de un selector ubicado en la parte posterior del sensor para cambiar el tipo de salida del sensor, normalmente cerrado (NC) o normalmente abierto (NO).

3.7. Pruebas de ajuste del sensor QS18VP6R (RX) y el sensor QS18SE (TX)

El par de sensores fotoeléctricos (RX Y TX) fueron sometidos a pruebas de ajuste colocándolos de tal forma que los dos puntos de la ventana queden en línea de vista para obtener un rendimiento óptimo, como se indica en la Figura 3.5.



Figura 3.5: Posición en línea de vista del sensor QS18VP6R (RX) y el sensor QS18SE (TX)

3.8. Pruebas de ajuste sensor fotoeléctrico G30-3A70NA

El sensor fotoeléctrico G30-3A70NA fue sometido a pruebas de ajuste con respecto a la detección de la presencia o ausencia de cajas en el contenedor, mediante la variación de un potenciómetro ubicado en la parte posterior del sensor de acuerdo a la distancia que haya sido ubicado el sensor de las cajas, como se indica en la Figura 3.6.

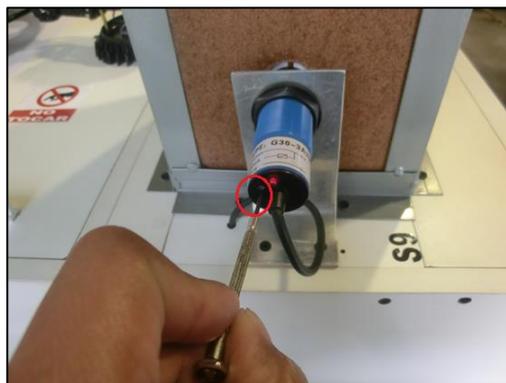


Figura 3.6: Ajuste del sensor fotoeléctrico en una ventana de un solo punto

En la Figura 3.7 indica los pulsadores de programación analógica y digital, además los indicadores de los mismos tanto de la programación como del suministro se identifican.

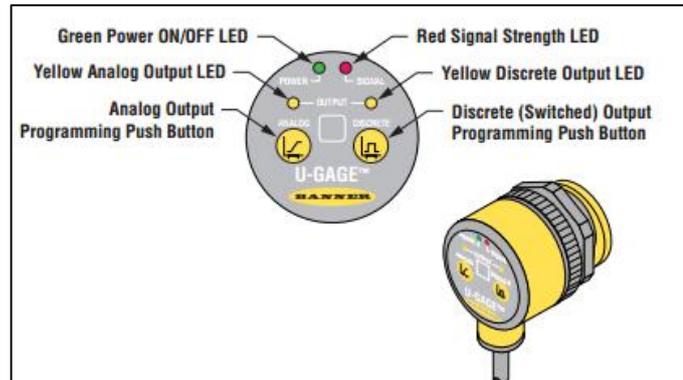


Figura 3.7: U-GAGE T30 pulsadores de programación del sensor e indicadores

Aquí las notas generales sobre la programación:

- EL sensor volverá al modo RUN si la primera condición TEACH no está registrado dentro de los 120 segundos.
- Después de enseñar el primer límite, el sensor permanecerá en modo PROGRAM hasta que finalice la secuencia TEACH.
- Mantener pulsado los botones pulsadores de programación mayores a 2 segundos (antes de enseñar el segundo límite) para salir del modo PROGRAM.

3.9. Pruebas de ajuste del sensor ultrasónico U-GAGE U30

El sensor ultrasónico U-GAGE U30 fue sometido a pruebas de ajuste de acuerdo a los procedimientos detallados en la hoja de datos de dicho sensor, de la siguiente manera:

- Pulsar sobre el botón de programación de la salida analógica para establecer la primera condición, si no se establece la primera condición el sensor volverá al modo RUN dentro de 120 segundos.
- Pulsar nuevamente sobre el botón de programación de la salida analógica después de establecer el primer límite, el sensor permanecerá en modo PROGRAM hasta que finalice la secuencia TEACH.

- Mantener pulsado el botón de programación por 2 segundos aproximadamente para salir del modo PROGRAM.

3.10. Calibración de las balanzas en la Etapa 2 y Etapa 5

La calibración de las etapas 2 y 5 fue realizada de manera similar mediante pesos patrones (Figura 3.8) provistos por el Laboratorio de Química de la Universidad de las Fuerzas Armadas Espe Extensión Latacunga.



Figura 3.8: Pesos patrón

Gracias a la facilidad de manejo del módulo HX711 implementado en la Etapa 2 y Etapa 5 se logró calibrar las balanzas para determinar variedad de pesos exactos vía software, con un número de calibración que es ingresado en la programación de cada etapa y modificado variando los pesos patrones, como se indica en la Figura 3.9 y la Figura 3.10

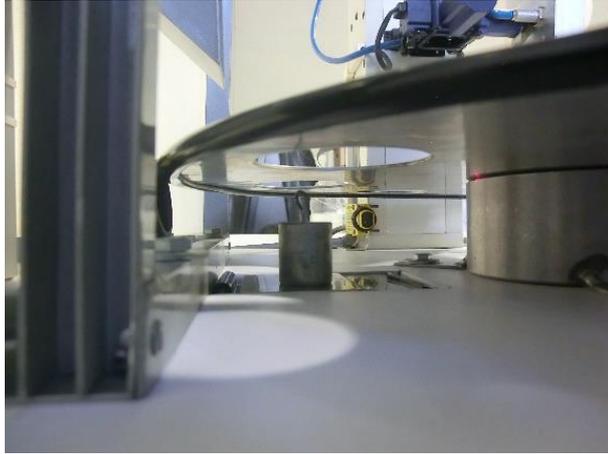
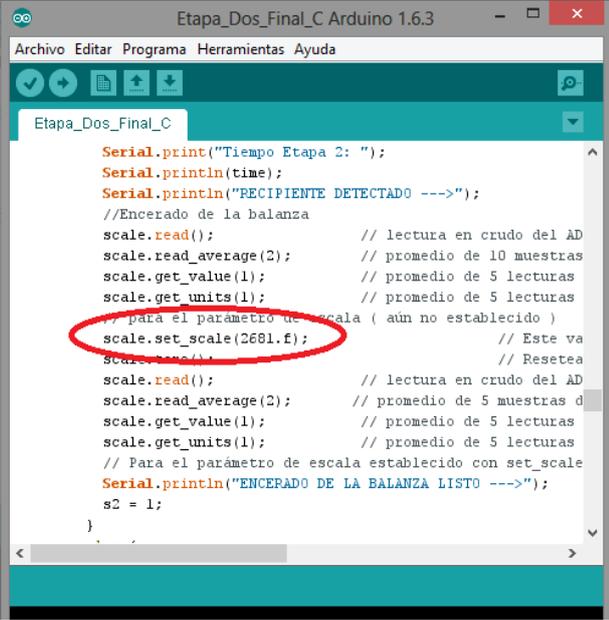


Figura 3.9: Ubicación de pesos patrón en la balanza de la Etapa 2



Figura 3.10: Ubicación de pesos patrón en la balanza de la Etapa 5

El número anteriormente mencionado se obtiene con la variación de pesos patrones en la balanza correspondiente a la Etapa 2, el cual fue de 2681 el momento de encerrar la balanza, como se indica en la Figura 3.11.



