

# UNIVERSIDAD CATÓLICA SANTA MARÍA

FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERÍA MECÁNICA,

MECÁNICA ELÉCTRICA Y MECATRÓNICA



“Diseño e implementación de un sistema de clasificación  
por visión artificial basado en redes neuronales  
y servo posicionamiento lineal industrial”

Tesis para optar el Título de **Ingeniero Mecatrónico**, que presentan los bachilleres:

Bch. Elliot Grover Rodríguez Sánchez

Bch. Mario Eduardo Zárate Cáceres

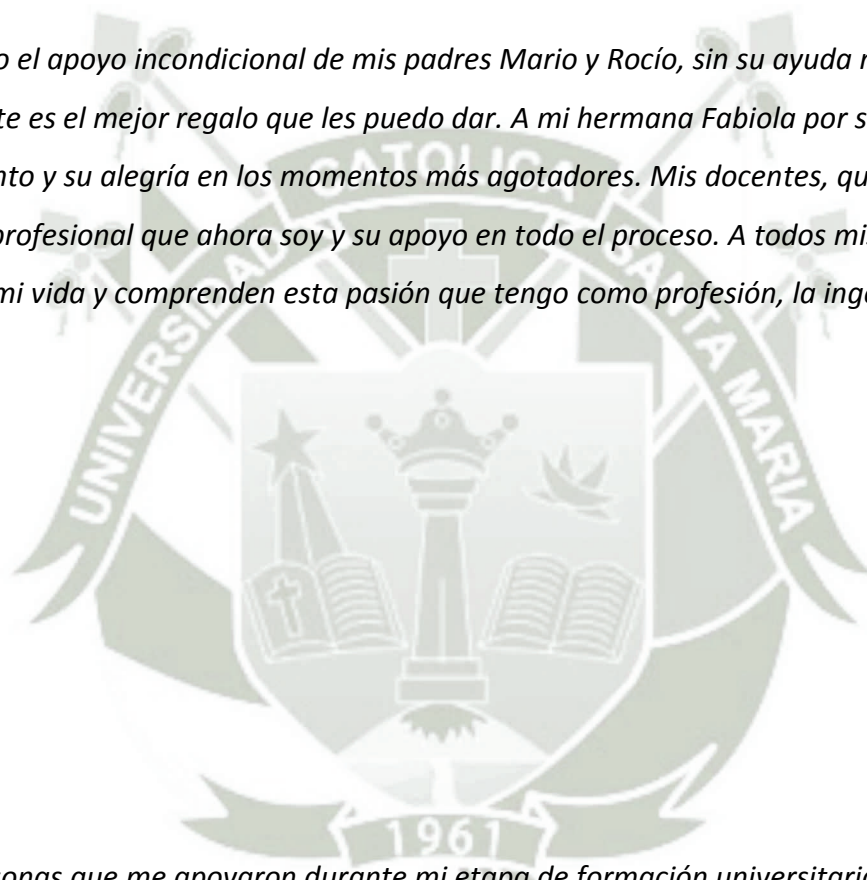
**Dirigida por:** Ing. Sergio Mestas Ramos

Arequipa, Marzo del 2015

## Agradecimientos

*Agradezco el apoyo incondicional de mis padres Mario y Rocío, sin su ayuda nada de esto sería posible, este es el mejor regalo que les puedo dar. A mi hermana Fabiola por su comprensión en todo momento y su alegría en los momentos más agotadores. Mis docentes, que contribuyeron a formar el profesional que ahora soy y su apoyo en todo el proceso. A todos mis amigos, que son parte de mi vida y comprenden esta pasión que tengo como profesión, la ingeniería, mi locura.*

*Edu Zárate C.*



*A todas las personas que me apoyaron durante mi etapa de formación universitaria, a los que estuvieron desde un principio y seguirán conmigo, a quienes por diferentes motivos ya no se encuentran presentes y a quienes se sumaron en el proceso.*

*Elliot Rodríguez S.*

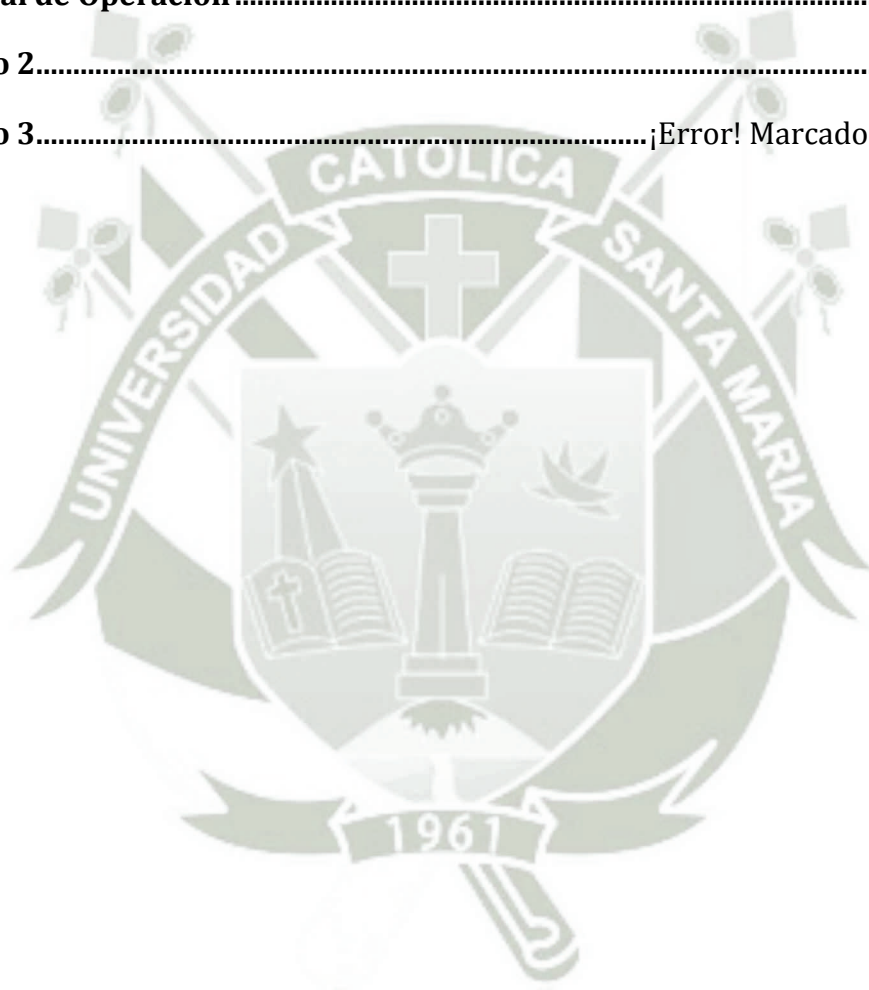
# Índice general

<b>Agradecimientos</b> .....	<b>2</b>
<b>Índice general</b> .....	<b>i</b>
<b>Índice de figuras</b> .....	<b>v</b>
<b>Índice de tablas</b> .....	<b>x</b>
<b>Resumen</b> .....	<b>xii</b>
<b>Abstract</b> .....	<b>xiii</b>
<b>Capítulo 1</b> .....	<b>1</b>
<b>Metodología</b> .....	<b>1</b>
1.1 Introducción.....	1
1.2 Objetivos.....	2
1.2.1. Objetivo General .....	2
1.2.2 Objetivos Específicos.....	2
1.3 Identificación del problema.....	2
1.3.1 Descripción del problema .....	3
1.3.2 Hipótesis.....	5
1.4 Justificación .....	5
1.4.1 Justificación Técnica.....	5
1.4.2 Justificación Económica .....	5
1.4.3 Justificación Social.....	5
1.5 Alcances .....	6
<b>Capítulo 2</b> .....	<b>7</b>
<b>Marco Teórico</b> .....	<b>7</b>
2.1 Sistemas de Visión y Hardware: Visión artificial .....	7

2.1.1	Propiedades básicas de las imágenes digitales.....	8
2.1.2	Introducción a los sensores de imagen CMOS.....	8
2.1.3	Conceptos básicos sobre procesamiento digital de la imagen.....	8
2.1.4	Estrategias para el procesamiento de imágenes digitales.....	9
2.1.5	Software dedicado para el procesamiento.....	9
2.2.	Inteligencia artificial .....	12
2.2.1	Redes neuronales.....	13
2.2.2	Algoritmo K-Nearest Neighbors .....	15
2.2.3	Algoritmo.....	15
2.2.4	Elección del $k$ .....	17
2.2.5	Posibles variantes del algoritmo básico.....	18
2.3	Controladores industriales .....	19
2.4	Servo Posicionamiento industrial.....	22
2.4.1	Motor síncrono de imán permanente .....	24
2.5	Electroimán de solenoide.....	25
2.5.1	Electroimán e imán permanente .....	26
2.5.2	Funcionamiento .....	27
2.5.3	Usos.....	28
<b>Capítulo 3</b>	.....	<b>29</b>
<b>Diseño e implementación</b>	.....	<b>29</b>
3.1	Diseño de sistema.....	29
3.1.1	Estado del Arte.....	30
3.1.2	Secuencia de operaciones y estructura de funciones.....	39
3.1.3	Descripción de procesos. ....	40
3.1.4	Agrupación de funciones .....	41

3.1.6 Selección de Cámara .....	52
3.1.7 Selección de procesador gráfico .....	58
3.1.8 Amplificación de señal de control.....	60
3.1.9 Selección de servomotor.....	64
3.1.10 Selección de controlador .....	69
3.1.11 Selección de actuador .....	70
3.1.12 Diseño y selección de elementos mecánicos.....	74
3.1.13 Listado de elementos y costo de proyecto .....	75
3.2 Implementación del sistema .....	77
3.2.1 Arquitectura de control.....	77
3.2.2 Controlador: Raspberry Pi.....	79
3.2.3 Interfaz electrónica .....	95
<b>Capítulo 4 .....</b>	<b>100</b>
<b>Programación del sistema .....</b>	<b>100</b>
4.1 Diagrama de flujo del funcionamiento.....	100
4.2 Proceso de aprendizaje .....	101
4.2.1 Procesamiento de la imagen.....	101
4.2.2 Aprendizaje del sistema .....	107
4.3 Clasificación de piezas .....	108
4.4 Ubicar las piezas .....	115
4.5 Separar las piezas .....	118
4.5.1 Parametrización de controlador industrial .....	118
4.5.2 Parametrización driver servomotor .....	146
<b>Capítulo 5 .....</b>	<b>156</b>
<b>Pruebas y resultados.....</b>	<b>156</b>

<b>Conclusiones .....</b>	<b>168</b>
<b>Recomendaciones .....</b>	<b>170</b>
<b>Limitaciones.....</b>	<b>171</b>
<b>Bibliografía.....</b>	<b>173</b>
<b>Anexos.....</b>	<b>176</b>
<b>Anexo 1.....</b>	<b>177</b>
<b>Manual de Operación .....</b>	<b>177</b>
<b>Anexo 2.....</b>	<b>180</b>
<b>Anexo 3.....</b>	<b>¡Error! Marcador no definido.</b>



# Índice de figuras

<b>FIGURA 1:</b>	ESQUEMA DEL SISTEMA A MONTAR.....	4
<b>FIGURA 2:</b>	ESQUEMA DE UBICACIONES.....	4
<b>FIGURA 3:</b>	PARTES DE UNA RED NEURONAL.....	14
<b>FIGURA 4:</b>	EJEMPLO DE K-NN.....	15
<b>FIGURA 5:</b>	ARQUITECTURA DE PLC.....	19
<b>FIGURA 6:</b>	FLUJO DE CORRIENTE.....	26
<b>FIGURA 7:</b>	PROCESO DE DISEÑO DE UN PROYECTO.....	30
<b>FIGURA 8:</b>	ESQUEMA DEL FUNCIONAMIENTO DEL SISTEMA DE VISIÓN.....	39
<b>FIGURA 9:</b>	ESQUEMA DEL SISTEMA DE VISIÓN.....	39
<b>FIGURA 10:</b>	ESQUEMA DEL SISTEMA MECÁNICO.....	39
<b>FIGURA 11:</b>	FUNCIONES SECUENCIALES.....	41
<b>FIGURA 12:</b>	FUNCIONAMIENTO DEL SISTEMA.....	41
<b>FIGURA 13:</b>	SISTEMA MECÁNICO DE SELECCIÓN.....	42
<b>FIGURA 14:</b>	SELECCIÓN, SISTEMA DE VISIÓN.....	42
<b>FIGURA 15:</b>	DIAGRAMA DE SOLUCIONES.....	51
<b>FIGURA 16:</b>	ESQUEMA DE FUNCIONAMIENTO.....	52
<b>FIGURA 17:</b>	CARACTERÍSTICAS DE PROCESADOR.....	53
<b>FIGURA 18:</b>	POSIBLE DISPOSICIÓN DE LA CÁMARA.....	54
<b>FIGURA 19:</b>	CÁMARA C615 LOGITECH.....	57
<b>FIGURA 20:</b>	ANÁLISIS DE CORRIENTE EN CIRCUITO DE SALIDA DE SEÑAL.....	60
<b>FIGURA 21:</b>	ANÁLISIS DE CORRIENTE EN CIRCUITO DE SALIDA DE ENTRADA.....	62
<b>FIGURA 22:</b>	MOTOR SEW CMP40M.....	65
<b>FIGURA 23:</b>	CARACTERÍSTICAS TÉCNICAS DEL MOTOR.....	65
<b>FIGURA 24:</b>	EXPLICACIÓN DE LAS CARACTERÍSTICAS DEL MOTOR.....	66
<b>FIGURA 25:</b>	LA ESTRUCTURA INTERNA DEL SERVO SEW:.....	67
<b>FIGURA 26:</b>	VARIADOR DE FRECUENCIA SEW EURODRIVE.....	69
<b>FIGURA 27:</b>	CONTROLADOR LÓGICO PROGRAMABLE VIPA.....	70
<b>FIGURA 28:</b>	ELECTROIMÁN.....	71
<b>FIGURA 29:</b>	WIKIPEDIA: ELECTROIMÁN.....	71
<b>FIGURA 30:</b>	ELECTROIMÁN INDUSTRIAL.....	72
<b>FIGURA 31:</b>	GEOMETRÍA DEL ELECTROIMÁN.....	73
<b>FIGURA 32:</b>	DISTRIBUCIÓN DEL SISTEMA EN VISTA ISOMÉTRICA.....	74
<b>FIGURA 33:</b>	HUSILLO SIN FIN.....	74
<b>FIGURA 34:</b>	DETALLE DE B EN EJE HUSILLO.....	75

<b>FIGURA 35:</b>	ETAPAS DE UNA APLICACIÓN.....	78
<b>FIGURA 36:</b>	RASPBERRY PI.MODEL A.....	79
<b>FIGURA 37:</b>	SISTEMA OPERATIVO RASPBIAN.....	82
<b>FIGURA 38:</b>	ARCHIVO DE IMAGEN DEL SISTEMA. ....	83
<b>FIGURA 39:</b>	CARPETA DE WIN32DISKIMAGER.....	84
<b>FIGURA 40:</b>	VENTA DEL PROGRAMA WIN32. ....	84
<b>FIGURA 41:</b>	PORTE INFERIOR DE RASPBERRY PI.....	84
<b>FIGURA 42:</b>	PANTALLA DE INICIO DEL RASPBERRY EN PRIMERA CONEXIÓN.....	85
<b>FIGURA 43:</b>	VENTANA DE INICIO. ....	85
<b>FIGURA 44:</b>	PARTICIÓN DEL SISTEMA. ....	86
<b>FIGURA 45:</b>	OPCIÓN "OVERSCAN".....	86
<b>FIGURA 46:</b>	CONFIGURACIONES LOCALES.....	87
<b>FIGURA 47:</b>	CONFIGURACIÓN DE IDIOMAS.....	88
<b>FIGURA 48:</b>	SELECCIONAR EL IDIOMA ESPAÑOL.....	88
<b>FIGURA 49:</b>	IDIOMAS AGREGADOS AL SISTEMA.....	89
<b>FIGURA 50:</b>	VENTANA DE SELECCIÓN DE UBICACIÓN.....	89
<b>FIGURA 51:</b>	SELECCIÓN DE ZONA HORARIO.....	90
<b>FIGURA 52:</b>	SELECCIÓN DE MEMORIA GPU.....	90
<b>FIGURA 53:</b>	VENTANA DE ADVERTENCIA DEL OVERCLOCKING.....	91
<b>FIGURA 54:</b>	SELECCIÓN DE FRECUENCIA DEL SISTEMA.....	91
<b>FIGURA 55:</b>	VENTANA PARA INICIAR EL SISTEMA.....	92
<b>FIGURA 56:</b>	INICIACIÓN DEL SISTEMA POR PRIMERA VEZ.....	92
<b>FIGURA 57:</b>	MENÚ DE CONFIGURACIÓN.....	94
<b>FIGURA 58:</b>	CONFIGURACIONES AVANZADAS.....	95
<b>FIGURA 59:</b>	CIRCUITO DE CONEXIONES PARA LAS SALIDAS DE RASPBERRY.....	97
<b>FIGURA 60:</b>	CIRCUITO DE CONEXIONES PARA LAS ENTRADAS AL RASPBERRY.....	97
<b>FIGURA 61:</b>	POTENCIÓMETRO DIGITAL. MCP41X1.....	98
<b>FIGURA 62:</b>	CONEXIÓN DE RASPBERRY CON EL INTEGRADO.....	98
<b>FIGURA 63:</b>	VISTA EN 3D DE TARJETA.....	99
<b>FIGURA 64:</b>	VISTA SUPERIOR DE TARJETA.....	99
<b>FIGURA 65:</b>	CONEXIONES RASPBERRY.....	99
<b>FIGURA 66:</b>	DIAGRAMA DE FLUJO.....	100
<b>FIGURA 67:</b>	IMAGEN PARA INICIAR EL APRENDIZAJE.....	102
<b>FIGURA 68:</b>	ESPECTRO ELECTROMAGNÉTICO.....	102
<b>FIGURA 69:</b>	IMAGEN DE COLOR A ESCALA DE GRISES.....	103
<b>FIGURA 70:</b>	FILTRO DE LA MEDIANA.....	104
<b>FIGURA 71:</b>	RESULTADO DE APLICAR EL FILTRO EN LA IMAGEN.....	104

<b>FIGURA 72:</b>	ESCALA DE GRISES. ....	104
<b>FIGURA 73:</b>	RESULTADO DE HABER UTILIZADO UMBRAL. ....	105
<b>FIGURA 74:</b>	IMAGEN BINARIA DE ENTRENAMIENTO. ....	105
<b>FIGURA 75:</b>	ALGORITMO DE DETECCIÓN DE BORDES. ....	105
<b>FIGURA 76:</b>	IMAGEN CON BORDES DETECTADOS. ....	106
<b>FIGURA 77:</b>	DIAGRAMA DE FLUJO DEL PROCESO DE APRENDIZAJE. ....	106
<b>FIGURA 78:</b>	BORDE IDENTIFICADO. ....	107
<b>FIGURA 79:</b>	IMAGEN 640 X 480. ....	108
<b>FIGURA 80:</b>	IMAGEN DE RGB A ESCALA DE GRISES. ....	109
<b>FIGURA 81:</b>	IMAGEN CON UMBRAL. ....	109
<b>FIGURA 82:</b>	FILTRO CANNY SOBRE LA IMAGEN. ....	110
<b>FIGURA 83:</b>	DETECCIÓN DE BASE. ....	110
<b>FIGURA 84:</b>	ÁREA DE INSPECCIÓN. ....	111
<b>FIGURA 85:</b>	IMAGEN PROCESADA. ....	111
<b>FIGURA 86:</b>	IMAGEN FILTRADA EN ROJO. ....	112
<b>FIGURA 87:</b>	IMAGEN PROCESADA LISTA PARA UBICAR LAS PIEZAS. ....	112
<b>FIGURA 88:</b>	MATRIZ 100 X 100 DE LA FIGURA. ....	112
<b>FIGURA 89:</b>	ANÁLISIS DE COLOR. ....	114
<b>FIGURA 90:</b>	BASE DE LAS PIEZAS. ....	115
<b>FIGURA 91:</b>	SALIDA SPI. ....	116
<b>FIGURA 92:</b>	TOMA DE DATOS DE SALIDA MCP4151. ....	117
<b>FIGURA 93:</b>	RESULTADO FINAL DEL PROCESAMIENTO. ....	118
<b>FIGURA 94:</b>	VENTANA DE INICIO. ....	119
<b>FIGURA 95:</b>	COMO INSETAR UN MODULO. ....	120
<b>FIGURA 96:</b>	INSTALAR ARCHIVOS GSD. ....	120
<b>FIGURA 97:</b>	BUSCAR ARCHIVOS DESCARGADOS. ....	121
<b>FIGURA 98:</b>	CONFIGURAR BASTIDOR. ....	121
<b>FIGURA 99:</b>	INSTALAR PROFIBUS. ....	122
<b>FIGURA 100:</b>	CONFIGURAR RED PROFIBUS. ....	122
<b>FIGURA 101:</b>	MÓDULOS AGREGADOS. ....	123
<b>FIGURA 102:</b>	ASIGNACION DE MODULO 238. ....	123
<b>FIGURA 103:</b>	MÓDULO 238, CONTADOR RÁPIDO. ....	124
<b>FIGURA 104:</b>	RED DE PROFIBUS. ....	124
<b>FIGURA 105:</b>	ORGANIZACIÓN DEL PROGRAMA. ....	125
<b>FIGURA 106:</b>	DIAGRAMA PRINCIPAL. ....	126
<b>FIGURA 107:</b>	SUBROUTINA DE CONDICIONES INICIALES. ....	127
<b>FIGURA 108:</b>	SUBROUTINA DE REFERENCIAMIENTO. ....	128

<b>FIGURA 109:</b>	SUBROUTINA DE CLASIFICACIÓN.....	129
<b>FIGURA 110:</b>	BLOQUES UTILIZADOS EN EL PROGRAMA.....	130
<b>FIGURA 111:</b>	PROGRAMA PRINCIPAL.....	131
<b>FIGURA 112:</b>	REVISIÓN DE CONDICIONES INICIALES.....	132
<b>FIGURA 113:</b>	REFERENCIAMIENTO.....	134
<b>FIGURA 114:</b>	CLASIFICACIÓN.....	136
<b>FIGURA 115:</b>	CLASIFICACIÓN.....	138
<b>FIGURA 116:</b>	CÁLCULO DE SET POINT.....	140
<b>FIGURA 117:</b>	ESCALAMIENTO DE LA ENTRADA ANALÓGICA.....	141
<b>FIGURA 118:</b>	ESCALAMIENTO DE PULSOS DE ENCODER.....	141
<b>FIGURA 119:</b>	SEÑAL DE SALIDA.....	142
<b>FIGURA 120:</b>	LAZO DE CONTROL PID.....	143
<b>FIGURA 121:</b>	SEÑALES ASOCIADAS AL BLOQUE PID.....	143
<b>FIGURA 122:</b>	PROGRAMA DE PAUSE.....	145
<b>FIGURA 123:</b>	PROGRAMACIÓN DEL STOP.....	145
<b>FIGURA 124:</b>	ESTRUCTURA DEL PROGRAMA.....	145
<b>FIGURA 125:</b>	VENTANA DE ESTADO.....	146
<b>FIGURA 126:</b>	MODOS DE CONTROL DEL DRIVER.....	147
<b>FIGURA 127:</b>	LAZOS DE CONTROL POR POSICIÓN Y VELOCIDAD.....	148
<b>FIGURA 128:</b>	SERVOMOTOR MÁS TARJETA LTX.....	149
<b>FIGURA 129:</b>	DIAGRAMA DE CONEXIÓN DRIVER- SERVO MOTOR.....	150
<b>FIGURA 130:</b>	CONEXIÓN DEL CONECTOR X14.....	150
<b>FIGURA 131:</b>	LINE CHOKE, SEW.....	151
<b>FIGURA 132:</b>	ACOMETIDA ELÉCTRICA.....	152
<b>FIGURA 133:</b>	BORNERAS DE DRIVER.....	153
<b>FIGURA 134:</b>	CONFIGURACIÓN DEL MODO 1.....	154
<b>FIGURA 135:</b>	SECUENCIA DE SEÑALES DE ENCODER.....	155
<b>FIGURA 136:</b>	CIRCUITO DE CONEXIÓN DEL ENCODER.....	155
<b>FIGURA 137:</b>	CIRCUITO DE ENCODER ENSAMBLADO.....	155
<b>FIGURA 138:</b>	MÓDULO ENSAMBLADO.....	156
<b>FIGURA 139:</b>	ARCHIVOS EN EL RASPBERRY PI.....	157
<b>FIGURA 140:</b>	SUBPROGRAMAS DENTRO DE LA CARPETA “LIB”.....	158
<b>FIGURA 141:</b>	VENTANA DE INICIO.....	159
<b>FIGURA 142:</b>	VENTANA DE APRENDIZAJE.....	159
<b>FIGURA 143:</b>	IMAGEN CENTRADA EN LA PANTALLA.....	161
<b>FIGURA 144:</b>	RESULTADO DEL UMBRAL ADAPTATIVO.....	162
<b>FIGURA 145:</b>	RESULTADO DE UMBRAL ADAPTATIVO.....	162

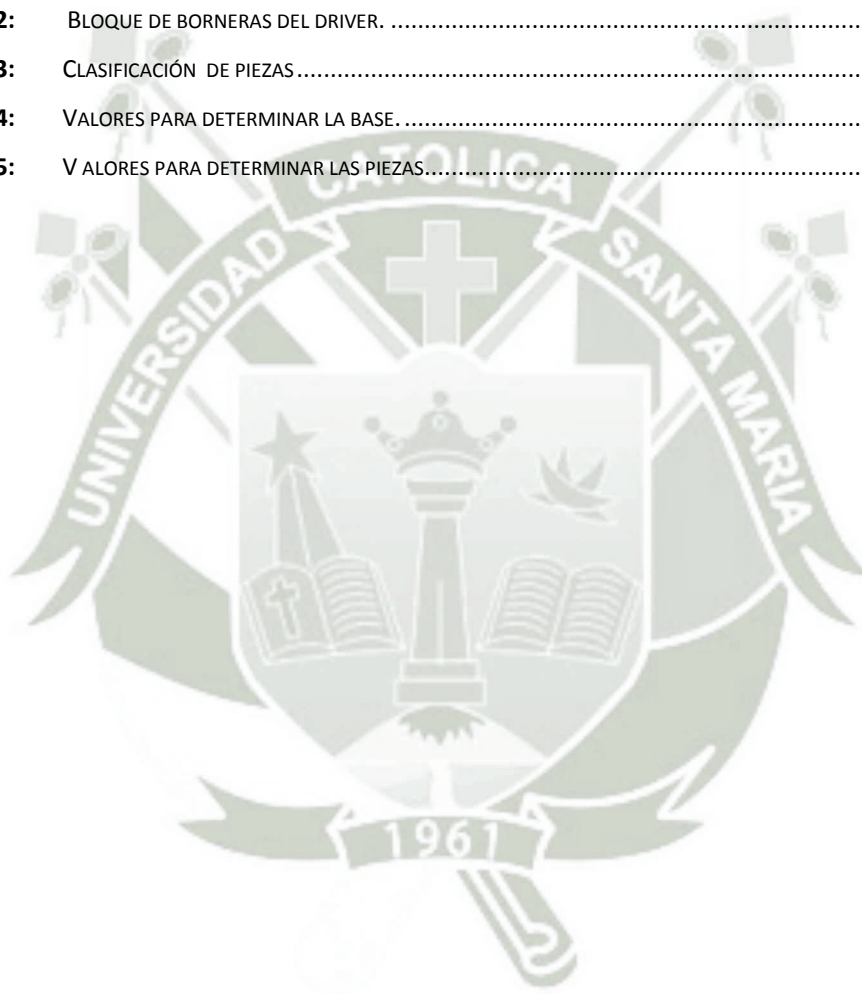
<b>FIGURA 146:</b>	VENTA DE LA OPCIÓN “CALIBRACIÓN” .....	163
<b>FIGURA 147:</b>	ARCHIVO .TXT DE PROCESAMIENTO. ....	164
<b>FIGURA 148:</b>	SELECCIÓN DE CARACTERÍSTICAS. ....	164
<b>FIGURA 149:</b>	PIEZAS CLASIFICADAS. ....	165
<b>FIGURA 150:</b>	ACTUADOR CLASIFICANDO PIEZAS.....	167



# Índice de tablas

<b>TABLA 1:</b>	CÁMARAS SEMI INDUSTRIALES. ....	31
<b>TABLA 2:</b>	CÁMARAS INDUSTRIALES.....	32
<b>TABLA 3:</b>	CÁMARAS MULTIUSO. ....	33
<b>TABLA 4:</b>	COMPAÑÍAS DE SOLUCIONES DE CLASIFICACIÓN. ....	34
<b>TABLA 5:</b>	FUNCIONES REALIZADAS POR OPERARIOS.....	34
<b>TABLA 6:</b>	LISTA DE EXIGENCIAS.....	38
<b>TABLA 7:</b>	PORTADORES DE FUNCIONES .....	43
<b>TABLA 8:</b>	ASPECTOS POSIBLES DE EVALUACIÓN DEL CONCEPTO DE SOLUCIÓN.....	44
<b>TABLA 9:</b>	FORMATO DE EVALUACIÓN DE CONCEPTO DE SOLUCIÓN .....	44
<b>TABLA 10:</b>	SISTEMA DE VISIÓN-CONTROL.....	45
<b>TABLA 11:</b>	TABLA DE EVALUACIÓN DEL SISTEMA DE VISIÓN.....	46
<b>TABLA 12:</b>	EVALUACIÓN DEL SISTEMA ELECTRÓNICO.....	46
<b>TABLA 13:</b>	SISTEMA ELECTRÓNICO.....	47
<b>TABLA 14:</b>	ENVÍO DE SEÑALES DIGITALES.....	48
<b>TABLA 15:</b>	EVALUACIÓN DEL SISTEMA ELECTRÓNICO PARA ENVÍO DE SEÑALES.....	49
<b>TABLA 16:</b>	EVALUACIÓN TÉCNICA DE LAS SOLUCIONES.....	50
<b>TABLA 17:</b>	EVALUACIÓN ECONÓMICA DEL AS SOLUCIONES.....	51
<b>TABLA 18:</b>	IMÁGENES TOMADAS POR LA WEBCAM.....	54
<b>TABLA 19:</b>	EVALUACIÓN Y TIEMPO DEL PROCESAMIENTO.....	55
<b>TABLA 20:</b>	EVALUACIÓN DE IMAGEN DE 1920 X 1080.....	56
<b>TABLA 21:</b>	HARDWARE IMPLEMENTADO.....	58
<b>TABLA 22:</b>	RESULTADOS DE PROCESAMIENTO.....	58
<b>TABLA 23:</b>	CARACTERÍSTICAS DEL HCNE.....	73
<b>TABLA 24:</b>	SISTEMA POR VISIÓN.....	75
<b>TABLA 25:</b>	TARJETA PARA SEÑALES.....	75
<b>TABLA 26:</b>	TARJETA PARA ENCODER.....	76
<b>TABLA 27:</b>	ELEMENTOS DE CONTROL.....	76
<b>TABLA 28:</b>	ELEMENTOS MECÁNICOS.....	76
<b>TABLA 29:</b>	ACCESORIOS PARA PODER TERMINAR TODO EL CABLEADO.....	76
<b>TABLA 30:</b>	RESUMEN FINAL DEL COSTO DEL PROYECTO.....	77
<b>TABLA 31:</b>	CARACTERÍSTICAS TÉCNICAS DEL MODELO RASPBERRY.....	81
<b>TABLA 32:</b>	PINOUTS RASPBERRY.....	96

<b>TABLA 33:</b>	NOMENCLATURA DE CONEXIONES. ....	96
<b>TABLA 34:</b>	INFORMACIÓN PARA EL APRENDIZAJE. ....	107
<b>TABLA 35:</b>	CARACTERÍSTICAS DE LA BASE. ....	111
<b>TABLA 36:</b>	REQUERIMIENTOS DEL COLOR ROJO. ....	111
<b>TABLA 37:</b>	CARACTERÍSTICAS DE LAS FIGURAS. ....	112
<b>TABLA 38:</b>	CORRESPONDENCIA NUMÉRICA DE FORMA. ....	113
<b>TABLA 39:</b>	DESIGNACIÓN DE TAMAÑOS. ....	114
<b>TABLA 40:</b>	DESIGNACIÓN DE TAMAÑOS. ....	114
<b>TABLA 41:</b>	CONEXIÓN DE CABLES X14.....	151
<b>TABLA 42:</b>	BLOQUE DE BORNERAS DEL DRIVER. ....	153
<b>TABLA 43:</b>	CLASIFICACIÓN DE PIEZAS.....	160
<b>TABLA 44:</b>	VALORES PARA DETERMINAR LA BASE.....	161
<b>TABLA 45:</b>	V ALORES PARA DETERMINAR LAS PIEZAS.....	162



# Resumen

El desarrollo tecnológico global viene creciendo a grandes pasos, a su vez el desarrollo en el Perú sigue avanzando, cada vez se necesita más tecnología, más investigación, más proyectos. Del mismo modo, uno de los rubros importantes en nuestro país es el de la agricultura, muchos de los procesos industriales de productos como la palta, chirimoya, arándano o uva, requieren de estrictos procesos de clasificación por *forma*, *tamaño* y *color*, así cumplir con normas y alcanzar estándares internacionales de calidad. Estos procesos en su mayoría son manuales, lamentablemente las personas están sujetas a fatiga y cansancio visual, razón por la cual no producen del mismo modo durante sus horas de trabajo. De este modo, se propone un *sistema automatizado de clasificación* que agilice este proceso y asegure el cumplimiento de estándares de calidad. En el presente trabajo de tesis se plantea el *diseño y la implementación de un sistema clasificación por visión artificial* para realizar la selección de pequeñas piezas metálicas, validando *métodos inteligentes y algoritmos en el procesamiento de imágenes* para así realizar una adecuada clasificación utilizando un controlador lógico programable y un servomotor, elementos comunes en la *industria*. Este mismo sistema se puede replicar en otros procesos, tan solo acondicionando los actuadores para la manipulación de los productos, se lograría mejorar los tiempos de clasificación.

# Abstract

The global technology development is growing by big steps, similarly, development in Peru continues in progress. More technology, more research, more projects are needed. Similarly, one of the important fields in our country is agriculture, many industrial processes of products such as avocado, custard apple, blueberry or grape require strict classification by shape, size and color, therefore, accomplish local standards and accomplish international standards of quality. These processes are mostly manual, unfortunately people are subject to fatigue and eyestrain, whereby they do not produce the same way during his working hours. In this way, it intends an *automated classification system* to expedite this process and ensure quality standards. In this thesis the design and implementation of a *classification system by artificial vision* are proposed for selection of small metal parts, validating intelligent methods and algorithms in image processing in order to make a proper classification using a programmable logic controller and a servomotor, common elements in the industry. The same system is replicable in other processes, only with adapting the actuators for handling the products, classification times would improve.

# Capítulo 1

## Metodología

### 1.1 Introducción

Actualmente la industria peruana viene incrementando los procesos productivos debido a la mayor demanda del mercado local y también a las crecientes tasas de exportación, entre las tareas comúnmente realizadas en los procesos productivos se encuentra las de clasificación, pero en muchos casos, estas no se puede realizar mediante sensores o instrumentación convencional, teniendo que utilizarse costosos métodos de clasificación por visión artificial o en muchos casos siendo esta clasificación realizada por operarios, limitando la producción y descuidando la relación calidad - costo de los procesos implícitos.

Existe entonces un mercado potencial para este tipo de soluciones tecnológicas; sin embargo, estas no se encuentran en nuestra industria debido al alto costo que involucra importar esta tecnología del extranjero, el conocimiento limitado en su uso y la escasa mano de obra especializada que se requiere para su puesta a punto son las razones por las cuales se desarrolló este tipo de solución.

El presente trabajo está orientado a diseñar y construir un sistema de clasificación utilizando visión artificial y servo posicionamiento lineal, procesando imágenes mediante sistemas inteligentes capaces de aprender y listos para integrarse con elementos utilizados en la industria.

## 1.2 Objetivos

### 1.2.1. Objetivo General

Diseñar e implementar un sistema de clasificación, usando reconocimiento por visión artificial basado en redes neuronales y servo posicionamiento lineal industrial.

### 1.2.2 Objetivos Específicos

- Realizar el diseño mecánico del equipo, generando planos y especificaciones técnicas.
- Ensamblar el sistema propuesto, adjuntar lista de componentes y procedimientos de construcción.
- Definir los algoritmos inteligentes para el procesamiento de imágenes y definir el proceso de servo posicionamiento lineal.
- Enlazar el proceso de visión con redes neuronales en hardware dedicado “open-source<sup>1</sup>” con proceso de posicionamiento lineal industrial estándar.

## 1.3 Identificación del problema

El problema surge a partir de la necesidad de clasificar objetos por características que no pueden ser diferenciados y/o detectados con sensores convencionales, es por ello que se ve la necesidad de implementar el uso de visión artificial, pero que a su vez sea un sistema capaz de enlazarse con elementos industriales que presentan distintas características, en este caso lo haremos con un servomotor y un Controlador Lógico Programable.

---

<sup>1</sup> **Open-Source:** Código abierto, es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

### 1.3.1 Descripción del problema

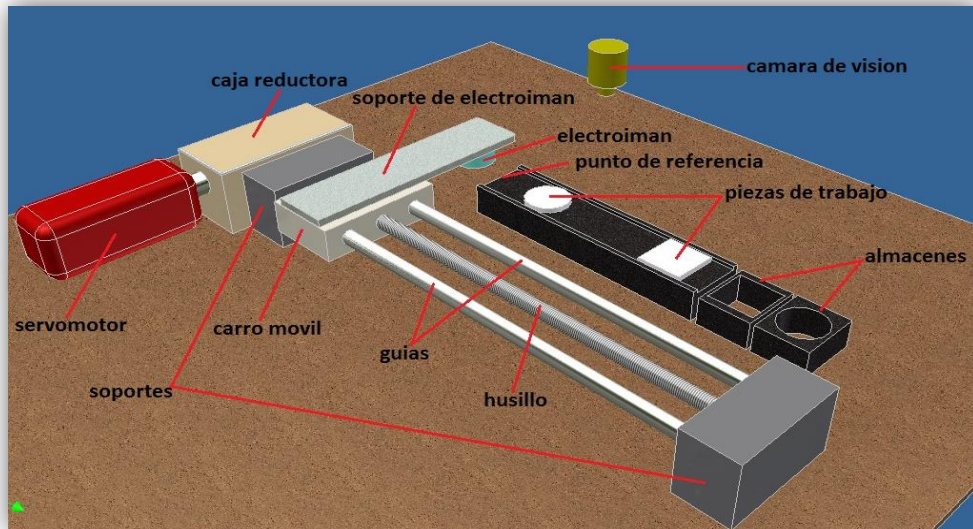
Actualmente en la pequeña y mediana industria del país se observa un crecimiento acelerado, debido principalmente al mayor consumo en el mercado local y a las mayores tasas de exportaciones que se vienen registrando, para sostener este crecimiento las industrias deben invertir en diversas tecnologías que les permitan aumentar su producción manteniendo estándares de calidad en sus procesos.

En la industria, uno de los procesos comúnmente observados es el de clasificación de objetos, en muchos casos la naturaleza de estos hace difícil la utilización de sensores convencionales para su reconocimiento teniendo que utilizar soluciones con visión artificial que se basen en algoritmos inteligentes para poder discriminar entre diferentes formas, esto requiere de un previo aprendizaje. Si bien en el mercado existen soluciones a este problema, los pocos fabricantes que lo ofrecen tienen tarifas que difícilmente pueden ser pagados por pequeños o medianos empresarios. Otra de las razones por la cual no se puede acceder a estos equipos, es porque las industrias que fabrican este tipo de soluciones, lo realizan con empresas que tienen altos volúmenes de producción, volúmenes que actualmente no tienen los productores. Además, como parte de los sistemas de clasificación, en la industria se manejan procesos estandarizados de posicionamiento mediante servomotores; por lo tanto lo que se necesita es lograr una implementación de clasificación de bajo costo con reconocimiento por visión artificial y que sea capaz de adaptarse a sistemas industriales existentes.