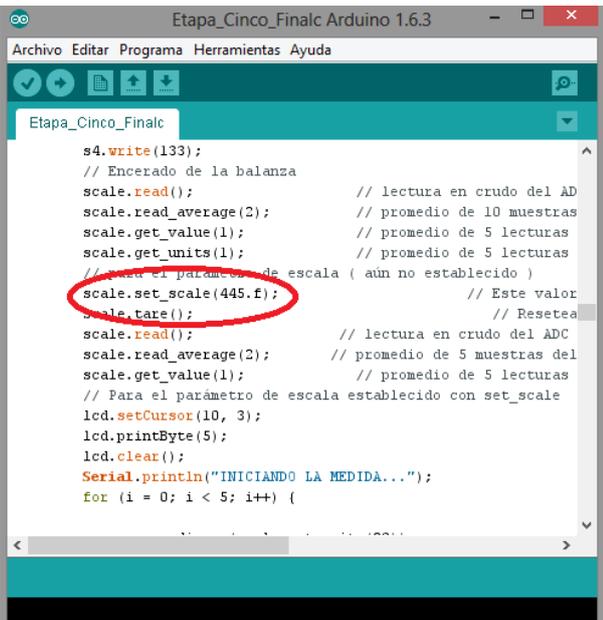
```

Etapa_Dos_Final_C
Serial.print("Tiempo Etapa 2: ");
Serial.println(time);
Serial.println("RECIPIENTE DETECTADO --->");
//Encerado de la balanza
scale.read(); // lectura en crudo del AD
scale.read_average(2); // promedio de 10 muestras
scale.get_value(1); // promedio de 5 lecturas
scale.get_units(1); // promedio de 5 lecturas
// Para el parámetro de escala ( aún no establecido )
scale.set_scale(2681.f); // Este va
scale.tare(); // Resetea
scale.read(); // lectura en crudo del AD
scale.read_average(2); // promedio de 5 muestras d
scale.get_value(1); // promedio de 5 lecturas
scale.get_units(1); // promedio de 5 lecturas
// Para el parámetro de escala establecido con set_scale
Serial.println("ENCERADO DE LA BALANZA LISTO --->");
s2 = 1;
}

```

Figura 3.11: Número de calibración obtenido para la balanza de la Etapa 2

De igual manera que en la balanza de la Etapa 2 el número obtenido con la variación de pesos patrones en la balanza correspondiente a la Etapa 5 fue de 445 el momento de encerar la balanza, como se indica en la Figura 3.12.



```

Etapa_Cinco_Finalc
s4.write(133);
// Encerado de la balanza
scale.read(); // lectura en crudo del AD
scale.read_average(2); // promedio de 10 muestras
scale.get_value(1); // promedio de 5 lecturas
scale.get_units(1); // promedio de 5 lecturas
// Para el parámetro de escala ( aún no establecido )
scale.set_scale(445.f); // Este valor
scale.tare(); // Resetea
scale.read(); // lectura en crudo del ADC
scale.read_average(2); // promedio de 5 muestras del
scale.get_value(1); // promedio de 5 lecturas
// Para el parámetro de escala establecido con set_scale
lcd.setCursor(10, 3);
lcd.printByte(5);
lcd.clear();
Serial.println("INICIANDO LA MEDIDA...");
for (i = 0; i < 5; i++) {

```

Figura 3.12: Número de calibración obtenido para la balanza de la Etapa 5

3.11. Pruebas de accionamiento de alarmas

Para verificar el correcto funcionamiento de las alarmas que existen en el módulo didáctico de un sistema de llenado y envasado de sólidos se procede a vaciar los contenedores de recipientes, cubiertas, cajas y el material contenido en la tolva, observando la activación de las luces pilotos en el panel de control, como se indica en la Figura 3.13.



Figura 3.13: Accionamiento de alarmas del módulo didáctico de llenado y envasado de sólidos

3.12. Pruebas de funcionamiento del control de peso aplicado en la Etapa 2

La medida de material sólido a dispensar en los recipientes, en este caso cereal, se controla mediante un control difuso implementado para obtener un mínimo de error, de manera que cumpla con los requisitos establecidos previamente por el operario.

El Set point o punto de ajuste establecido previamente es de 11 gramos, medida que debe ser proporcionada por el operador. La Etapa 2 fue sometida a pruebas, obteniendo un análisis estadístico de un conjunto de muestras tomadas y comparadas para así calcular una media de lo calculado

previamente y el error medido en dicha etapa. En la Tabla 3.1 se observa la comparativa entre los valores cálculos y los valores medidos con distintos puntos de ajuste o set points.

Tabla 3.1: Comparativa de valores calculados y valores medidos en el dispensador de sólidos de la Etapa 2

RESULTADOS ETAPA 2						
REVOLUCIONES DE LA MESA GIRATORIA	SET POINT (GRAMOS)					
	VALORES CALCULADOS			VALORES MEDIDOS		
	5 GR	11 GR	18 GR	5 GR	11 GR	18 GR
1°	5	11	18	5,11	11,013	18,261
	5	11	18	5,095	11,129	18,173
	5	11	18	5,16	11,095	18,075
	5	11	18	5,046	11,087	18,04
	5	11	18	5,122	11,058	18,138
	5	11	18	5,233	11,12	18,042
	5	11	18	5,063	11,083	18,048
	5	11	18	5,096	11,125	18,135
2°	5	11	18	5,019	11,046	18,153
	5	11	18	5,129	11,219	18,021
	5	11	18	5,142	11,165	18,092
	5	11	18	5,167	11,022	18,2
	5	11	18	5,023	11,098	18,176
	5	11	18	5,124	11,111	18,09
	5	11	18	5,101	11,019	18,16
	5	11	18	5,009	11,026	18,011
3°	5	11	18	5,096	11,017	18,149
	5	11	18	5,045	11,19	18,065
	5	11	18	5,165	11,034	18,034
	5	11	18	5,159	11,081	18,019
	5	11	18	5,112	11,038	18,07
	5	11	18	5,041	11,134	18,122
	5	11	18	5,178	11,086	18,131
	5	11	18	5,197	11,107	18,196
4°	5	11	18	5,04	11,022	18,205
	5	11	18	5,115	11,137	18,12
	5	11	18	5,098	11,178	18,045
	5	11	18	5,086	11,182	18,023
	5	11	18	5,229	11,04	18,228
	5	11	18	5,147	11,22	18,134
	5	11	18	5,235	11,018	18,078

	5	11	18	5,131	11,025	18,066
PROMEDIO	5,000	11,000	18,000	5,116	11,091	18,109
ERROR	0,000	0,000	0,000	0,116	0,091	0,109

Como se puede observar en la Tabla 3.1 se realizó un análisis estadístico para determinar el error en el dispensado del material sólido sobre el recipiente, en resumen se realizó un muestreo de un total de 32 muestras (recipientes llenos) las cuales fueron pesadas en dicha etapa, generando como resultados una media de error de 0,116 gramos, que representa un +2,32% del punto de ajuste de 5 gramos, una media de error de 0,091 gramos, que representa el +0,82% del punto de ajuste de 11 gramos y una media de error de 0,109 gramos, que representa un +0,60% del punto de ajuste de 18 gramos.

El análisis realizado en la etapa 2 demuestra que el material vertido en los recipientes dispone de un error positivo mesurablemente pequeño, debido a la inercia del proceso de dispensado de sólidos, lo cual permite observar a nivel didáctico que el error medido en el presente proyecto no es de relevancia.

Adicionalmente se observó gráficamente el desempeño del dosificador de sólidos el momento de verter el material sólido en el recipiente con los puntos de ajuste anteriormente mencionados, mediante el software de Acimut Monitoriza for Arduino.

En la Figura 3.14 se puede observar el desempeño del dispensador de sólidos con un set point establecido de 5 gramos, de igual forma el tiempo que toma en llenar el recipiente con dicho set point, así también las gráficas de la variable de control (velocidad del motor vibrador) y la gráfica de la variable del proceso (peso del material).

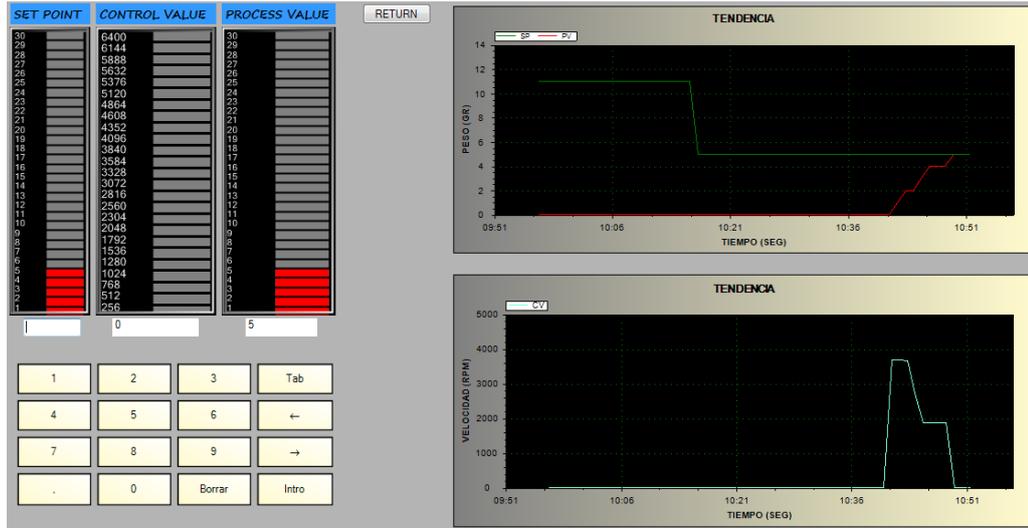


Figura 3.14: Gráfica de funcionamiento del dispensador de sólidos con un set point de 5 gramos

En la Figura 3.15 se observa el desempeño del dispensador de sólidos trabajando de manera continua verificando las formas de onda de las variables manejadas en el proceso (set point, control value y process value).

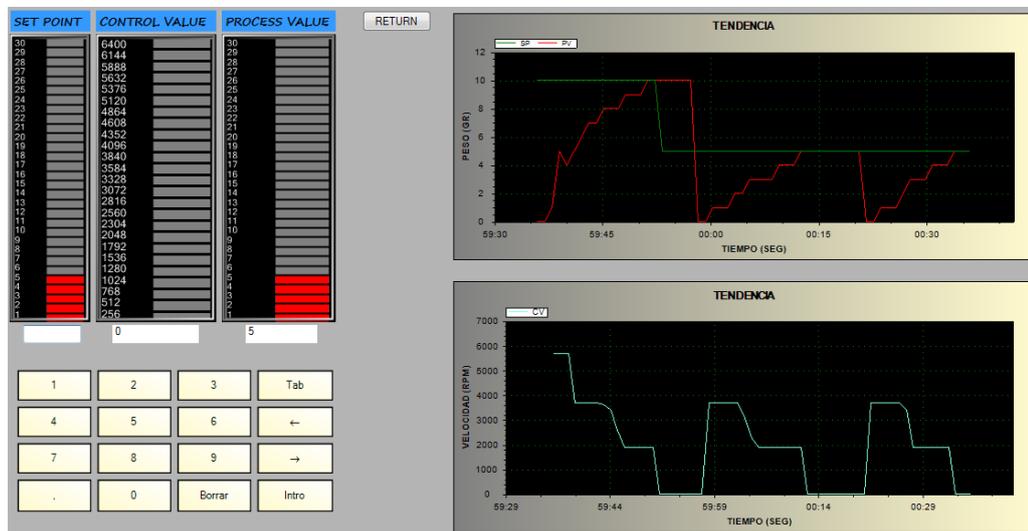


Figura 3.15: Gráfica de funcionamiento del dispensador de sólidos con un set point de 5 gramos para 3 muestras continuas

En la Figura 3.16 se puede observar el desempeño del dispensador de sólidos con un set point establecido de 11 gramos, de igual forma el tiempo que toma en llenar el recipiente con dicho set point.

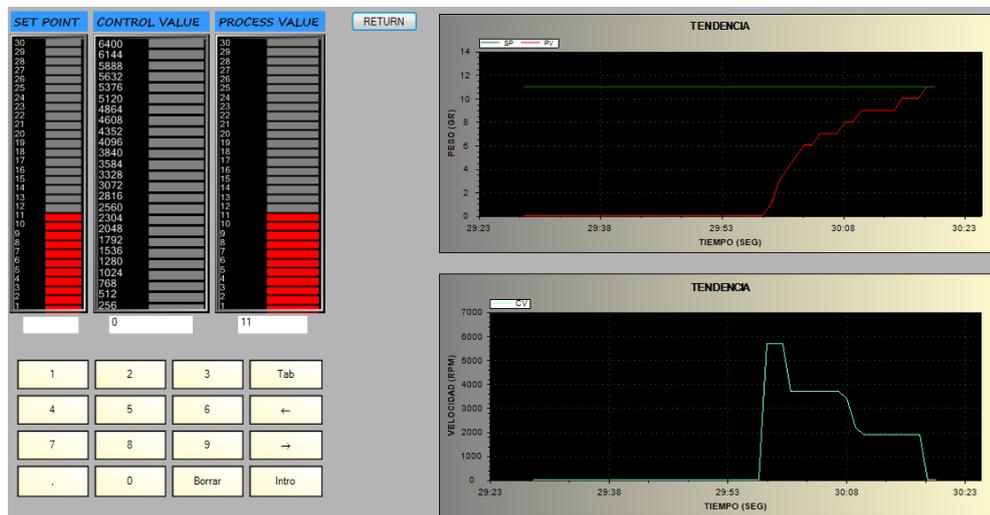


Figura 3.16: Gráfica de funcionamiento del dispensador de sólidos con un set point de 11 gramos

En la Figura 3.17 se observa el desempeño del dispensador de sólidos trabajando de manera continua verificando las formas de onda de las variables manejadas en el proceso (set point, control value y process value).

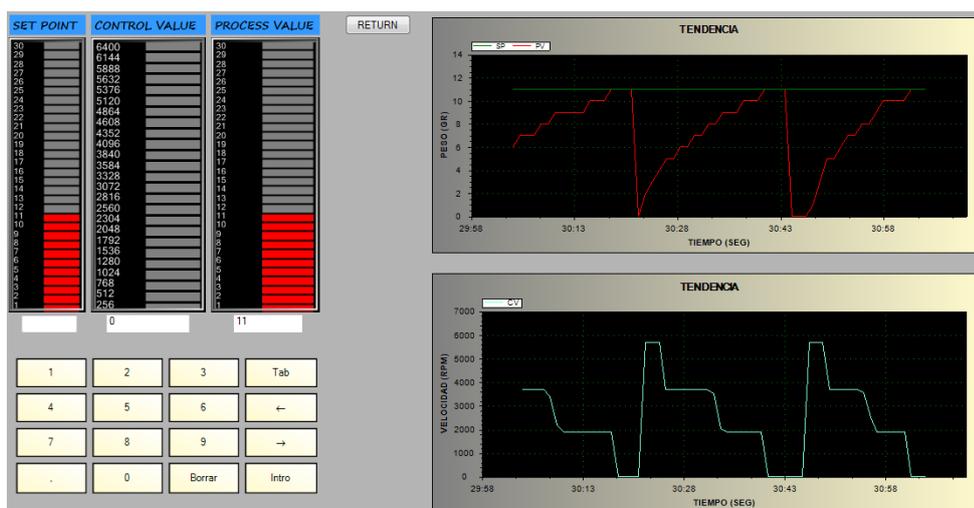


Figura 3.17: Gráfica de funcionamiento del dispensador de sólidos con un set point de 11 gramos para 3 muestras continuas

En la Figura 3.18 se puede observar el desempeño del dispensador de sólidos con un set point establecido de 18 gramos, de igual forma el tiempo que toma en llenar el recipiente con dicho set point.