Uno de los problemas a resolver es el de comunicación entre ambos sistemas propuestos. El sistema de visión estará basado en electrónica de bajo nivel, de 5V. Para poder enviar la información de clasificación al PLC, se tiene que utilizar algunos principios electrónicos básicos y así asegurar la comunicación entre ambos. Posteriormente una vez que el controlador industrial tiene la información, este puede procesarla y realizar la correcta clasificación de los objetos mediante el control del

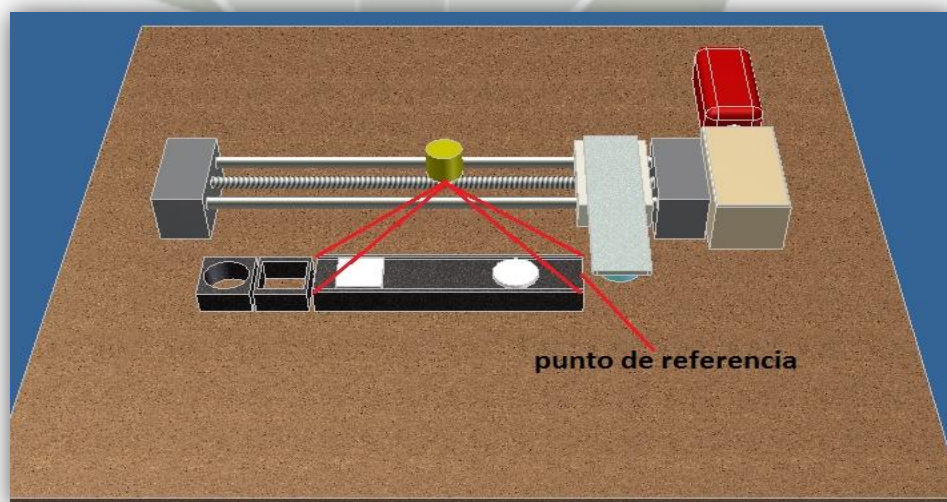
servo-motor. Solo es necesario revisar muy bien la programación que hará posible el movimiento del eje.

Para poder ejecutar todo el procesos de clasificación de manera exitosa, es necesario que el sistema este mecánicamente bien ensamblado, reducir fricción y que el movimiento sea uniforme y suave, de lo contrario se tendrá problemas de fricción, vibraciones y poca precisión.



**Figura 1:**Esquema del sistema a montar.

*Fuente: Elaboración propia.*



**Figura 2:**Esquema de ubicaciones.

*Fuente: Elaboración propia.*

### 1.3.2 Hipótesis

Es posible identificar características físicas de objetos, como forma, tamaño y color empleando algoritmos de visión artificial sobre las imágenes obtenidas con la cámara, para así poder clasificarlas según las características mencionadas. Si se desarrolla el sistema, entonces se conseguirán una solución que podría adaptarse a elementos industriales y ayudar al desarrollo de la pequeña y mediana empresa en el sur del país.

## 1.4 Justificación

### 1.4.1 Justificación Técnica

- Estudio de aplicaciones con visión artificial.
- Reconocimiento de patrones basado en algoritmos con redes neuronales.
- Implementación de procesos de posicionamiento lineal con elementos industriales.
- Necesidad de una solución con visión artificial adaptable a aplicaciones con buses de comunicación industrial (Servomotores).

### 1.4.2 Justificación Económica

- Búsqueda de una solución económica para la industria local con sistemas de visión artificial integrables a procesos industriales con servo posicionamiento.
- Los clientes industriales que usen este sistema se verían beneficiados con el aumento de producción, lo que les generaría mayores ingresos.
- Se disminuirían significativamente los costos de producción elevando las ganancias de los potenciales usuarios.

### 1.4.3 Justificación Social

- Se generará mayor interés de investigación y desarrollo en áreas de visión artificial y redes neuronales poco estudiadas.
- Los sistemas de clasificación automáticos mejoran las condiciones de trabajo de los operarios, haciendo las tareas más seguras y de condiciones ergonómicas.

## 1.5 Alcances

El presente proyecto tiene como alcance el diseñar e implementar un sistema de clasificación, generando los planos y especificaciones técnicas requeridas, basado en el reconocimiento de patrones. El proceso de clasificación está en base al posicionamiento lineal. Para la visión artificial y procesamiento de imágenes se utilizó hardware basado en el sistema Raspberry Pi, el lenguaje de programación es Python™ y se aplicaron algoritmos inteligentes basados en el aprendizaje supervisado.

Para el posicionamiento lineal se utilizó un servomotor y driver industrial marca SEW®, el control se realizó a través de un PLC industrial de gama media-alta VIPA®, ambos de origen Alemán líderes en la industria nacional e internacional, la comunicación y enlace de los elementos se hizo con señales digitales. La herramienta utilizada para manipular los objetos procesados es un electroimán magnético plano.

Las señales de los periféricos del computador Raspberry Pi están adaptadas mediante elementos electrónicos para que puedan comunicarse con el controlador industrial, de esta forma se completa el sistema de clasificación planteado, resumiendo en un solo proyecto el uso de sistemas de visión artificial, procesamiento de imágenes utilizando algoritmos inteligentes, servo posicionamiento industrial, uso de hardware y software Open-Source y su integración con elementos industriales.

Como en el mercado existe una gran variedad de productos con los cuales se podría demostrar su eficiencia, se implementó un módulo de aprendizaje en el cual se pueda clasificar pequeñas figuras geométricas por sus características de forma, *tamaño o color*. Además se enlazó este sistema en conjunto a un PLC y servomotor industrial, cumpliendo las exigencias y requerimientos de la industria en cuanto a sistemas de automatización.

# Capítulo 2

## Marco Teórico

### 2.1 Sistemas de Visión y Hardware: Visión artificial

La visión artificial, también conocida como visión por computador (*del inglés computer vision*) o visión técnica, es un sub-campo de la inteligencia artificial. El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características de una imagen. Los objetivos típicos de la visión artificial incluyen:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas).
- La evaluación de los resultados (por ejemplo, segmentación, registro).
- Registro de diferentes imágenes de una misma escena u objeto, es decir, hacer concordar un mismo objeto en diversas imágenes.
- Seguimiento de un objeto en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena; este modelo podría ser usado por un robot para navegar por la escena.
- Estimación de las posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido.

Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesamiento de imágenes, teoría de grafos y otros campos. La visión artificial cognitiva está muy relacionada con la psicología cognitiva y la computación biológica. [1]

### **2.1.1 Propiedades básicas de las imágenes digitales**

Las imágenes de señal continua se reproducen mediante dispositivos electrónicos analógicos que registran los datos de la imagen con precisión utilizando varios métodos, como una secuencia de fluctuaciones de la señal eléctrica o cambios en la naturaleza química de la emulsión de una película, que varían continuamente en los diferentes aspectos de la imagen. Para procesar o visualizar en el ordenador una señal continua o una imagen analógica, se debe convertir primero a un formato comprensible para el ordenador o formato digital. Este proceso se aplica a todas las imágenes, independientemente de su origen, de su complejidad y de si son blanco y negro (escala de grises) o a todo color. Una imagen digital se compone de una matriz rectangular (o cuadrada) de píxeles que representan una serie de valores de intensidad ordenados en un sistema de coordenadas  $(x,y)$ .

### **2.1.2 Introducción a los sensores de imagen CMOS**

Los sensores de imagen CMOS se han diseñado con la capacidad para integrar un número de funciones de procesamiento y control directamente en el circuito integrado del sensor, lo que se extiende más allá de la tarea fundamental de recopilación de fotones. Estos nuevos aspectos incluyen generalmente lógica temporal, control de exposición, conversión de analógico a digital, obturación, balance de blancos, ajuste del aumento y algoritmos de procesamiento inicial de la imagen. Se están introduciendo sensores de imagen CMOS económicos en el campo de la microscopía óptica en instrumentos para fines educativos que combinan una calidad óptica aceptable con paquetes de software de control e imagen fáciles de usar.

### **2.1.3 Conceptos básicos sobre procesamiento digital de la imagen**

El procesamiento digital de la imagen permite una modificación reversible prácticamente libre de ruido en forma de una matriz de enteros, en vez de las clásicas manipulaciones en el cuarto oscuro o filtración de voltajes dependientes del tiempo necesarios para las imágenes analógicas y señales de video. Incluso aunque muchos algoritmos de procesamiento de imágenes son extremadamente potentes, el usuario

medio a menudo aplica operaciones a imágenes digitales sin tener en cuenta los principios subyacentes tras dichas manipulaciones. Las imágenes que resultan de una manipulación descuidada están a menudo distorsionadas en comparación con las que podrían producirse si la potencia y versatilidad del software de procesamiento digital se utilizarán correctamente.

#### **2.1.4 Estrategias para el procesamiento de imágenes digitales**

Dependiendo de las condiciones de iluminación, la integridad de la muestra y los métodos de preparación, las imágenes capturadas con el microscopio óptico pueden requerir una cantidad considerable de rehabilitación/reinserción/renovación para conseguir un equilibrio entre precisión científica y composición estética. Las imágenes digitales que se obtienen mediante un CCD (charge-coupled device) o un CMOS (complementary metal oxide semiconductor) a menudo presentan señales a ruido pobres, iluminación irregular, impurezas de enfoque, deslumbramiento, cambios en los colores y otros problemas que distorsionan la calidad global de la imagen.

#### **2.1.5 Software dedicado para el procesamiento**

##### *a) Python*

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.



Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable. Aunque la programación en Python podría considerarse en algunas situaciones, hostil a la programación funcional tradicional del Lisp<sup>2</sup>, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme. [2]

#### b) OPENCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se



---

<sup>2</sup> **Lisp** (o LISP) es una familia de lenguajes de programación de computadora de tipo multiparadigma con una larga historia y una sintaxis completamente entre paréntesis. El nombre LISP deriva del "LIST Processing" (Proceso de LISTas). Las listas encadenadas son una de las estructuras de datos importantes de Lisp, y el código fuente de Lisp en sí mismo está compuesto de listas.

debe a que su publicación se da bajo licencia BSD<sup>3</sup>, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multi núcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel (IPP). [3]

### c) Pygame

Pygame es una herramienta que brinda Python, que se conforma de un conjunto de módulos para poder crear videojuegos en dos dimensiones fácilmente y está orientado al manejo de sprites.<sup>4</sup> También puede utilizarse para crear otros programas multimedia o interfaces gráficas de usuario. Funciona como interfaz de las bibliotecas SDL. [4]



---

<sup>3</sup> La **licencia BSD** es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

<sup>4</sup> Los **sprites** (del inglés 'duendecillos'). Se trata de un tipo de mapa de bits dibujados en la pantalla de ordenador por hardware gráfico especializado (MSX, Atari 400/800, Commodore 64 y Commodore Amiga fueron unos de los pocos ordenadores que soportan sprites reales) sin cálculos adicionales de la CPU.

#### d) Step7

STEP 7 (Steuerungen Einfach Programmieren, controlador simple de programación.) es un Software de Programación de PLC (Controladores Lógicos Programables el SIMATIC-S7 de Siemens, es el sucesor de SIMATIC S5 STEP 7 está ampliamente extendido en toda Alemania, Los autómatas SIMATIC constituyen un standard en la zona, compitiendo en primera línea con otros sistemas de programación y control lógico de autómatas, según la norma IEC 61131-3. STEP 7 domina el mercado de lenguajes de programación según la norma DIN EN 61131-3 disponiendo de tres lenguajes de programación:

- **FBS** - Funktionsbausteinsprache FUP (Función Diagrama de funciones)
- **KOP** - Kontaktplan englisch LD o LAD, diagrama de contactos
- **AWL** - Anweisungsliste englisch STL, lista de instrucción. [5]

## 2.2. Inteligencia artificial

La inteligencia artificial (IA) es una combinación de la ciencia del computador, fisiología y filosofía, tan general y amplio como eso, es que reúne varios campos (robótica, sistemas expertos, por ejemplo), todos los cuales tienen en común la creación de máquinas que pueden pensar. En ciencias de la computación se denomina inteligencia artificial a la capacidad de razonar de un agente no vivo. John McCarthy<sup>5</sup>, acuñó el término en 1956, la definió: "Es la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes."

- Búsqueda del estado requerido en el conjunto de los estados producidos por las acciones posibles.

---

<sup>5</sup> **John McCarthy** (Boston, Massachusetts, 4 de septiembre de 1927 - Stanford, California, 24 de octubre de 2011), fue un prominente informático que recibió el Premio Turing en 1971 por sus importantes contribuciones en el campo de la Inteligencia Artificial. De hecho, fue el responsable de introducir el término "inteligencia artificial", concepto que acuñó en la Conferencia de Dartmouth en 1956.

- Algoritmos genéticos (análogo al proceso de evolución de las cadenas de ADN).
- Redes neuronales artificiales (análogo al funcionamiento físico del cerebro de animales y humanos).
- Razonamiento mediante una lógica formal análogo al pensamiento abstracto humano.

También existen distintos tipos de percepciones y acciones, pueden ser obtenidas y producidas, respectivamente por sensores físicos y sensores mecánicos en máquinas, pulsos eléctricos u ópticos en computadoras, tanto como por entradas y salidas de bits de un software y su entorno software.

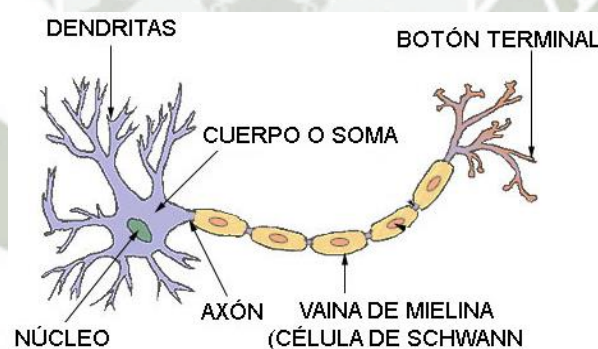
Varios ejemplos se encuentran en el área de control de sistemas, planificación automática, la habilidad de responder a diagnósticos y a consultas de los consumidores, reconocimiento de escritura, reconocimiento del habla y reconocimiento de patrones. Los sistemas de IA actualmente son parte de la rutina en campos como economía, medicina, ingeniería y la milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia como ajedrez de computador y otros videojuegos. [6]

### **2.2.1 Redes neuronales**

Las redes de neuronas artificiales (denominadas habitualmente como RNA o en inglés como: "ANN") son un modelo de aprendizaje y procesamiento sistematizado inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales. [7]

Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

- Una función de propagación (también conocida como función de excitación), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina *excitatoria*; si es negativo, se denomina *inhibitoria*.
- Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida la misma función de propagación.
- Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la función sigmoidea (para obtener valores en el intervalo  $[0,1]$ ) y la tangente hiperbólica (para obtener valores en el intervalo  $[-1,1]$ ).



**Figura 3:** Partes de una red neuronal.

Fuente: *Institution of Human & Machine Cognition (IHMC)*.

#### Algoritmos de aprendizaje

- Corrección de Error
- Basado en Memoria (Vecino más cercano [K-nn])
- Hebbian
- Competitivo
- Boltzmann

### 2.2.2 Algoritmo K-Nearest Neighbors

Este algoritmo  $k$ -nn [1] es de clasificación supervisada (Aprendizaje, estimación basada en un conjunto de entrenamiento y prototipos) que sirve para estimar la función de densidad:

$$F(x/C_j) \text{ de las predictoras } x \text{ por cada clase } C_j$$

Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento  $x$  pertenezca a la clase  $C_j$  a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras.

En el reconocimiento de patrones, el algoritmo  $k$ -nn es usado como método de clasificación de objetos (elementos) basado en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos.  $k$  - nn es un tipo de "Lazy Learning", donde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación. [9]

### 2.2.3 Algoritmo

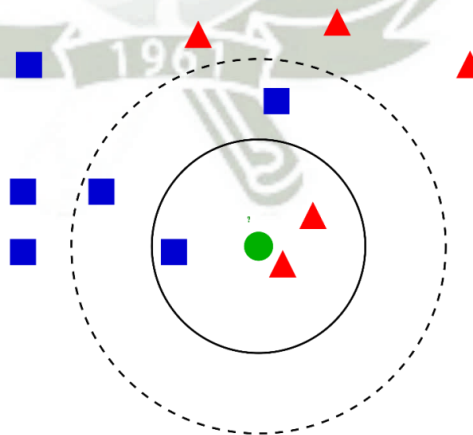


Figura 4: Ejemplo de K-nn

Fuente: Wikipedia, K-Nearest Neighbor.

*Ejemplo del algoritmo Knn.:* El ejemplo que se desea clasificar es el círculo verde. Para  $k = 3$  este es clasificado con la clase *triángulo*, ya que hay solo un cuadrado y 2 triángulos, dentro del círculo que los contiene. Si  $k = 5$  este es clasificado con la clase *cuadrado*, ya que hay 2 triángulos y 3 cuadrados, dentro del círculo externo.

Los ejemplos de entrenamiento son vectores en un espacio característico multidimensional, cada ejemplo está descrito en términos de " $p$ " atributos considerando " $q$ " clases para la clasificación. Los valores de los atributos del  $i$ -ésimo ejemplo (donde  $1 \leq i \leq n$  se representan por el vector  $p$ -dimensional)

$$x_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \in X$$

El espacio es particionado en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto en el espacio es asignado a la clase  $C$  si esta es la clase más frecuente entre los  $k$  ejemplos de entrenamiento más cercano. Generalmente se usa la distancia euclidiana.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ir} - x_{jr})^2} \dots (1)$$

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación del ejemplo (del que no se conoce su clase) es representada por un vector en el espacio característico. Se calcula la distancia entre los vectores almacenados y el nuevo vector, se seleccionan los  $k$  ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados.

Este método supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos; el problema de dicha suposición es que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación: dos atributos relevantes perderían peso entre otros veinte irrelevantes.

Para corregir el posible sesgo se puede asignar un peso a las distancias de cada atributo, dándole así mayor importancia a los atributos más relevantes. Otra posibilidad consiste en tratar de determinar o ajustar los pesos con ejemplos conocidos de entrenamiento. Finalmente, antes de asignar pesos es recomendable identificar y eliminar los atributos que se consideran irrelevantes. En síntesis, el método  $k$ -nn se resumen en dos algoritmos:

*a) Algoritmo de entrenamiento*

Para cada ejemplo  $\langle x, f(x) \rangle$ , donde  $x \in X$ , agregar el ejemplo a la estructura representando los ejemplos de aprendizaje.

*b) Algoritmo de clasificación*

Dado un ejemplar  $x_q$  que debe ser clasificado, sean  $x_1, \dots, x_k$  los  $k$  vecinos más cercanos a  $x_q$  en los ejemplos de aprendizaje, regresar:

$$\hat{f}(x_q) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k \delta(v, f(x_i)) \dots (2)$$

Dónde:

$$\delta(a, b) = 1 \text{ si } a = b; \text{ y } 0 \text{ en cualquier otro caso.}$$

El valor  $\hat{f}(x_q)$  devuelto por el algoritmo como un estimador de  $f(x_q)$  es solo el valor más común de  $f$  entre los  $k$  vecinos más cercanos a  $x_q$ . Si elegimos  $k = 1$ ; entonces el vecino más cercano a  $x_i$  determina su valor.

## 2.2.4 Elección del $k$

La mejor elección de  $k$  depende fundamentalmente de los datos; generalmente, valores grandes de  $k$  reducen el efecto de *ruido* en la clasificación, pero crean límites entre clases parecidas. Un buen  $k$  puede ser seleccionado mediante una optimización de uso. El caso especial en que la clase es predicha para ser la clase más cercana al ejemplo de entrenamiento (cuando  $k = 1$ ) es llamada *Nearest Neighbor Algorithm*, Algoritmo del vecino más cercano.

La exactitud de este algoritmo puede ser severamente degradada por la presencia de ruido o características irrelevantes, o si las escalas de características no son consistentes con lo que uno considera importante. Muchas investigaciones y esfuerzos fueron puestos en la selección y crecimiento de características para mejorar las clasificaciones. Particularmente una aproximación en el uso de algoritmos que evolucionan para optimizar características de escalabilidad. Otra aproximación consiste en escalar características por la información mutua de los datos de entrenamiento con las clases de entrenamiento.

### 2.2.5 Posibles variantes del algoritmo básico

#### a) Vecinos más cercanos con distancia ponderada

Se puede ponderar la contribución de cada vecino de acuerdo a la distancia entre él y el ejemplar a ser clasificado  $x_q$ , dando mayor peso a los vecinos más cercanos. Por ejemplo podemos ponderar el voto de cada vecino de acuerdo al cuadrado inverso de sus distancias

$$\hat{f}(x_q) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k \omega_i \delta(v, f(x_i)) \dots (3)$$

Dónde:

$$\omega_i = \frac{1}{d(x_q, x_i)^2}$$

De esta manera se ve que no hay riesgo de permitir a todos los ejemplos entrenamiento contribuir a la clasificación de  $x_q$ , ya que al ser muy distantes no tendrían peso asociado. La desventaja de considerar todos los ejemplos sería su lenta respuesta (método global). Se quiere siempre tener un método local en el que solo los vecinos más cercanos son considerados.

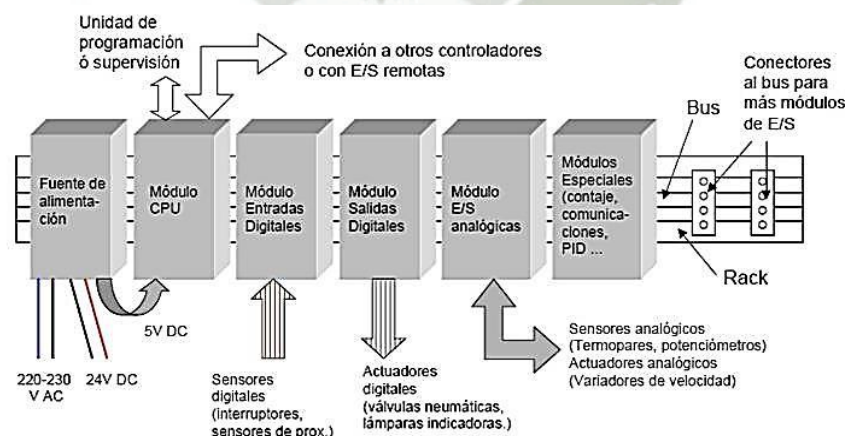
Esta mejora es muy efectiva en muchos problemas prácticos. Es robusto ante los ruidos de datos y suficientemente efectivo en conjuntos de datos grandes. Se puede

ver que al tomar promedios ponderados de los  $k$  vecinos más cercanos el algoritmo puede evitar el impacto de ejemplos con ruido aislados.

## 2.3 Controladores industriales

Un controlador lógico programable, más conocido por sus siglas en inglés *PLC* (*Programmable Logic Controller*), es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas. [10]

Los PLC's son utilizados en muchas industrias y máquinas. A diferencia de las computadoras de propósito general, el PLC está diseñado para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto. Los programas para el control de funcionamiento de la máquina se suelen almacenar en baterías copia de seguridad o en memorias no volátiles. Un PLC es un ejemplo de un sistema de tiempo real *duro* donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, que de lo contrario no producirá el resultado deseado.



**Figura 5:**Arquitectura de PLC.

Fuente: Blog de David Rojas Cubides

#### a) Selección de los PLC

La selección de un PLC como sistema de control depende de las necesidades del proceso productivo que tiene que ser automatizado, considerando como más importantes los aspectos que a continuación se enlistan:

**Espacio reducido.-** Cuando el lugar donde se tiene que instalar el sistema de control dentro de la planta es muy pequeño el PLC es la mejor alternativa, ya que aún con todos sus aditamentos necesarios llegan a ocupar un mínimo de espacio sin que esto vaya en detrimento de la productividad y la seguridad del personal y las instalaciones.

**Procesos de producción periódicamente cambiantes.-** Existen industrias como es la automotriz que año con año se ve en la necesidad de cambiar el modelo del vehículo que sale de sus plantas, razón por la cual se tiene que modificar tanto la secuencia de armado como el reajustar los valores de tolerancia de las partes con las que se arma el vehículo. Siendo el arma principal de estos cambios, las modificaciones que sufren las instrucciones del programa que controla la lógica de operación del PLC.

**Procesos secuenciales.-** Es bien conocido que cuando una actividad que se repite una gran cantidad de veces durante cierto intervalo de tiempo, se convierte en una actividad monótona para el hombre, produciendo en determinado momento fatiga del tipo emocional, provocando la desconcentración y la inducción involuntaria de errores que pueden ser fatales, tanto para la integridad del hombre como para las instalaciones. Con un PLC se puede evitar lo anterior con tan solo implementar secuencias de control, que aunque se repitan muchas veces durante el día, no se perderá la precisión con la que tienen que hacerse.

**Actuadores distintos en un mismo proceso industrial.-** Con un solo PLC se cuenta con la posibilidad de manipular actuadores de diferente naturaleza entre sí, y todavía más, con un mismo PLC se pueden dirigir diferentes líneas de producción en las que cada una tiene asignada a sus propios actuadores, esto último depende de la cantidad de salidas y en general del tamaño en cuanto a su capacidad para alojar el programa de usuario.

**Verificación de las distintas partes del proceso de forma centralizada.**- Existe una gran cantidad de industrias en que la planta de producción se encuentra alejada de la sala de control, o también por ejemplo, como es en las plantas petroleras, se tiene la necesidad de verificar la operación a distancia de todas las refinerías. Con un PLC se tiene de manera natural el diseño de redes de comunicación, para que se canalice la información a una central.

**Cantidad de I/Os Modularidad.**- Es importante que el PLC seleccionado cuente con la cantidad adecuada de I/Os o que estas puedan ser ampliadas por módulos.

**Ventajas y desventajas en el empleo de los PLC.**- Para aquellas personas que comienzan a adentrarse en el mundo de los PLC, es oportuno darles la información de lo bueno y lo malo de los PLC, para que de esta manera se cuente con todos los elementos a la hora de seleccionar el sistema de control más conveniente.

Cabe aclarar que aunque se puede automatizar cualquier proceso con un PLC, no se debe caer en la tentación de convertirlo en la panacea para solucionar todos los problemas que se nos puedan presentar, por ejemplo, si queremos controlar el llenado de un tanque de agua del inodoro que tenemos instalado en nuestra casa, el empleo de un PLC para realizar esta actividad sería un desperdicio tecnológico además de representar un costo muy alto para una tarea muy sencilla.

La utilización de un PLC debe ser justificada para efectos de optimizar sobre todo los recursos económicos que en nuestros días son muy importantes y escasos. A continuación se enlistan las ventajas y desventajas que trae consigo el empleo de un PLC.

#### Ventajas

- Control más preciso.
- Mayor rapidez de respuesta.
- Flexibilidad Control de procesos complejos.
- Facilidad de programación.

- Seguridad en el proceso.
- Empleo de poco espacio.
- Fácil instalación.
- Menos consumo de energía.
- Mejor monitoreo del funcionamiento.
- Menor mantenimiento.
- Detección rápida de averías y tiempos muertos.
- Menor tiempo en la elaboración de proyectos.
- Posibilidad de añadir modificaciones sin elevar costos.
- Menor costo de instalación, operación y mantenimiento.
- Posibilidad de gobernar varios actuadores con el mismo autómeta.

#### Desventajas

- Mano de obra especializada.
- Centraliza el proceso.
- Condiciones ambientales apropiadas.
- Mayor costo para controlar tareas muy pequeñas o sencillas.

## 2.4 Servo Posicionamiento industrial

Un servomotor es un actuador giratorio que permite un control preciso de la posición angular, velocidad y aceleración. Se compone de un motor apropiado acoplado a un sensor para la realimentación de posición. También se requiere un controlador relativamente sofisticado, a menudo un módulo dedicado específicamente diseñado para su uso con servomotores.

Servomotores no son una clase diferente de motor, sobre la base de principio de funcionamiento fundamental, pero el uso de servomecanismo para lograr el control de

bucle cerrado con un motor de bucle abierto genérico. Los servomotores se utilizan en aplicaciones tales como la robótica, CNC máquinas o de fabricación automatizada. Como su nombre indica, un servomotor es un servomecanismo, más específicamente, es un bucle cerrado servomecanismo que utiliza realimentación de posición para controlar su movimiento y posición final. La entrada a su control es alguna señal, ya sea analógica o digital, que representa la posición ordenada por el eje de salida.

El motor está emparejado con algún tipo de codificador (encoder, sensores efecto hall, etc) para proporcionar retroalimentación de posición y velocidad. En el caso más simple, sólo se mide la posición. La posición medida de la salida se compara con la posición de comando que es la entrada externa al controlador. Si la posición de salida distinta de la exigida, una señal de error que se genera a continuación, hace que el motor gire en cualquier dirección, según sea necesario para llevar el eje de salida a la posición apropiada. Como el enfoque posiciones, la señal de error se reduce a cero y se detiene el motor.

Los servomotores muy simples utilizan posiciones fijas detectadas a través de un potenciómetro y bang-bang de control de su motor, el motor gira siempre a máxima velocidad (o se detiene). Este tipo de servomotor no es ampliamente utilizado en la industria de control de movimiento, sino que es la base de los sencillos y baratos servos utilizados para los modelos radio-controlados.

Servomotores más sofisticados miden tanto la posición como también la velocidad del eje de salida. También pueden controlar la velocidad de su motor, en lugar de un recorrido a toda velocidad. Estas mejoras, por lo general en combinación con un control PID algoritmo, permita que el servomotor al ser llevado a su posición ordenada, llegué más rápido y con mayor precisión, además de que así tiene menos overshooting.

### 2.4.1 Motor síncrono de imán permanente

Las máquinas de imán permanente son extensivamente usadas en servomotores, accionamiento eléctrico para posicionamiento robótico, máquinas herramienta, ascensores, etc. Se han llegado a construir máquinas de una potencia por encima de 1 MW, por ejemplo para el accionamiento de submarinos. También es posible su aplicación en generación y bombeo a partir de energía solar fotovoltaica o energía eólica.

La construcción de los rotores de los servomotores sincrónicos de imán permanente pueden adoptar una forma cilíndrica con un bajo diámetro y gran longitud (*cylinder rotor*) llamados de flujo radial, o pueden tener un rotor en forma de disco más liviano rotor de disco (*disk rotor*), también llamadas máquinas de flujo axial, resultando así en ambos casos un bajo momento de inercia y una constante de tiempo mecánica baja. Por otra parte, para aplicaciones industriales con arranque de línea o mediante arrancadores de voltaje reducido, los motores poseen un damper que protege los imanes de la des-magnetización durante los transitorios asociados en el arranque, y además amortigua las oscilaciones pendulares.

En aplicaciones en que el motor es operado electrónicamente desde un "invertir", no es necesario el devanado amortiguador para el arranque pues este lo realiza el control electrónico, y además el devanado amortiguador (*damper*) produce pérdidas de energía adicionales debido a las forma de onda no senoidales.

Se analizará el caso de estátor trifásico, el cual es similar a uno de una máquina sincrónica trifásica clásica, debiendo destacarse dos tipos de PMSM según el tipo de rotor:

- Imanes montados en la superficie del rotor (*Surface-mounted magnets*)
- Imanes insertos en el rotor (*Buried Magnets*)

### *a) Imanes montados en la superficie del rotor*

En el caso que los imanes van montados (pegados o zunchados) en la superficie del rotor, estos por el espacio que ocupan obligan a tener un entrehierro relativamente grande, además los imanes cerámicos tienen efectos de saliencia despreciables. En estos casos no existe devanado amortiguador. El gran entrehierro hace que el flujo de la reacción de armadura (RA) tenga efectos atenuados sobre el rotor, es decir la inductancia sincrónica  $L_d$  es pequeña pues tiene una componente de reacción de armadura  $L_{ad}$  pequeña y por consiguiente los efectos de la RA son muy atenuados. Por otra parte se deduce que el gran entrehierro resulta en una constante de tiempo eléctrica del estator  $T = L/R$  pequeña.

### *b) Imanes insertos en el rotor*

Si los imanes están insertos en el rotor, quedan físicamente contenidos y protegidos, pero el espacio de hierro del rotor eliminado para insertar los imanes hace que no puede considerarse que en este caso se tenga un entrehierro uniforme, se tiene un efecto de saliencia, y aparece una componente de reluctancia del par. El criterio de diseño en el caso de servomotores deben encuadrar los siguientes requerimientos:

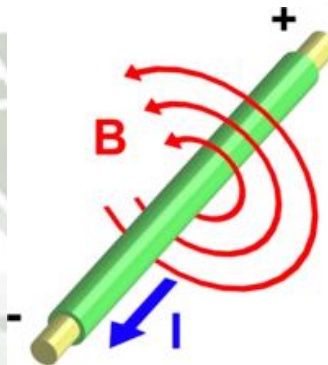
- ✓ Velocidad de operación y par controlado a todas las velocidades
- ✓ Alta relación [Potencia / peso] y [Par / inercia]
- ✓ Par electromagnético suave: sin pares pulsantes debido a las armónicas, ni efectos de posicionamiento preferencial (cogging) debido a las ranuras
- ✓ Alta densidad de flujo en el entrehierro
- ✓ Diseño compacto con alto rendimiento y factor de potencia

## **2.5 Electroimán de solenoide**

El tipo más simple de electroimán es un trozo de alambre enrollado. Una bobina con forma de tubo recto (parecido a un tornillo) se llama solenoide y cuando además se curva de forma que los extremos coincidan se denomina toroide. Pueden producirse

campos magnéticos mucho más fuertes si se sitúa un «núcleo» de material paramagnético o ferromagnético (normalmente hierro dulce o ferrita, aunque también se utiliza el llamado acero eléctrico) dentro de la bobina. El núcleo concentra el campo magnético, que puede entonces ser mucho más fuerte que el de la propia bobina. [11]

Los campos magnéticos generados por bobinas se orientan según la regla de la mano derecha. Si los dedos de la mano derecha se cierran en torno a la dirección del campo magnético  $B$ , el pulgar indica la dirección de la corriente  $I$ . El lado del electroimán del que salen las líneas de campo se define como «polo norte».



**Figura 6:** Flujo de corriente.

*Fuente: Wikipedia, electroimán.*

Dentro de la bobina se crean corrientes inducidas cuando esta está sometida a un flujo variable. Estas corrientes son llamadas corrientes de Foucault y en general son indeseables, puesto que calientan el núcleo y provocan una pérdida de potencia de sí mismo.

### 2.5.1 Electroimán e imán permanente

La principal ventaja de un electroimán sobre un imán permanente es que el campo magnético puede ser rápidamente manipulado en un amplio rango controlando la cantidad de corriente eléctrica. Sin embargo, se necesita una fuente continua de energía eléctrica para mantener el campo.

Cuando una corriente pasa por la bobina, pequeñas regiones magnéticas dentro del material, llamadas dominios magnéticos, se alinean con el campo aplicado, haciendo que la fuerza del campo magnético aumente. Si la corriente se incrementa, todos los dominios terminarán alineándose, condición que se denomina saturación. Cuando el núcleo se satura, un mayor aumento de la corriente sólo provocará un incremento relativamente pequeño del campo magnético. En algunos materiales, algunos dominios pueden realinearse por sí mismos. En este caso, parte del campo magnético original persistirá incluso después de que se retire la corriente, haciendo que el núcleo se comporte como un imán permanente. Este fenómeno, llamado remanencia, se debe a la histéresis del material. Aplicar una corriente alterna decreciente a la bobina, retirar el núcleo y golpearlo o calentarlo por encima de su punto de Curie reorientará los dominios, haciendo que el campo residual se debilite o desaparezca.

En aplicaciones donde no se necesita un campo magnético variable, los imanes permanentes suelen ser superiores. Además, es posible fabricar imanes permanentes que producen campos magnéticos más fuertes que un electroimán de tamaño similar.

### **2.5.2 Funcionamiento**

El material del núcleo del imán (generalmente hierro) se compone de pequeñas regiones llamadas dominios magnéticos que actúan como pequeños imanes. Antes que la corriente en el electroimán este activada, los dominios en el núcleo de hierro están en direcciones al azar, por lo que sus campos magnéticos pequeños se anulan entre sí, el hierro aún no tiene un campo magnético de gran escala. Cuando una corriente pasa a través del alambre envuelto alrededor de la plancha, su campo magnético penetra en el hierro y hace que los dominios giren, alineándose en paralelo al campo magnético, por lo que sus campos magnéticos diminutos se añaden al campo del alambre, creando un campo magnético que se extiende en el espacio alrededor del imán. Cuanto mayor es la corriente que pasa a través de la bobina de alambre, más dominios son alineados, aumentando la intensidad del campo magnético. Finalmente,

todos los dominios estarán alineados, nuevos aumentos en la corriente sólo causan ligeros aumentos en el campo magnético: este fenómeno se denomina saturación. Cuando la corriente en la bobina está desactivada, la mayoría de los dominios pierden la alineación y vuelven a un estado aleatorio y así desaparece el campo. Sin embargo en algunos la alineación persiste, ya que los dominios tienen dificultades para perder su dirección de magnetización, dejando en el núcleo un imán permanente débil. Este fenómeno se llama histéresis y el campo magnético restante se llama magnetismo remanente. La magnetización residual del núcleo se puede eliminar por desmagnetización.

### 2.5.3 Usos

Los electroimanes se usan en muchas situaciones en las que se necesita un campo magnético variable rápida o fácilmente. Muchas de estas aplicaciones implican la defección de haces de partículas cargadas, como en los casos del tubo de rayos catódicos y el espectrómetro de masa.

Los electroimanes son los componentes esenciales de muchos interruptores, siendo usados en los frenos y embragues electromagnéticos de los automóviles. En algunos tranvías, los frenos electromagnéticos se adhieren directamente a los rieles. Se usan electroimanes muy potentes en grúas para levantar pesados bloques de hierro y acero, para separar magnéticamente metales en chatarrerías y centros de reciclaje. Los trenes de levitación magnética usan poderosos electroimanes para flotar sin tocar la pista. Algunos trenes usan fuerzas atractivas, mientras otros emplean fuerzas repulsivas.

Los electroimanes se usan en los motores eléctricos rotatorios para producir un campo magnético rotatorio y en los motores lineales para producir un campo magnético itinerante que impulse la armadura. Aunque la plata es el mejor conductor de la electricidad, el cobre es usado más a menudo debido a su relativo bajo costo, a veces se emplea aluminio para reducir el peso.

# Capítulo 3

## Diseño e implementación

### 3.1 Diseño de sistema

“El diseño ha sido una tarea que se ha venido desarrollando en base a la propia experiencia del diseñador. La necesidad de desarrollar rápidamente nuevos productos ha hecho que se estudie e investigue esta actividad dando como resultado diversas metodologías. Hoy en día el diseño es una técnica que se puede enseñar y aprender.”  
[12]

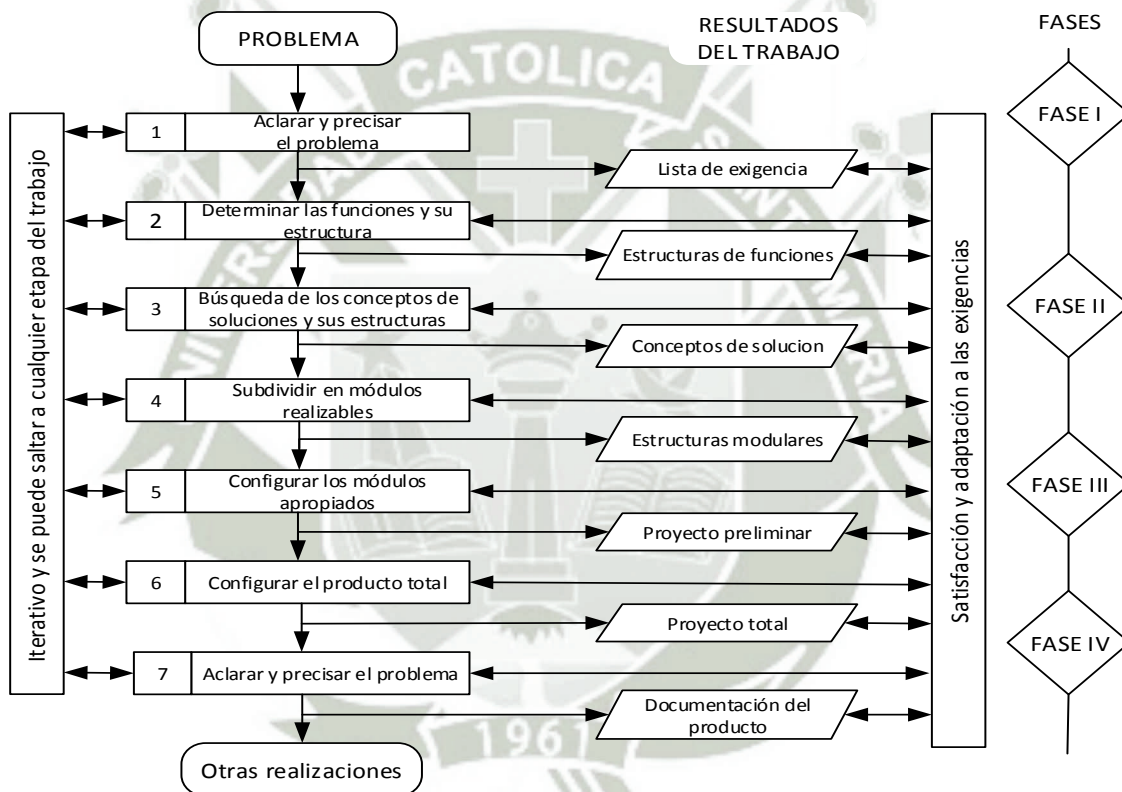
El sistema de clasificación por visión se diseñó teniendo como guía la directriz VDI 2221<sup>6</sup> desarrollada por la Asociación Alemana de Ingeniería. Esta directriz propone un proceso estructurado para el diseño de sistemas y productos técnicos que contempla las siguientes fases:

- **Comprensión de la solicitud:** La primera fase consiste en identificar el problema para que a partir de él se puedan determinar las características del sistema a diseñar, las restricciones y recursos disponibles.
- **Concepto de solución:** Esta fase permite obtener los conceptos de solución que se emplearán y el diseño cualitativo del producto. Asimismo, permite plantear un bosquejo de solución completa, la forma en la que van a trabajar el sistema de visión en conjunto con el sistema de servo posicionamiento industrial y los posibles materiales a utilizar.

---

<sup>6</sup> VDI: Verein Deutscher Ingenieure.

- **Elaboración del proyecto:** Esta tercera fase permite obtener el listado de las piezas requeridas, los cálculos, planos de dichas piezas y los planos de ensamble. Del mismo modo estarán definidos los circuitos electrónicos, los programas, las comunicaciones y la estrategia de control.
- **Elaboración de Detalles:** En esta última fase se elaboran todos los planos necesarios para la fabricación y montaje de la máquina. Así mismo se documentarán los cálculos realizados durante dichos procesos; se programará las tarjetas de control electrónico y se aplicará el algoritmo de control. La interrelación de estas fases se muestra en la figura 7.



**Figura 7:** Proceso de diseño de un proyecto.

*Fuente: Directriz VDI 2221.*

### 3.1.1 Estado del Arte

Como parte del proceso de diseño, es necesario determinar si existe en el mercado soluciones sobre el tipo de sistema que se desea desarrollar, como el objetivo

principal es el poder reconocer patrones de *forma, tamaño y color*, características que se detectan de manera visual, una de las mejores opciones es la de usar cámaras.

Básicamente se ha dividido la investigación sobre cámaras en tres grupos, primero las cámaras semi industriales, estas cámaras cumplen con los estándares de seguridad y protección IP, tienen buenas características que permiten tomarlas como alternativa para desarrollar soluciones industriales, las ventajas es que son de costo relativamente accesibles, pero necesitan de un controlador externo, esto es una ventaja pues da libertad al usuario para utilizar el tipo de programación y controlador que desee, nada más es necesario una adecuada selección.

**Tabla 1:** Cámaras Semi industriales.

Marca	Cámara
	
	

El segundo grupo son las cámaras industriales de software cerrado, los cuales necesitan de un software de la misma empresa para poder realizar la programación, la ventaja de estos equipos es que poseen un controlador incorporado y permiten obtener la señal directamente de la cámara, una desventaja es que son muy costosas. En el mercado existen distintas marcas de cámaras enfocadas a la industria, como las

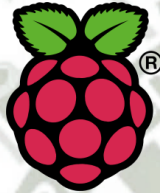



FESTO, Cognex, National Instrument, TOMRA, etc, todas ellas robustas y enfocadas a trabajar bajo condiciones extremas. Estas cámaras se utilizaron en distintas aplicaciones de clasificación, muchas de ellas a alta velocidad, lo cual justifica su precio, pero en muchos otros, no se requiere de tanta velocidad y estas cámaras quedan sobre dimensionadas y no justifican su costo.

Tabla 2: Cámaras Industriales.

Marca	Cámara
	
	
	
	

Finalmente, están el grupo de cámaras más sencillas, enfocadas a educación y proyectos simples, este grupo necesita de un procesador externo, pero tiene como desventaja que presentan una velocidad lenta de FPS (Frames per second), estas cámaras son buenas para poder iniciar con temas de visión artificial y probar algoritmos básicos. El costo de estas cámaras es accesible y se pueden encontrar en el mercado local, la mayoría de ellos se comunican por USB (Universal Serial Bus) y sus controladores son sencillos de instalar.

**Tabla 3:**Cámaras multiuso.

Marca	Cámara
	
	

Internacionalmente se tiene empresas grandes que fabrican soluciones integrales para toda una línea de producción, un claro ejemplo de ella es la empresa Compac Sorting Equipment [13], de Nueva Zelanda, los cuales tienen equipos que realizan una clasificación de paltas por peso en balanzas dinámicas y a volúmenes de producción que superan las 30 TN/hr. Otra empresa grande es TOMRA [14] (Empresa Noruega), trabajan con volúmenes mayores a 60 TN/hr, valor que está muy por encima de los volúmenes actuales de producción que tienen los pequeños y medianos productores de nuestro país.

Tabla 4: Compañías de soluciones de clasificación.

Empresa	Equipo
	
	

A raíz de los equipos encontrados fabricados por las empresas antes mencionadas, se observó mecanismos y sistemas importantes en el funcionamiento, los cuales se intentarán replicar de forma sencilla para diseñar un sistema más accesible económicamente. Aún en estos sistemas, hay funciones realizadas por operarios, como se ve en el siguiente cuadro.

Tabla 5: Funciones realizadas por operarios.

Imagen	Proceso
	<p>Aún en estos equipos, el encajonado del producto sigue siendo un proceso manual.</p>



También se tiene un sistema de visión para control de calidad, donde se pueden desechar los productos que tienen cortes, manchas, fallas, etc.



Los procesos de selección por *tamaño* a veces son necesarios para uniformizar las cajas del producto.

Un punto para resaltar en esta parte, es el hecho de que muchas de las funciones que realizan los operarios manualmente, se basan en su percepción visual, reconociendo forma de los productos, el tamaño que tienen y el color que presentan, todo esto para poder determinar si tienen buena calidad y cumplen con los estándares de producción nacionales establecidos por norma en INDECOPI o estándares internacionales.

Ahora como se conoce el problema a solucionar, se optimizará los procesos de selección. Primero debemos determinar los requisitos mínimos que debería cumplir nuestro sistema, se espera diseñar un sistema económico, eficaz y que realice una clasificación adecuada de cualquier tipo de producto, con un simple entrenamiento de por medio.

En lo referente a servo posicionamiento, existen servomotores Sew, Festo, ABB, Allen Bradley, Siemens, entre otros, los cuales tiene muchas aplicaciones en la industria, en el desarrollo del sistema se utilizó servomotores dependiendo a su disposición en el mercado y costo.

Sobre la programación en Python, se revisaron direcciones web con proyectos realizados utilizando visión artificial y procesamiento de imágenes basado en algoritmos inteligentes. La ventaja de utilizar un software Open Source o un software libre, es que se tiene bastante información disponible en la web a la cual se puede acceder y se cuentan con foros internacionales para resolver dudas en el proceso de ejecución.

También se realizó una investigación de tesis relacionadas al tema mencionado, en universidades importantes del país y artículos internacionales, de los cuales se puede aprender sobre el tema de visión e inteligencia artificial. Respecto el tema de reconocimiento de características por visión, en Colombia se planteó un proyecto basado en visión para determinar el grado de madurez de palta por su color, se han obtenido resultados aceptables en el reconocimiento de patrones como tamaño y color, alcanzado hasta un 87.85% de precisión en los resultados [15], este trabajo nos sirve como punto de partida sobre sistemas de visión, la selección de hardware es un paso importante para tener éxito en el proyecto. Sistemas parecidos se desarrollaron en otro tipos de frutos, en china se realizó una clasificación de manzanos por color [16]. También se desarrollaron en Indonesia sistemas portables de clasificación de naranjas basados en visión por computadora enfocados a la pequeña agro-industria, el sistema clasifica las naranjas por su tamaño y grado de madurez, es un sistema de bajo costo y puede llegar a procesar hasta 2 frutos por segundo [17]. Del mismo modo, en Alemania se utilizó sistemas de visión para ubicar plantas y evaluar la calidad de sus hojas, este es otro ejemplo de una aplicación agro-industrial [18]. También es necesario optimizar tiempo de procesamiento utilizando lenguajes de programación ligeros, según métodos experimentales es posible determinar el tiempo de

procesamiento de las instrucciones del programa [19], estos nos sirvió como punto de partida para estimar el costo computacional requerido y así seleccionar el procesador ideal para nuestro sistema.

Se tiene como antecedentes, trabajos realizados en la Universidad Nacional de Ingeniería sobre sistemas de visión artificial, "Algoritmos de aprendizaje en visión artificial para la captura de objetos en un espacio tridimensional" [20], en el trabajo mencionado se utilizó una cámara para reconocer una ubicación en el espacio, para después enviarla a un robot y este pueda sujetar el objeto de la manera más adecuada. Respecto al tema de Inteligencia Artificial, se desarrolló el tema "Control por aprendizaje Neuro-fuzzy de un sistema servo-hidráulico de alta frecuencia" [21], en esta tesis, se entrenó al sistema servo-hidráulico con datos experimentales, una vez que el sistema fue correctamente entrenado, se optimizó el control como tal. En el sur del país, también se viene desarrollando estos temas, en la Universidad Católica Santa María, se sustentó el tema de "Reconocimiento de patrones" [22], en el cual se utilizó una cámara y el LabView para obtener la imagen e identificar ciertos parámetros. Estos trabajos realizados son para nosotros una referencia para así iniciar con nuestra investigación y desarrollo.

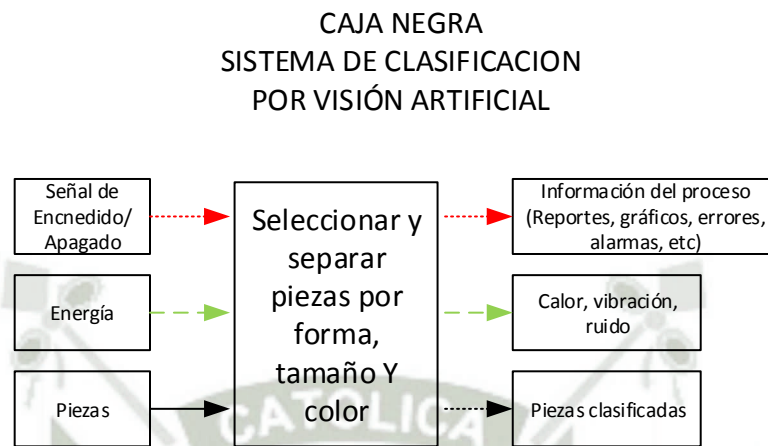
A continuación iniciaremos con el procedimiento de diseño del sistema, nuestro primer paso es el de determinar los requerimientos básicos del sistema que debemos cumplir con el desarrollo.

**Tabla 6:**Lista de Exigencias

Fecha		Deseo o Exigencia	Descripción	Responsables
Proyecto:		Diseño e implementación de un sistema de clasificación por visión artificial basado en redes neuronales y servo posicionamiento lineal industrial		Edición: Rev. 1
Cliente:		UNIVERSIDAD CATOLICA DE SANTA MARIA		Fecha: 26/10/14
				Revisado:
				Elaborado: E.R./M.Z.
Fecha		Deseo o Exigencia	Descripción	Responsables
26-10-14	E	<b>Función Principal</b>	Sistema de selección de piezas geométricas por forma, tamaño y color deseado.	E.R./M.Z.
26-10-14	E	<b>Materia Prima de la máquina:</b>	Se trabajará con materiales accesible en el mercado local, de alta rotación y del menor costo posible.	E.R./M.Z.
26-10-14	E	<b>Fuerza:</b>	Utilizaremos fuerza magnética para la selección de las piezas.	E.R./M.Z.
26-10-14	E	<b>Control:</b>	El control de la selección será por medio de un PLC.	E.R./M.Z.
26-10-14	E	<b>Seguridad:</b>	El diseño del equipo debe asegurar la integridad física de los alumnos al momento de operarlo y ver su funcionamiento.	E.R./M.Z.
26-10-14	E	<b>Montaje:</b>	La instalación, montaje y desmontaje de la máquina serán de la manera más simplificada posible.	E.R./M.Z.
26-10-14	E	<b>Mantenimiento:</b>	La máquina será diseñada de tal manera que se requiera el menor mantenimiento posible, asegurando así la fluidez de operación.	E.R./M.Z.
26-10-14	E	<b>Geometría:</b>	Las dimensiones de la máquina deben ser diseñadas de modo que el producto final sea un equipo compacto en tamaño y que al mismo tiempo sea dimensionalmente accesible para la estatura promedio de los usuarios.	E.R./M.Z.
26-10-14	D	<b>Fabricación:</b>	El diseño debe estar basado en materiales que se puedan encontrar y manufacturar localmente.	E.R./M.Z.
26-10-14	D	<b>Transporte:</b>	Se optará por una fácil movilización del equipo. Las dimensiones de la máquina deben ser tales que esta pueda ser transportada fácilmente en una unidad de transporte mediano.	E.R./M.Z.
26-10-14	E	<b>Energía:</b>	Para garantizar el buen funcionamiento de la máquina se debe contar con una fuente estable de voltaje de 220 VAC.	E.R./M.Z.
26-10-14	E	<b>Ergonomía:</b>	La máquina debe tener las dimensiones adecuadas para brindar comodidad y facilidad de uso.	E.R./M.Z.

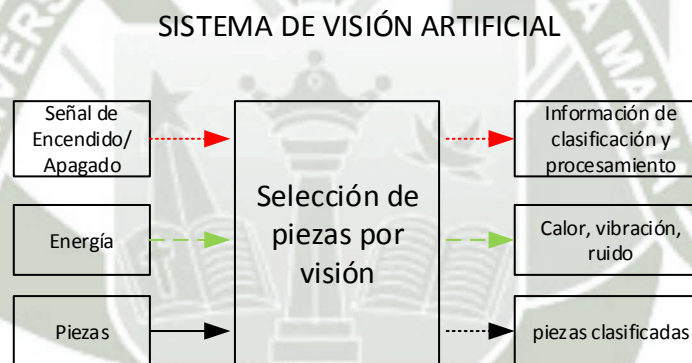
### 3.1.2 Secuencia de operaciones y estructura de funciones

#### a) Abstracción de caja negra



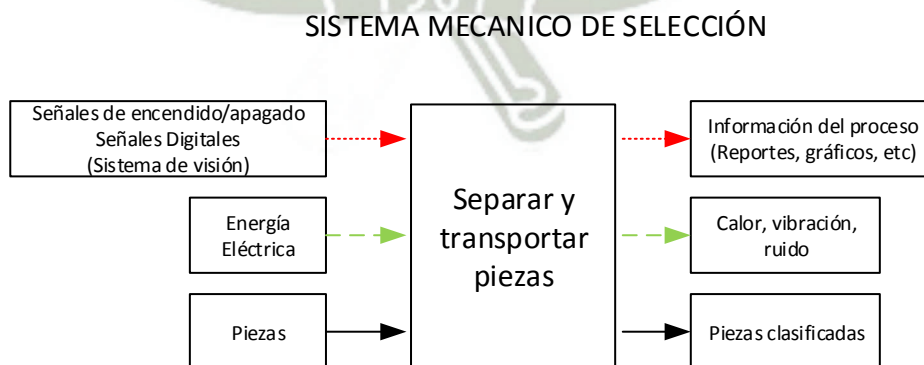
**Figura 8:**Esquema del funcionamiento del sistema de visión.

*Fuente: Elaboración propia.*



**Figura 9:**Esquema del sistema de visión.

*Fuente: Elaboración propia.*



**Figura 10:**Esquema del sistema mecánico.

*Fuente: Elaboración propia.*

### 3.1.3 Descripción de procesos.

#### 1) Preparación:

- a) *Mover* todo el sistema a sus condiciones iniciales para poder iniciar la clasificación.
- b) *Recepcionar* las piezas y ubicarlas sobre una superficie adecuada.

#### 2) Ejecución

- a) *Observar* las piezas para obtener información de ella.
- b) *Analizar* la información obtenida para poder medir los parámetros necesarios. (tamaño, color, forma).
- c) *Clasificar* las piezas por tamaño, forma o color.
- d) *Enviar* información para que se realice la selección de las piezas, en esta etapa se incluye un acondicionamiento de señal y amplificación de la misma si los elementos eléctricos lo necesitan.
- e) *Separar* las piezas basados en la información enviada.
- f) *Descargar* las piezas clasificadas en sus respectivas posiciones, cajetines.

#### 3) Control

- a) *Verificar* que la cámara se encuentre bajo las condiciones iniciales establecidas.
- b) *Encender* la máquina e iniciar el proceso de selección.
- c) *Controlar* el procesamiento y el tiempo de ejecución de la clasificación.
- d) *Controlar* la descarga adecuada de las piezas en su respectivo depósito de selección.
- e) *Controlar* la parada de emergencia en caso de atasco y problemas
- f) *Controlar* el proceso de inicio a fin.

#### 4) Fase final

- a) *Ubicar* los elementos en condiciones iniciales para poder verificar la clasificación.
- b) *Retirar* las piezas de los cajetines, verificando que estos hayan sido clasificados adecuadamente y se pueda volver a iniciar la clasificación.

### 3.1.4 Agrupación de funciones

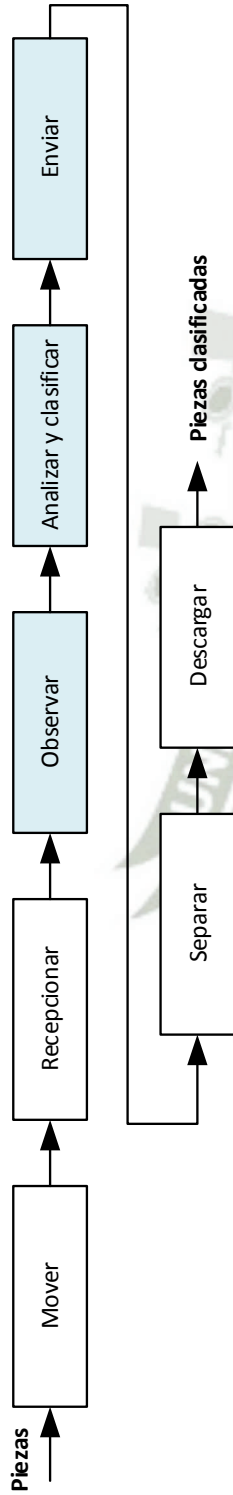


Figura 11: Funciones secuenciales.

Fuente: Elaboración propia.

### 3.1.5 Estructura de funciones

#### ESTRUCTURA DE FUNCIONES SISTEMA DE CLASIFICACION POR VISIÓN ARTIFICIAL

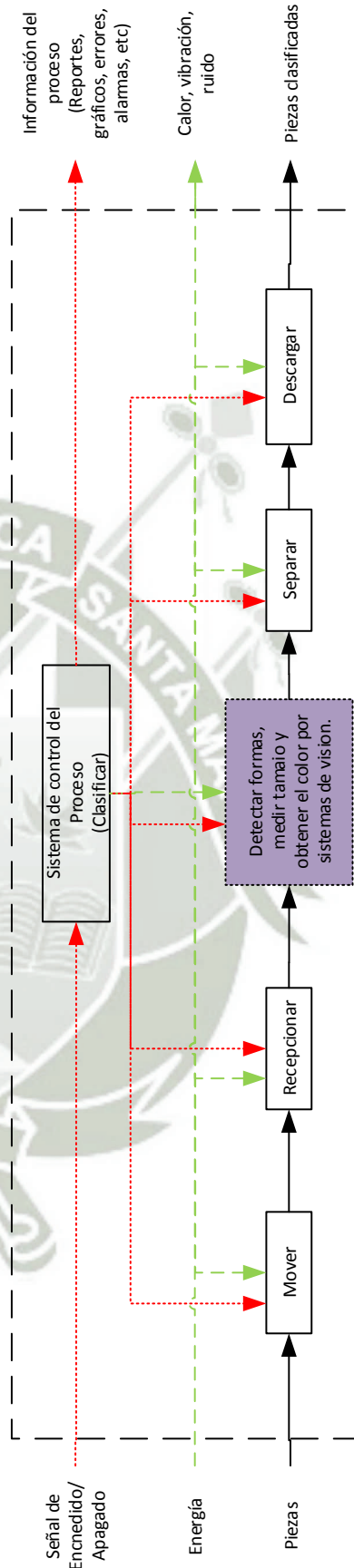


Figura 12: Funcionamiento del sistema.

Fuente: Elaboración propia.

SISTEMA MECANICO DE SELECCIÓN

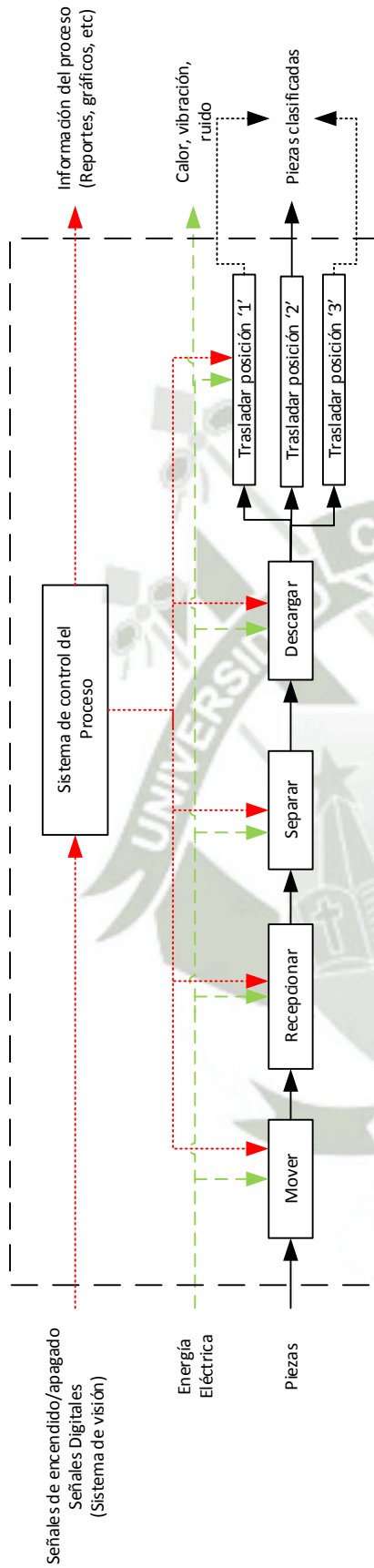


Figura 13: Sistema Mecánico de Selección.

Fuente: *Elaboración propia.*

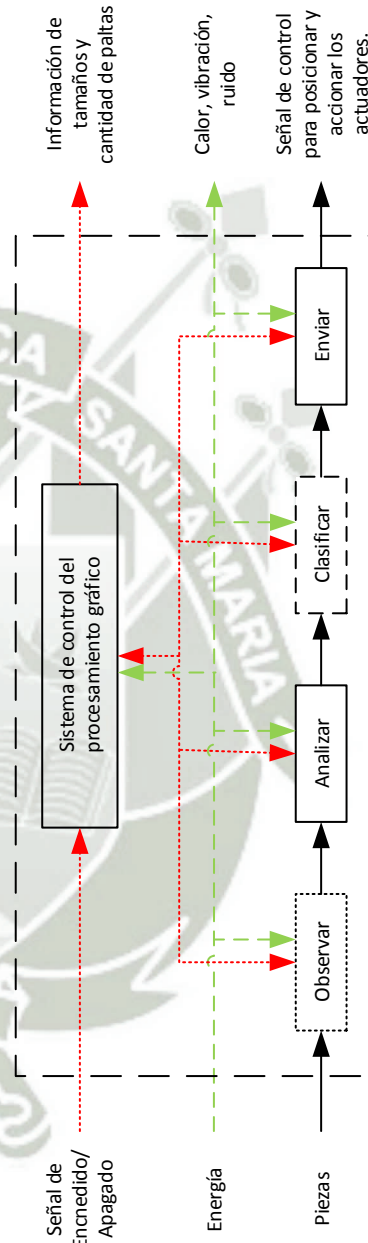

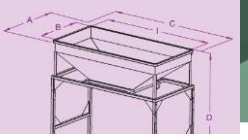



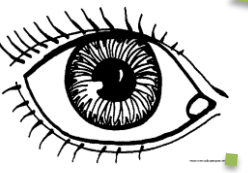
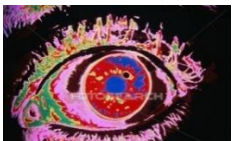









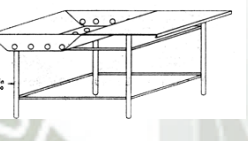



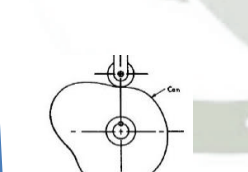


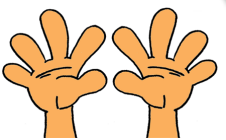









Figura 14: Selección, Sistema de visión

Fuente: *Elaboración propia.*

a) Matriz Morfológica del sistema integrado

Funciones	Tabla 7: Portadores de funciones				
	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4	Alternativa 5
Mover y Recepcionar	 <p>Operarios</p>	 <p>Bandeja de recepción por gravedad</p>	 <p>Canaletas</p>	 <p>Fajas Transportadoras</p>	 <p>Brazos robóticos</p>
Observar		 <p>Infrarrojo</p>	 <p>ULTRASONIDO</p>		
Analizar		 <p>Discernir mecánicamente por tamaño</p>	 <p>CPU</p>	 <p>GPU</p>	 <p>PLC</p>
Clasificar	 <p>Inteligencia humana</p>	 <p>Clasificarlos mecánicamente por tamaño.</p>	 <p>Inteligencia artificial</p>	 <p>Balanza dinámica</p>	
Enviar	 <p>Sistema nervioso humano</p>	 <p>Transmisión mecánica (Engranajes, levas, etc)</p>	 <p>Amplificación electrónica</p>	 <p>Bus de campo</p>	
Separar	 <p>Manualmente</p>	 <p>Servomotores</p>	 <p>Balanza dinámica</p>	 <p>Brazos robóticos</p>	
Descargar	 <p>Manualmente</p>	 <p>Electroimán</p>	 <p>Accionamiento neumático</p>	 <p>Ventosas de vacío</p>	

Solución 1: 

Solución 3: 

Solución 2: 

Solución 4: 

**Tabla 8:** Aspectos posibles de evaluación del concepto de solución.

Aspectos Técnicos	Aspectos Económicos
Uso adecuado de la energía	Número de piezas
Seguridad	Fácil adquisición de materiales
Rapidez	Costo de la tecnología
Estabilidad	Productividad
Confiabilidad	Número de operarios
Manipulación	Facilidad de mantenimiento
Portabilidad	Costo de operación
Grado de contaminación del medio ambiente	
Resistencia a interferencias externas (Robustez)	

Puntaje para calificar los criterios de 0 a 4.

0=No satisface

3=Bien

1=Aceptable a las justas

4=Muy Bien (puntaje reservado

2=Suficiente

para la solución ideal)

**Tabla 9:** Formato de evaluación de concepto de solución

















Nro	Propiedades técnicas y económicas	Soluciones				
		Solución 1	Solución 2	Solución 3	Solución 4	Solución Ideal
1	Función	3	3	3	2	4
2	Precisión	2	3	3	2	4
3	Velocidad	1	4	3	3	4
4	Productividad	1	4	3	3	4
5	Dimensiones	3	2	3	2	4
6	Confiabilidad	2	3	2	2	4
7	Portabilidad	2	2	2	2	4
8	Robustez	1	3	2	3	4
9	Seguridad	1	3	3	2	4
10	Ergonomía	2	3	3	3	4
11	Fabricación	3	2	3	1	4
12	Montaje	3	2	3	1	4
13	Mantenimiento	3	3	2	1	4
14	Costo de tecnología	3	1	4	3	4
15	Costo de operación	1	1	3	3	4
	<b>Suma Total:</b>	31	39	42	33	52

Después de una evaluación técnica económica, concluimos que nuestra solución

óptima es la solución 3.

c) Matriz morfológica en el dominio de Sistema de visión-Control (VDI2221)

Tabla 10: Sistema de visión-control.

<b>Observar</b>	 Cámaras embebedas	 Webcam	 Cámaras semi industriales	 Cámaras industriales
<b>Analizar</b>	 Rapsberry Pi	 Beaglebone	 PcDuino v8	 PLC
<b>Clasificar</b>	 python OpenCV	 OpenCV Microsoft Visual C++	 PyBrain	 ISION SOFTWARE BUSINESS TECHNOLOGY
<b>Enviar</b>	 Transistores y Réles	 SPI	 Serial RS 232	 Ethernet TCP/IP

Solución 1: 

Solución 2: 

Solución 3: 



c) Evaluación de matriz Morfológica en el dominio de Sistema de visión-Control

**Tabla 11:**Tabla de evaluación del sistema de visión.

Nro	Propiedades técnicas y económicas	Soluciones			
		Solución 1	Solución 2	Solución 3	Solución Ideal
1	Función	3	3	3	4
2	Precisión	3	3	3	4
3	Velocidad	1	3	3	4
4	Productividad	1	3	3	4
5	Diseño y forma	2	2	2	4
6	Portabilidad	2	2	2	4
7	Robustez	1	3	3	4
8	Seguridad	3	3	3	4
9	Ergonomía	3	3	3	4
10	Fabricación	3	2	1	4
11	Montaje	3	2	2	4
12	Mantenimiento	3	2	2	4
13	Costo de tecnología	3	1	1	4
14	Costo de operación	3	1	1	4
	<b>Suma Total:</b>	<b>34</b>	33	32	56









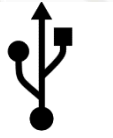




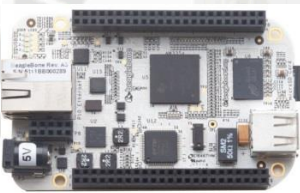











d) Evaluación de matriz Morfológica en el dominio del Sistema electrónico

**Tabla 12:**Evaluación del Sistema electrónico.

Nro	Propiedades técnicas y económicas	Soluciones			
		Solución 1	Solución 2	Solución 3	Solución Ideal
1	Función	3	3	3	4
2	Precisión	2	3	3	4
3	Velocidad	1	3	4	4
4	Productividad	2	3	3	4
5	Diseño y forma	3	2	2	4
6	Portabilidad	2	2	2	4
7	Robustez	1	3	3	4
8	Seguridad	3	3	3	4
9	Ergonomía	3	3	3	4
10	Fabricación	3	2	1	4
11	Montaje	3	1	2	4
12	Mantenimiento	3	2	2	4
13	Costo de tecnología	2	1	1	4
14	Costo de operación	3	1	1	4
	<b>Suma Total:</b>	<b>34</b>	32	33	56

e) Matriz morfológica en el dominio del Sistema electrónico

Tabla 13: Sistema electrónico.

Energizar	 Alimentación 5 VDC	 Fuente 24 VDC			
Observar	 Cámaras embebidas	 Webcams	 Cámaras industriales OpenSources	 Cámaras industriales OpenSources	 Cámaras industriales
Comunicar	 FireWire 800	 USB 2.0	 USB 3.0	 GigE	 TCP/IP
Analizar y Clasificar	 Raspberry Pi	 Beaglebone	 PcDuino v8	 PC	 PLC
Enviar	 VGA	 HDMI	 DeviceNet™	 TCP/IP	 Ethernet TCP/IP
Mostrar	 Pantalla LCD	 Pantalla	 HMI		

Solución 1:   
 Solución 2: 

Solución 3: 

Matriz Morfológica en el dominio del Sistema electrónico y control (Sub funciones de enviar)

Tabla 14: Envío de señales digitales.

Acondicionar señal	Mosfet	Relé de 5 VDC	Relé de 24 VDC	
Activar	Válvulas electro-neumáticas	Actuadores eléctricos	Solenoides electromagnéticos	Electroimán
Conectar/ arrancar	Contactor	Variador de frecuencia	Drivers de potencia	Drivers de control
Iluminar	Iluminación artificial	Iluminación industrial	Iluminación natural	



g) Evaluación de matriz Morfológica en el dominio del Sistema electrónico y control (Sub funciones de enviar)

**Tabla 15:** Evaluación del sistema electrónico para envío de señales.

Nro	Propiedades técnicas y económicas	Soluciones				Solución Ideal
		Solución 1	Solución 2	Solución 3	Solución 3	
1	Función	3	3	3	4	
2	Precisión	3	3	3	4	
3	Velocidad	3	3	3	4	
4	Productividad	3	3	3	4	
5	Diseño y forma	3	3	2	4	
6	Portabilidad	3	2	3	4	
7	Robustez	1	3	2	4	
8	Seguridad	1	3	3	4	
9	Ergonomía	2	2	2	4	
10	Fabricación	2	2	3	4	
11	Montaje	3	2	2	4	
12	Mantenimiento	3	3	2	4	
13	Costo de tecnología	3	1	2	4	
14	Costo de operación	3	2	2	4	
	<b>Suma Total:</b>	<b>36</b>	<b>35</b>	<b>35</b>	<b>56</b>	

h) Determinar el proyecto preliminar óptimo

Tabla 16: Evaluación técnica de las soluciones.

Variantes / Proyectos		EVALUACIÓN TÉCNICA													
		Solución S1			Solución S2			Solución S3			Solución S4			Sol. Ideal	
Nº	Criterios de evaluación	g	p	gp	p	gp	p	gp	p	gp	p	gp	p	gp	
1	Seguridad	2	0	0	2	4	2	4	2	4	3	6	4	8	
2	Aprovechamiento de la energía	2	2	4	3	6	3	6	3	6	3	6	4	8	
3	Control del desplazamiento	1.5	1	1.5	4	6	4	6	3	4.5	3	4.5	4	6	
4	Control de la presión ejercida	1	1	1	4	4	4	4	3	3	3	3	4	4	
5	Rapidez	2	4	8	2	4	2	4	3	6	2	4	4	8	
6	Rigidez	1	4	4	2	2	2	2	2	2	2	2	4	4	
7	Estabilidad	1	3	3	2	2	2	2	2	2	2	2	4	4	
8	Posibilidad de automatización	1	1	1	4	4	2	2	2	2	3	3	4	4	
9	Facilidad de manipulación	2	3	6	2	4	3	6	1	2	1	2	4	8	
10	Productividad	2	3	6	1	2	3	6	2	4	2	4	4	8	
11	Facilidad de montaje	1.5	3	4.5	2	3	2	3	0	0	0	0	4	6	
12	Mantenimiento	1	2	2	1	1	3	3	1	1	1	1	4	4	
<b>Puntaje máximo</b>			27	41	29	42	31	47.5	25	37.5	48	72			
<b>Valor económico Xi</b>			0.6			0.6			0.7			0.5			1.0

Tabla 17: Evaluación económica del as soluciones.

Variantes / Proyectos		EVALUACIÓN ECONÓMICA											
		Solución S1		Solución S2		Solución S3		Solución S4		Sol. Ideal			
Nº	Criterios de evaluación	g	p	gp	p	gp	p	gp	p	gp	p	gp	
1	Facilidad/Costo de transporte	1	3	3	2	2	4	4	0	0	4	4	
2	Costo de materiales	2	2	4	2	4	3	6	1	2	4	8	
3	Costo de fabricación	1	3	3	2	2	3	3	1	1	4	4	
4	Costo de componentes	2	3	6	1	2	3	6	1	2	4	8	
5	Costo de la tecnología	2	2	4	2	4	3	6	1	2	4	8	
6	Costo adquisición de repuestos	0.5	2	1	2	1	2	1	1	0.5	4	2	
7	Consumo energético	1	2	2	2	2	2	2	1	1	4	4	
<b>Puntaje máximo</b>			17	23	13	17	20	28	6	8.5	28	38	
<b>Valor económico Yi</b>			0.6		0.4		0.7		0.2		1.0		

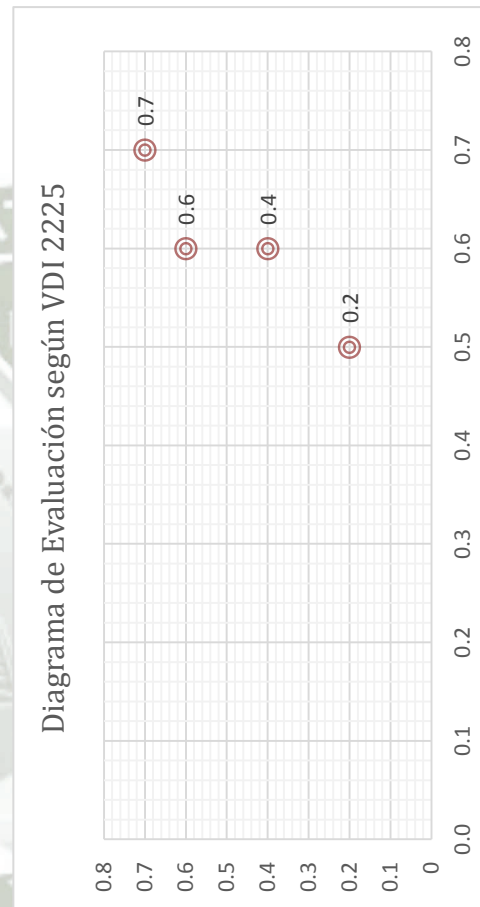


Figura 15: Diagrama de soluciones.

Fuente: Elaboración propia.