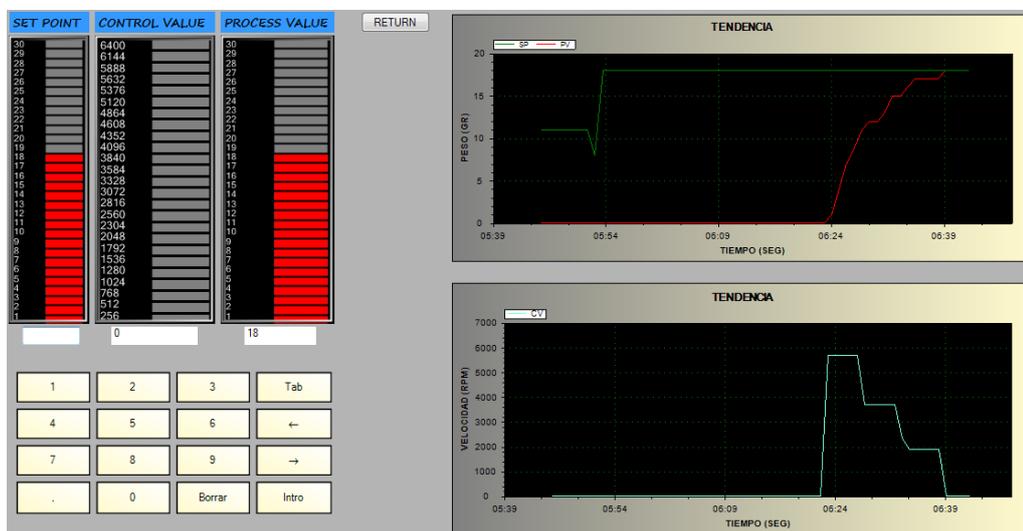


Figura 3.18: Gráfica de funcionamiento del dispensador de sólidos con un set point de 18 gramos

En la Figura 3.19 se observa el desempeño del dispensador de sólidos trabajando de manera continua verificando las formas de onda de las variables manejadas en el proceso (set point, control value y process value).

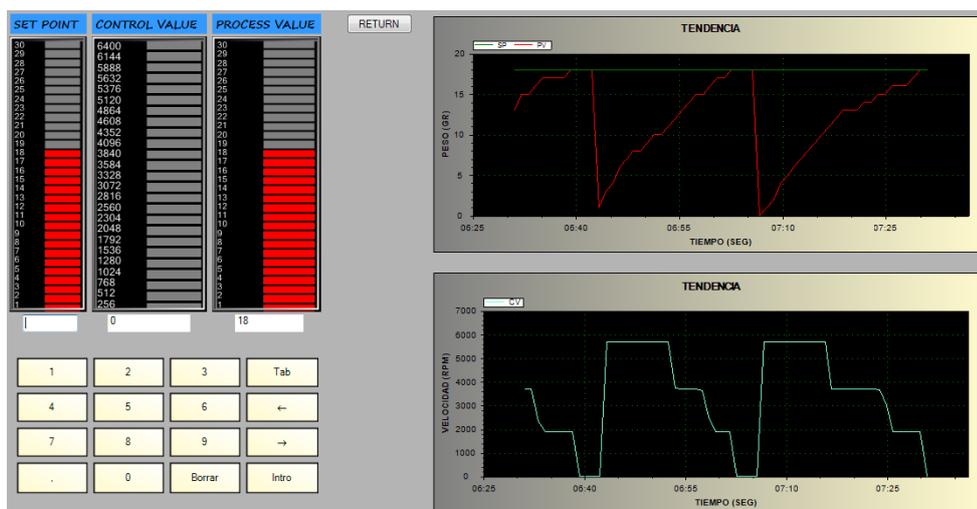


Figura 3.19: Gráfica de funcionamiento del dispensador de sólidos con un set point de 18 gramos para 3 muestras continuas

Una vez verificado y observado gráficamente el comportamiento del dispensador de sólidos perteneciente a la Etapa 2 se puede concluir que, para un set point de 18 gramos toma más tiempo en llenar el recipiente pero controla de mejor manera el error resultante, mientras que para un set point de 5 gramos toma menos tiempo en llenar el recipiente pero el error incrementa en un pequeño porcentaje de 0,05 gramos que es un valor considerablemente pequeño a nivel didacticó, el cual puede ser comparado a nivel industrial siendo un muy buen resultado ya que no genera grandes pérdidas a lo largo de la producción del producto final , calificando al control difuso de peso aplicado el dispensador de sólidos como sobresaliente.

3.13. Pruebas de calidad de la Etapa 5

La revisión del producto final se realiza mediante el pesaje de los contenedores o recipientes los cuales fueron depositados en una caja anteriormente en la Etapa 4 y del contenido o peso neto de cada recipiente que fue llenado en la Etapa 2.

Los pesos de cada uno de los elementos que constan del producto final fueron pesados en la balanza de la Etapa 5, vale recalcar que los pesos de las cajas varía un pequeño porcentaje el cual se muestra en la Tabla 3.2.

Tabla 3.2: Taras o pesos de los recipientes empleados en el producto final

TARAS(PESO DE RECIPIENTES EN GR)	
MUESTRA 1	158,38
MUESTRA 2	158,35
MUESTRA 3	158,6
MUESTRA 4	158,33
MUESTRA 5	157,64
MUESTRA 6	158,11
MUESTRA 7	158,36
MUESTRA 8	157,06
MUESTRA 9	152,72
MUESTRA 10	159,39

MUESTRA 11	155,18
MUESTRA 12	158,76
PROMEDIO CAJA	158,298
RECIPIENTE	4,8
TAPA	1,6

Se determinó que las cajas tienen un peso promedio por unidad de 158,298 gramos, cabe recalcar que se eliminaron 2 muestras (muestra 9 y muestra 11) o cajas defectuosas que no constataban un peso aproximado o igual al de las demás 10 cajas, por ello se debieron retirar del contenedor de cajas, en la Tabla 3.2 se puede determinar visulamente las cajas defectuosas.

Se genera una fórmula, la cual será el punto principal para realizar la comparativa entre lo calculado y lo medido en la Etapa 5 y de esta forma clasificar el producto final.

La fórmula encontrada se detalla de la siguiente manera:

$$PESO BRUTO = 4 * (RECIPIENTE Y TAPA) + CAJA + 4 * (SET POINT)$$

- Peso bruto: Peso total del producto final
- Peso de recipiente: Peso determinado por el fabricante (4,8 gramos).
- Peso de la tapa: Peso determinado por el fabricante (1,6 gramos).
- Peso de la caja: Peso determinado de acuerdo a un análisis estadístico de 12 cajas o muestras (158,298 gramos).
- Set point: Peso neto del recipiente, establecido en la Etapa 2 (11gramos) y modificable de acuerdo al operador.

De acuerdo a la fórmula descrita el operador puede definir el porcentaje de error que puede admitir para que el producto final apruebe la inspección. El porcentaje de error o tolerancia es definido de acuerdo a los porcentajes anteriormente calculados en la Etapa 2, los cuales son:

- Un error promedio de + 0,2 gramos en el dispensador de sólidos con un factor de seguridad de 0,5 que da un total de + 0,3 gramos de error admitido.
- Un error promedio de +/- 1 gramo en la medida del peso de la caja de acuerdo a los valores medidos.

Nota: El error promedio de un recipiente lleno es multiplicado por 4 debido a la cantidad de recipientes que contiene la caja lo que incrementa el error total a considerar.

En conclusión, la tolerancia mínima resultante queda establecida en +/- 2,2 gramos del peso en bruto del producto final, el cual puede ser modificado por el operador dependiendo de los elementos manipulados en el proceso.

Se realizó una comparativa de los valores medidos del producto final con las muestras anteriores, repartidas en 8 cajas para comprobar si cumplen con la tolerancia establecida, como se indica en la Tabla 3.3.

Tabla 3.3: Valores medidos del producto final del módulo didáctico de llenado y envasado de sólidos

RESULTADOS ETAPA 5 O DE INSPECCIÓN									
PRODUCTO FINAL									
CAJA	VALORES MEDIDOS			VALORES CALCULADOS			ERROR		
	SET POINT			SET POINT			SET POINT		
	5 GR	11 GR	18 GR	5 GR	11 GR	18 GR	5 GR	11 GR	18 GR
1	204,22	229,06	254,08	203,898	227,898	255,898	0,322	1,162	-1,818
2	204,47	228,12	256,09	203,898	227,898	255,898	0,572	0,222	0,192
3	204,95	228,12	255,88	203,898	227,898	255,898	1,052	0,222	-0,018
4	204,35	227,74	255,3	203,898	227,898	255,898	0,452	-0,158	-0,598
5	203,6	228,4	255,23	203,898	227,898	255,898	-0,298	0,502	-0,668
6	204,28	227,72	255,64	203,898	227,898	255,898	0,382	-0,178	-0,258
7	204,95	227,5	255,03	203,898	227,898	255,898	1,052	-0,398	-0,868
8	203,12	227,84	254,67	203,898	227,898	255,898	-0,778	-0,058	-1,228
\bar{X}	204,24	228,06	255,24	203,898	227,898	255,898	0,3445	0,1645	-0,658

De acuerdo a la Tabla 3.3 se comprueba que todas las muestras del producto final cumplen con los requisitos establecidos anteriormente, y también se puede observar el error promedio que comete la Etapa 5 con respecto a los siguientes set points establecidos:

- Set point = 5gr, el error es de 0,3445 gr
- Set point = 11gr, el error es de 0,1645 gr
- Set point = 18gr, el error es de -0,658 gr

La Etapa 5 dispone de luces indicadoras, que permiten al operador visualizar la aprobación o rechazo del producto final el cual se encuentra conformado por los pesos de los 4 recipientes con sus cubiertas, el peso de la caja y el peso neto vertido en cada recipiente anteriormente descrito y calculado.

En la Figura 3.20 se indica como el producto final pasa la inspección al visualizar en el panel de control una luz piloto de color verde.



Figura 3.20: Aprobación por parte de la etapa de inspección del producto final

De igual forma se puede observar en el panel frontal una luz indicadora en señal de rechazo del producto final sin haber cumplido el peso en bruto establecido, dicho producto tendrá que ser verificado por el operador, como se indica en la Figura 3.21.



Figura 3.21: Rechazo por parte de la etapa de inspección del producto final

3.14. Pruebas de comunicación

Para la comprobación de la correcta comunicación de la placa electrónica Arduino se comprueba que la PC donde se encuentra conectada haya reconocido cada uno de los dispositivos, en la se indica en la Figura 3.22 el reconocimiento y asignación automática del puerto serie.

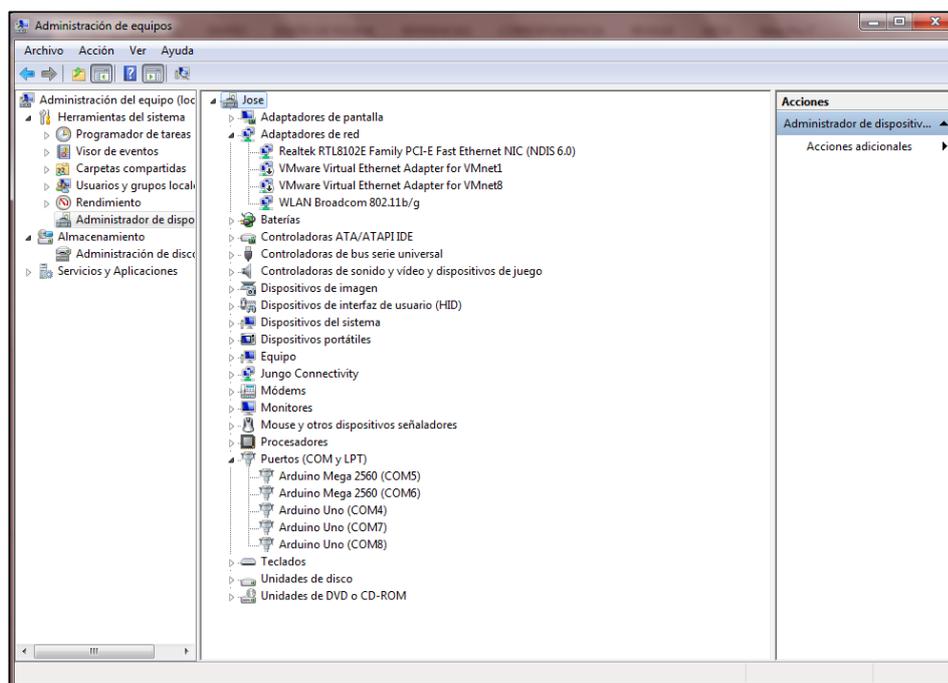


Figura 3.22: Asignación de puertos COM para las tarjetas Arduino

Se comprueba en el software APCAutomation de Acimut Monitoriza for Arduino que el led "Status" se encuentre parpadeando y se haya creado rutas de comunicación con los dispositivos conectados como se muestra en la Figura 3.23.