### 3.1.6 Selección de Cámara

Después de haber analizado las soluciones posibles, se decidió utilizar una cámara, pero para poder determinar el tipo de cámara, resolución, distancia y velocidad de procesamiento, es necesario determinar las características mínimas de una cámara que cumplirían las exigencias, para lo cual se realizaron pruebas y en base a ellas se estableció los requerimientos mínimos. Primero es necesario determinar el proceso que debe realizar la cámara, que sería el siguiente:

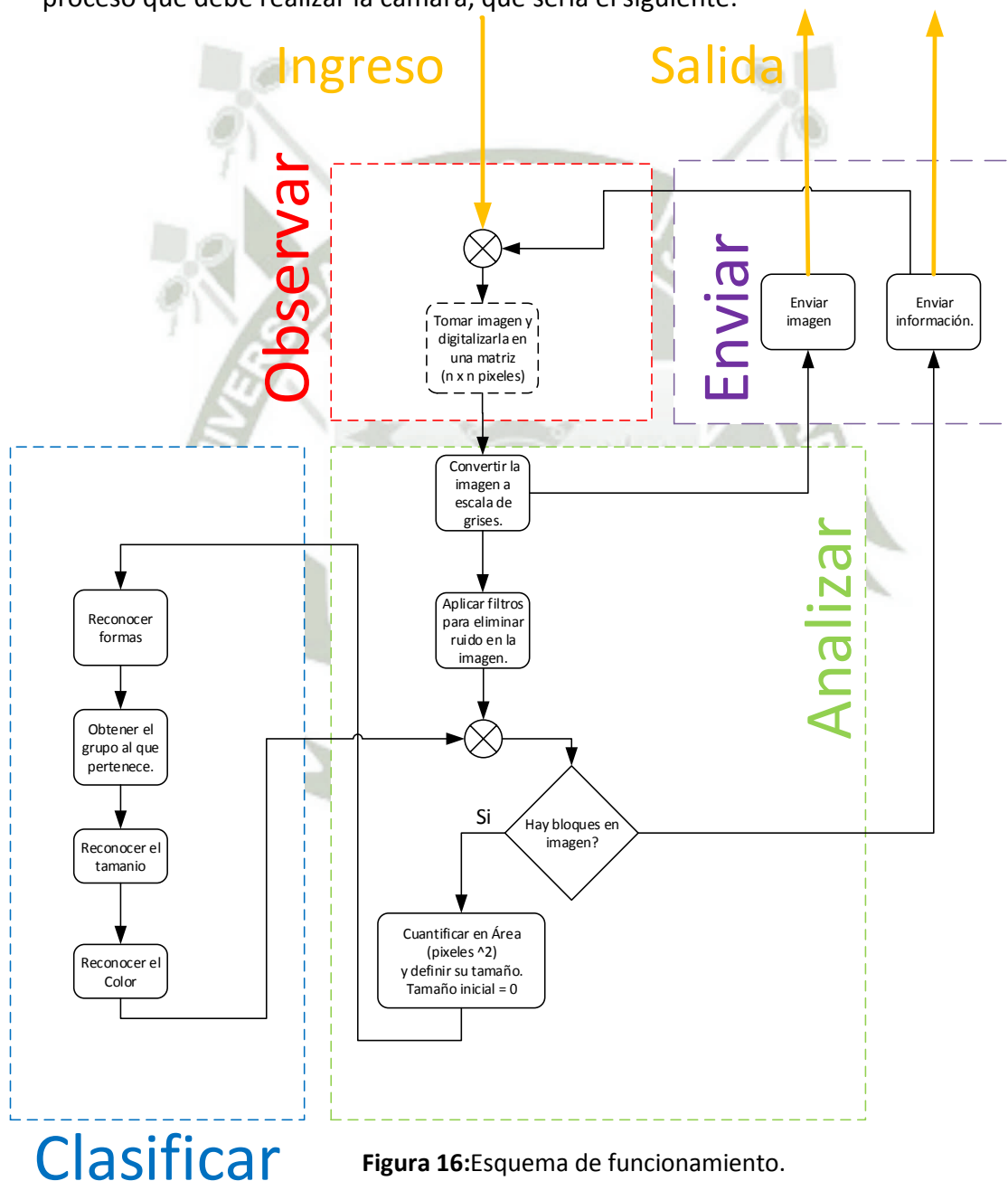
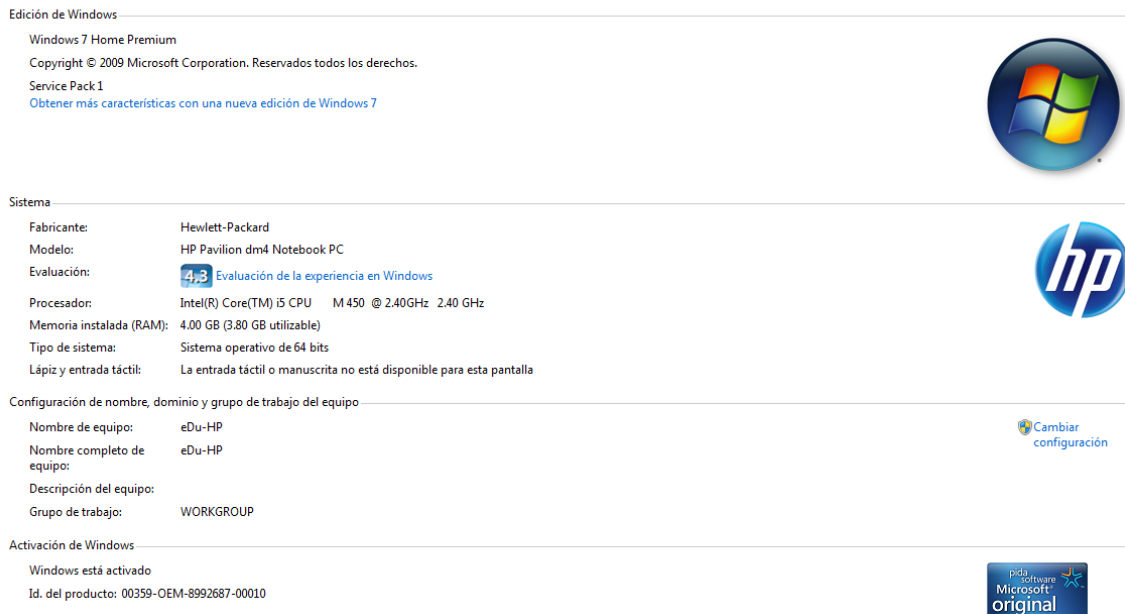


Figura 16: Esquema de funcionamiento.

Fuente: Elaboración propia.

Como ya tenemos determinado el proceso a realizar, se puede hacer un código con una imagen de prueba, para esto se utilizó una computadora con las siguientes características:

[Ver información básica acerca del equipo](#)



Edición de Windows

Windows 7 Home Premium  
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.  
Service Pack 1  
[Obtener más características con una nueva edición de Windows 7](#)

Sistema

Fabricante: Hewlett-Packard  
Modelo: HP Pavilion dm4 Notebook PC  
Evaluación: [Evaluación de la experiencia en Windows](#)  
Procesador: Intel(R) Core(TM) i5 CPU M 450 @ 2.40GHz 2.40 GHz  
Memoria instalada (RAM): 4.00 GB (3.80 GB utilizable)  
Tipo de sistema: Sistema operativo de 64 bits  
Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo: eDu-HP  
Nombre completo de equipo: eDu-HP  
Descripción del equipo:  
Grupo de trabajo: WORKGROUP

Activación de Windows

Windows está activado  
Id. del producto: 00359-OEM-8992687-00010

**Figura 17:**Características de Procesador.

*Fuente: Elaboración Propia.*

El ítem más resaltante de las características anteriores es la siguiente:

**Procesador Intel® Core™ i5 CPU M450 @2.40 GHz**

Pues de ella depende la velocidad de procesamiento de la imagen. Para realizar la evaluación de las imágenes se utilizó el software de libre programación Python® en conjunto con la librería OpenCV, una herramienta para realizar procesamiento de imágenes.



Ambos sistemas son de uso libre, Open Source.

a) Prueba de velocidad de procesamiento

Para esta prueba, se capturó imágenes a una altura de 20 cm aproximadamente, sin iluminación controlada, con esta altura se llega a ver los 22 cm de base donde se encuentran las piezas desordenadas.

$$h = 200 \text{ mm}$$

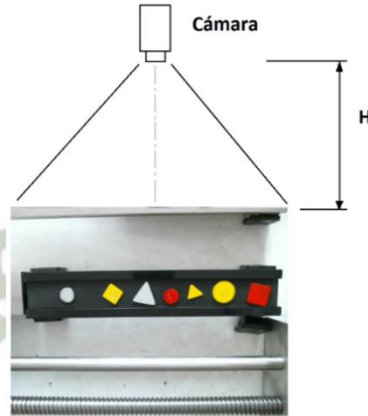


Figura 18: Posible disposición de la cámara.

Fuente: Elaboración propia.

La iluminación, es de aproximadamente de 780 lúmenes y se encontraba a una altura de 2100 mm. Esta iluminación sin control o enfoque en el área de trabajo, puede presentar problemas que los resolveremos más adelante si los encontramos. Para las pruebas se tomaron imágenes de 640 x 480 pixeles, con lo cual a una altura de 20 cm, se tiene una resolución aproximada de 0.05 mm por pixel.



Tabla 18: Imágenes tomadas por la WebCam.

Imagen RGB <sup>7</sup> (640 x 480)	Desenfoque (3) + Umbral Adaptativo

<sup>7</sup> Red, Green, Blue, denominación vectorial de los colores

En las siguientes imágenes, se analizó las piezas y se tomó el tiempo en que demoró ejecutar las instrucciones necesarias.

**Tabla 19:** Evaluación y tiempo del procesamiento.

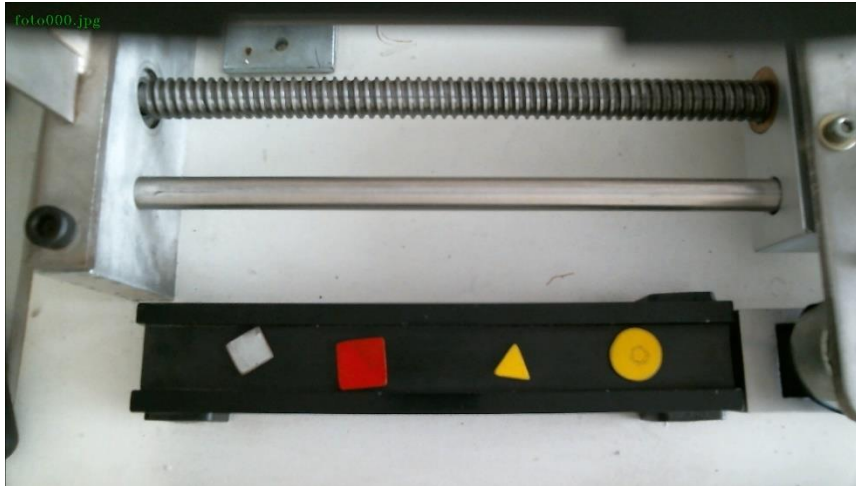
Imagen Procesada	Resultado
	RunTime: 937.58 ms
	RunTime: 1086.50 ms

*Fuente: Elaboración propia*

En el caso del “RunTime”, el tiempo varia, pues depende del uso del procesador en ese momento, muchas veces disminuye cuando se tiene mayor disponibilidad de la memoria RAM.

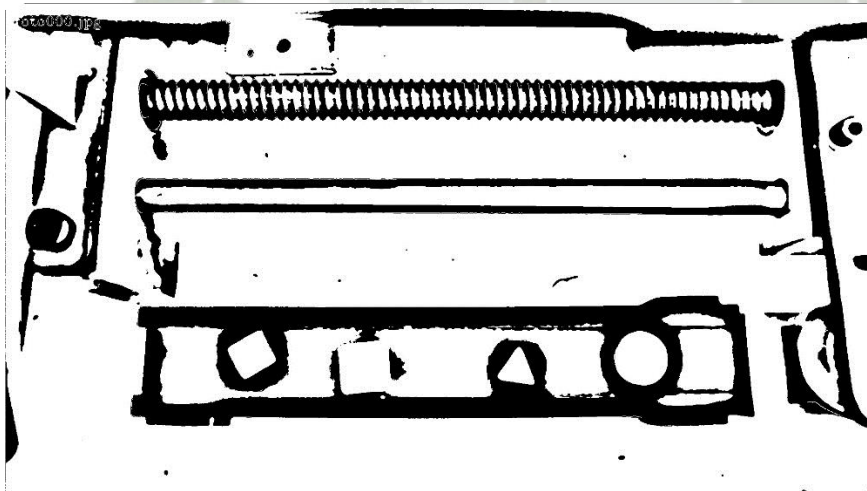
Posteriormente se realizó otra prueba, bajo las mismas condiciones de posición de cámara y de iluminación que la anterior, pero en este caso se tomó una imagen mucho más grande, para aumentar de esta manera la resolución. Para las pruebas se tomaron imágenes de 1920 x 1080 pixeles a una altura de 200 mm.

**Tabla 20:** Evaluación de imagen de 1920 x 1080.



**Imagen RGB**  
(1920x 1080)

**RunTime:**  
1330.96 ms



Desenfoco  
(3) + Umbral  
Adaptativo

*Fuente: Elaboración propia.*

En esta última prueba, se observa que se tiene una mayor resolución, pero esto implica un mayor tiempo de procesamiento, pues se tiene una matriz casi 3 veces más grande.

### b) Cálculo de velocidad de cámara

Como ya se observó, la velocidad mínima en la que el sistema procesa una imagen de 640 x 480 pixeles fue de 937.58 ms, entonces con este valor, calcularemos la velocidad de la cámara.

$$FPS = \frac{1}{937.58 \cdot 10^{-3}} \rightarrow FPS = 1.066 \dots (4)$$

El programa debe ser optimizado, para reducir velocidad de procesamiento, como nuestro sistema es un módulo de pruebas, no existen velocidades que alcanzar, por lo cual con una cámara que logre tomar fotos, es suficiente., pero es necesario una resolución mínima de 640 x 480.

### c) Requerimientos de Cámara

- Velocidad mayor a 1 FPS o superior.
- Tamaño de imagen de 640 x 480 pixeles.
- Conexión con velocidad mayor a 1 Gb/s.
- Campo de visión de 700 mm x 480 mm.
- Lente, solo si la cámara requiere.
- Protección IP30, (opcional)

Revisando las cámaras que se pueden encontrar en el mercado, se decidió utilizar una WebCam Logitech C615, pues cumple con la mayoría de requerimientos, tiene conexión USB y es de un costo accesible.



**Figura 19:**Cámara c615 Logitech

*Fuente: Logitech*

Especificaciones:

- Captura de vídeo Full HD 1080p (hasta 1920 x 1080 píxeles) con el sistema recomendado
- Videoconferencias HD (1280 x 720 píxeles) con el sistema recomendado
- Tecnología Logitech Fluid Crystal™
- Enfoque automático
- Fotos: Hasta 8 megapíxeles (mejora por software)
- Micrófonos integrados con reducción de ruido automática
- Certificación USB 2.0 de alta velocidad (se recomienda)
- Clip universal para monitores LCD, CRT o portátiles

**3.1.7 Selección de procesador gráfico**

Para definir un procesador, se tiene que asegurar que el tiempo de procesamiento sea el mínimo, es por eso que se realizaron pruebas con elementos que se disponen, para poder estimar y predecir que procesador sería capaz de cumplir nuestro requerimiento.

**Tabla 21:**Hardware implementado.

WebCam Logitech C650	Software	HP Pavilion dm4
		
USB 2.0		CPU: Intel i5 2.4 GHz

Se realizó un programa rápido que nos permite detectar formas y colores, el procesamiento fue realizado en tiempo real, contrastándolo con un reloj, a continuación se muestran los resultados.

**Tabla 22:** Resultados de procesamiento.

Fotos	Resultado
	<p>Latency: 31.28 ms Tiempo Real: 5600 ms Retraso en TR: 0 ms</p>
	<p>Latency: 31.36 ms Tiempo Real: 5700 ms Retraso en TR: 100 ms</p>

En esta prueba se observó que la respuesta es bastante rápida, recordar que aún las pruebas son en una Laptop i5 con 2.4 Ghz de frecuencia, como una meta es hacer un sistema “Plug & Play”<sup>8</sup> y utilizar sistemas embebidos, debemos hacer pruebas en uno de ellos y contrastar tiempos.

#### a) *Requerimientos de Procesador*

Los requerimientos mínimos que debería tener nuestro procesador son los siguientes:

- Sistema embebido con procesador mayor a 500 MHz.
- GPU (Procesador Gráfico), para optimizar el procesamiento.
- Conexión HDMI para mostrar en pantalla.
- Conexión mayor a 1 Gb/s.
- Conexión mouse y teclado.

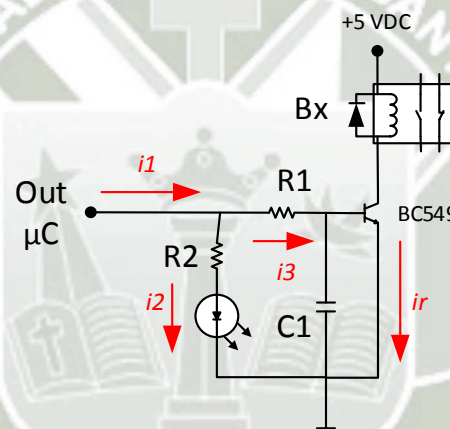
<sup>8</sup> Plug & Play: denominación para sistemas que están listos para utilizar apenas se energice.

- GPIO y buses de comunicación SPI<sup>9</sup>, I2C y serial.
- Energía 5 VDC y 2000 mA.

Con esta información, después de haber observado los requerimientos y considerando las matrices morfológicas explicadas anteriormente, la opción ideal para implementar el sistema es un Raspberry PI, model B, pues cumple con los requerimientos de conexión USB, Ethernet y velocidad de procesamiento. En el punto 3.2.2 se explica a mayor detalle las características del equipo.

### 3.1.8 Amplificación de señal de control

#### a) Amplificación de señal de salida del controlador



**Figura 20:** Análisis de corriente en circuito de salida de señal.

*Fuente: Elaboración propia.*

Para conmutar la bobina del Relé Bx se necesita una corriente aproximada de 50 mA. La ganancia del transistor ( $h_{fe} = 100$ ,  $V_{BE} = 0.7 V$ )

Entonces podemos calcular que corriente se necesita en la base para tener los 50mA de corriente entre el emisor y colector

<sup>9</sup> El **Bus SPI** (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

$$i_B \cdot 100 = i_r \dots (5)$$

$$i_B \cdot 100 = 0.050 A$$

$$i_B = 0.5 mA$$

Como ya conocemos la corriente que necesitamos allí, calculamos el valor  $R_1$  que nos permite alcanzar este valor.

$$\sum V = 0 \dots (6)$$

$$i_3 \cdot R_1 + 0.7 = 5 \quad ; \quad i_3 = i_B$$

$$\therefore R_1 = 8.6 k\Omega \rightarrow R_1 \approx 8 k\Omega$$

La corriente en el led no debe ser mayor a 20 mA, para lo cual calculamos el valor de  $R_2$ .

$$i_2 \leq 20 mA$$

$$\sum V = 0; V_{Led} = 1.9 V$$

$$i_2 \cdot R_2 + 1.9 = 5 \dots (7)$$

$$\therefore R_2 = 132 \Omega \rightarrow R_2 \approx 330 \Omega$$

Finalmente se halla la corriente total del circuito, para asegurarnos que no exceda el máximo posible de 40 mA.

$$i_1 = i_2 + i_3 \dots (8)$$

$$i_1 = \frac{5 - 1.9}{330} + \frac{5 - 0.7}{10000}$$

$$i_1 = 9.82 mA \ll 40 mA \dots \dots Ok$$

El Capacitor C1, solo tiene función de filtro, motivo por el cual se selecciona un valor estándar "22" 22 pF.

Calculando la potencia,  $P = VI \dots (9)$

$$P = 5 \cdot 9.82 \cdot 10^{-3}$$

$$P = 0.0491 W$$

La resistencias pueden tener una potencia de 0.25 W 1/8W.

b) Acondicionamiento de Entradas con Relés de Bobina de 24 VDC

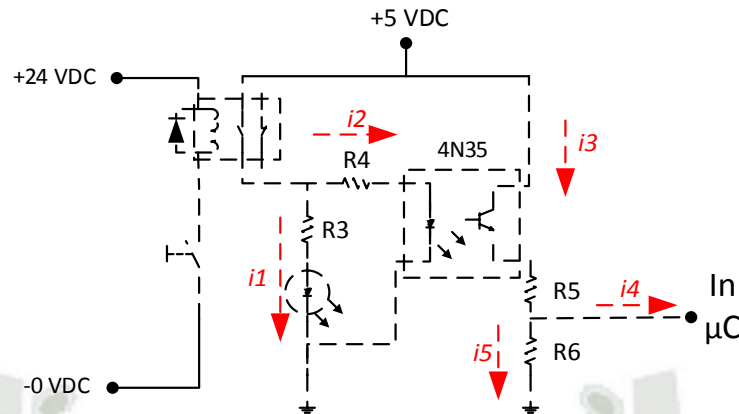


Figura 21: Análisis de corriente en circuito de salida de entrada.

Fuente: Elaboración propia.

La corriente en el led no debe ser mayor a 20 mA.

$$i_1 \leq 20 \text{ mA}$$

$$\sum V = 0; V_{Led} = 1.9 \text{ V}$$

$$i_1 \cdot R_3 + 1.9 = 5 \dots (10)$$

$$\therefore R_3 = 132 \Omega \rightarrow R_3 \approx 330$$

La corriente en el Optocoplador no debe ser mayor a 20 mA.

$$i_2 \leq 20 \text{ mA}$$

$$\sum V = 0; V_{oLed} = 1.5 \text{ V}$$

$$i_2 \cdot R_4 + 1.5 = 5 \dots (11)$$

$$\therefore R_4 = 175 \Omega \rightarrow R_4 \approx 330 \Omega$$

Pero para el óptimo funcionamiento se debe asegurar que se cumpla:

$$i_2 \gg i_1$$

$$\therefore R_3 = 1 \text{ k}\Omega; R_4 = 330 \Omega$$

La corriente de entrada en el Micro Controlador no debe superar los 10 mA.

$$i_{\mu C} < 10 \text{ mA}$$

$$R_6 \gg R_5$$

Cuando el optocoplador está cerrado, la corriente en la  $R_6$ , debe ser mínima:

$$i_5 \approx 0; \therefore i_3 = i_4$$

$$\sum V = 0$$

$$i_3 \cdot R_5 = 5 \dots (12)$$

$$\therefore R_5 = 500 \Omega \rightarrow R_5 \approx 560 \Omega$$

Ahora asignamos un valor de  $R_6$  mayor al de  $R_5$ .

$$R_6 = 20R_5 \dots (13)$$

$$\therefore R_6 = 10000 \Omega \rightarrow R_6 = 10 \text{ k}\Omega$$

Hallando la potencia de los componentes:

$$i_t = i_1 + i_2 + i_3 \dots (14)$$

$$i_t = \frac{5 - 1.9}{1000} + \frac{5 - 1.5}{330} + \frac{5}{560 + 10000}$$

$$i_t = 14.18 \text{ mA} ; P = 0.07 \text{ W}$$

La resistencias pueden tener una potencia de 0.25 W.

### c) Consumo de Energía eléctrica

#### Fuente de +5VDC

- Micro Procesador para el procesamiento de las imágenes: 200 mA x 1
- Consumo de elementos de acondicionamiento de entrada: 14.18 mA x 4
- Activación de relés, 14 entradas de 40 mA cada uno: 40 mA x 4

Nuestra fuente de 5VDC, debe tener un límite de corriente mayor a 254.18 mA.

Así entonces necesitamos una fuente de +5 VDC y 1.3 W o superior.

#### Fuente de +24VDC

- Controlador Lógico Programable VIPA: 1.5 A máx
- Entradas PLC, Relé. 10 mA x 4

Nuestra fuente de +24VDC, la corriente debe de ser mayor a 1.54 A, el valor inmediato superior es 2 A, por lo que en resumen se necesitaría una fuente de +24 VDC y una potencia de 48W.

### 3.1.9 Selección de servomotor

El servomotor a utilizar es uno de tipo síncrono, marca Sew, este motor se eligió por disponibilidad, no presenta características especiales sobre algún otro servomotor, pero se usa como parte del desarrollo del sistema. Las características industriales son similares a las que se pueden encontrar en el mercado en otras marcas.

*Tipo: Servomotor síncrono de imanes permanentes SEW EURODRIVE  
Modelo: CMP40M/KY/AK0H/SM1*

#### Especificaciones Técnicas:

- Velocidad nominal del motor [1/min] : 4500
- Posición de montaje / IM: B5
- Posición de la caja de bornas [°] : 270
- Entrada de cable/ Posición del conector : radial
- Eje de salida [mm] : 9x20
- Diámetro de la brida [mm] : 55x55
- Par de motor [Nm] : 0.80
- Factor de duración : S1-100%
- Tensión del motor [V] : 230
- Esquema de conexionado : DT11
- Corriente estática (I<sub>o</sub>) [A] : 6.00
- Corriente permitida máxima [A] : 9.00
- Corriente nominal [A] : 1.50
- Aislamiento térmico : F
- Tipo protección del motor : IP65
- Peso neto [Kg] : 1.6
- Eje con canal chavetero
- AK0H- Encoder absoluto rotativo con Hiperface
- KTY- Sensor de temperatura
- SM1 - Conector radial

En la imagen se puede apreciar que el motor tiene 2 partes: La primera parte donde se puede ver el eje y la señal que indica superficie caliente; es donde está el rotor y el estator del motor síncrono, en la parte posterior están las conexiones para el encoder y la alimentación del motor, ambas poseen muescas para su correcta instalación.



**Figura 22:**Motor Sew CMP40M.

*Fuente: SEW Eurodrive.*

Datos de placa:

<b>SEW-EURODRIVE</b>		<b>CE</b>
76646 Bruchsal/Germany		
CMP71M/BP/KY/AKOH/SB1		
01.1297864407.0001.09		3~IEC60034
Motor Mo	9.4 Nm Io	10.9
nN	0- 4500 r/min	I max 57.0
U sys	400 V	
Bremse	24= V br 14 Nm	
	IMB5	kg 10.000
1333 930 3.11	Umrichterbetrieb	Made in Germany

**Figura 23:**Características técnicas del motor.

*Fuente: Sew Eurodrive.*



1	Rotor	312	Carcasa del conector
7	Brida de acople	313	Placa de bloqueo
11	Rodamiento de bolas acanalado	316	Enchufe de potencia, cpl.
16	Estator	317	Contacto del enchufe
42	Tapa de rodamiento lado no accionamiento	318	Enchufe de montaje-brida, cpl.
44	Rodamiento de bolas acanalado	321	Enchufe de señal, cpl.
105	Arandela de ajuste	550	Freno, cpl.
106	Retén de aceite	660	Palanca de desbloqueo
304	Cubierta de la carcasa	737	Carcasa del encoder
305	Resolver		

**Figura 25:**La estructura interna del servo SEW:

*Fuente: SEW Eurodrive*

Este Servomotor requiere de un controlador para poder manejar su posición, velocidad de giro, torque, entre otras características, el fabricante recomienda el siguiente controlador para el motor elegido:

*Tipo: Variador de frecuencia SEW EURODRIVE  
Modelo: MC LTX 0008-2B1-4-00 (18251382)*

Especificaciones Técnicas:

Datos de entrada del variador:

- Tensión nominal de la red [Vmains]: 200–240 V ± 10%, 1-fase
- Frecuencia nominal de la red [fmains]: 50/ 60 Hz ± 5%
- Corriente nominal de entrada [IN]: AC 10.5 A

Datos de salida del variador:

- Potencia del motor recomendado [PMot]: 0.75 kW / 1 HP
- Corriente nominal de salida [IN]: AC 4.3 A
- Valor mínimo permitido de resistencia de frenado [RBW\_min]: 27 Ohm

Información general:

- Masa [Kg]: 1.8
- Talla: 2
- Grado de protección: IP20

- Dimensiones (ancho x alto x profundidad): 110 x 220 x 185 mm
- Longitud máxima del cable del motor al variador: 100 m (apantallado), 150 m (no apantallado)

Funcionalidad:

- Modo de control: V/f (escalar), VFC sensorless, Ahorro de energía, Control de motores de imanes permanentes en lazo abierto, Control de servomotores (con módulo servo LTX).
- Control PID integrado.
- Capacidad de sobrecarga (Imax): 150% (60 seg) / 175% (2 seg)
- Frecuencia de modulación PWM [kHz]: 2/4/8/12/16
- Modo de comunicación: SBus.
- Operación en cuatro cuadrantes.
- Transistor chopper de frenado incorporado.
- Filtro de entrada EMC incorporado
- Puesta en marcha desde panel o PC.
- 03 Entradas digitales
- 02 Entradas analógica/digital
- 02 Salidas digital/analógico:
  - Digital: "1 lógico": DC 8... 30 V, "0 lógico": DC 0... 2 V.
  - Analógico: DC 0... 10 V o 0/0... 20 mA/4... 20 mA
- 02 Salida de relé programable: AC 250 V / DC 30 V @ 5A.
- Funciones de protección integradas:
  - Falta y desbalanceo de fase en la alimentación
  - Corto circuito a la salida, fase-fase, fase-tierra
  - Sobre-corriente a la salida
  - Sobre-carga

- Sobre-tensión, baja-tensión
- Sobre-temperatura, baja-temperatura (-10°C...50°C)

**Incluye:**

- Reactancia de línea ND020-013 (8260125).



- Módulo para control de servomotores LTX-H1A (18239226).

**Figura 26:**Variador de frecuencia SEW EURODRIVE

*Fuente: SEW Eurodrive.*

### 3.1.10 Selección de controlador

Para poder seleccionar un controlador adecuado, primero se necesita tener los requerimientos del mismo, los cuales son los siguientes:

Entradas al controlador

- Inicio/Stop (1)
- Pause (1)
- Emergencia (1)
- Limit Switch (2)
- Entradas digitales de comunicación Raspberry Pi (4)
- Encoder (1)
- Entrada Análoga (1)

Se necesitan 9 entradas digitales, entradas rápidas para lectura de encoder, una entrada análoga.

### Salidas del controlador

- Lámpara (1)
- Control de Electroimán (1)
- Salidas digitales de comunicación Raspberry Pi (4)
- Salida Análoga(1)

Se necesita entonces 6 salidas digitales y una salida análoga. En el mercado muchos de los PLC tienen las mismas características, por lo cual es necesario configurar según los requerimientos del sistema. En este sistema implementaremos un PLC VIPA por su disposición en el mercado. La configuración adecuada para el PLC es la siguiente:

- CPU 214-1BC02
- Mod. I/O 221-1BF00 DI 8 x 24VDC
- Mod. I/O 222-1HF00 DO 8 x relé
- Mod. I/O 238-2BC00 AI 4, AO 2 x 12Bits, Fast Counter x 3



Con estas características se satisfacen los requerimientos esenciales.

**Figura 27:**Controlador Lógico Programable VIPA.

*Fuente: VIPA a Yaskawa Company.*

### **3.1.11 Selección de actuador**

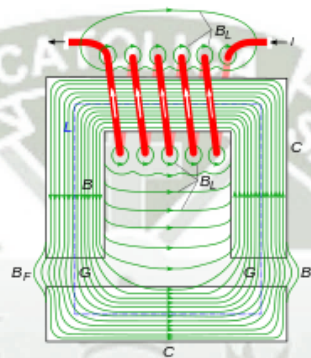
Calcular la fuerza sobre materiales ferromagnéticos es, en general, bastante complejo. Esto se debe a las líneas de campo de contorno y a las complejas geometrías. Este sistema puede simularse usando análisis de elementos finitos. Sin embargo, es posible estimar la fuerza máxima bajo condiciones específicas. Si el campo

magnético está confinado dentro de un material de alta permeabilidad, como es el caso de ciertas aleaciones de acero, la fuerza máxima viene dada por: [23]

$$F = \frac{B^2 A}{2\mu_0} \dots (15)$$

Dónde:

- $F$  es la fuerza en newtons;
- $B$  es el campo magnético en teslas;
- $A$  es el área de las caras de los polos en  $m^2$ ;



- $\mu_0$  es la permeabilidad magnética del espacio libre.

**Figura 28:**Electroimán.

**Figura 29:**wikipedia: electroimán

En el caso del espacio libre (aire),  $\mu_0 = 4\pi \cdot 10^{-7} H \cdot m^{-1}$ , siendo la fuerza por unidad de área (presión):

$$P \approx 398 \text{ kPa} , \text{ para } B = 1 \text{ tesla}$$

$$P \approx 1592 \text{ kPa} , \text{ para } B = 2 \text{ teslas}$$

En un circuito magnético cerrado:

$$B = \frac{\mu NI}{L} \dots (16)$$

Dónde:

- $N$  es el número de vueltas del cable en torno al electroimán;
- $I$  es la corriente en amperios;
- $L$  es la longitud del circuito magnético.

Sustituyendo (16) en (15) se obtiene:

$$F = \frac{\mu N^2 I^2 A}{2L^2} \dots (17)$$

Por su fuerza, los electroimanes se utilizan para levantar contenedores de más de 25 toneladas, más el peso de la carga y vehículos. Para construir un electroimán fuerte, se prefiere un circuito magnético corto con una gran superficie. La mayoría de los materiales ferromagnéticos se saturan sobre 1 a 2 teslas. Esto sucede a una intensidad de campo de  $H \approx 787 A \times \frac{\text{vueltas}}{\text{metro}}$

Por esta razón, no hay motivos para construir un electroimán con una intensidad de campo mayor. Los electroimanes industriales usados para levantar peso se diseñan con las caras de ambos polos en un lado (el inferior). Eso confina las líneas de campo para maximizar el campo magnético. Es como un cilindro dentro de otro. Muchos altavoces usan una geometría parecida, aunque las líneas de campo son radiales al cilindro interior más que perpendicular a la cara.



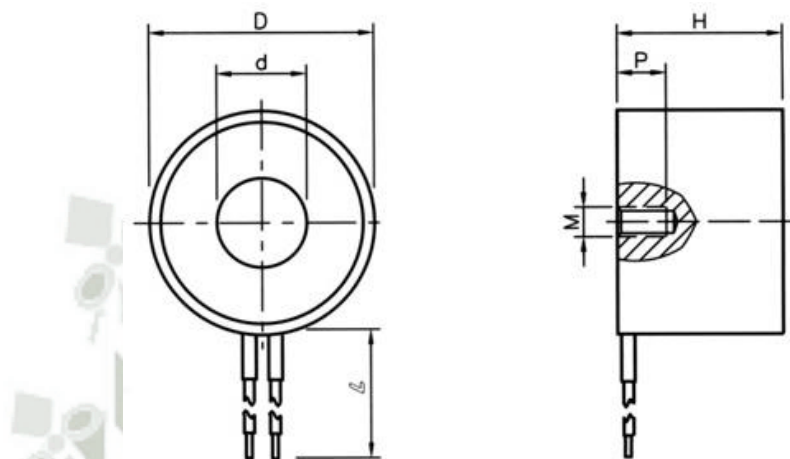
**Figura 30:** Electroimán industrial.

*Fuente: AliExpress.*

Una bobina solenoide en el estado excitado puede producir una fuerza adhesiva fuerte, se puede utilizar en equipos automatizados para detener o mover un objeto. Ampliamente utilizado en las líneas de producción automatizadas y distribución, maquinaria de clasificación, los robots de transporte, equipos de prueba, médico, esmerilado, corte y otros materiales de la línea de producción de

procesamiento automatizado o producto, control sencillo, ahorro de energía, segura y fiable, que puede ser de operación remota.

**Precauciones** para ser sujeta la superficie deben ser lo más lisas posibles y una zona de contacto no menor al área de las caras de extracción de solenoide.



**Figura 31:** Geometría del electroimán.

Tipo	D	D	H	M	P	L	Potencia	Atracción	Peso
	mm	Mm	mm	-	Mm	mm	W	N	g
HCNE 1 -P40	∅ 40	∅ 18	20	M5	8	250	8	250	130

*Fuente: Hojas técnicas de HCNE.*

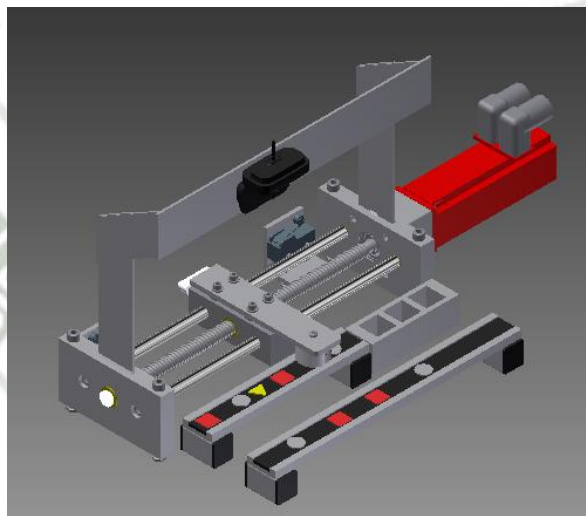
**Tabla 23:** Características del HCNE.

Explicación

1. DC24V100% TD
2. H se pueden personalizar según las necesidades del usuario agujeros tipo de montaje.

### 3.1.12 Diseño y selección de elementos mecánicos

El diseño mecánico del sistema tiene que cumplir básicamente con un requisito, cumplir con las disposiciones geométricas óptimas para el funcionamiento, al ser un módulo de pruebas y simulaciones, las fuerzas que actúan sobre el sistema son mínimas. El Servomotor antes seleccionado se encuentra sobredimensionado para la

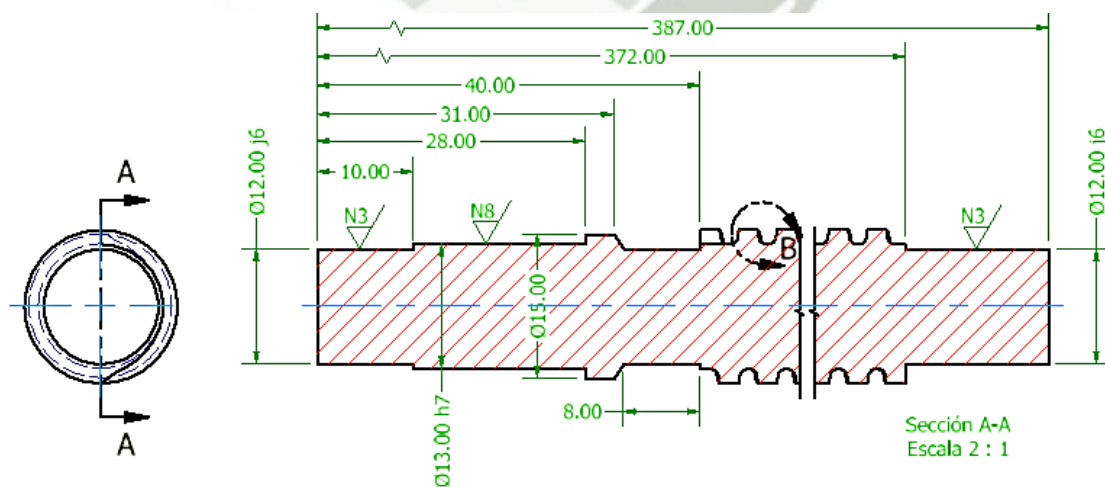


función que cumple en el sistema.

**Figura 32:** Distribución del sistema en vista isométrica.

*Fuente: Elaboración propia.*

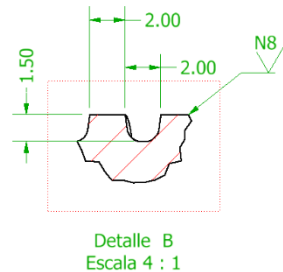
El mecanismo funcional del sistema es el husillo sin fin:



**Figura 33:** Husillo sin fin.

*Fuente: Plano "Husillo sin fin" adjunto en anexos.*

Los planos a detalle se adjuntan como anexos al final de este documento.



**Figura 34:**Detalle de B en Eje husillo.

Fuente: Plano "Husillo sin fin" adjunto en anexos.