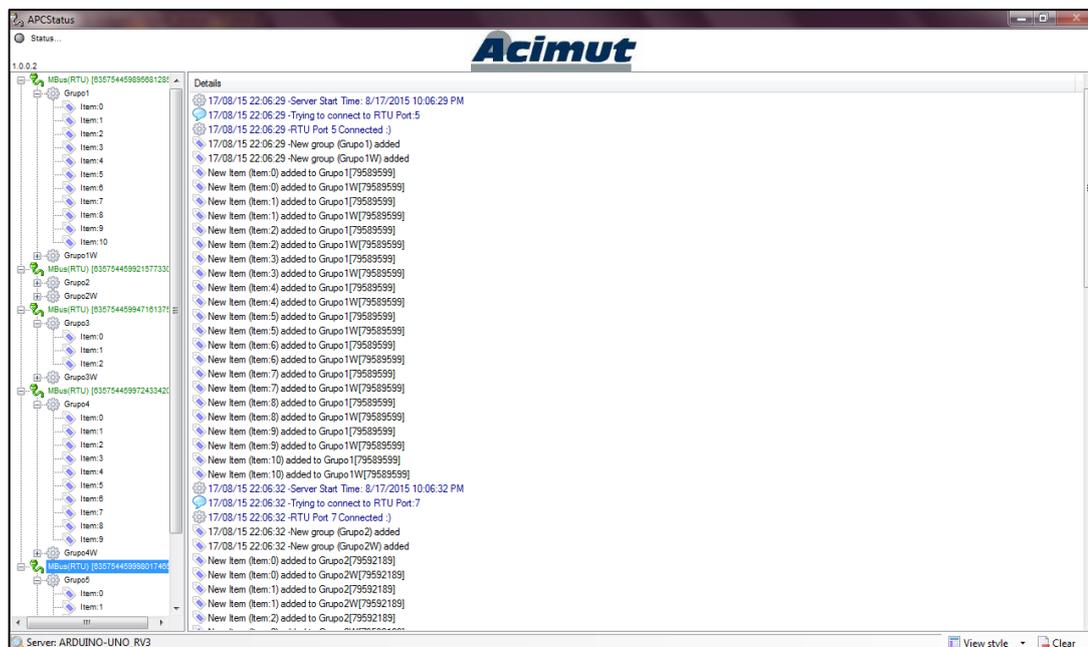


Figura 3.23: Estado de la conexión de las tarjeta Arduino en Acimut Monitoriza for Arduino

Existe la posibilidad de que no haya sido configurado correctamente el puerto de comunicación serie de uno o varios dispositivos conectados, como resultado una comunicación errónea como se muestra en la Figura 3.24.

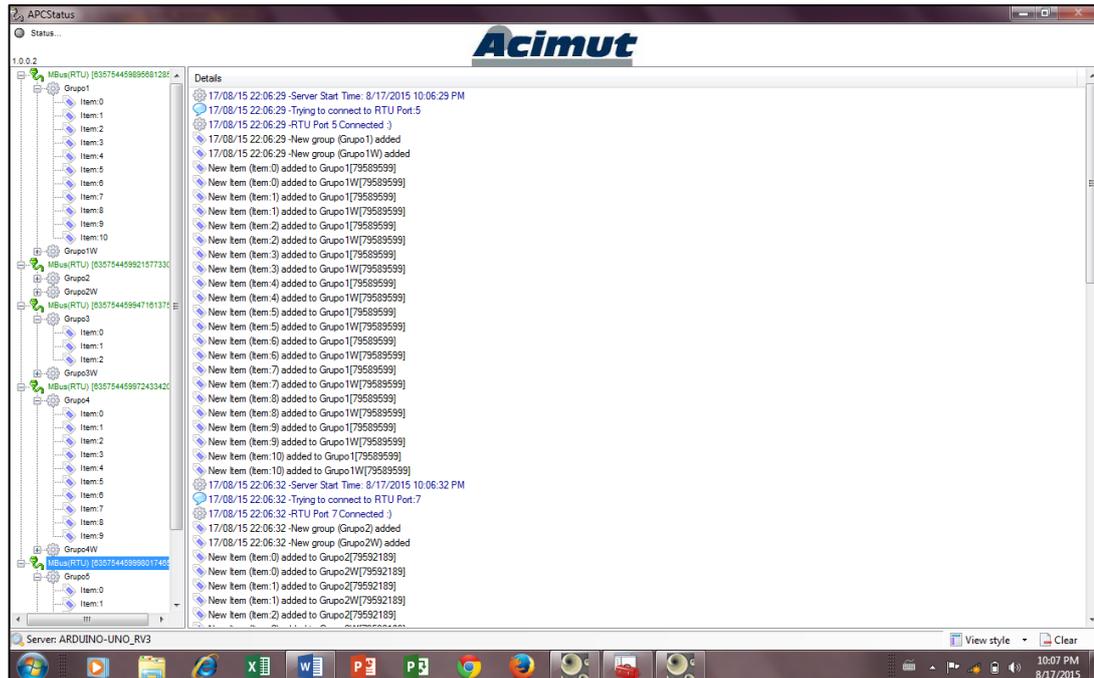


Figura 3.24: Comunicación errónea

3.15. Diagrama de tiempos del módulo didáctico de llenado y envasado de sólidos

Al realizar las respectivas pruebas y mediciones de tiempos de cada una de las etapas que conforman el módulo didáctico de llenado y envasado de sólidos, mediante la ayuda de la función `millis()` en el software arduino, que provee del tiempo exacto que utiliza el microcontrolador para ejecutar las instrucciones establecidas en la programación, devolviendo un valor de tipo entero largo el cual está establecido en milisegundos, los mismos datos que se obtuvieron en cada una de las distintas etapas del módulo didáctico generando un diagrama de tiempos como se indica en la Figura 3.25.

Después de haber obtenido el diagrama de tiempos se puede observar que el tiempo de producción del módulo didáctico queda limitado por el tiempo de envasado de la Etapa 3, ya que el tiempo de envasado no puede ser reducido debido a que contiene una bomba de vacío y el tiempo medido en purgar el aire después de cada activación es de 9 segundos, dejando un restante de 8,708 segundos para los demás actuadores de la Etapa 3.

Diagrama de tiempos del Módulo didáctico de llenado y envasado de sólidos

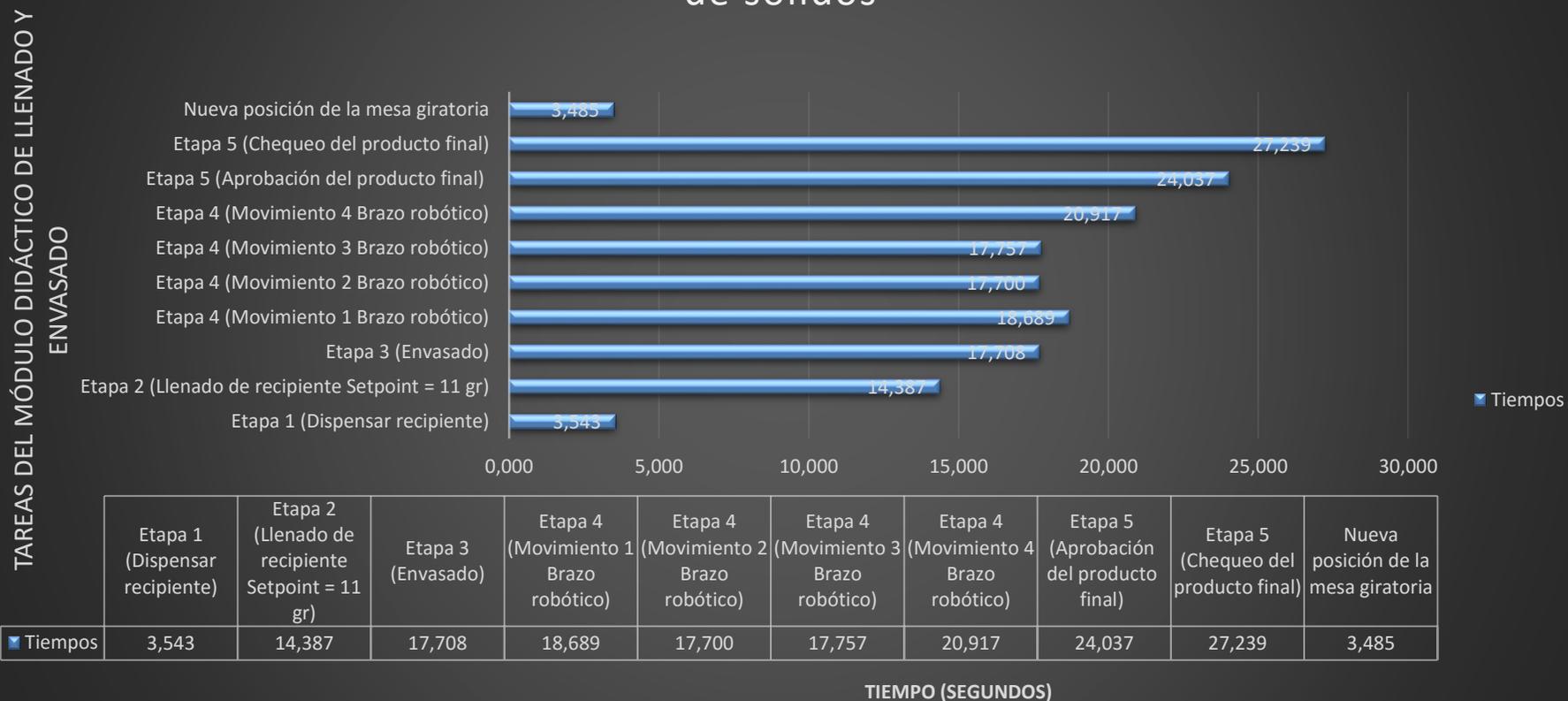


Figura 3.25: Diagrama de tiempos del funcionamiento del Módulo didáctico de llenado y envasado de sólidos

3.16. Pruebas y funcionamiento del HMI

Se verifica el funcionamiento de la interfaz HMI en el software Monitoriza for Arduino Acimut–Scada, variando parámetros de configuración para la comunicación como el rango de actualización en milisegundos y se comprobó que los datos a visualizar no contenían errores, pero al contrario la visualización de eventos, alarmas y tendencias se torno lenta. Como solución a este problema se minimizó el tiempo de actualización de envío y recepción de datos en el software Monitoriza for Arduino, como se aprecia en la Figura 3.26 el UpdateRate (velocidad de actualización) al crear el o los grupos de cada servidor ModbusRTU por defecto viene en 1000 ms, esto quiere decir que actualizará los datos cada 1 segundo provocando dicho retardo en el envío y recepción de datos.



Figura 3.26: UpdarateRate por defecto

Cambiando este parámetro a 1 milisegundo, es decir a 0,001 segundos la velocidad de actualización de datos mediante el protocolo de comunicación ModBus RTU entre el dispositivo Arduino a controlar y el software Acimut Monitoriza for Arduino para la visualización de las variables será mucho mas rápida en comparación al UpdateRate que viene por defecto al momento de crear los grupos de dicho servidor (Figura 3.27).

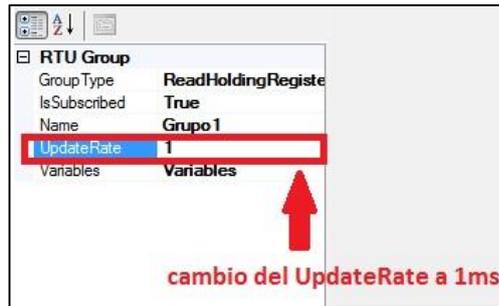


Figura 3.27: UpdateRate modificado a 1ms

Se realizan pruebas modificando valores de escritura en la etapa 2, como el set point de la cantidad a dispensar en gramos, de la misma forma trasladar este dato a la Etapa 5 para poder actualizar el peso del producto final y verificar si es aprobada o no la inspección, la cual fue satisfactoria y se constató la correcta interacción de la interfaz HMI con los esclavos de la red ModBus RTU.

CAPÍTULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- El diseño y construcción del módulo didáctico de un sistema de llenado y envasado de sólidos, permitió verificar el funcionamiento de un proceso industrial a pequeña escala, el cual será de gran ayuda para los estudiantes de la carrera de Ing. Electrónica e Instrumentación, incrementando los conocimientos prácticos y reforzando los conocimientos teóricos.
- La operación en paralelo de las etapas del módulo didáctico optimiza los recursos y aumenta la velocidad de producción, ya que estos son los principales parámetros a tomar en consideración en la automatización de un proceso industrial.
- El manejo de tarjetas Arduino en la automatización del proceso de llenado y envasado de sólidos no requiere de licencia alguna ya que está basado en una plataforma libre, pero para implementarlo a nivel industrial se necesita de acondicionamiento de hardware para evitar el mal funcionamiento de los dispositivos.
- Fue necesario acoplar filtros RC pasa bajos en las entradas digitales e interrupciones, para rechazar las frecuencias altas que normalmente se producen por ruidos inductivos de la red eléctrica, cabe recalcar que las tarjetas utilizadas no disponen de ningún sistema de protección frente a la presencia de dichos ruidos.
- El ruido inductivo generado por el accionamiento de relés provocó un mal funcionamiento en las distintas etapas del módulo didáctico, reseteando contadores y generando señales falsas en las entradas digitales de las tarjetas Arduino, por lo que se vio necesario implementar filtros RC paso bajos para eliminar las frecuencias altas generadas por dichos ruidos.
- El uso del protocolo de comunicación ModBus RTU en las tarjetas Arduino permitió recolectar información en tiempo real del proceso en

funcionamiento del módulo didáctico, lo que fue de gran ayuda para monitorear el proceso y administrar eventos externos que suscitan, que serán manejados por el operador.

- La velocidad de producción alta, media o baja, está limitada por los actuadores que se encuentran en el proceso de llenado y envasado de sólidos.
- El módulo didáctico de llenado y envasado de sólidos se limita a envasar sólidos como los cereales, debido a la composición densidad y peso de dicho material, lo que conlleva a un óptimo desempeño en la etapa de transporte.
- En la industria se requiere de controles exactos para evitar pérdidas en la producción de materiales líquidos o sólidos, debido a esto se implementó un control difuso en la etapa de llenado para que la medida sea lo más exacta posible y no exista pérdidas en la producción.
- El uso de dispositivos de medida y el correcto acondicionamiento de los mismos, acorde a las condiciones presentes en el módulo didáctico fue de gran importancia, ya que, de las señales o respuestas a cambios producidas por los mismos, depende el sincronismo de cada etapa en el módulo didáctico.
- El uso de interrupciones generadas por eventos externos en las tarjetas Arduino, facilitó e incrementó la velocidad del proceso obteniendo la independencia de cada etapa en el módulo didáctico.
- El uso de brazos robóticos fue una limitante para utilizar diversos materiales sólidos para el llenado, debido a la inestabilidad de los mismos y falencia de torque en los servomotores que conforman los mismos.
- Gracias a la implementación de los diversos tipos de sensores banner provistos por el laboratorio de redes y control de procesos se obtuvo mejoras y grandes ventajas en el funcionamiento del módulo didáctico.