### 3.1.13 Listado de elementos y costo de proyecto

**Tabla 24:** Sistema por visión.

Item	Cantidad	Elementos	Precio Unit	Precio
<b>Sistema de Vision</b>				
1	1	Raspberry PI model B	S/. 100.00	S/. 100.00
2	1	Camara WEB 1080 HD	S/. 320.00	S/. 320.00
3	1	Tarjeta SD 8 GB	S/. 74.25	S/. 74.25
4	1	Fuente 5V	S/. 49.22	S/. 49.22
5	1	Cable HDMI	S/. 21.31	S/. 21.31
6	1	Mouse	S/. 25.00	S/. 25.00
7	1	Teclado	S/. 28.45	S/. 28.45
8	1	Pantalla Led de 15"	S/. 400.00	S/. 400.00
9	1	Cable de comunacion de I/O	S/. 15.00	S/. 15.00
<b>Total:</b>				S/. 1,033.23

**Tabla 25:**Tarjeta para señales.

<b>Tarjeta de Acondicionamiento de Señal</b>				
1	8	Resistencia (10k $\Omega$ 1.4W)	S/. 0.10	S/. 0.80
2	8	Resistencia (330 $\Omega$ 1.4W)	S/. 0.10	S/. 0.80
3	4	Capacitor (22 pF)	S/. 0.10	S/. 0.40
4	4	Resistencia (1 k $\Omega$ 1.4W)	S/. 0.20	S/. 0.80
5	4	Resistencia (560 $\Omega$ 1.4W)	S/. 0.20	S/. 0.80
6	4	Optoacoplador 4N35	S/. 2.00	S/. 8.00
7	4	Diodo 1N4001	S/. 0.50	S/. 2.00
8	4	Diodo 1N4002	S/. 0.50	S/. 2.00
9	8	Diodo LED	S/. 0.40	S/. 3.20
10	4	Transistor BC549	S/. 0.70	S/. 2.80
11	1	Pines 2x13	S/. 5.00	S/. 5.00
12	2	Borneras de 5 pines	S/. 4.00	S/. 8.00
13	3	Borneras de 2 pines	S/. 4.00	S/. 12.00
14	8	Socket de 16 pines	S/. 3.00	S/. 24.00
15	1	Fabricacion	S/. 95.00	S/. 95.00
<b>Total:</b>				S/. 165.60

**Tabla 26:**Tarjeta para encoder.

Tarjeta de Acondicionamiento de Encoder				
1	2	Resistencia (1 k $\Omega$ 1.4W)	S/. 0.20	S/. 0.40
2	2	Resistencia (2,2k $\Omega$ 1.4W)	S/. 0.10	S/. 0.20
3	2	Diodo LED	S/. 0.40	S/. 0.80
4	2	Optoacoplador 4N35	S/. 2.00	S/. 4.00
5	2	Socket de 16 pines	S/. 3.00	S/. 6.00
6	2	Borneras de 3 pines	S/. 4.00	S/. 8.00
7	1	Fabricacion	S/. 40.00	S/. 40.00
<b>Total:</b>				S/. 59.40

**Tabla 27:**Elementos de control.

Elementos Industriales de control				
1	1	Fuente de 24 VDC, 1A	S/. 400.00	S/. 400.00
2	2	Fuente de 12 VDC, 2A	S/. 60.00	S/. 120.00
3	1	PLC VIPA CPU - 214 1BC02	S/. 800.00	S/. 800.00
4	1	VIPA 221-1BF00 SM 221 DI 8x24VDC	S/. 600.00	S/. 600.00
5	1	VIPA 222-1HF00 SM 222 DO 8xRelais	S/. 600.00	S/. 600.00
6	1	VIPA 238-2BC00 SM 238C	S/. 800.00	S/. 800.00
7	1	Electroiman de 12 VDC	S/. 80.00	S/. 80.00
8	2	Final de carrera tipo rodillo	S/. 18.00	S/. 36.00
9	1	Pulsador Start/Pause/Emergencia	S/. 150.00	S/. 150.00
10	1	Variador MC LTX - 0.75kW - 220V	S/. 1,769.62	S/. 1,769.62
11	1	Servomotor CMP40M - 4500 RPM - 220V	S/. 4,353.78	S/. 4,353.78
12	1	Cables de fuerza y control (5m c/u)	S/. 958.11	S/. 958.11
13	1	Reactancia de línea ND020-013	S/. 193.64	S/. 193.64
<b>Total:</b>				S/. 10,861.15

**Tabla 28:**Elementos mecánicos.

Elementos mecanicos				
1	1	Eje electrico lineal	S/. 1,500.00	S/. 1,500.00
2	1	Accesorios de montaje mecanico	S/. 400.00	S/. 400.00
3	1	Mecanizado de elementos	S/. 2,000.00	S/. 2,000.00
4	1	Fabricacion de base de modulo	S/. 500.00	S/. 500.00
<b>Total:</b>				S/. 3,900.00

**Tabla 29:**Accesorios para poder terminar todo el cableado.

Accesorios electricos				
1	1	Interruptores automáticos 230 VAC 30 KA In 10 A 2 Polos Protegidos	S/. 76.50	S/. 76.50
2	4	CANAleta RANURA A 33X45 KSS	S/. 18.00	S/. 72.00
3	4	RIEL DIN SIMET 5 2 MTS PERF TELEMEC.	S/. 13.03	S/. 52.12
4	50	GPT CABLE COTR GPT 18AWG. NEGRO INCO	S/. 0.43	S/. 21.50
5	20	GPT CABLE COTR GPT 18AWG. AZUL INCO	S/. 0.43	S/. 8.60
6	20	GPT CABLE COTR GPT 18AWG. ROJO INCO	S/. 0.43	S/. 8.60
7	5	GPT CABLE COTR GPT 18AWG. BLANCO	S/. 0.43	S/. 2.15
8	5	GPT CABLE COTR GPT 18AWG. TIERRA	S/. 0.43	S/. 2.15
9	1	CV-100 AMARRACA LE BLANCO 100X2.5 KSS	S/. 1.00	S/. 1.00
10	1	TERMINAL SOBREM LADO 18 AWG. C/AMARILLO	S/. 3.50	S/. 3.50
11	1	PRESOSTOPA PVC 3/4" KSS	S/. 4.43	S/. 4.43
12	1	CINTA AISLANTE	S/. 3.30	S/. 3.30
13	60	BORNE UNIVERSAL UT 2.5 mm	S/. 1.72	S/. 103.20
14	20	TAPA FINAL PARA BORNE	S/. 1.08	S/. 21.60
15	10	SOPORTE FINAL PARA BORNE	S/. 1.35	S/. 13.50
16	2	PUENTE ENCHUFABLE	S/. 7.26	S/. 14.52
17	20	REGLETAS KLM-A(MARCADOR PARA GRUPO DE BORNE)	S/. 2.70	S/. 54.00
18	10	TIRA ZACK HORIZONTAL (MARCADORES P/BORNE)	S/. 3.29	S/. 32.90
19	1	Cableado electrico	S/. 800.00	S/. 800.00
			<b>Total:</b>	S/. 1,295.57

Tabla 30: Resumen final del costo del proyecto.

Costo de Sistema de clasificacion por visión		
Item	Elemento	Costo
1	Sistema de Vision	S/. 1,033.23
2	Tarjeta de acondicionamiento de senial	S/. 165.60
3	Tarjeta de Acondicionamiento de Encoder	S/. 59.40
4	Elementos Industriales de control	S/. 10,861.15
5	Elementos mecanicos	S/. 3,900.00
6	Accesorios electricos	S/. 1,295.57
<b>Total parcial:</b>		S/. 14,198.26
<b>IGV:</b>		S/. 3,116.69
<b>Total:</b>		S/. 17,314.95

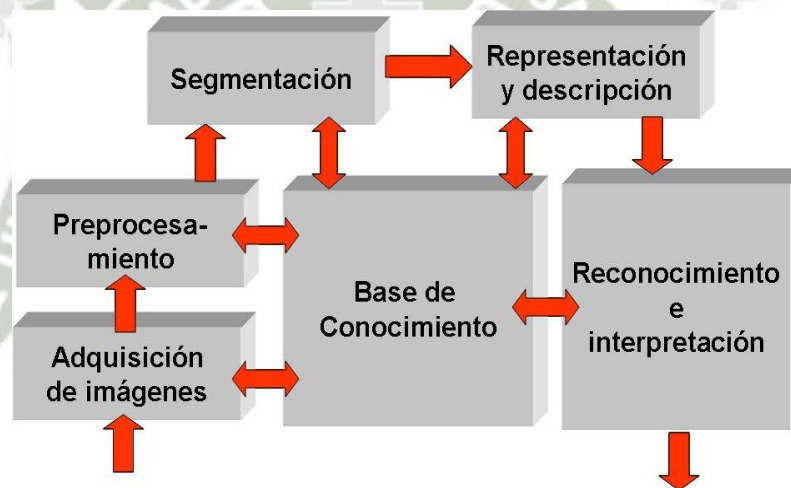
## 3.2 Implementación del sistema

### 3.2.1 Arquitectura de control

Para que el proceso se realice con éxito se utilizó una cámara para obtener la imagen, después procesarla, para este análisis se utiliza el software Python™, que es un software mediante el cual se puede programar en forma práctica aplicaciones con visión artificial y algoritmos inteligentes. En este software se utilizó herramientas de la librería OpenCV, esta es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Open CV es multiplataforma, existiendo versiones

para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica. Entre sus herramientas también se tiene comandos que permiten realizar entrenamiento de redes neuronales, con estos comandos entrenamos el sistema y después se reconoce los patrones basados en esta información.

El control industrial se realizará con un PLC VIPA y un servomotor industrial SEW y un lazo de control cerrado con encoder realizado mediante un Driver, de la misma marca. Después, se enlazarán ambos procesos mediante señales digitales y análogas del proceso de visión hacia el de servo posicionamiento.



**Figura 35:** Etapas de una aplicación

*Fuente: Web, Visión artificial e interacción sin mandos.*

Una vez que el PLC ya tiene la información de la ubicación y el tipo de objetos que se tienen, se realiza el control del servomotor por medio de su controlador para así realizar movimientos de alta precisión. Esta comunicación se realizó utilizando una salida análoga del controlador y algunas salidas digitales.

### 3.2.2 Controlador: Raspberry Pi

Raspberry Pi es un ordenador de placa reducida o (placa única) (SBC) de bajo costo, desarrollado en Reino Unido por la fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. [24]



**Figura 36:** Raspberry Pi.model A.

*Fuente: Raspberry Pi org.*

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos “Turbo” para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MiB de memoria RAM (aunque originalmente al ser lanzado eran 256 MiB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de programación Python, y otros lenguajes como Tiny BASIC, C y Perl.

### *a) Hardware*

Las ventas iniciales fueron del modelo B. El modelo A solo tiene un puerto USB, carece de controlador Ethernet y cuesta menos que el modelo B, el cual tiene dos puertos USB y controlador Ethernet 10/100. A pesar que el Modelo A no tiene un puerto RJ45, se puede conectar a una red usando un adaptador USB-Ethernet suministrado por el usuario. Por otro lado, a ambos modelos se puede conectar un adaptador Wi-Fi por USB, para tener acceso a redes inalámbricas o internet. El sistema cuenta con 356 MiB de memoria RAM en su modelo A, y con 512 MiB de memoria RAM en su modelo B. Como es típico en los ordenadores modernos, se pueden usar teclados y ratones con conexión USB compatible con Raspberry Pi.

El Raspberry Pi no viene con reloj en tiempo real, por lo que el sistema operativo debe usar un servidor de hora en red, o pedir al usuario la hora en el momento de arrancar el ordenador. Sin embargo se podría añadir un reloj en tiempo real (como el DS1307) con una batería mediante el uso de la interface I<sup>2</sup>C.

La aceleración por hardware para la codificación de vídeo (H.264) se hizo disponible el 24 de agosto de 2012, cuando se informó que la licencia permitiría su uso gratuitamente; antes se pensó en anunciarlo cuando se lanzara el módulo de cámara. También se puso a la venta la capacidad para poder usar la codificación-decodificación de MPEG-2 y Microsoft VC-1. Por otro lado se hizo saber que el ordenador soportaría CEC, permitiendo que pudiera ser controlado mediante un mando a distancia de televisión.

**Tabla 31:**Características técnicas del modelo Raspberry.

	<b>Modelo A</b>	<b>Modelo B</b>
SoC:	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB)	
CPU:	ARM 1176JZF-S a 700 MHz (familia ARM11)	
Juego de instrucciones:	RISC de 32 bits	
GPU:	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC3	
Memoria (SDRAM):	256 MiB (compartidos con la GPU)	512 MiB (compartidos con la GPU)
Puertos USB 2.0:	1	2 (vía hub USB integrado)
Entradas de vídeo:	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RPF	
Salidas de vídeo:	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD	
Salidas de audio:	Conector de 3.5 mm, HDMI	
Almacenamiento integrado:	SD / MMC / ranura para SDIO	
Conectividad de red:	5 Ninguna	10/100 Ethernet (RJ-45) via hub USB
Periféricos de bajo nivel:	8 x GPIO, SPI, I <sup>2</sup> C, UART	
Reloj en tiempo real:	Ninguno	
Consumo energético:	500 mA, (2.5 W)	700 mA, (3.5 W)
Fuente de alimentación	5 V vía Micro USB o GPIO header	
Dimensiones:	85.60mm × 53.98mm (3.370 × 2.125 inch)	
Sistemas operativos soportados:	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux. RISC OS	

### *b) Software*

El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.

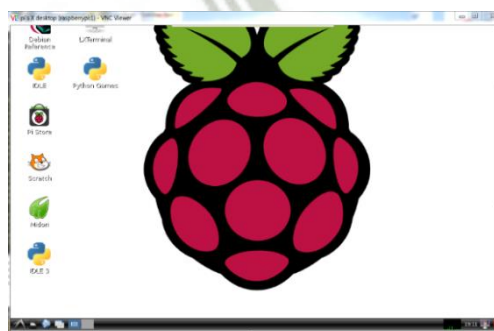
Slackware ARM (también llamada ARMedslack) versión 13.37 y posteriores arranca sin ninguna modificación. Los 128-496 MiB de memoria RAM disponible en la Raspberry Pi, cubren los necesarios 64 MiB de RAM para arrancar esta distribución en sistemas ARM y i386 sin usar interfaz gráfica (el administrador de ventanas Fluxbox que funciona bajo X Window System requiere 48 MiB de memoria RAM adicional).

A la GPU se accede mediante una imagen del firmware de código cerrado, llamado blob binario, que se carga dentro de la GPU al arrancar desde la tarjeta SD. El blob binario está asociado a los drivers Linux que también son de código cerrado. Las aplicaciones hacen llamadas a las librerías de tiempo de ejecución que son de código abierto y estas librerías hacen llamadas a unos drivers de código abierto en el kernel de Linux.

### *c) Configuración de sistema*

Para poder utilizar el sistema Raspberry Pi, el primer paso es darle un sistema operativo sobre el cual se pueda trabajar, se tienen varias opciones:

- RASPBIAN
- PIDORA
- OPENELEC
- RASPBMC
- RISC OS
- ARCH LINUX

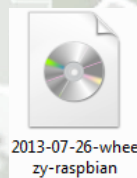


**Figura 37:** Sistema operativo Raspbian.

*Fuente: Elaboración propia.*

También se cuenta con dos sistemas “NOOB” que te permiten seleccionar el sistema al momento de instalarlo, pero es mucho más sencillo. La desventaja de esta última opción, es la cantidad de memoria que ocupa, se recomienda utilizarlo en caso se inicie en programación con un Raspberry Pi. Para realizar el proceso de clasificación, se utilizó el RASPBIAN, por ser el más ligero y tener mayor soporte en la RED. Para instalarlo es necesario de una memoria SD y contar con una PC o Laptop Windows XP o superior. No se realizó una instalación de sistema como tal, sino en cambio se copió una imagen en la memoria SD.

Primero se debe descargar el archivo “Torrent” directamente de la página oficial de Raspberry Pi<sup>10</sup>. Una vez descargado el archivo se ubica en cualquier carpeta, el archivo tiene un tamaño aproximado de de 1.8 GB y tiene la extensión *.img*. [13]



**Figura 38:** Archivo de imagen del sistema.

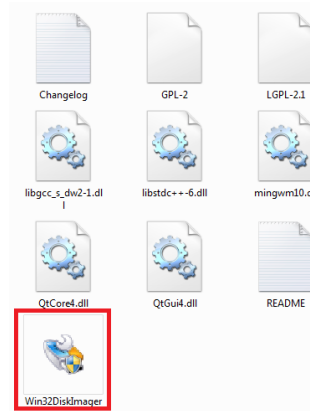
*Fuente: Elaboración propia.*

Posteriormente se busca el programa “Win32 Disk Imager”<sup>11</sup>, una aplicación que nos ayudará para copiar la imagen de Raspbian en la memoria SD. Abrimos el archivo descargado y extraemos su contenido en cualquier ubicación. El archivo para ejecutar es el Win32DiskImager.

---

<sup>10</sup> <http://www.raspberrypi.org/downloads/>

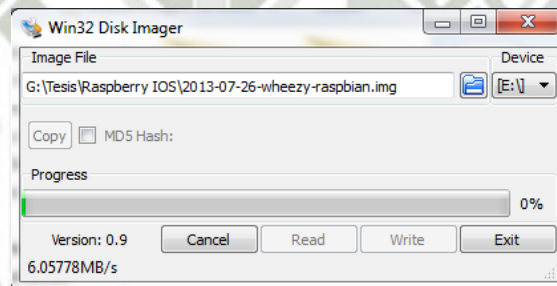
<sup>11</sup> <http://sourceforge.net/projects/win32diskimager/files/latest/download>



**Figura 39:** Carpeta de Win32DiskImager.

*Fuente: Elaboración propia.*

Se introduce la tarjeta SD en el Slot del ordenador, se busca la imagen que se descargó y damos clic en “Write”, la unidad de destino se selecciona automáticamente.



**Figura 40:** Venta del programa Win32.

*Fuente: Elaboración propia.*

Cuando ya se tiene la imagen en la Memoria SD, se introduce en el Rapsberry Pi y se inicia el sistema.



**Figura 41:** Parte inferior de Raspberry Pi.

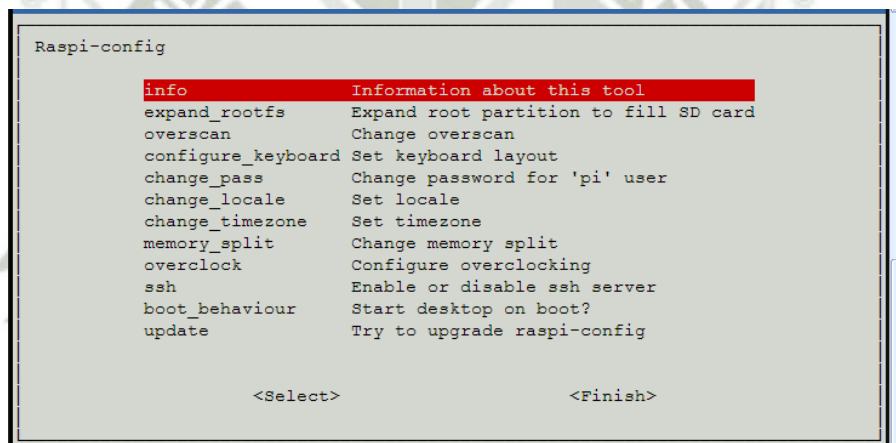
*Fuente: Elaboración propia.*



**Figura 42:** Pantalla de Inicio del Raspberry en primera conexión.

*Fuente: Elaboración propia.*

Una vez iniciado el sistema por primera vez debería iniciar con una pantalla azul similar a la de BIOS como en la imagen.



**Figura 43:** Ventana de inicio.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

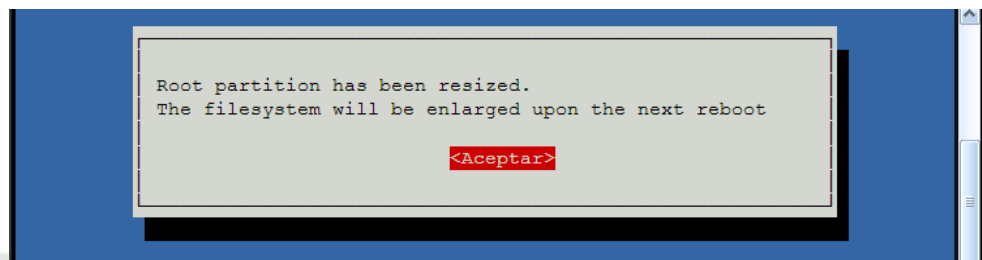
En caso no se abra este menú, se puede ingresar a él escribiendo:

```
Sudo raspi-config
```

Ahora explicaremos las opciones relevantes y las cuales son necesarias configurar para iniciar el sistema:

### 1. “expand\_rootfs – Expand root partition to fill SD card”

La tarjeta SD después del arranque tendrá dos particiones: una de boot que hace las veces de BIOS y otra, donde está instalado Raspbian y que solo es visible en un PC con Linux. Como por defecto se utiliza una imagen para instalar el Raspbian es de 2 GB, si se utilizó una SD más grande el resto del espacio estará sin utilizar. Al utilizar esta opción, se consigue que Raspbian aproveché el espacio libre de la tarjeta de manera eficiente

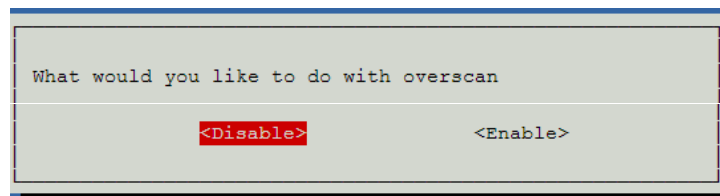


**Figura 44:**Partición del sistema.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

### 2. “overscan – Change overscan”

Si la imagen que se observa en la pantalla no ocupa completamente el monitor, podemos intentar desactivando esta opción y así ver si se soluciona el problema.



**Figura 45:** Opción “overscan”.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

### 3. “configure\_keyboard – Set keyboard layout”

Para poder utilizar el sistema, es necesario configurar el teclado, para lo cual hacemos lo siguiente:

1. El tipo de teclado, lo más usual es “Generic 105-key (Intl) PC”
2. Después el idioma en el que se utilizará el teclado. Por defecto está configurado en inglés británico. Seleccionamos “Other” y buscamos la opción “Spanish”.

3. Ahora se debe selección el idioma en el que está el teclado. Esto básicamente se diferencia si e teclado tiene Ñ o no. Lo usual es tener el teclado también en “Spanish”.

4. En seguida, la ubicación de la tecla “Alt”. Lo normal es escoger “The default for the keyboard layout”.

5. Entre las últimas opciones, pedirá la tecla para componer “AltGr” (compose key). Igualmente lo normal suele ser usar “No compose key”.

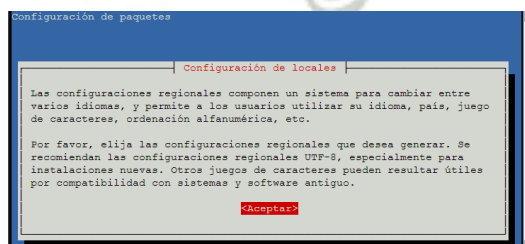
6. Por último hay que indicar si deseamos usar la combinación de teclas “Ctrl + Alt + Borrar” para salir del modo de interfaz gráfica. Por defecto, la opción es NO.

#### 4. “change\_pass – Change password for ‘pi’ user”

Para cambiar la contraseña de fábrica. Por defecto es: user ‘pi’ y pass ‘raspberry’. Por seguridad es muy recomendable cambiarla.

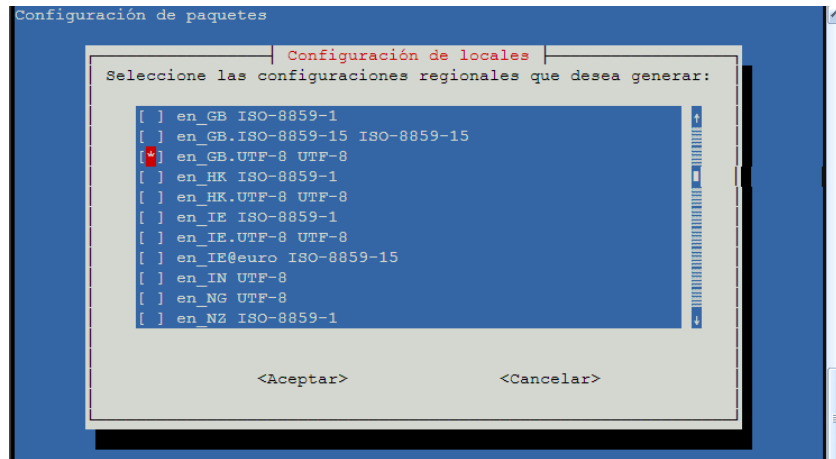
#### 5. “change\_locale – Set locale”

En esta opción se configura el idioma para el sistema operativo, además del modo es el cual se mostraran los números, moneda y fechas de acuerdo a lo estándar en ese idioma. Las ayudas y aplicaciones también se cargaran en ese idioma. Por defecto se tiene la opción “en\_GB”, se deja marcado y después se agrega el español, la opción se busca moviendo los cursores (las flechas del teclado) y cuando se encuentra la opción, pulsas la tecla “espacio”. Si eliges “es\_ES” se refiere a español de España o según donde te encuentras elige es\_CO o lo que convenga. Y luego el mapa de códigos recomendado que es UTF-8. Una vez que se eligió, pulsamos la tecla “Enter”. Y lo siguiente será elegir cuál de los instalados es el principal. En Perú se marca “es\_PE”, lo seleccionamos y luego seleccionamos aceptar.



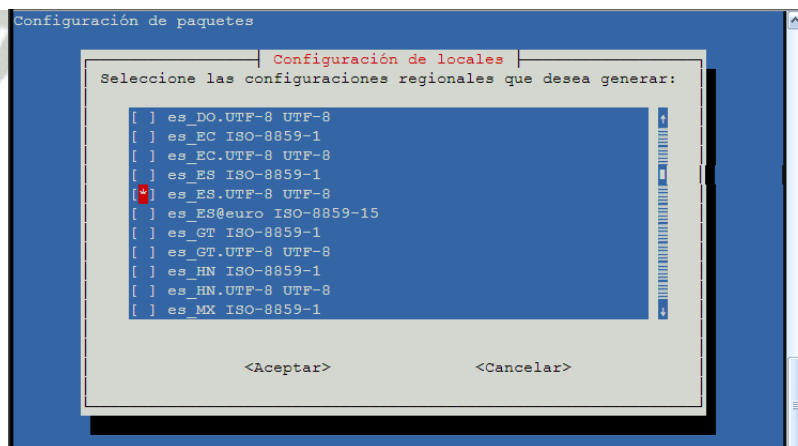
**Figura 46:**Configuraciones locales.

*Fuente: Raspberry para torpes, pero para torpes torpes.*



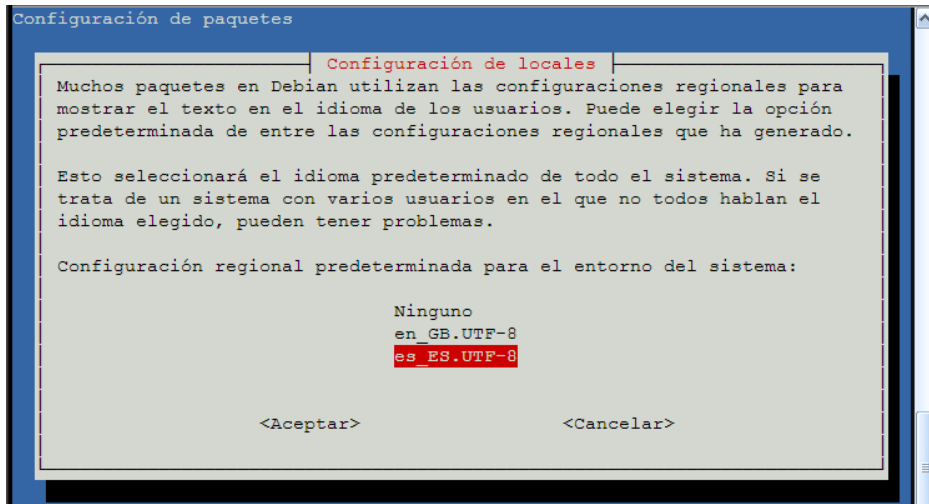
**Figura 47:**Configuración de idiomas.

*Fuente: Raspberry para torpes, pero para torpes torpes.*



**Figura 48:**Seleccionar el idioma español.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

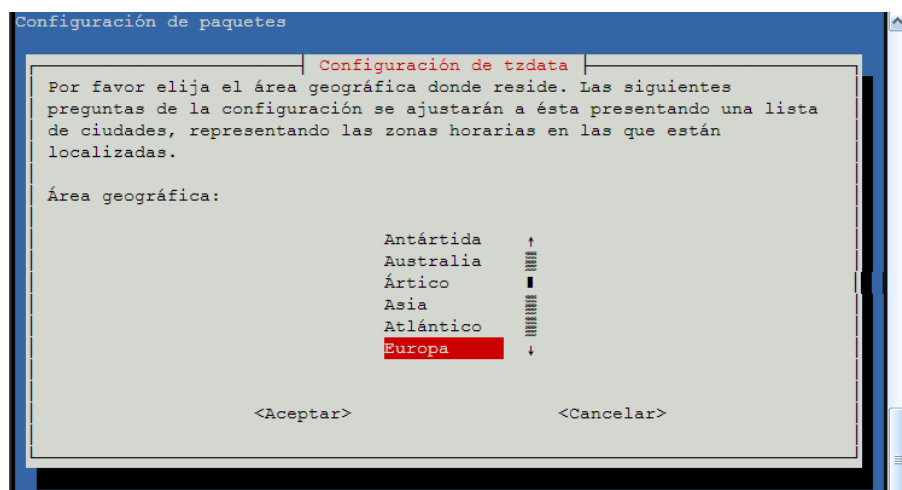


**Figura 49:** Idiomas agregados al sistema.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

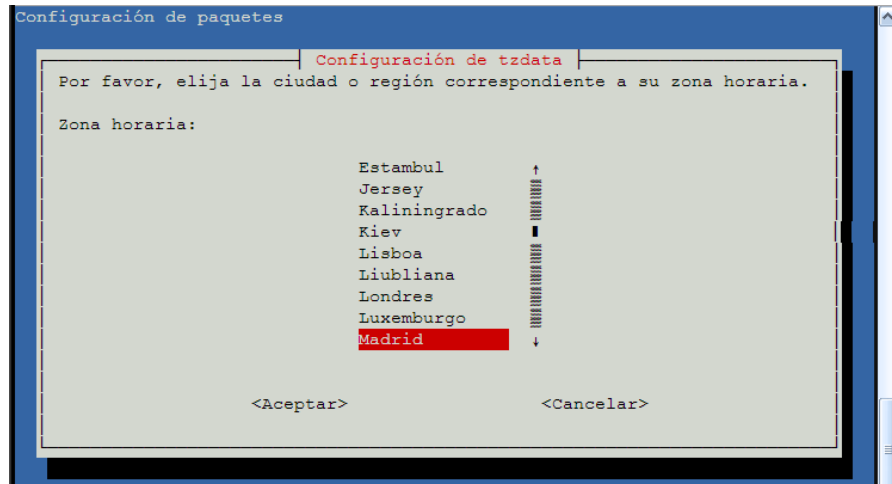
## 6. “change timezone – Set timezone”

Esta opción es para determinar la hora del Raspberry Pi. Primero se elige la opción “América” y después se busca la ciudad “Lima”. Considerar que como el Raspberry no tiene una batería ni un reloj interno RTC, esta hora se usa como referencia para añadir o quitar horas sobre la hora estándar que el Raspberry recopila del internet al conectarse, sin conexión no se tiene hora en tiempo real.



**Figura 50:** Ventana de selección de ubicación.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

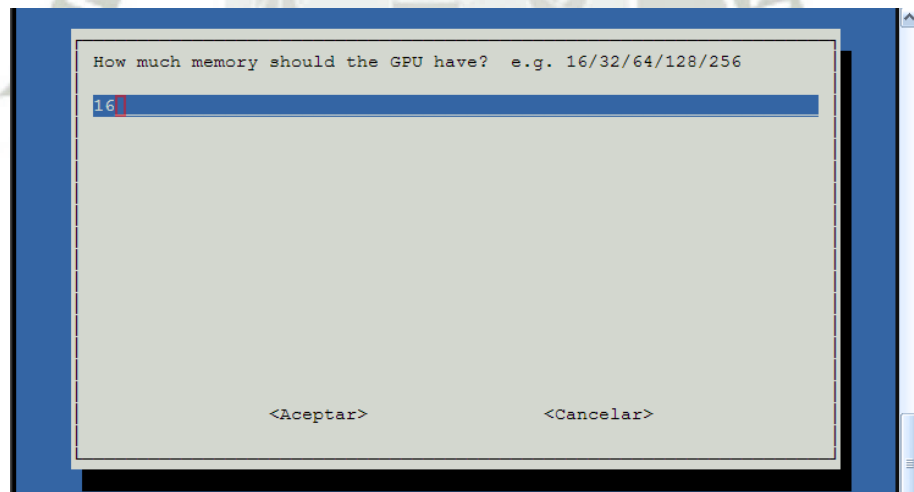


**Figura 51:**Selección de zona horario.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

### 7. “memory\_split – Change memory split”

En este punto, se elige la cantidad de memoria para la tarjeta gráfica y cual será utilizada por la memoria principal. En caso de usarlo como servidor es mejor a opción 16. En nuestro caso estará conectado a una pantalla y la opción más adecuada es de 128.



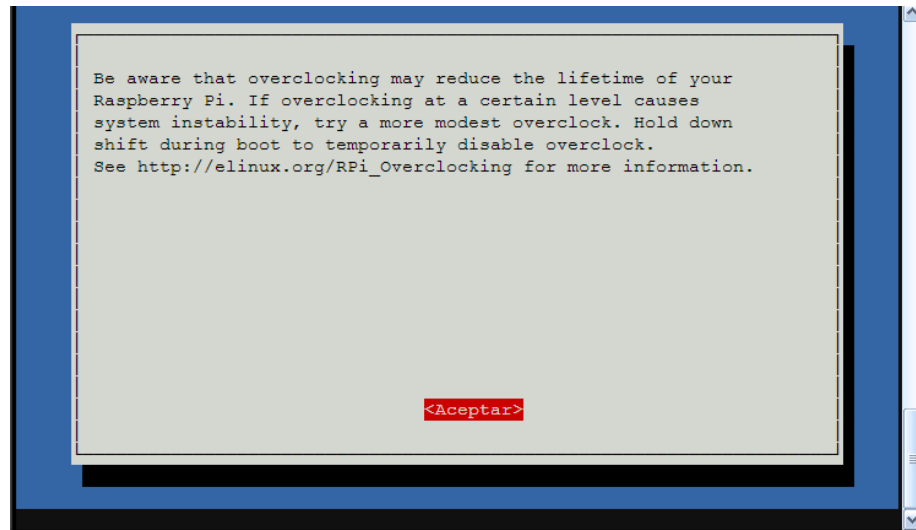
**Figura 52:**Selección de memoria GPU.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

### 8. “overclock – Configure overclocking”

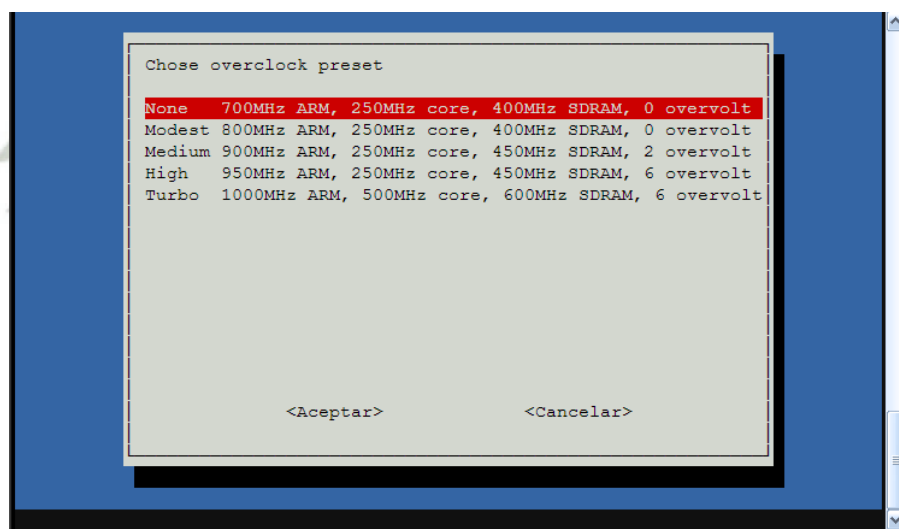
Aquí se puede escoger entre varias opciones de configuración para aumentar la velocidad del procesador, memoria, SD, etc. Su capacidad varía mucho de 700MHz a 1GHZ, pero es cuestión de suerte cómo sea de estable, lamentablemente es cuestión

de prueba y error. En caso de fallo, considerar que no se pierde la garantía, pero utilizando en “Overclock”, disminuye su vida a largo plazo.



**Figura 53:** Ventana de advertencia del Overclocking.

*Fuente: Raspberry para torpes, pero para torpes torpes.*



**Figura 54:** Selección de Frecuencia del sistema.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

### 9. “boot\_behaviour – Start desktop on boot?”

Al seleccionar la opción “Yes”, el sistema iniciará el escritorio tipo Windows directamente al arrancar la Raspberry. Si se selección “No”, arrancará en modo

Terminal y luego de forma manual con el comando `startx`, se inicia el sistema cuando se desee.

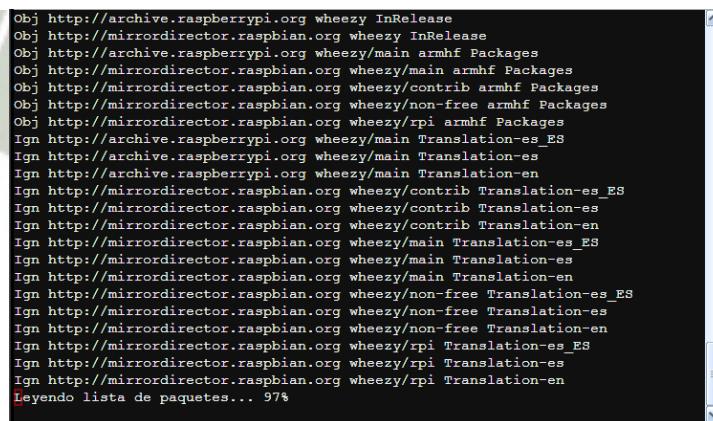


**Figura 55:** Ventana para iniciar el sistema.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

## 10. “update – Try to upgrade raspbi-config”

Finalmente, se le da click en la opción de actualizar el sistema operativo Raspbian, estar seguro de tener la conexión a internet por cable o USB-Wifi.



**Figura 56:** Iniciación del sistema por primera vez.

*Fuente: Raspberry para torpes, pero para torpes torpes.*

Para seguir con la configuración se dirige a “Finish”. Termina la configuración de Raspbian y se guardan los cambios realizados. Solo resta reiniciar para confirmar que todo está conforme. Una vez que se tiene instalado el sistema operativo, el sistema inicia automáticamente en modo Terminal, pero como deseamos ingresar

directamente a la ventana de inicio similar a Windows, para lo cual se hace lo siguiente, ingresamos el siguiente comando en el terminal:

```
sudo nano /root/.bash_profile
```

y allí se escribe:

```
startx
```

Se presiona “ctrl+o”,”enter”,”ctrl+x” y ya está listo, la siguiente vez que se reinicie el sistema, iniciará de frente en el escritorio. Antes de actualizar el sistema, se cambian los repositorios que ya vienen instalados:

```
sudo nano /etc/apt/sources.list
```

Agregar:

```
deb http://archive.raspbian.org/raspbian wheezy main contrib non-free
deb-src http://archive.raspbian.org/raspbian wheezy main contrib non-free
```

Ahora se procede a actualizar el sistema y también instalar los softwares necesarios:

Actualizar Firmware	sudo rpi-update
Actualizaciones	sudo apt-get install update
Busca de mejoras en el Sistema	sudo apt-get install upgrade

#### d) Instalar OpenCV

Para instalar el openCV se utilizan los siguientes comandos:

```
sudo apt-get install libopencv-dev python-opencv
```

El Python 2.7 ya viene instalado por defecto en el sistema operativo Raspbian.

### e) Instalar cámara

Es necesario instalar los drivers [26] necesarios para poder utilizar la WebCam y se hace con los siguientes comandos:

```
git clone
git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-
2.6.git uvcvideo
cd uvcvideo
git remote add uvcvideo git://linuxtv.org/pinchartl/uvcvideo.git
git remote update
git checkout -b uvcvideo-master uvcvideo/uvcvideo-next
```

### f) Instalar SPI

Como se utilizará la comunicación SPI (*Interface Serial Periférica*), tenemos que activar este periférico y lo se hace de la siguiente manera:

```
sudo raspi-config
```

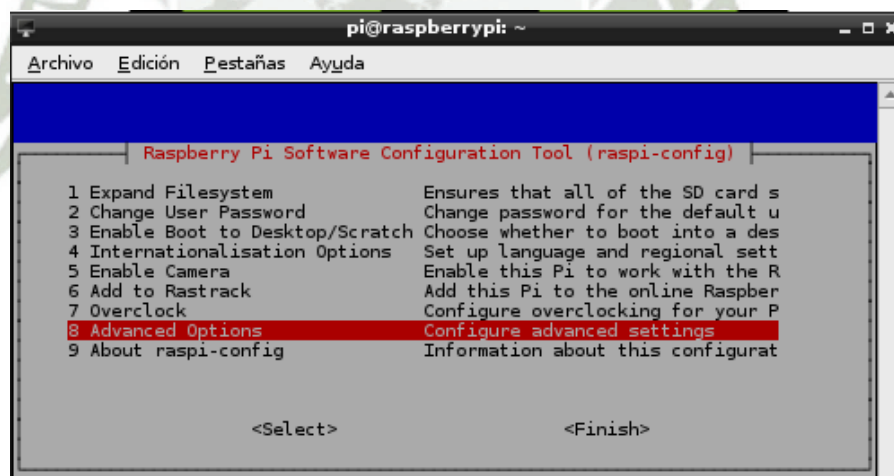
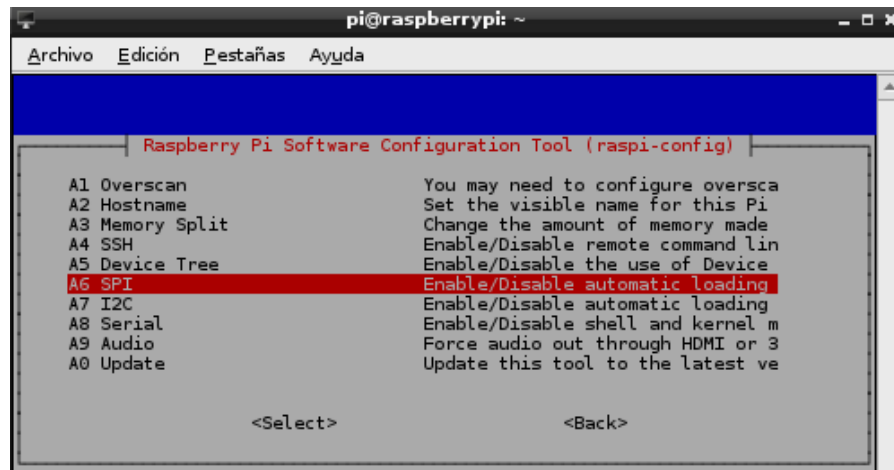


Figura 57:Menú de configuración.

Fuente: Elaboración propia.

Seleccionar "Advances options" en el menú, entonces marcar "SPI" o "I2C" para permitir que las interfaces inicien. Responder "Yes" a continuación para cargar automáticamente los controladores.



**Figura 58:**Configuraciones avanzadas.

*Fuente: Elaboración propia.*

Una vez que se indicó que se utilizará el puerto SPI, reiniciar el sistema y volver a abrir la ventana del LXTerminal, ingresar al siguiente fichero para configurarlo:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Agregar '#' en frente de la línea "spi-bcm2708" y guardar los cambios. Para poder utilizarlo, se debe instalar la librería "SPI" en Python:

```
git clone https://github.com/lthiery/SPI-Py.git
cd SPI-Py
sudo python setup.py install
```

Finalmente el sistema está listo para poder realizar el programa de clasificación.

### 3.2.3 Interfaz electrónica

Es necesario elaborar una tarjeta y así acondicionar las señales de salida del Raspberry y comunicarse con el PLC, este desarrollo debe contemplar robustez, es decir procurar asegurar el buen funcionamiento de los equipos y reducir las interferencias exteriores que se puedan producir.

**Tabla 32:** PinOuts Raspberry.

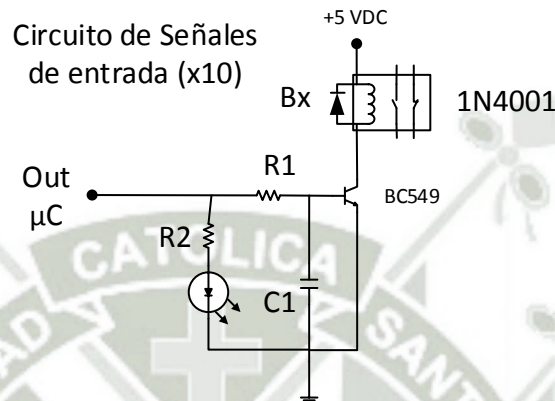
Physical / Raspberry Pi Name			
3.3v	1	2	5v
SDA0	3	4	DNC
SCL0	5	6	0v
GPIO 7	7	8	TX
DNC	9	10	RX
GPIO 0	11	12	GPIO 1
GPIO 2	13	14	DNC
GPIO 3	15	16	GPIO 4
DNC	17	18	GPIO 5
SPI MOSI	19	20	DNC
SPI MISO	21	22	GPIO 6
SPI SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

**Tabla 33:** Nomenclatura de conexiones.

Pin Numbers	RPI.GPIO	Raspberry Pi Name	BCM2835	Description
P1_01	1	3V3		3.3 Volts
P1_02	2	5V0		5 Volts
P1_03	3	SDA0	GPIO0	IC2 Bus
P1_04	4	DNC		Do not connect
P1_05	5	SCL0	GPIO1	IC2 Bus
P1_06	6	GND		Ground 0 Volts
P1_07	7	GPIO7	GPIO4	General Input output header
P1_08	8	TXD	GPIO14	UART - Transmit
P1_09	9	DNC		Do not connect
P1_10	10	RXD	GPIO15	UART - Receive
P1_11	11	GPIO0	GPIO17	General Input output header
P1_12	12	GPIO1	GPIO18	General Input output header
P1_13	13	GPIO2	GPIO21	General Input output header
P1_14	14	DNC		Do not connect
P1_15	15	GPIO3	GPIO22	General Input output header
P1_16	16	GPIO4	GPIO23	General Input output header
P1_17	17	DNC		Do not connect
P1_18	18	GPIO5	GPIO24	General Input output header
P1_19	19	SPI_MOSI	GPIO10	Serial Peripheral Interface Bus
P1_20	20	DNC		Do not connect
P1_21	21	SPI_MISO	GPIO9	Serial Peripheral Interface Bus
P1_22	22	GPIO6	GPIO25	General Input output header
P1_23	23	SPI_SCLK	GPIO11	Serial Peripheral Interface Bus
P1_24	24	SPI_CE0_N	GPIO8	Serial Peripheral Interface Bus

P1_25	25	DNC		Do not connect
P1_26	26	SPI_CE1_N	GPIO7	Serial Peripheral Interface Bus

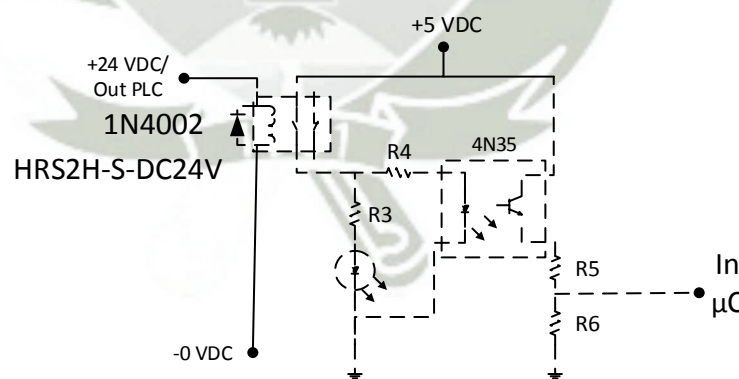
Se utilizan los GPIO<sup>12</sup> 1, 4, 5 y 6 como salidas, estas salidas están conectadas al siguiente circuito, calculado en la etapa de diseño, para poder enviar señales digitales al controlador PLC que recibe señales de 24 VDC.



**Figura 59:** Circuito de conexiones para las salidas de Raspberry.

*Fuente: Elaboración propia.*

Los GPIO 7, 0, 2 y 3 se utilizan como señales de entrada para el sistema, estas señales no deben ser superiores a los 5 VDC, para lo cual también se necesita acondicionarla, pues si se conecta a una señal de 24 VDC, esta dañaría al procesador del Raspberry.



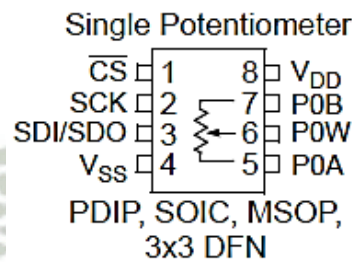
**Figura 60:** Circuito de conexiones para las entradas al Raspberry.

*Fuente: Elaboración propia.*

<sup>12</sup> *General Purpose Input/Output*, Entradas y salidas de propósito general.

Como ya se tiene el interfaz electrónico de las señales digitales, ahora se tienen 4 bits de entrada y 4 de salida para poder recibir y enviar información al PLC. Ahora se debe enviar una señal analógica, para esto se utiliza un integrado, un potenciómetro digital que permite modificar su valor con una señal SPI.

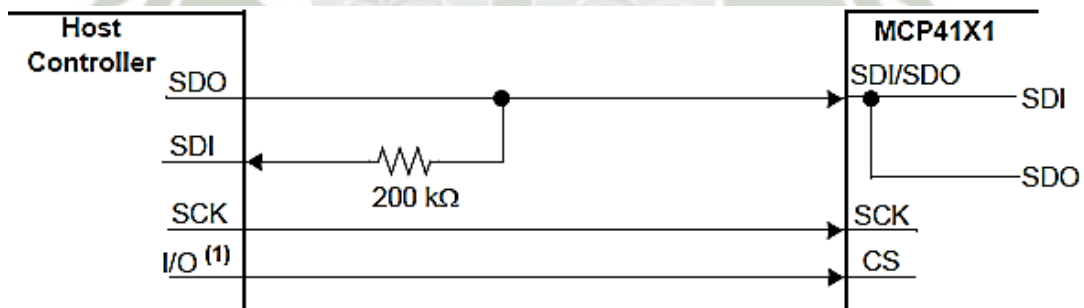
Se ha elegido el integrado *Microchip MCP4151*, un potenciómetro digital de 8 bits de resolución con memoria volátil.



**Figura 61:** Potenciómetro digital. MCP41X1

*Fuente: Hoja de datos Microchip MCP41X1.*

Este integrado debe conectarse al Raspberry, típica conexión SPI en modo "Host" para tener el integrado como esclavo del controlador.



**Figura 62:** Conexión de Raspberry con el integrado.

*Fuente: Hoja de datos Microchip MCP41X1.*

Finalmente ya se tiene todas las conexiones de controlador, ahora realizamos las conexiones y elaboramos la tarjeta electrónica, los planos y la distribución se adjuntan en los anexos.

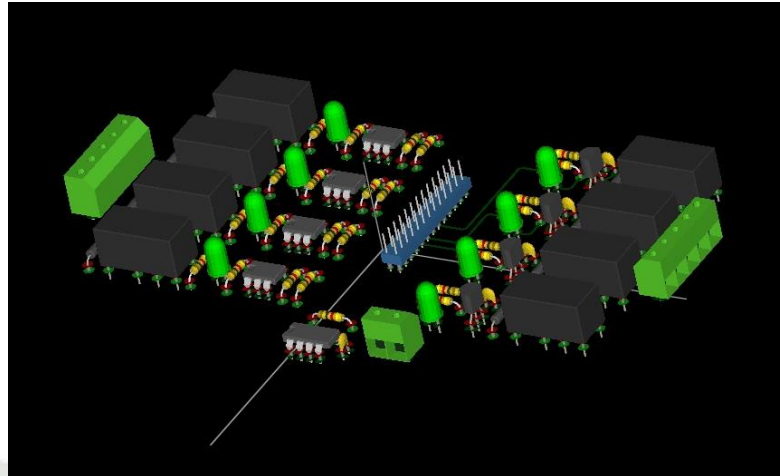


Figura 63: Vista en 3D de tarjeta.

Fuente: Elaboración propia.

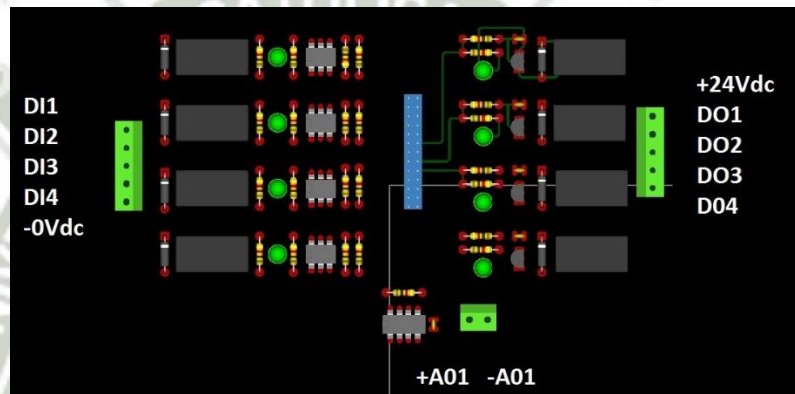


Figura 64: Vista superior de tarjeta.

Fuente: Elaboración propia.

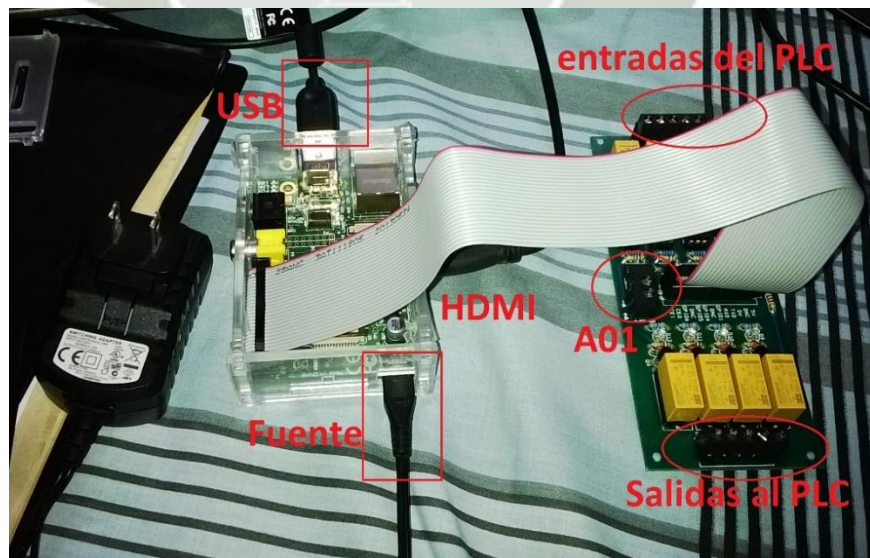


Figura 65: Conexiones Raspberry.

Fuente: Elaboración propia.

## Capítulo 4

# Programación del sistema

### 4.1 Diagrama de flujo del funcionamiento

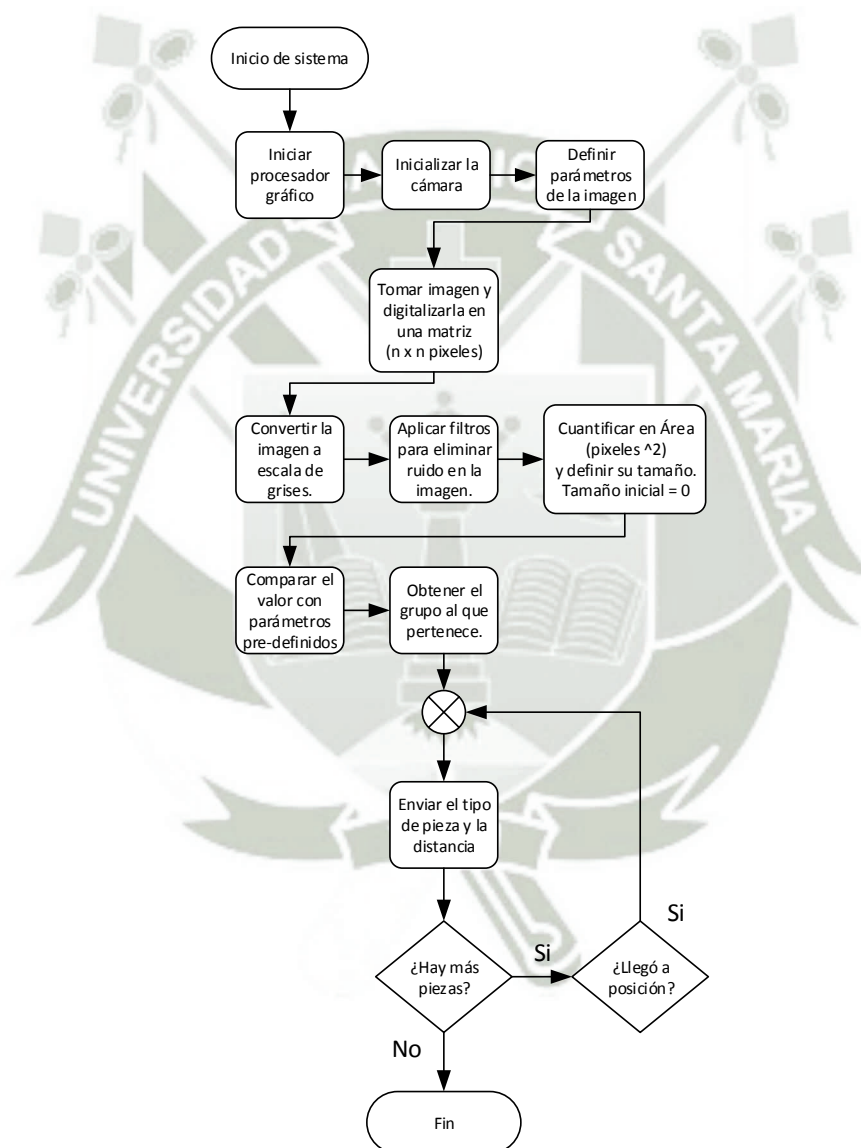


Figura 66:Diagrama de flujo.

Fuente: Elaboración propia.

El sistema necesita una secuencia de programación, pero para esto es necesario establecer una lógica la cual se debe seguir. En la figura 66 se explica en bloques generales la secuencia. Para el funcionamiento se realizó el programa en la Python utilizando varias herramientas que nos permiten adquirir imágenes, procesarlas como operaciones matriciales y vectoriales para procesar las imágenes. Después se envía la información al PLC y una vez que el Controlador ya tiene todas las señales, debe mover el motor, para lo cual se utiliza una salida análoga de +10V a -10V, con lo cual se le puede indicar el sentido de giro del motor. Una vez que estos se encuentren en posición, activaremos el electroimán para así mover las piezas de su posición.

## 4.2 Proceso de aprendizaje

Para que el sistema tenga un aprendizaje exitoso se utilizó el algoritmo K-nn (K-nearest neighbors o el vecino más cercano). El primer paso para lograr el aprendizaje es realizar un procesamiento de la imagen, después a partir de esta imagen se tiene información, tomando como punto de partida esta información, se entrena el sistema; es decir, se le da la información adecuada y el segmento o grupo de parámetros al que pertenece. A partir de esta información almacenada, el algoritmo permite obtener a que grupo pertenece la nueva entrada basado en información estadística.

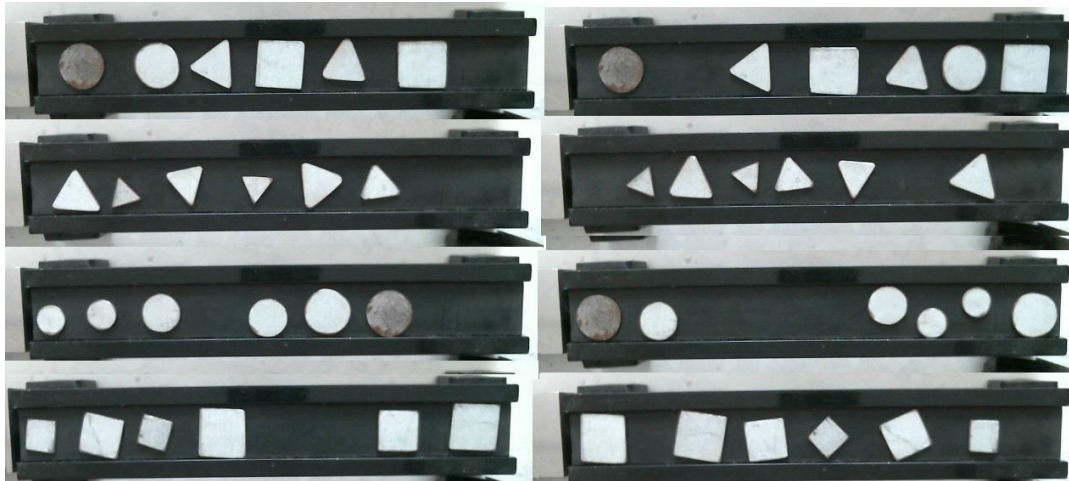
### 4.2.1 Procesamiento de la imagen

El primer paso de este proceso es obtener la matriz de la imagen como dato de entrada se tiene una imagen en formato .jpg<sup>13</sup> de 640 x 480 pixeles, pero de los cuales se recortan a un área menor en el cual se observe solo lo necesario.

La imagen utilizada para entrenar el sistema es la siguiente:

---

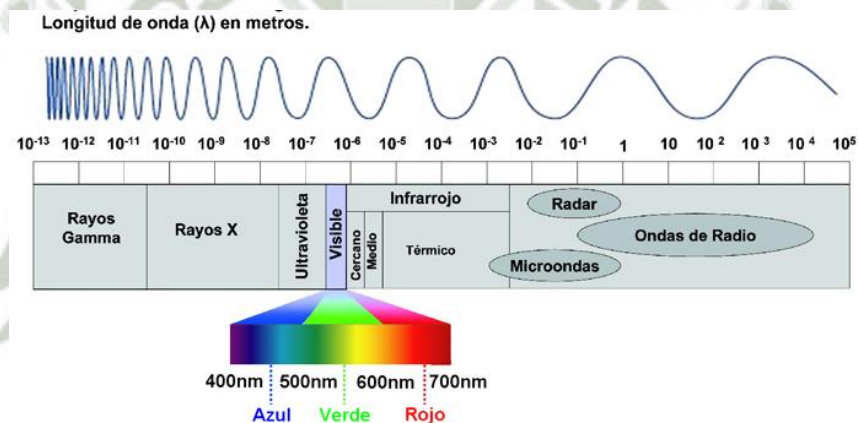
<sup>13</sup> **JPEG** (del inglés *Joint Photographic Experts Group*, *Grupo Conjunto de Expertos en Fotografía*) es el nombre de un comité de expertos que creó un estándar de compresión y codificación de archivos e imágenes fijas.



**Figura 67:**Imagen para iniciar el aprendizaje.

*Fuente: Elaboración propia.*

Se observa en la figura 67 una imagen, la cual utilizaremos para entrenar al sistema.



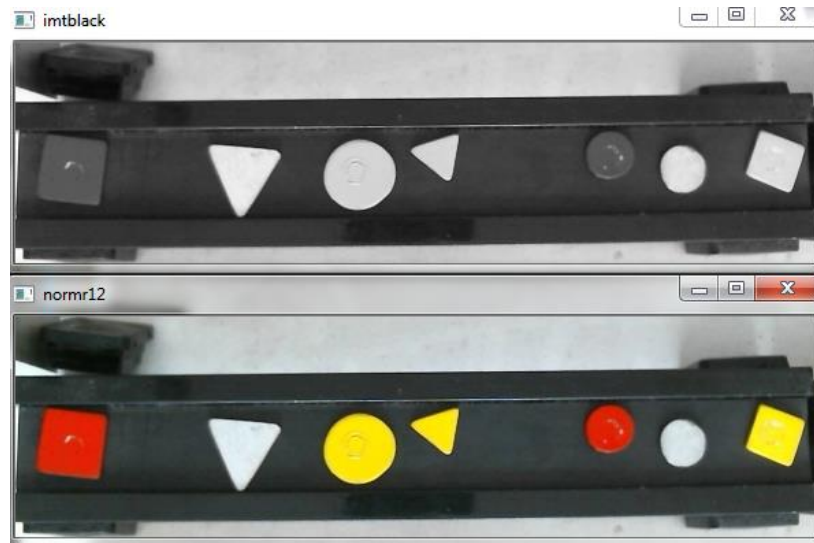
**Figura 68:** Espectro electromagnético.

*Fuente: Astronomía 2009.*

El ojo percibe distintas intensidades de luz en función del color que se observa, esto es debido a la respuesta del ojo al espectro, figura 68, por esa razón el cálculo del equivalente blanco y negro (escala de grises o luminancia) de la imagen debe realizarse como una media ponderada de las distintas componentes de color de cada pixel.

$$G = 0.30 \cdot R + 0.59 \cdot G + 0.11 \cdot B \dots (18)$$

Repitiendo esta misma operación a cada pixel de la imagen, se obtiene una a escala de grises.



**Figura 69:** Imagen de color a escala de grises.

*Fuente: Elaboración propia.*

Este procedimiento permite reducir una matriz de  $[n \times m \times 3]$  a  $[n \times m \times 1]$ , es decir por cada nueva posición de la matriz se tiene un solo valor  $[G]$  que varía de 0 a 255. Esta reducción permite disminuir los tiempos de procesamiento de la imagen. Después de este paso se utilizó un filtro de la mediana para poder quitar las imperfecciones o ruidos de la imagen, pero se debe evitar modificar la imagen y perder estructuras de la imagen que se consideren importantes.

Como explica Cuevas [27], la estadística de la mediana es el valor de la variable que deja el mismo número de datos antes y después que él. Considerando que  $x_1, x_2, x_3 \dots x_n$  son los datos de una muestra ordenada en orden creciente, la mediana quedaría definida como:

$$M_e = \frac{x_{n+1}}{2} \dots (19)$$

El filtro de la mediana sustituye cada pixel de la imagen por la mediana de los valores de intensidad dentro de la región de influencia  $R(x,y)$  definida por el filtro, esto expresado formalmente quedaría como:

$$I'(x, y) = M_e(R(x, y)) \dots (20)$$

Para efectos de este proceso, la región de influencia es "7", con lo cual se toma los 49 valores alrededor de un pixel y su mediana reemplaza al valor anterior.



**Figura 70:** Filtro de la mediana.

*Fuente: Environmental software and modelling group*

Hasta el momento se tiene la imagen representada en una matriz con valores que varían de 0 a 255, pues es una imagen en grises.



**Figura 71:** Resultado de aplicar el filtro en la imagen.

*Fuente: Elaboración propia.*

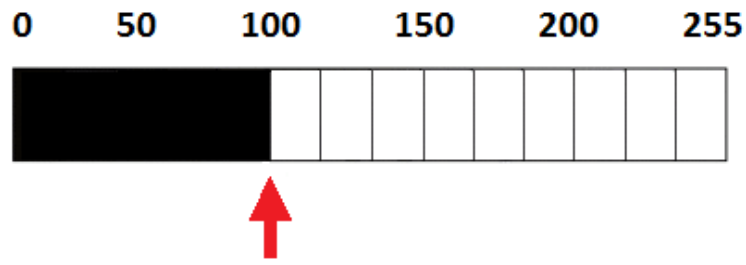
Posteriormente se resalta las diferencias entre el blanco y el negro, pues hasta el momento la imagen es gris, para esto se utilizó la herramienta del umbral.



**Figura 72:** Escala de grises.

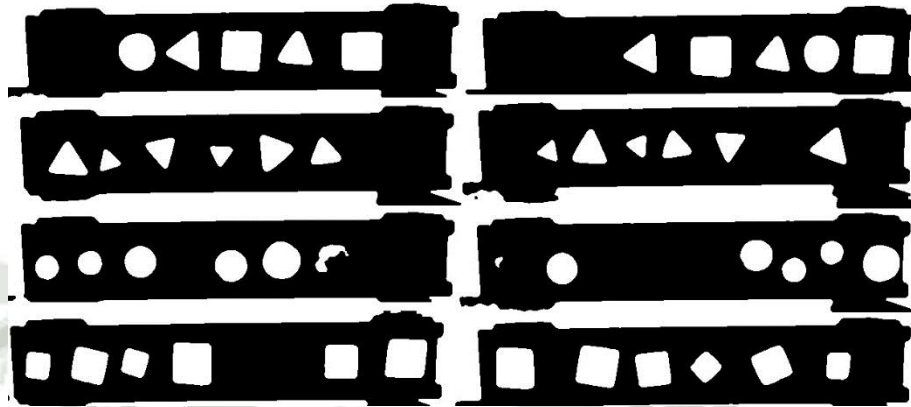
*Fuente: Elaboración propia.*

Marcando un valor umbral, todos los valores por encima o por debajo de este son reemplazados por 0 o por 255 (negro o blanco), para este caso se estableció un valor de comparación en 100, así que todos los valores por encima de este valor, se hace 255 y los menores 0, es decir nuestra imagen se convierte en una imagen binaria de dos colores, 0 y 255, como se muestra a continuación.



**Figura 73:** Resultado de haber utilizado umbral.

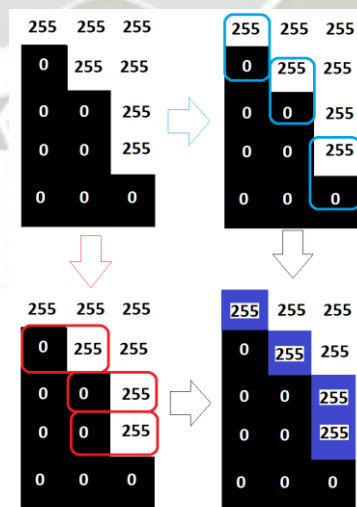
*Fuente: Elaboración propia.*



**Figura 74:** Imagen binaria de entrenamiento.

*Fuente: Elaboración propia.*

Como se tiene la imagen en blanco y negro, es más sencillo detectar los bordes, pues estos se dan en los puntos donde existe un cambio de 0 a 255, tanto en las filas como en las columnas.



**Figura 75:** Algoritmo de detección de bordes.

*Fuente: Elaboración propia.*

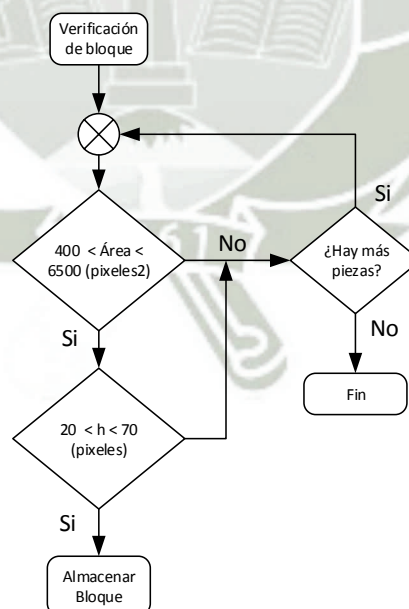
Ahora utilizando herramientas de programación, se ubican los bordes de todos los bloques encontrados, obteniendo así la siguiente imagen.



**Figura 76:**Imagen con bordes detectados.

*Fuente: Elaboración propia.*

Cada una de ellas encierra un área en píxeles, ahora solo se toma aquellas que tienen un área entre 1000 y 4000 píxeles<sup>2</sup>, pues es el tamaño aproximado de cada bloque en una imagen de 640x480 píxeles. Pero esto no es suficiente para encontrar los bloques, si no también hay que determinar que el tamaño este dentro del rango de 50 a 60 píxeles de ancho para asegurar que es un bloque. Este procedimiento se basa en comparación de términos y se rigen bajo la siguiente lógica.



**Figura 77:** Diagrama de flujo del proceso de aprendizaje.

*Fuente: Elaboración propia.*

Ahora se bordea la imagen y se tiene una matriz completa que define una forma, con esta información se realiza el aprendizaje.









**Figura 78:** Borde identificado.

*Fuente: Elaboración propia.*

#### 4.2.2 Aprendizaje del sistema

Como ya se tiene la imagen preparada, solo queda especificar a qué grupo pertenece, mientras más muestras se tomen, se tendrá un sistema más preparado.

**Tabla 34:** Información para el aprendizaje.

Bloque	Denominación
	Circulo
	Cuadrado
	Circulo
	Triangulo
	Cuadrado
	Triangulo

*Fuente: Elaboración propia.*

La información almacenada es la matriz que se muestra a continuación, esta matriz sirve como base para poder reconocer las formas posteriormente.

✓ Circulo

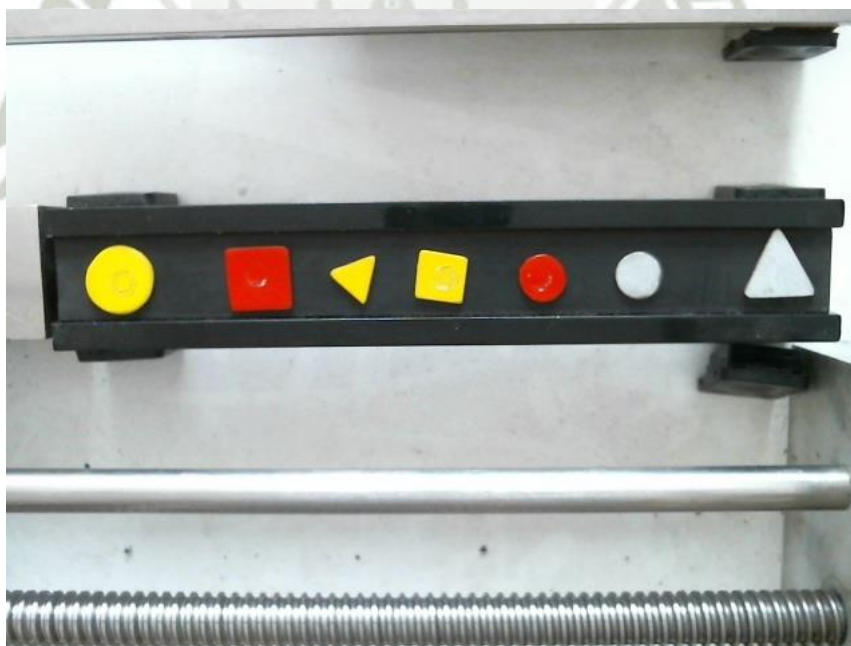
```

[[255 234 0 0 0 0 219 255 255]
 [255 0 0 0 0 0 0 0 255]
 [255 150 0 0 0 0 0 0 29]
 [255 0 0 0 0 0 0 0 0]
 [ 38 0 0 0 0 0 0 0 0]
 [255 0 0 0 0 0 0 0 0]
 [ 38 0 0 0 0 0 0 0 0]
 [ 10 0 0 0 0 0 0 0 0]
 [255 63 0 0 0 0 0 0 255]
 [255 255 255 0 0 0 0 255 255]]
    
```

### 4.3 Clasificación de piezas

a) *Determinar la forma*

Para realizar la clasificación, se parte de la imagen adquirida por la cámara.



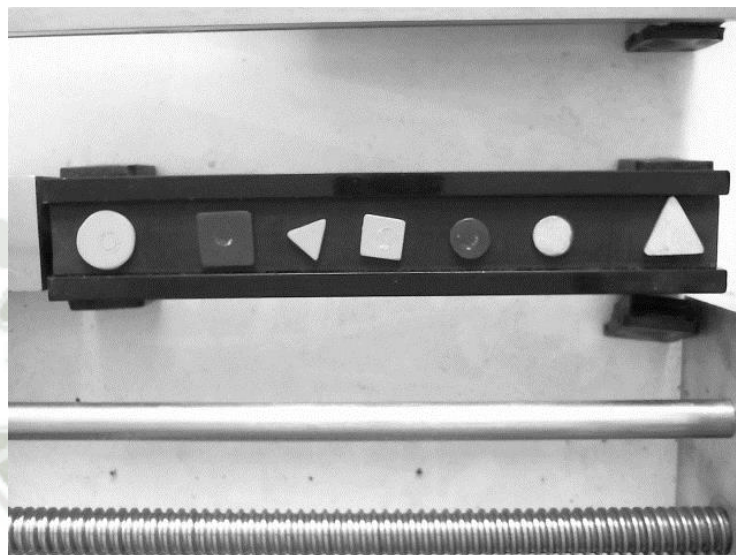
**Figura 79:** Imagen 640 x 480.

*Fuente: Elaboración propia.*

El proceso de clasificación está basado en cuatro etapas, primero es necesario detectar la base sobre las cuales están situadas las piezas, obtener la matriz de información de cada pieza, reconocer el tipo de pieza que es, tanto en forma, tamaño

y color, para finalmente enviar la información y las mismas sean separadas en 3 grupos.

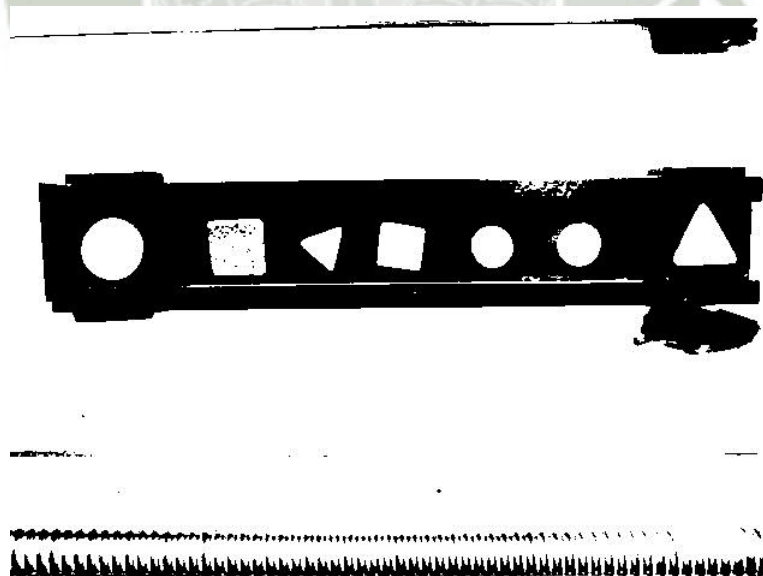
Primero para obtener la base se sigue un pre procesamiento de imagen como el hecho para entrenar al sistema, convertimos la imagen RGB en escala de grises.



**Figura 80:** Imagen de RGB a escala de grises.

*Fuente: Elaboración propia.*

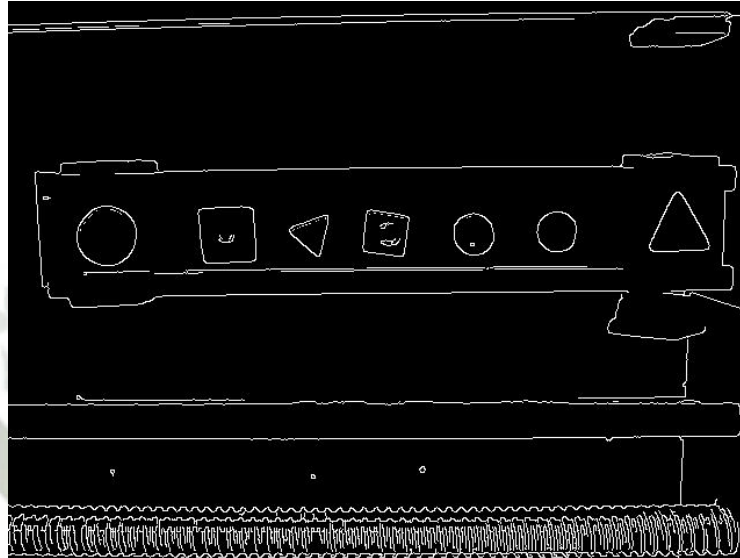
Una vez que la imagen ya esta en grises, se aplica un umbral, para así detectar bloques y bordes



**Figura 81:** Imagen con Umbral.

*Fuente: Elaboración propia.*

Después se aplica el filtro Canny para detectar bordes, el resultado de la imagen es la siguiente.



**Figura 82:**Filtro canny sobre la imagen.

*Fuente: Elaboración propia.*

Luego estas dos últimas imágenes se suman y se tiene como resultado la siguiente:



**Figura 83:**Detección de base.

*Fuente: Elaboración propia.*

Para poder detectar la base, se verifica todas las áreas encontradas, en esta imagen la base ya fue separada del resto de imágenes, por lo cual se determina la base por comparación, debe cumplir las siguientes características:

**Tabla 35:**Características de la base.

Área Máx	Área min	Ancho Máx	Ancho min	Largo Máx	Largo min
15000	80000	550	640	120	165

Estos valores se determinaron experimentalmente y nos dieron buenos resultados. Ahora como se ha detectó la base se recorta la imagen, para trabajar sobre ella.



**Figura 84:**Área de inspección.

*Fuente: Elaboración propia.*

Después se utilizan los filtros, como lo hicimos para la base (gris, mediana y umbral adaptativo), pero en una imagen más pequeña, entonces aquí se puede diferenciar claramente todas las formas.



**Figura 85:**Imagen procesada.

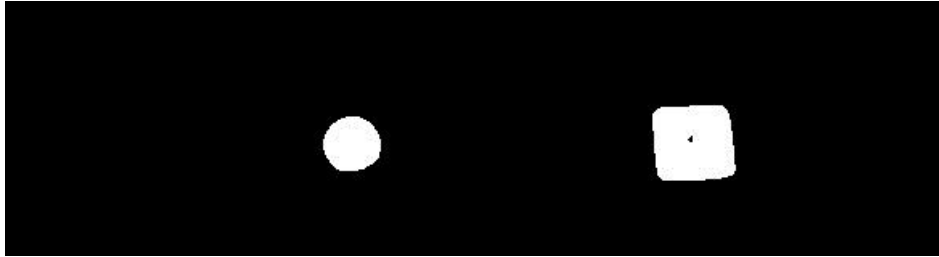
*Fuente: Elaboración propia.*

Si se observa bien, se puede constatar que las herramientas utilizadas no perciben el color rojo, por este motivo es necesario realizar un algoritmo para detectar las piezas de este color, esto se hizo comparando los valores [R,G,B] de la imagen bajo las siguientes características:

**Tabla 36:**Requerimientos del color rojo.

R>80	G<60	B<60
------	------	------

Se tiene como resultado la figura 86.



**Figura 86:** Imagen filtrada en rojo.

*Fuente: Elaboración propia.*

Para finalizar se suman ambas imágenes y de esta manera se sigue el procedimiento.



**Figura 87:** Imagen procesada lista para ubicar las piezas.

*Fuente: Elaboración propia.*

Para considerar un bloque de imagen como una pieza, estas deben cumplir ciertas características de tamaño, sobre todo para evitar el ruido de la imagen:

**Tabla 37:** Características de las figuras.

Área Máx	Área mín	Ancho Máx	Ancho mín	Largo Máx
400 pix	6500 pix	20 pix	70 pix	100 pix

Finalmente, se puede separar las imágenes por bloques y así ya se tendrían las matrices de 100x100 para poder utilizar el algoritmo.