4.2. Recomendaciones

- El implementar tarjetas electrónicas mucho más robustas aumenta la fiabilidad en el proceso, pero conlleva un gasto mayor, por esta razón se recomienda aislar la parte electrónica de la parte de potencia para obtener un rendimiento óptimo.
- El aprovechar la comunicación serie que dispone la tarjeta Arduino para utilizar el protocolo ModBus RTU fue de gran ayuda para la recolección de información, pero se pudo haber utilizado “shields ethernet” para obtener la información del proceso más rápido y poder añadir aplicaciones web para hacer más amigable el entorno con el operador, esto gracias a los diversos complementos que dispone la plataforma Arduino.
- La importancia de la implementación en el módulo didáctico de una etapa de inspección es de gran relevancia, debido a que se asemeja a la parte de comprobación de la mercadería antes de salir a la venta en una empresa, por lo que se clasifican los productos aceptados y rechazados para su expendio.
- El sobredimensionar parámetros como torque en servomotores es de gran importancia para procesos que manejan diversos materiales que van a ser procesados, en este caso el torque de los servomotores que constituyen los brazos robóticos no generaron el torque suficiente para transportar el producto, lo cual se solucionó sustituyendo dichos servomotores por unos de mayor torque.
- El realizar este tipo de proyectos en software libre abre una gran puerta de ideas a ser implementadas, se recomienda aportar a esta comunidad para que muchas más ideas como está se vean plasmadas a precios módicos y de gran ayuda para los estudiantes acorde al tema.
- Se recomienda implementar y diseñar módulos didácticos basados en procesos industriales, para incrementar los conocimientos de los estudiantes de la carrera Ing. Electrónica e Instrumentación, y así en el momento de buscar una fuente laboral disponer de experiencia en el manejo de dichos procesos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. G. Pere Ponsa, «Diseño y Automatización Industrial,» *Diseño Industrial*, pp. 2-30, Julio 2010.
- [2] Hurtado y Rivas, Robótica y Automatización Industrial, «<http://www.hurtadorivas.com>,» Diciembre 2014. [En línea]. Available: <http://www.hurtadorivas.com/productos.php>. [Último acceso: 10 Febrero 2015].
- [3] Control, Promoting Excellence in Process Automation, «<http://www.controlglobal.com/>,» 10 Abril 2012. [En línea]. Available: <http://www.controlglobal.com/articles/2012/montague-microbrewery-grows-micro-plc/>. [Último acceso: 12 Abril 2015].
- [4] Universidad del País Vasco, «Automatización,» sgiiker, [En línea]. Available: <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/Automatizacion/Automatizacion.htm>. [Último acceso: 13 Mayo 2015].
- [5] P. Abarca, «SISTEMAS DE CONTROL AUTOMÁTICO,» AIE, 2011.
- [6] National Instruments, "www.ni.com," 08 Enero 2014. [Online]. Available: <http://digital.ni.com/public.nsf/allkb/413F24FE159630C4862573620011BAF6>. [Accessed 19 Mayo 2015].
- [7] J. Cosco Grimaney, «UNIVERSIDAD NACIONAL DE INGENIERÍA: CONTROLES ELÉCTRICOS Y AUTOMATIZACIÓN,» 2011. [En línea]. Available:

<http://ecaths1.s3.amazonaws.com/ipgn/1755989761.1-definiciones-basicas-de-control.pdf>. [Último acceso: 2015 Mayo 19].

- [8] F. Torres, «<http://www.aurova.ua.es>,» [En línea]. Available: http://rua.ua.es/dspace/bitstream/10045/18432/1/Tema%201_Introduccion.pdf. [Último acceso: 18 Mayo 2015].
- [9] Solving Systems Engineering, [En línea]. Available: <http://s2e.es/diferencia-entre-scada-y-hmi/>. [Último acceso: 19 Mayo 2015].
- [10] J. Q. y. J. Aguilar-Martín, «pymes,» [En línea]. Available: <http://publicacionesymedios.net/articulo.php?ida=1057>. [Último acceso: 22 Noviembre 2014].
- [11] J. E. Morales Sánchez, . E. . V. Talavera Moctezuma, A. Suarez Calleja y A. Hernández López, *Definición de proyecto para la clase de Robótica*, pp. 1-2, 2014.
- [12] J. M. Angulo Usategui, S. Romero Yesa y I. Angulo Martínez, *Introducción a la robótica*, Madrid: Thomson, 2005.
- [13] Universidad de Guadalajara, «<http://proton.ucting.udg.mx>,» Octubre 2014. [En línea]. Available: <http://proton.ucting.udg.mx/materias/robotica/>. [Último acceso: 28 Mayo 2015].
- [14] A. Castro Gómez y A. Valero Gómez, «DISEÑO Y PUESTA EN FUNCIONAMIENTO DE UN BRAZO ROBÓTICO IMPRIMIBLE,» Madrid, España, Septiembre 2012.

- [15] R. Sanchis Llopis, J. A. Romero Pérez y C. V. Ariño Latorre, «Ingeniería de Sistemas Industriales y Diseño,» [En línea]. Available: <http://www.uji.es/bin/publ/edicions/auto.pdf>. [Último acceso: 08 08 2015].
- [16] OMEGA, «OMEGA,» 03 Agosto 2015. [En línea]. Available: <http://es.omega.com/prodinfo/galgas-extensiométricas.html>. [Último acceso: 10 Agosto 2015].
- [17] METAL ACTUAL, «Servomotores, la Misma Potencia con Ahorro para la Industria,» [En línea]. Available: http://www.metallactual.com/revista/25/maquinaria_servo.pdf. [Último acceso: 11 03 2015].
- [18] BIOINGENIERIA CO, «MOTORES DE DC,» [En línea]. Available: http://www.bioingenieria.edu.ar/academica/catedras/control/archivos/material/Anexos/anexo%20laboratorio%20control%20de%20posicion%20_motor%20cc_.pdf. [Último acceso: 11 08 2015].
- [19] Tecnología, «TECNOLOGIA,» [En línea]. Available: <http://www.tecnologia-informatica.es/tecnologia/los-motores-electricos.php>. [Último acceso: 11 08 2015].
- [20] Arduino, «Arduino.cc,» 2015. [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Último acceso: 11 08 2015].
- [21] Ó. T. Artero, Arduino : curso práctico de formación, San Fernando de Henares, Madrid: ALFAOMEGA, 2013.

- [22] R. L. Boylestad, «BOYLESTAD,» de *Introducción al análisis de circuitos*, México, PEARSON, 2011, pp. 713-715.
- [23] J. Quevedo y J. Aguilar Martín, «<http://publicacionesymedios.net/>,» Marzo 2008. [En línea]. Available: <http://publicacionesymedios.net/articulo.php?ida=1057>. [Último acceso: 20 Mayo 2015].
- [24] OMEGA, 03 Agosto 2015 . [En línea]. Available: <http://es.omega.com/prodinfo/galgas-extensiometricas.html>. [Último acceso: 10 Agosto 2015].
- [25] Arduino, «Arduino,» Arduino, 1 Abril 2015. [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: 2 Agosto 2015].
- [26] GitHub, «GitHub,» GitHub, 7 Julio 2015. [En línea]. Available: <https://github.com/bogde/HX711>. [Último acceso: 13 Mayo 2015].
- [27] Adafruit, «Arduino/Shields,» Adafruit, 5 Agosto 2015. [En línea]. [Último acceso: 14 Mayo 2015].
- [28] Pesquisiar, «Pesquisiar,» sexta-feira, 28 Septiembre 2012. [En línea]. Available: <http://www.zerokol.com/2012/09/arduinofuzzy-fuzzy-library-for-arduino.html>. [Último acceso: 13 Mayo 2015].
- [29] encoder, «encoder,» Electronic Projects, 5 Enero 2015. [En línea]. Available: http://www.pjrc.com/teensy/td_libs_Encoder.html. [Último acceso: 7 Marzo 2015].

ANEXOS

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERIA ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por los señores Sebastián Francisco Panchi Olivo y Washington Paúl Herrera Sinchiguano bajo nuestra supervisión.

.....

Ing. EDWIN PRUNA
DIRECTOR DEL PROYECTO

.....

Ing. GALO ÁVILA
CODIRECTOR DEL PROYECTO

.....

Ing. FRANKLYN SILVA
DIRECTOR DE LA CARRERA

.....

Dr. RODRIGO VACA
SECRETARIO ACADÉMICO
UNIDAD DE ADMISIÓN Y REGISTRO