**Figura 88:** Matriz 100 x 100 de la figura.

*Fuente: Elaboración propia.*

Como ya tenemos la matriz, es momento de utilizar el algoritmo K-nn. Primero se necesita leer la información adquirida en el entrenamiento, y se hace de la siguiente manera.

```
filememory=filepath("Memoria/generalsamples.data")
samples = np.loadtxt(filememory,np.float32)
filememory=filepath("Memoria/generalresponses.data")
responses = np.loadtxt(filememory,np.float32)
responses = responses.reshape((responses.size,1))
```

Una vez que ya se tiene la información, se entrena el algoritmo Knn.

```
#Llamamos a la funcion del KNearest para reconocer los datos
model = cv2.KNearest()
model.train(samples, responses)
```

Finalmente se obtiene la imagen, se reescala a una matriz de 10 x10 y después se convierte en un vector, una vez se tiene la información de esta manera, se utiliza el algoritmo.

```
#Para reconocer la forma
#Viendo el area
roi = thresh1[y:y+h,x:x+w]
roismall = cv2.resize(roi, (10,10))
roismall = roismall.reshape((1,100))
roismall = np.float32(roismall)
retval, results, neigh_resp, dists = model.find_nearest(roismall,
k = 1)
forma=int((results[0][0]))
```

Se tiene como salida del sistema la forma de la pieza en un número entero y que corresponde a la siguiente tabla:

**Tabla 38:**Correspondencia numérica de forma.

Triangulo	1
Cuadrado	2
Circulo	3

b) Determinar el tamaño de las piezas.

Para determinar el tamaño de las piezas, solo se hace por comparación de pixeles, como ya se tiene una matriz que nos describe las imágenes, tomando en cuenta la forma que tiene, se compara el área y así se reconoce el tamaño de la pieza.

**Tabla 39:**Designación de tamaños.

A = (pixeles <sup>2</sup> )	Triangulo	Cuadrado	Circulo
<b>Pequeño</b>	A<800	A<1500	A>1900
<b>Mediano</b>	A<1200 and A>800	A<2200 and A>1500	A<1900 and A>1100
<b>Grande</b>	A>1200	A>2200	A<1100

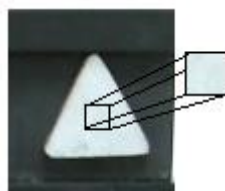
c) Determinar el color de las piezas.

Para poder saber el color de las piezas es sencillo, como se tiene una matriz (m x n x 3), se tiene información sobre el color de la imagen, para esto es necesario tomar un pequeño área de 10 x 10 pixeles del centro de la pieza, promediarlos y comparar sus valores según la siguiente tabla:

**Tabla 40:**Designación de tamaños.

	Rojo [r]	Verde [g]	Azul [b]
<b>Rojo</b>	r>80	b<60	g<60
<b>Plata</b>	r>100	b>100	g>100
<b>Amarillo</b>	r>200	b<50	g>150

Por ejemplo, este pedazo de imagen, tiene los siguientes valores RGB



[r,g,b]=[120,150,200]

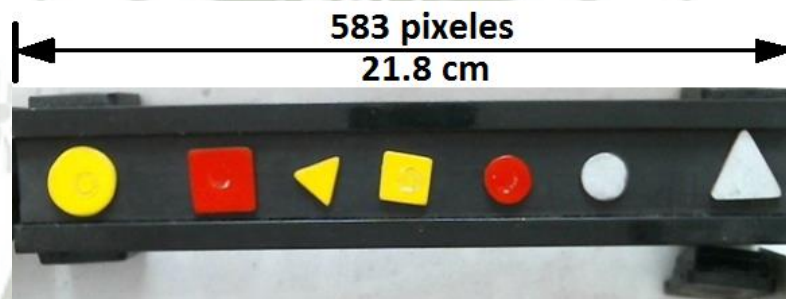
**Figura 89:**Análisis de color.

*Fuente: Elaboración propia.*

Al revisar la tabla superior, se observa que esta pieza es de color plata, este mismo procedimiento se repite para todas las piezas.

#### 4.4 Ubicar las piezas

Se revisa el área de inspección y se necesita determinar un punto fijo, en base a él se calcula la distancia, como se tiene una altura 20 cm entre la cámara y la base, se puede calcular la distancia cuantificando los pixeles y comparándolos con valores reales. Por ejemplo en la imagen que se muestra a continuación, se tiene 583 pixeles de largo.



**Figura 90:** Base de las piezas.

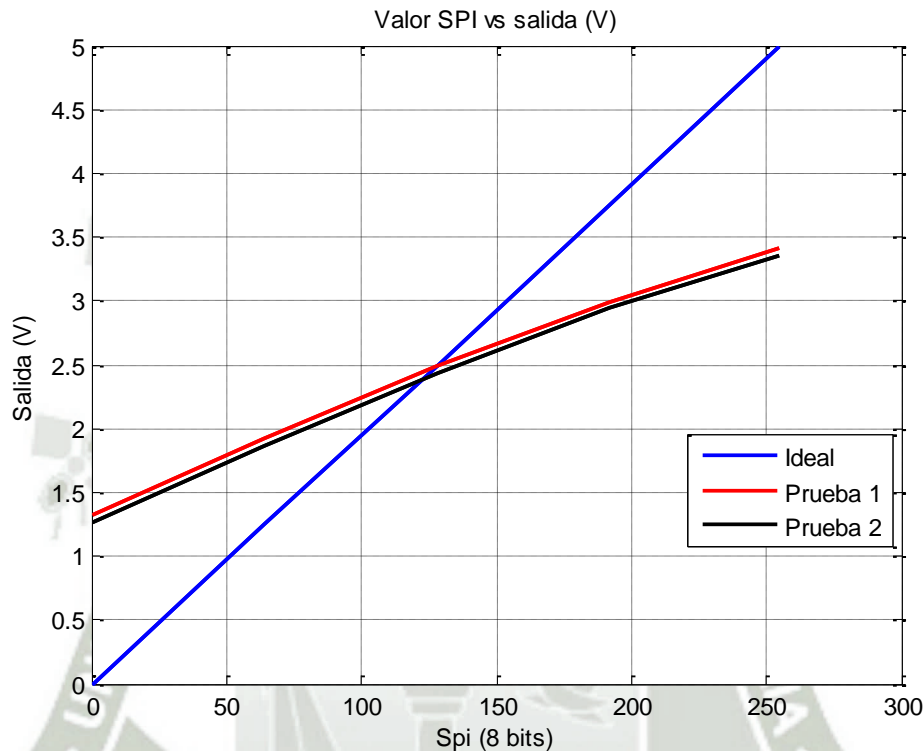
*Fuente: Elaboración propia.*

Como se conoce que esta distancia real mide 21.8 cm, entonces se puede obtener un factor que relacione ambos valores.

$$fs = \frac{21.8}{583} = 0.0374 \text{ cm/pixel}$$

Posteriormente se determinó el centro de área de cada bloque, gracias a herramientas que nos permita utilizar el OpenCV (`cv2.findContours`), después se multiplica el valor de pixel que nos indica la posición de cada pieza con el factor y así se obtiene la distancia real de cada pieza respecto al inicio. Posteriormente esta señal se escala en 8 bits de 0 a 255 para ser enviado por medio del puerto SPI al potenciómetro y generar un diferencial de potencial, el cual reconocerá el PLC, para que este identifique donde se encuentra la pieza.

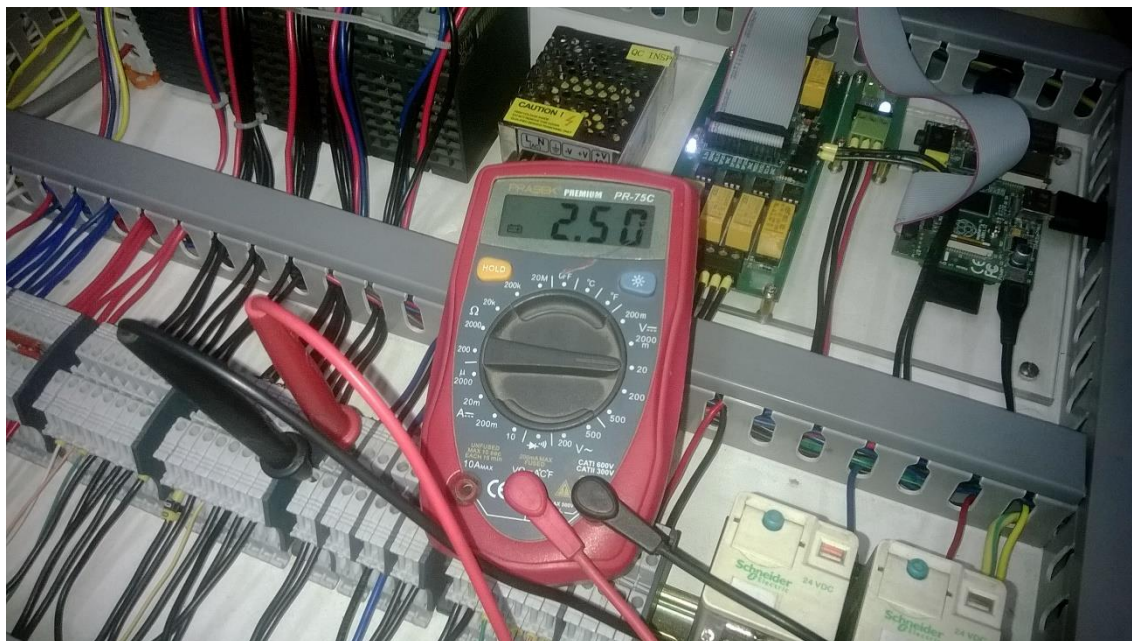
Es necesario mencionar que este potenciómetro digital no es lo suficientemente robusto y demostró ser susceptible al uso y a la temperatura, cuando se inicia su uso presenta variaciones de 0 a 5 V, pero en cuanto se va utilizando, varia de 1.3 a 3.3V, su comportamiento se explica en la siguiente figura.



**Figura 91:** Salida SPI.

*Fuente: Elaboración propia.*

Después de percatarnos de este comportamiento, el funcionamiento del potenciómetro digital se mantiene en los intervalos de 1.3 a 3.3 V, por lo cual escalamos estos valores tanto de salida, como de entrada en el PLC, pero es necesario mencionar que al iniciar el sistema, es necesario esperar unos minutos antes de iniciar la clasificación.



**Figura 92:** Toma de datos de salida MCP4151.

*Fuente: Elaboración propia.*

Después de realizar todo este procedimiento, se tiene como resultado final la figura 93 y 4 vectores

```
#Variables de forma
global t_f
global jf
#Variables de tamaño
global t_t
global jt
#Variables de color
global t_c
global jc
#Variables de posicion
global jp #Posicion en x en pixeles
global jpy #Posicion en y en pixeles

# Vector que tiene el numero de piezas ubicadas en forma
t_f=np.zeros((1, pi+1))

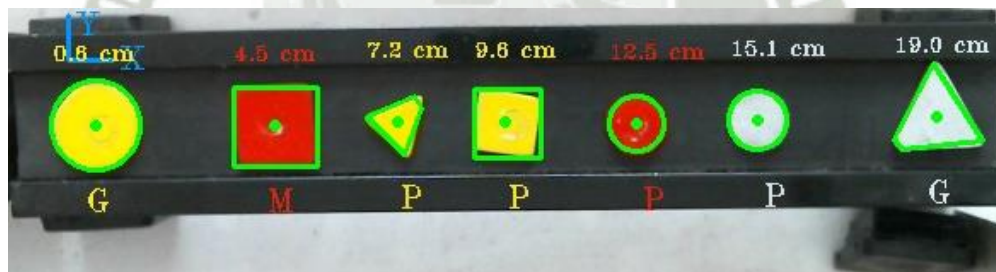
# Vector que tiene el numero de piezas ubicadas en tamaño
```

```
t_t=np.zeros((1, pi+1))

# Vector que tiene el numero de piezas ubicadas en color
t_c=np.zeros((1, pi+1))

#Lista de piezas en letras de la forma
jf=list(range(pi+1))
```

Esta información al final debe ser ordenada de derecha a izquierda, pues se debe recordar que el actuador no nos permite pasar una pieza encima de otra debido a la potencia de energía magnética. Al procesar la imagen se obtiene todos los parámetros en forma, tamaño y color, nada más resta verificar que tipo de información es la que se necesita y es solo está la información que se envía al PLC para su clasificación. En caso de seleccionarse 2 o 3 de estos parámetros, la clasificación se define en piezas buenas o piezas malas, clasificando las piezas como buenas, las que cumplen las condiciones, y malas, que son las que no las cumplen.



**Figura 93:**Resultado final del procesamiento.

*Fuente: Elaboración propia.*

El procedimiento a detalle se explica en el programa adjunto a este documento.

## 4.5 Separar las piezas

### 4.5.1 Parametrización de controlador industrial

Para la parte del posicionamiento en el proyecto se seleccionaron elementos de uso industria: PLC Modular VIPA y Driver Servomotor Sew.

### a) Programación PLC

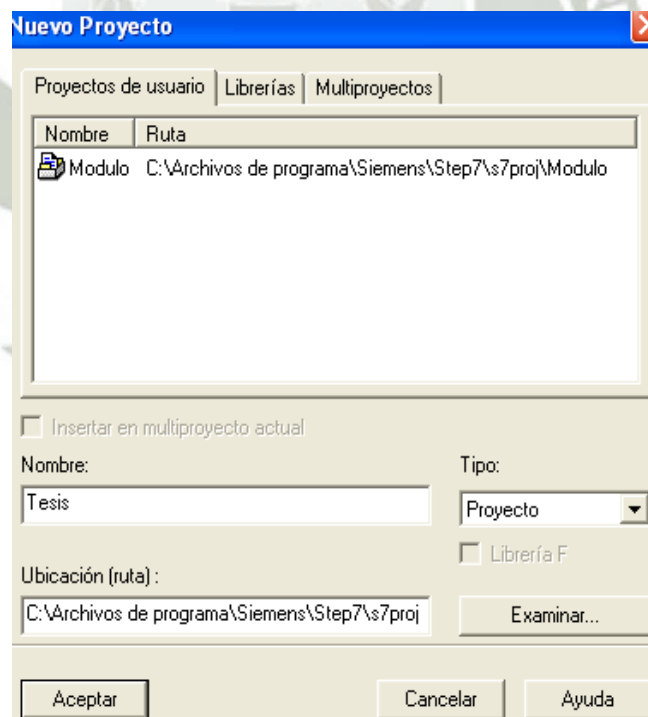
El PLC utilizado es de tipo modular marca Vipa, procedencia Alemana, se consideraron los siguientes módulos:

- CPU 214-1BC02
- Mod. I/O 221-1BF00 DI 8 x 24VDC
- Mod. I/O 222-1HF00 DO 8 x rele
- Mod. I/O 238-2BC00 AI 4, AO 2 x 12Bits, Fast Counter x 3

Este conjunto tiene las características mínimas necesarias para cumplir con los requerimientos que involucra la realización del proyecto, representando además un balance apropiado entre costo – producto para el mercado. El software utilizado para la programación del PLC es el Step7 v5.5 proveído por Siemens AG.

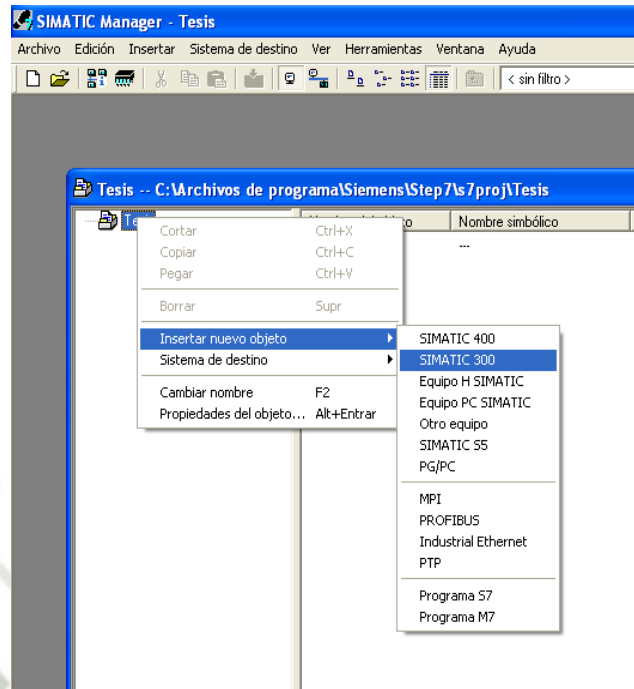
### b) Configuración de hardware

- Crear un proyecto nuevo y asignar nombre.
- Insertar un objeto Simatic 300.



**Figura 94:** Ventana de inicio.

*Fuente: Elaboración propia.*



**Figura 95:** Como insertar un modulo.

*Fuente: Elaboración propia.*

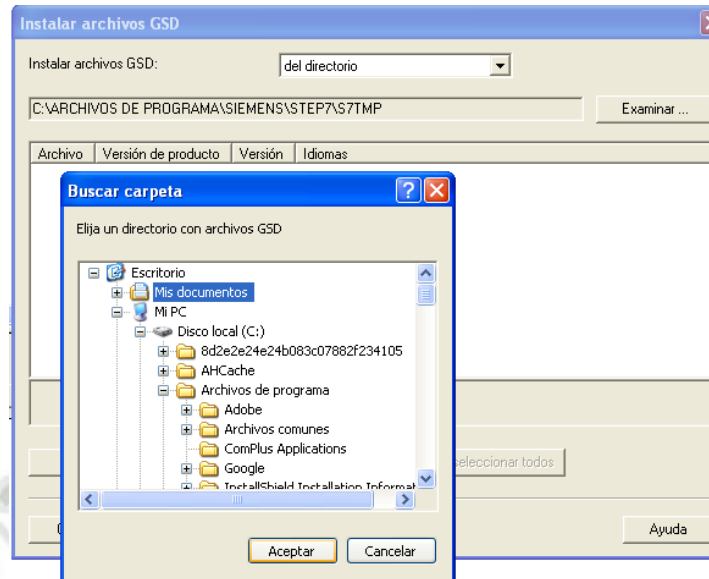
- Ir a Simatic 300(1) → Hardware → Herramientas → Instalar Archivos GSD.



**Figura 96:** Instalar archivos GSD.

*Fuente: Elaboración propia.*

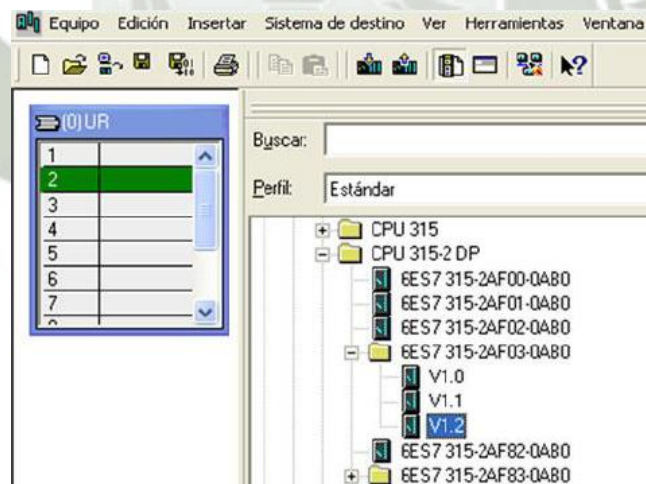
- Seleccionar ruta donde se encuentran los archivos GSD de VIPA (previamente descargados de la página oficial del fabricante), instalar.



**Figura 97:**Buscar archivos descargados.

*Fuente: Elaboración propia.*

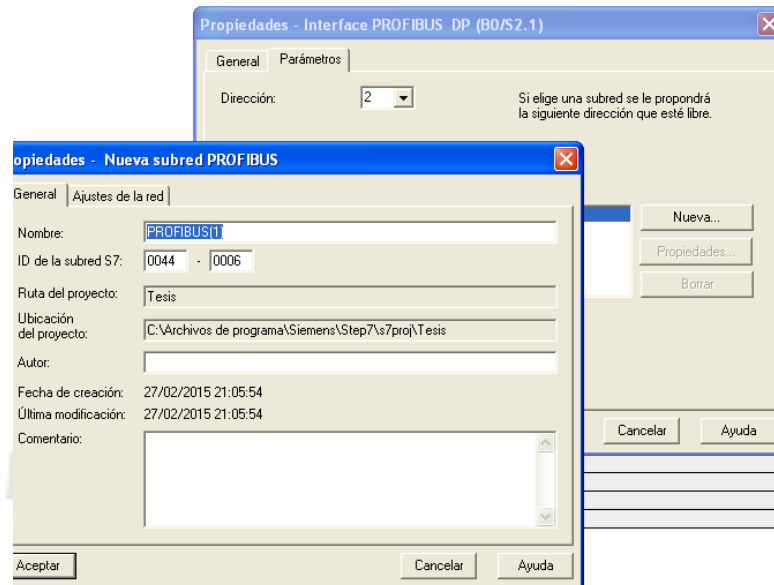
- Iniciar la adición de elementos desde el catálogo de hardware: Perfil de Soporte Simatic 300(Bastidor) y CPU 315–2DP (6ES7 315-2AF03-0A80 V1.2 )



**Figura 98:**Configurar bastidor.

*Fuente: Elaboración propia.*

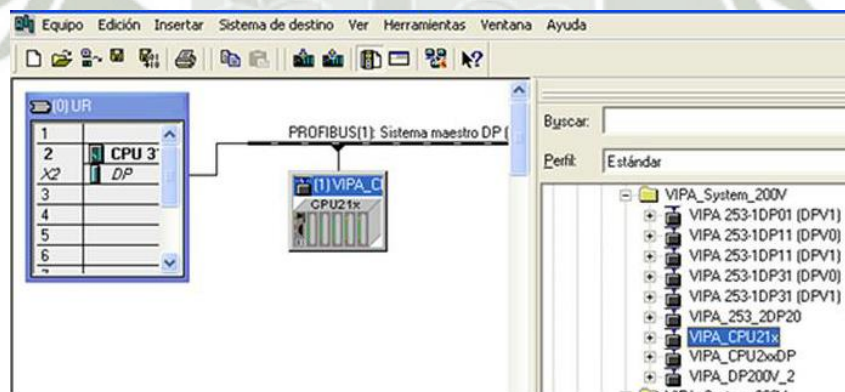
- En la ventana emergente seleccionamos “Nueva” para crear una subred Profibus y luego aceptamos los parámetros de fábrica.



**Figura 99:** Instalar Profibus.

*Fuente: Elaboración propia,*

- Ubicamos los elementos de VIPA instalados en el catálogo de Hardware, en la dirección: Profibus DP → Otros aparatos de campo → I/O, Seleccionamos Vipa\_System\_200V → Vipa\_CPU21x y arrastramos a la red Profibus, configurar como esclavo DP virtual.



**Figura 100:** Configurar red Profibus.

*Fuente: Elaboración propia.*

- Desde el catálogo de hardware seleccionar el CPU (*Unidad Central de Procesamiento*) adecuado así como los diferentes módulos de I/O con los que se ha de trabajar, configurar las direcciones y/o modos de funcionamiento de los mismos.

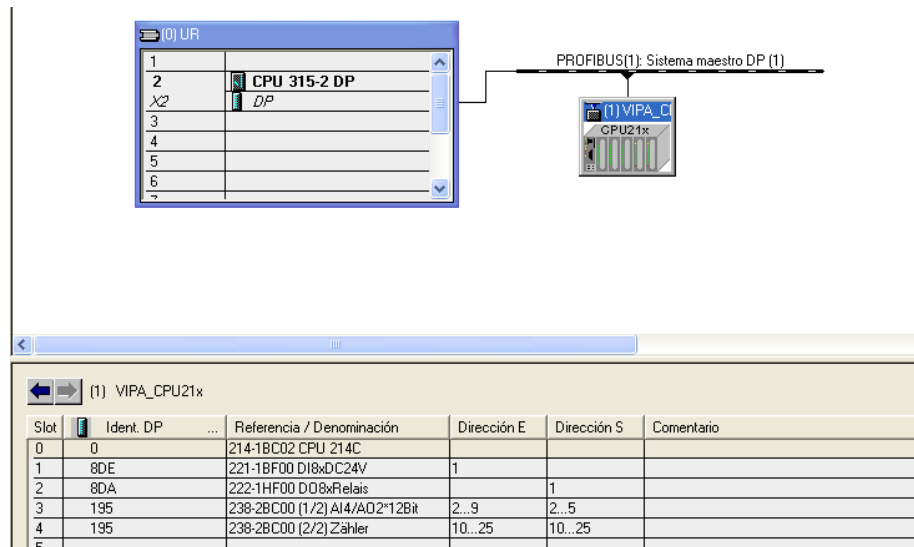


Figura 101: Módulos agregados.

Fuente: Elaboración propia.

- Los módulos se configuraron de la siguiente manera:
  - Mod. I/O 221-1BF00 DI 8 x 24VDC
  - Mod. I/O 222-1HF00 DO 8 x relé
  - Mod. I/O 238-2BC00 AI 1, AO 1 x 12Bits, Fast Counter x 1

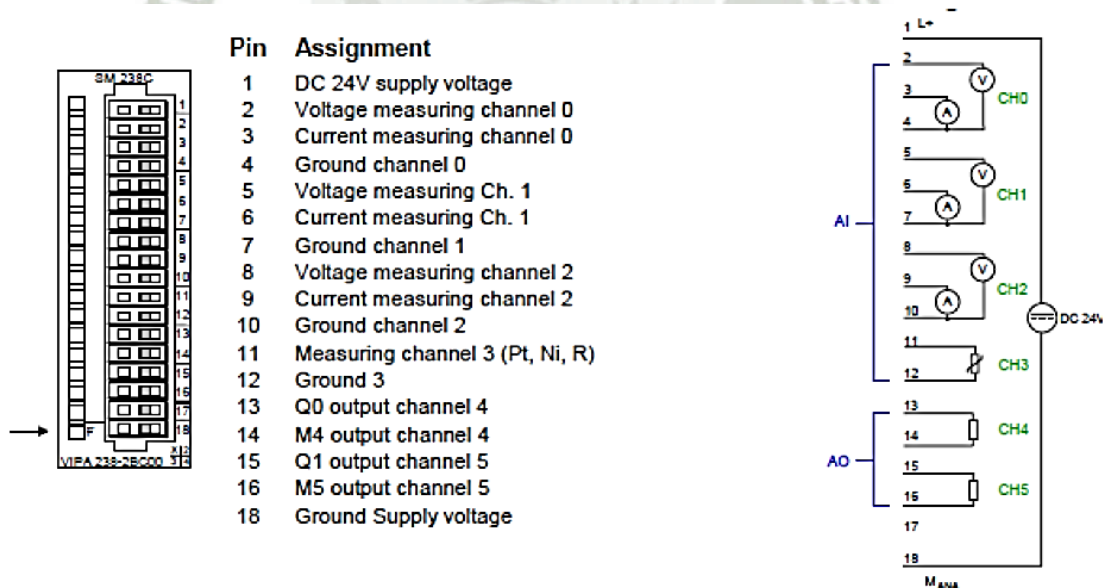


Figura 102: Asignación de módulo 238.

Fuente: Manual de VIPA.

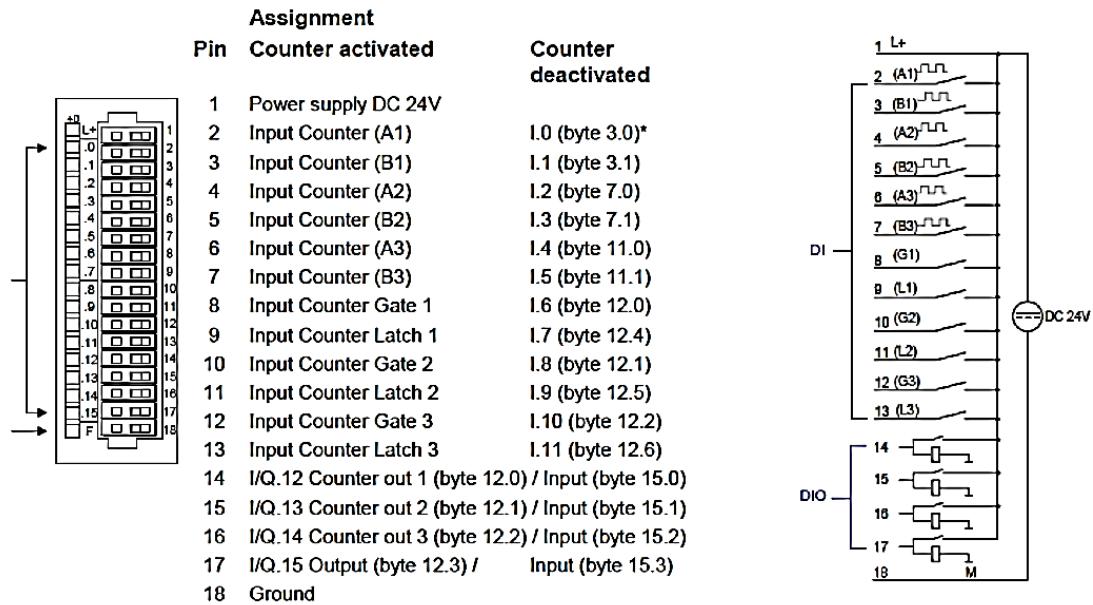


Figura 103: Módulo 238, contador rápido.

Fuente: Manual de VIPA.

- Configurar comunicación mediante cable MPI en ajustar interface PG/PC.
- Guardar – Compilar y cargar configuración en Modulo (PLC).

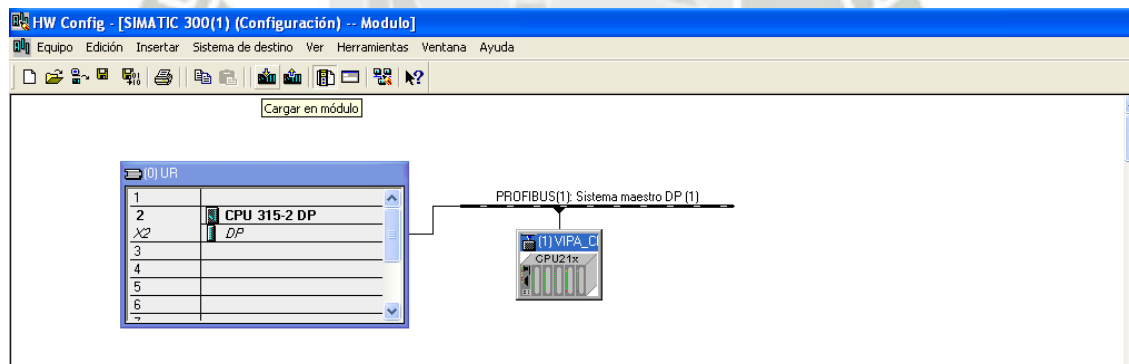


Figura 104: Red de PROFIBUS.

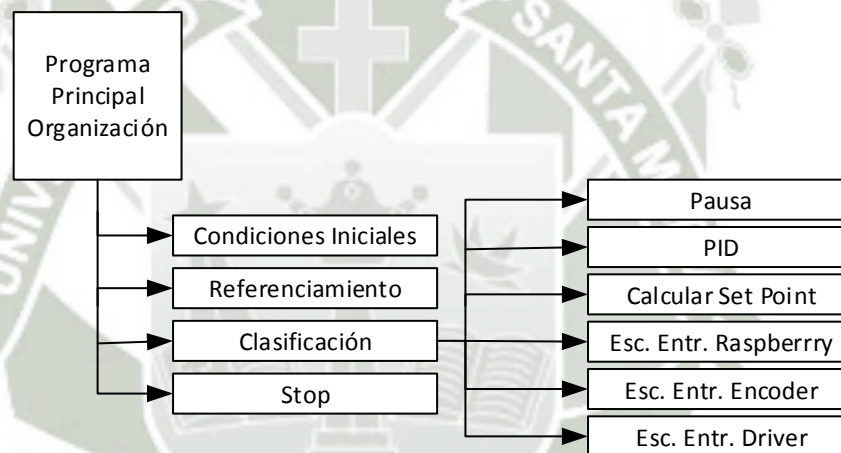
Fuente: Elaboración propia.

- Realizar pruebas de comunicación.

### c) Estructura general del programa

Se plantea agrupar las principales acciones de funcionamiento del sistema, se llega así a:

- **Condiciones iniciales:** al energizar el sistema para poder resetear posibles señales remanentes.
- **Referenciamiento:** como en todo sistema de posicionamiento se envía señal al driver hasta que la parte móvil encuentre un punto de referencia que será tomado como cero para el encoder.
- **Clasificación:** se realiza toda la lógica de posicionamiento para recoger las piezas y su posterior almacenamiento, cuando este programa se encuentra activo los anteriores se desactivan.
- **Stop:** desactiva los programas “Referenciamiento” - “Clasificación” y mantiene dicha condición hasta que se active el programa de “Condiciones Iniciales” y se libere Stop.



**Figura 105:** Organización del programa.

*Fuente: Elaboración propia.*

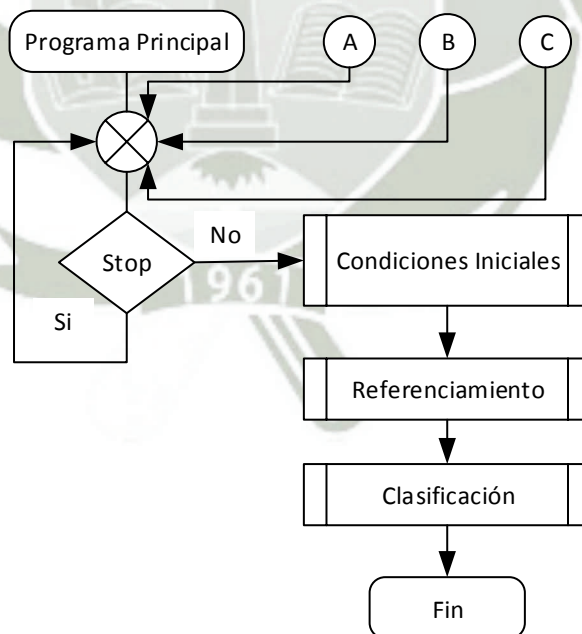
En “Clasificación” se realizan diversas acciones, para mantener el orden se plantea la creación de sub programas:

- **Pausa:** a diferencia de Stop que puede ser llamado en cualquier instante, Pausa puede ser activado únicamente cuando el sistema se encuentre en el programa Clasificación; Condiciones Iniciales y Referenciamiento son programas momentáneos y de acondicionamiento.

- **PID:** como estrategia de control con retroalimentación se sugiere realizarlo mediante el PID interno que provee el software, step7.
- **Calc. Set Point:** ya que existirán diversas piezas el *set point* para el posicionamiento será variable en el tiempo también por las posiciones fijas de almacenamiento.
- **Esc. Ent. Raspberry:** acondicionamiento de la señal de entrada del Raspberry para que el PID pueda procesarlo.
- **Esc. Ent. Encoder:** acondicionamiento de la señal de retroalimentación, encoder del servomotor para su procesamiento PID.
- **Esc. Sal. Driver:** escalamiento de variable manipulada PID, para salida análoga que será referencia de velocidad del Driver del servomotor.

d) Diagrama de flujo

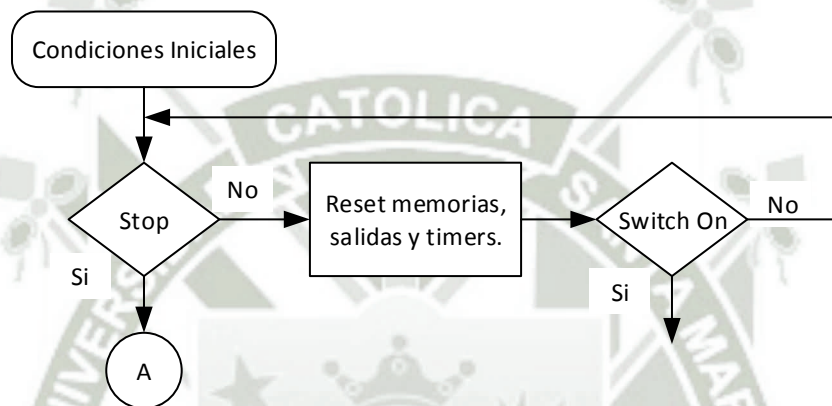
Estando definida la estructura del programa se procede a analizar la interacción entre sus elementos para definir el diagrama de flujo que guiará el proceso de implementación del código fuente con el software Step7 v5.5



**Figura 106:** Diagrama principal.

*Fuente: Elaboración propia.*

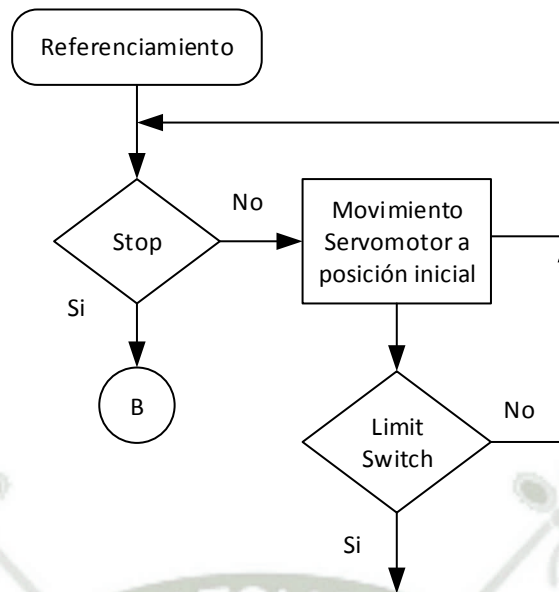
Cuando se encuentre activo el “Programa Principal” (equipo encendido) se evalúa si se encuentra activado “Stop” (Parada de emergencia) de estarlo el sistema queda en un bucle hasta que se libere esta condición, inmediatamente entonces se inicia subrutina “Condiciones Iniciales” cuando esta se cumple se inicia “Referenciamiento” y posteriormente “Clasificación”.



**Figura 107:** Subrutina de condiciones iniciales.

*Fuente: Elaboración propia.*

En “Condiciones Iniciales” se monitorea el estado de stop, de no estar activo se carga el valor 0 a memorias y salidas que puedan tener valores remanentes, esto se realizara en un solo ciclo de scan del PLC ya que son acciones en paralelo y sin condición inicial, se verificara también estado del Switch principal del equipo, de estar activado se pasa a la siguiente subrutina, si Stop es activado en cualquier instante se realiza un salto hasta el punto A en el programa principal.



**Figura 108:** Subrutina de referenciamiento.

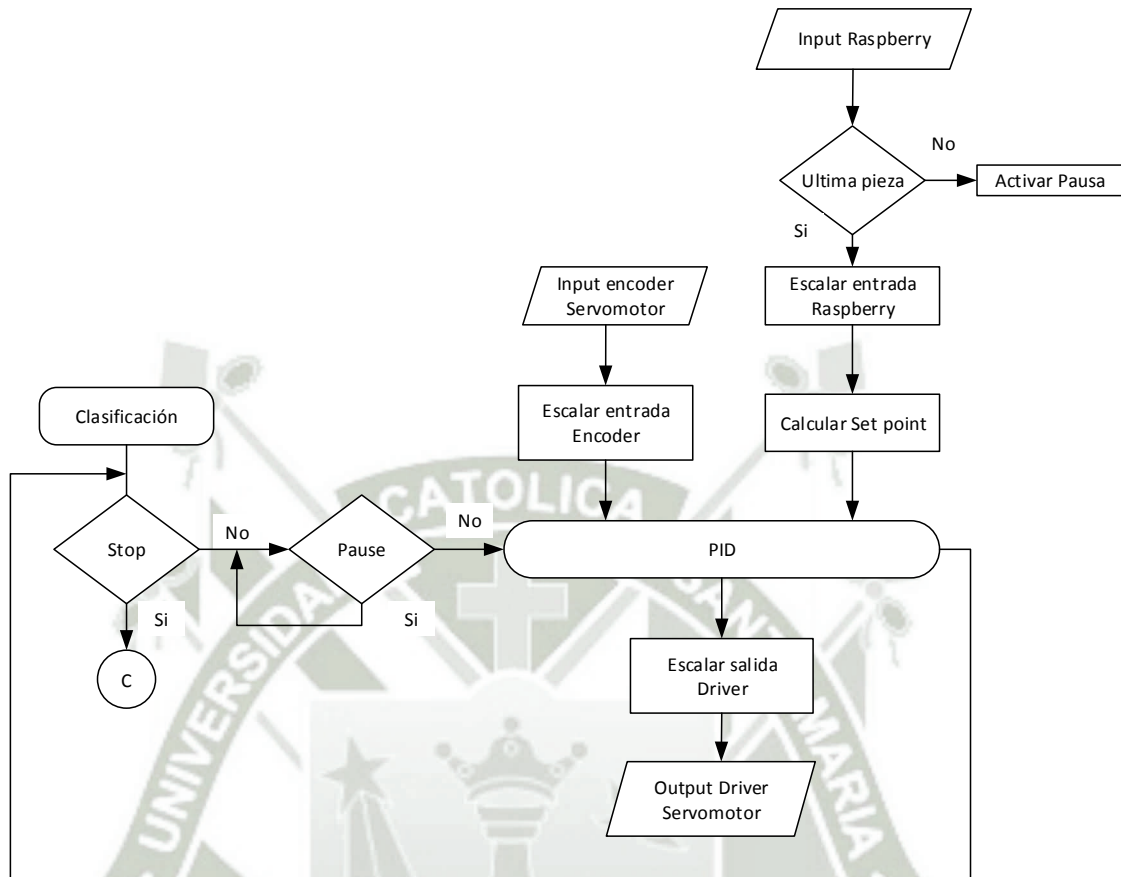
*Fuente: Elaboración propia.*

Si “Stop” se encuentra activo se realiza el salto hasta el punto A en el Programa Principal, al levantar esta condición necesariamente primero se ingresa a Condiciones Iniciales antes de volver a “Referenciamiento”. En esta subrutina se realizara en paralelo la evaluación del estado del limit switch (posición cero – lado izq.) y la carga de un valor constante en la salida de referencia de velocidad del Driver.

\*Debido a la ubicación física de los elementos luego de referenciado el carro del eje deberá ubicarse en la parte derecha para que se puedan colocar las piezas en la zona de captura de imagen.

Se evalúa la condición de Stop al igual que en las anteriores subrutinas, inmediatamente después se evalúa activación de pausa que funciona de forma semejante al stop pero de manera local, las I/O de: Raspberry (posición de pieza), encoder de servomotor (posición actual del carro – electroimán) y salida driver (ref. de velocidad para el driver) son parámetros del bloque de control PID, existen también bloques donde estos datos se escalan y uniformizan para que puedan ser utilizados.

Cuando no existan más piezas de trabajo el Raspberry enviara un bit particular que identifique esta situación lo cual pondrá automáticamente el sistema en pausa.



**Figura 109:** Subrutina de clasificación.

*Fuente: Elaboración propia.*

#### e) Implementación del programa

Teniendo en cuenta los diagramas de flujo elaborados se inicia programación en el proyecto previamente creado donde ya se configuro el hardware que se utilizó. En Programa S7 – Bloques, insertar los siguientes objetos:

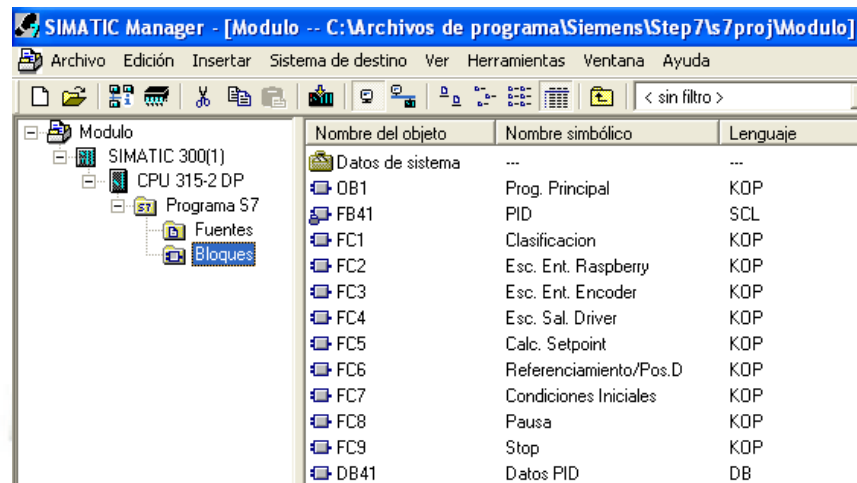


Figura 110: Bloques utilizados en el programa.

Fuente: Step 7.

- **OB1 – Programa Principal:** OB1 es el bloque de organización de Step7 desde aquí se realiza el llamado a las demás funciones.

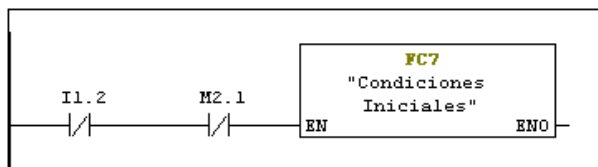
Para fines de este documento no se incluyen tags, símbolos y/o comentarios para no extender las dimensiones de la imagen, al final de esta sección se puede encontrar la Tabla de Símbolos

OB1 : "Main Program Sweep (Cycle)"

Organizacion de Programas

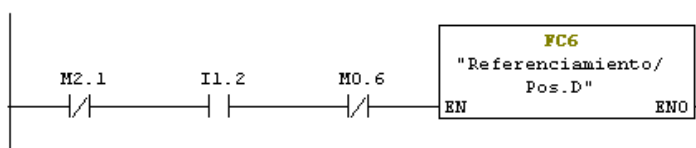
Segm. 1: Título:

Llamado a condiciones iniciales  
M2.1 Memoria Stop



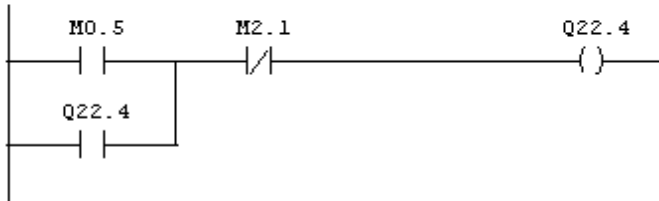
Segm. 2 : Título:

Llamado a referenciamiento



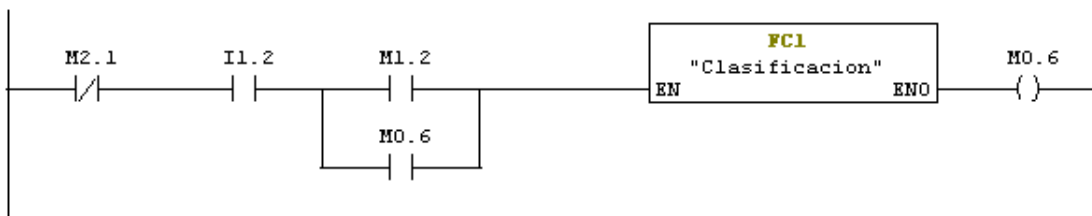
Segm. 3 : Título:

Habilitacion encoder, encoder pos\_cero:0mm (izq.)  
M0.5 se habilita en "Referenciamineto/Pos.D"



Segm. 4 : Título:

Activacion prog. principal  
M1.2 se habilita en "Referenciamiento/Pos.D"



Segm. 5 : Título:

Comentario:

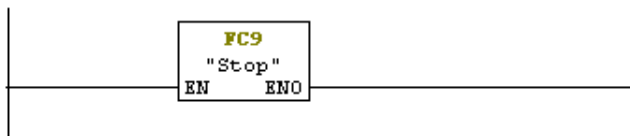


Figura 111: Programa principal.

Fuente: Elaboración propia.

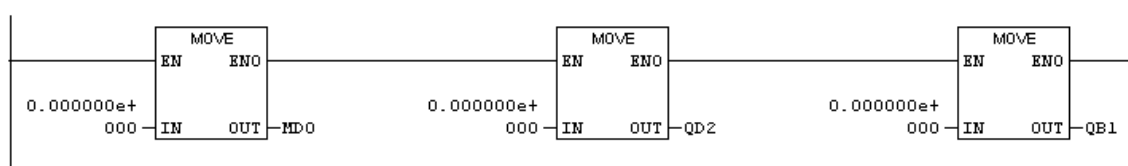
**FC7 – Condiciones Iniciales:** Carga con "0" todas las salidas y memorias para resetear valores remanentes.

FC7 : Título:

Condiciones iniciales al iniciar el sistema, stop  
Se carga con "0" todas las salidas y memorias utilizadas

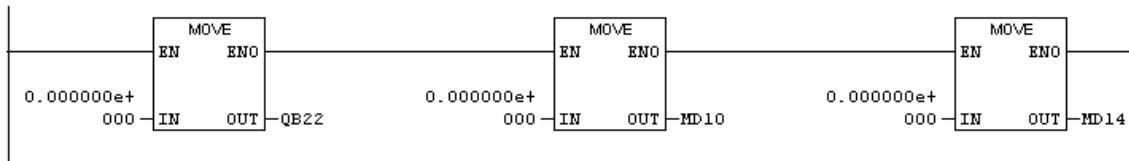
Segm. 1): Título:

Comentario:



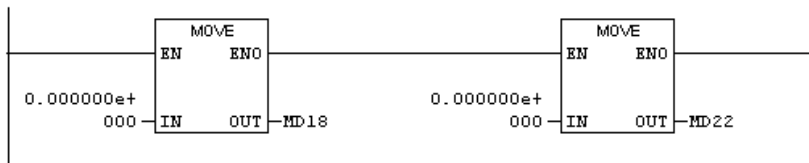
Segm. 2 : Título:

Comentario:



Segm. 3 : Título:

Comentario:



**Figura 112:** Revisión de condiciones iniciales.

*Fuente: Elaboración propia.*

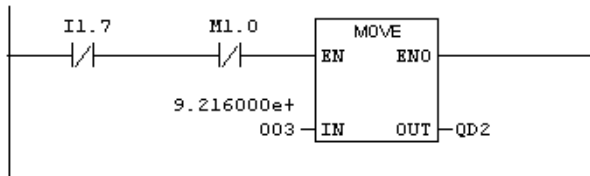
**FC6 – Referenciamiento/Pos.D:** La salida análoga del PLC que viene a ser la señal de velocidad de referencia para el Driver es cargada con un valor constante para su posicionamiento a la izquierda, al tener esta posición se inicia/resetea la cuenta del encoder, luego de referenciado se espera una habilitación manual para que el carro se posicione en el otro extremo, derecho, se realiza este procedimiento para que se puedan ubicar las piezas de trabajo en la zona de captura de imagen, luego de posicionado a la derecha se envía una señal al Raspberry para que se habilite el proceso de reconocimiento de imágenes, el sistema entra en pausa hasta que el Raspberry envía una señal luego terminar el proceso de aprendizaje y estar listo para enviar la data necesaria.

FC6 : Título:

Referenciamiento y posterior posicionamiento a la derecha para colocar piezas

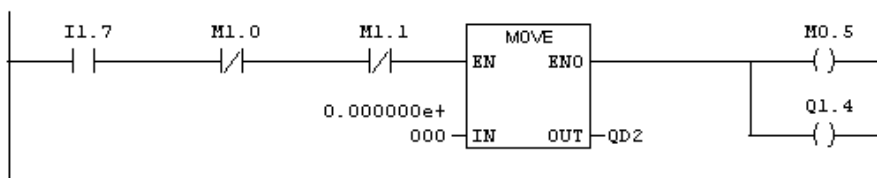
Segm. 1: Título:

Si el Limit Switch no es confirmado se carga la salida analogica con valor de velocidad positiva (hacia el lado izq)



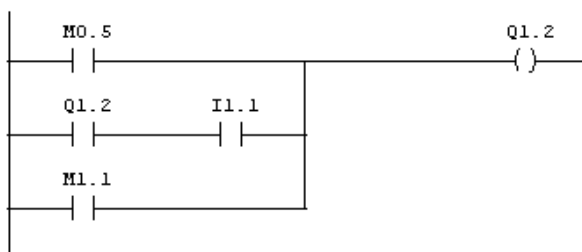
Segm. 2 : Título:

Confirmacion de referencia M0.5  
Desactiva Driver Q1.4



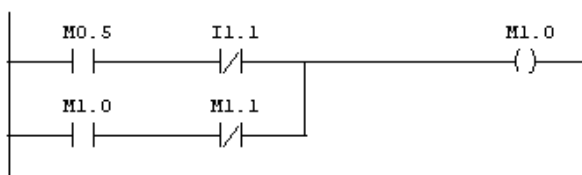
Segm. 3 : Título:

Modo Pausa activado despues de referenciamiento, luz testigo Q1.2  
Pulsador pausa I1.1 NC, usar contacto M0  
M1.1 activa luz de pausa cuando se llegue a pos. derecha



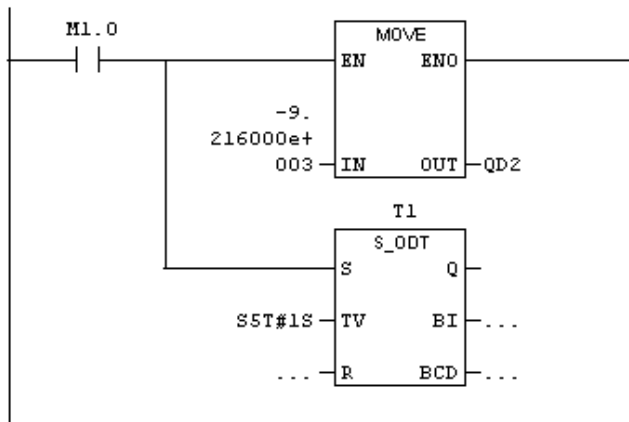
Segm. 4 : Título:

Al pulsar Pausa I1.1 se activa M1.0, confirmando liberacion para posicionamiento a la derecha



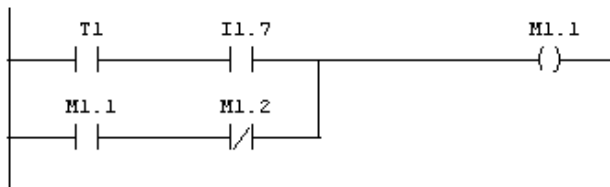
**Segm. 5 :** Título:

M1.0 carga velocidad negativa para posicionamiento a la derecha y activa tiempo para no sensar switch izquierdo



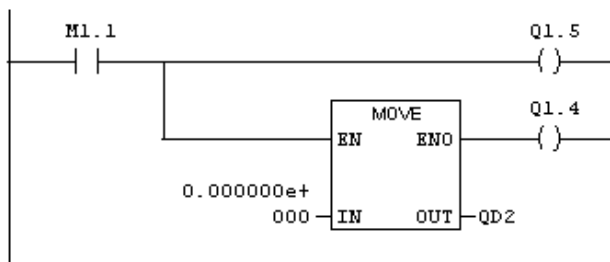
**Segm. 6 :** Título:

Confirma llegada a posicion derecha M1.1



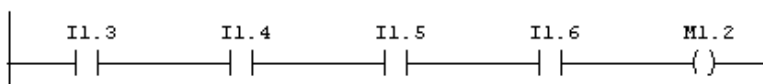
**Segm. 7 :** Título:

Envia señal a RB Q1.5, carga cero a salida QD2, desactiva driver Q1.4



**Segm. 8 :** Título:

RB puede iniciar proceso de aprendizaje luego de recibir Q1.5, confirma proceso enviando 4 bits al mismo tiempo, se activa M1.2 para salir de programa de referenciamiento e iniciar clasificacion



**Figura 113:** Referenciamiento.

*Fuente: Elaboración propia.*

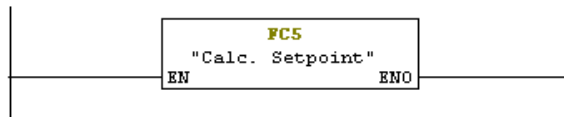
**FC1 – Clasificación:** Este programa se activa luego de que el Raspberry envía la señal de confirmación de que el proceso de aprendizaje ha concluido. Enseguida el PLC lee una señal análoga que indica la distancia desde la posición izquierda (referencia) de la primera pieza de trabajo reconocida y también se recibe un bit que indica la posición a la que esta pieza debe almacenarse. Este programa a su vez llama a otros en los que las señales son adaptadas, se discrimina el set point actual y se realiza el control PID de posicionamiento.

FC1 : Título:

Programa de Clasificacion de Piezas

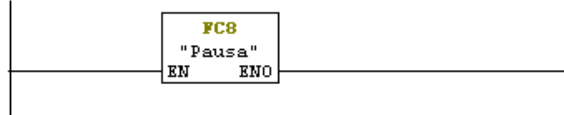
Segm. 1: Título:

Llamado a Calc. SetPoint



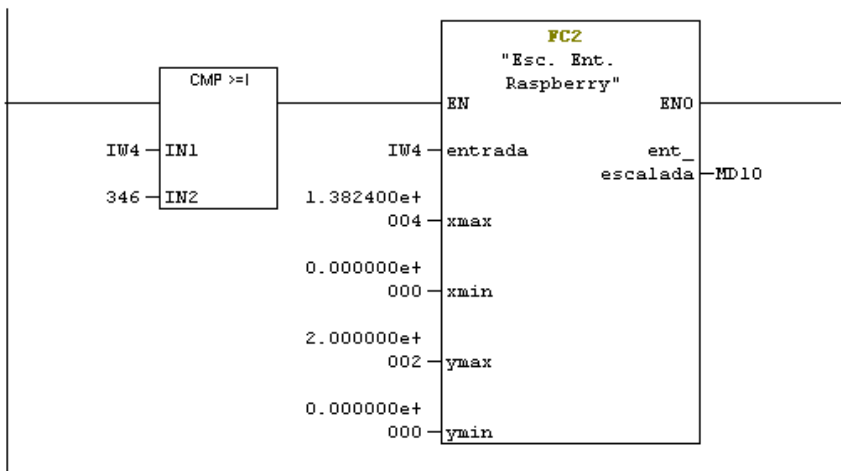
Segm. 2 : Título:

Llamado a Pausa



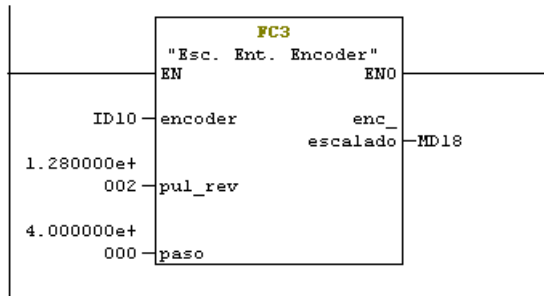
Segm. 3 : Título:

Escalamiento señal analogica de raspberry 0-5v a 0-200mm  
Filtro para activacion, el bloque solo es llamado a partir de una entrada mayor a 5mm; MD10 es la posicion que se desea alcanzar en mm.



Segm. 4 : Título:

Escala entrada encoder, se reciben 128 pulso por cada revolucion, el paso del eje es de 4mm; MD18 posicion actual en mm



Segm. 5 : Título:

Bloque PID para posicionamiento:  
MO.0, Deshabilita salida calculada de PID cargando "0" u otro valor seteable  
MD14, SetPoint designado en Calc. Setpoint con MD10 u otras posiciones fijas  
MD18, Retroalimentacion  
MD22, Salida calculada PID -100% a +100%

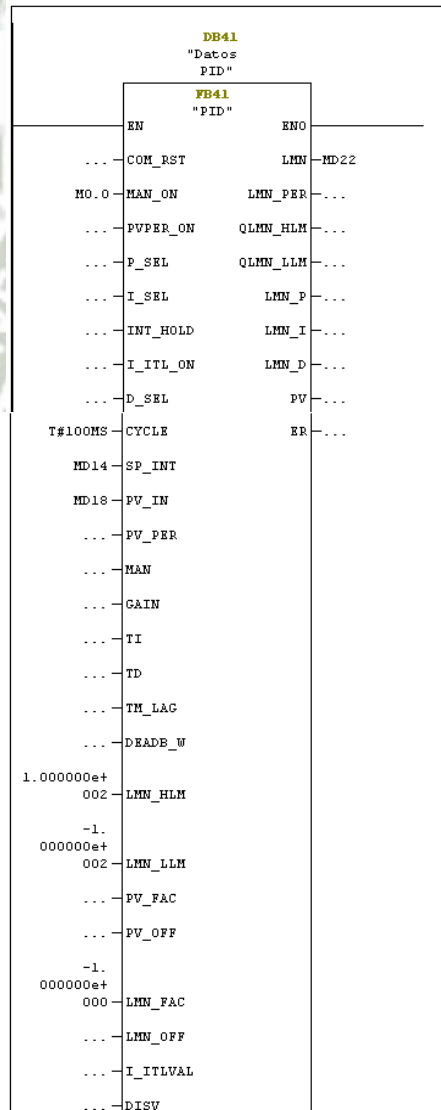
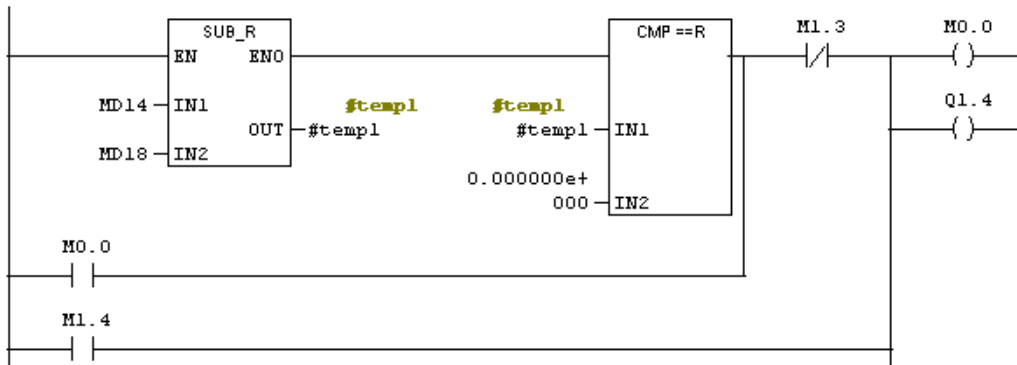


Figura 114: Clasificación.

Fuente: Elaboración propia.

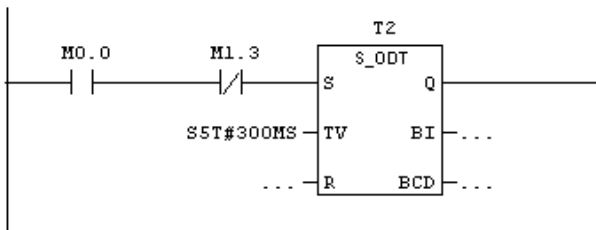
**Segm. 7 :** Título:

Modo Posicionamiento, si: SetPoint-Retroal.=0; M0.0=1; PID\_sal=0; conf. llegó a pos. de pieza  
M1.4 se activa en Modo traslado a almacen  
M1.5 se activa al presionar pausa  
M2.0 se activa cuando no hay mas piezas que procesar, RB envia I1.6



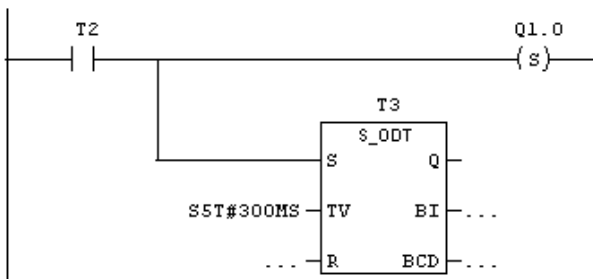
**Segm. 8 :** Título:

Al llegar a pos. se activa timer T2 de 0.3s (tiempo antes de activar electroiman)



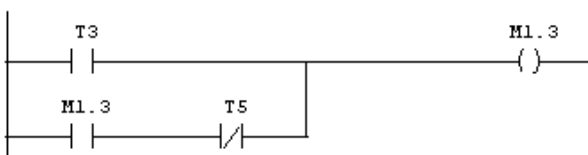
**Segm. 9 :** Título:

Luego de transcurrido T2 se setea el electroiman y se inicia otro timer T3 con 0.3s



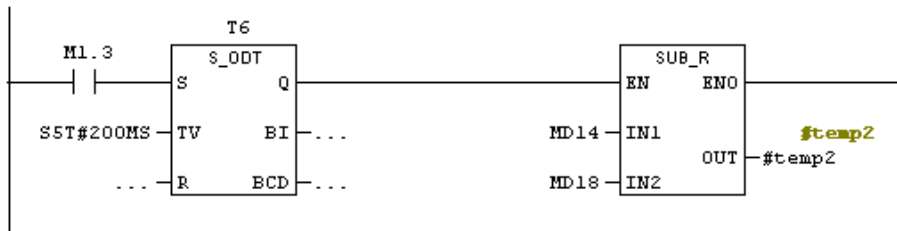
**Segm. 10 :** Título:

Transcurrido T3 se activa modo traslado a almacen M1.3  
M1.3 direcciona posicion de almacenaje en Calc.SetPoint hacia MD14



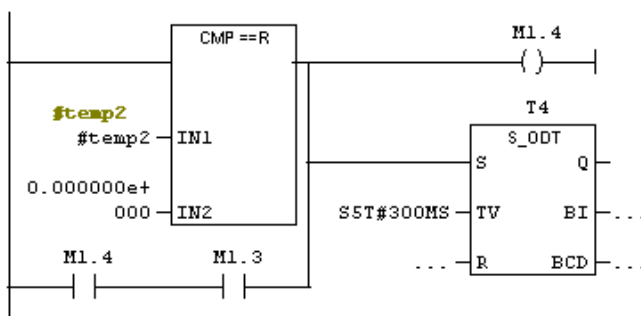
**Segm. 11:** Título:

Con Modo traslado a almacen, se activa el timer T6 con 0.2s para evitar una detencion por comparacion = 0 entre MD14 y MD18 al momento de partir



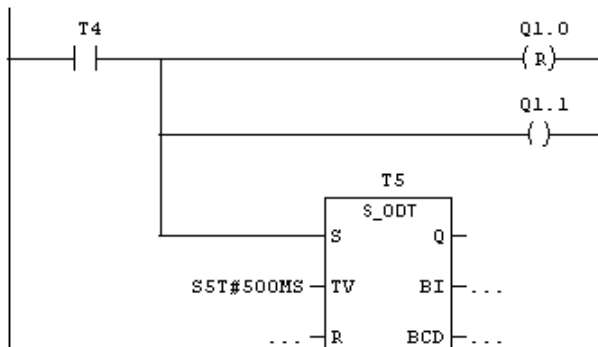
**Segm. 12:** Título:

Si en Modo Traslacion a Almacen la diferencia ente MD14 y MD18 es 0, se activa M1.4 que activara MO.0 y Q1.4 para obtener salida de PID=0 y desactivar Driver



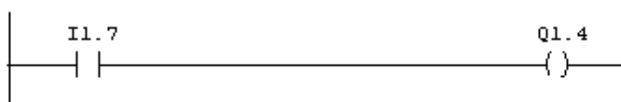
**Segm. 13:** Título:

Transcurrido T4 se resetea electroiman Q1.0 y se envia confirmacion de pieza procesada Q1.1 hacia RB, inmediatamente despues se recibe nuevos parametros de RB para siguiente pieza, se activa timer t5 con 0.5s antes de procesar sgte pieza.



**Segm. 14:** Título:

Deteccion de Limit Switch deshabilita Driver



**Figura 115:** Clasificación.

Fuente: Elaboración propia.

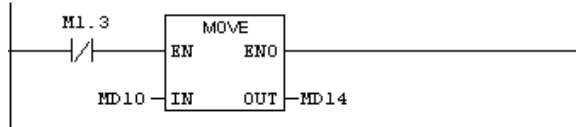
**FC5 – Calc. SetPoint:** Direcciona el Set Point para el PID con las señales de posición de las piezas de trabajo o las ubicaciones de almacén.

FC5 : Título:

Direccionamiento a MD14, posicionamiento de piezas ó posiciones de almacen

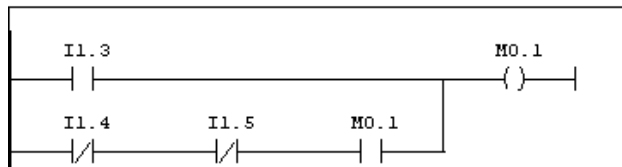
Segm. 1: Título:

Se carga a MD14, valor escalado proveniente de RB



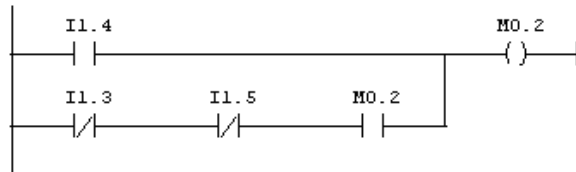
Segm. 2: Título:

Memoria habilitacion pos. 1



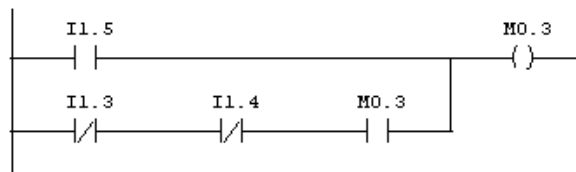
Segm. 3: Título:

Memoria habilitacion pos. 2



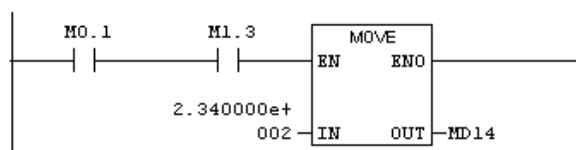
Segm. 4: Título:

Memoria habilitacion pos. 3



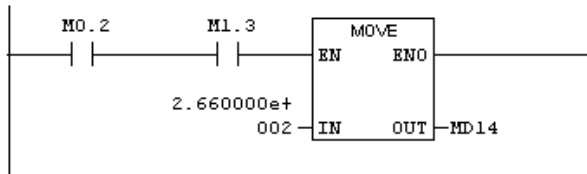
Segm. 5: Título:

Set point pos. 1



Segm. 6 : Título:

Set point pos. 2



Segm. 7 : Título:

Set point pos. 3

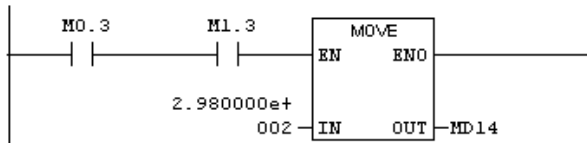


Figura 116: Cálculo de Set Point.

Fuente: Elaboración propia.

**FC2 - Esc. Ent. Raspberry:** escala señal del Raspberry 0-5V a 0-200mm, posición de piezas.

FC2 : Título:

Escala 0-5V a 0-200mm

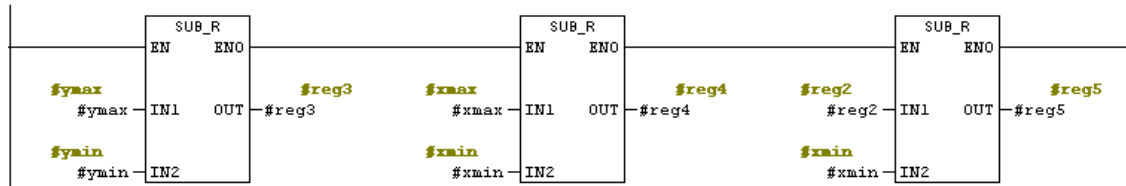
Segm. 1: Título:

Comentario:



Segm. 2 : Título:

Comentario:



Segm. 3 : Título:

Comentario:

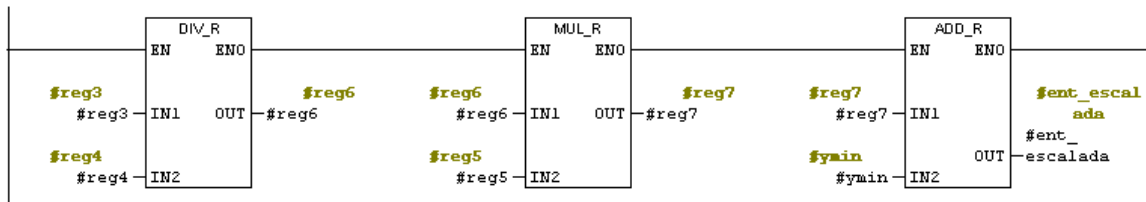


Figura 117: Escalamiento de la entrada analógica.

Fuente: Elaboración propia.

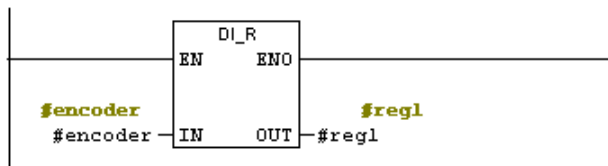
**FC3 – Esc. Ent. Encoder:** escala los pulsos del encoder del servomotor a mm, esta señal será usada como la retroalimentación del bloque PID.

FC3 : Título:

Escala pulsos del encoder a mm, para retroalimentacion del lazo de control.

Segm. 1: Título:

Comentario:



Segm. 2 : Título:

Comentario:

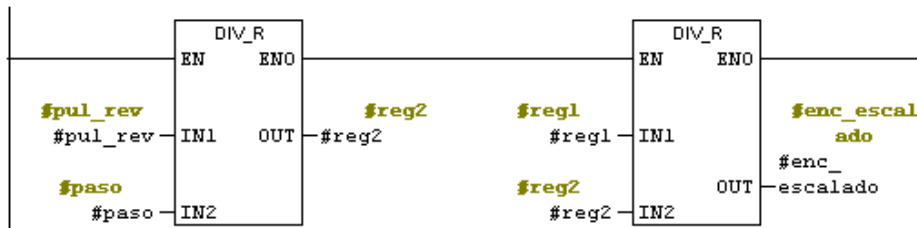


Figura 118: Escalamiento de pulsos de encoder.

Fuente: Elaboración propia.

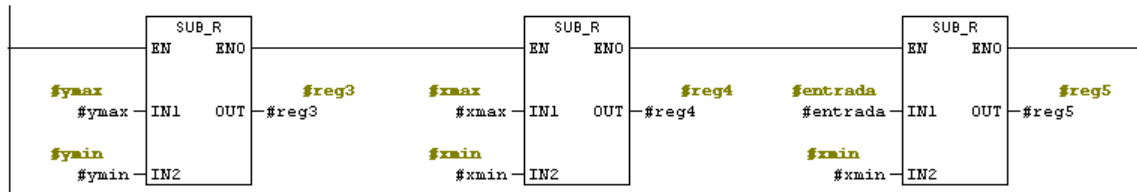
**FC4 – Esc. Sal. Driver:** escala salida del PID +/- 100% a +/-10V para salida analógica del PLC que se usa como referencia de velocidad del Driver de Servomotor.

FC4 : Título:

Escala señal de salida del PID, a +/-10V para referencia de velocidad del Driver.

Segm. 1: Título:

Comentario:



Segm. 2: Título:

Comentario:

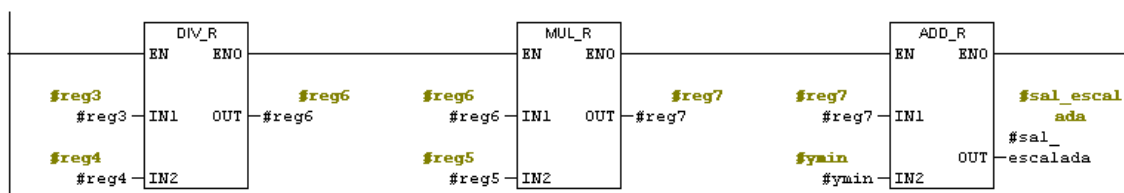


Figura 119: Señal de salida.

Fuente: Elaboración propia.

**FB41 – PID:** Bloque de Función Pre definido de Siemens: Continuous PID Controller, no se tiene acceso al código fuente de este bloque ya que se encuentra protegido, al ser llamado desde otro bloque es posible parametrizarlo.

**DB41 – Datos PID:** bloque de datos del FB41, se almacenan diversos parámetros del PID.

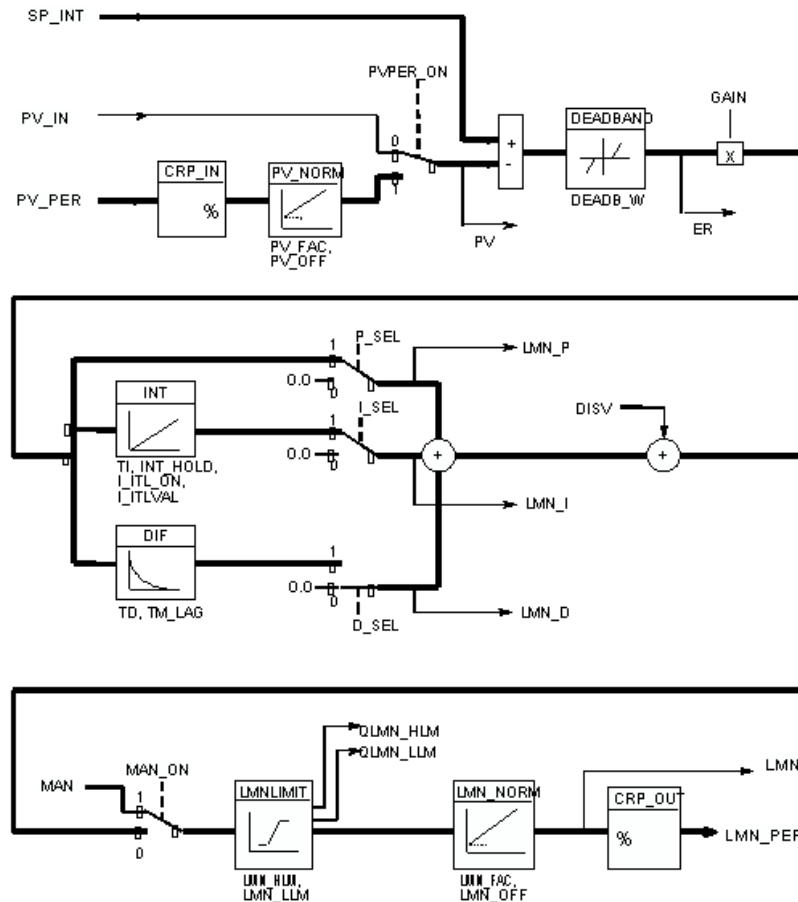


Figura 120: Lazo de control PID.

Fuente: Step 7.

DB-Param - [DB41 -- Modulo SIMATIC 300(1) CPU 315-2 DP]

Bloque de datos Edición Sistema de destino Test Ver Ventana Ayuda

	Dirección	Declaración	Nombre	Tipo	Valor inicial	Valor actual	Comentario
1	0.0	in	COM_RST	BOOL	FALSE	FALSE	complete restart
2	0.1	in	MAN_ON	BOOL	TRUE	TRUE	manual value on
3	0.2	in	PVPER_ON	BOOL	FALSE	FALSE	process variable peripherie on
4	0.3	in	P_SEL	BOOL	TRUE	TRUE	proportional action on
5	0.4	in	I_SEL	BOOL	TRUE	FALSE	integral action on
6	0.5	in	INT_HOLD	BOOL	FALSE	FALSE	integral action hold
7	0.6	in	I_ITL_ON	BOOL	FALSE	FALSE	initialization of the integral action
8	0.7	in	D_SEL	BOOL	FALSE	FALSE	derivative action on
9	2.0	in	CYCLE	TIME	T#1S	T#1S	sample time
10	6.0	in	SP_INT	REAL	0.000000e...	0.000000e+000	internal setpoint
11	10.0	in	PV_IN	REAL	0.000000e...	0.000000e+000	process variable in
12	14.0	in	PV_PER	WORD	W#16#0	W#16#0	process variable peripherie
13	16.0	in	MAN	REAL	0.000000e...	0.000000e+000	manual value
14	20.0	in	GAIN	REAL	2.000000e...	2.000000e+000	proportional gain
15	24.0	in	TI	TIME	T#20S	T#20S	reset time
16	28.0	in	TD	TIME	T#10S	T#10S	derivative time
17	32.0	in	TM_LAG	TIME	T#2S	T#2S	time lag of the derivative action
18	36.0	in	DEADB_WV	REAL	0.000000e...	0.000000e+000	dead band width
19	40.0	in	LMN_HLM	REAL	1.000000e...	1.000000e+002	manipulated value high limit
20	44.0	in	LMN_LLM	REAL	0.000000e...	0.000000e+000	manipulated value low limit
21	48.0	in	PV_FAC	REAL	1.000000e...	1.000000e+000	process variable factor
22	52.0	in	PV_OFF	REAL	0.000000e...	0.000000e+000	process variable offset
23	56.0	in	LMN_FAC	REAL	1.000000e...	1.000000e+000	manipulated value factor
24	60.0	in	LMN_OFF	REAL	0.000000e...	0.000000e+000	manipulated value offset
25	64.0	in	I_ITLVAL	REAL	0.000000e...	0.000000e+000	initialization value of the integral action
26	68.0	in	DISV	REAL	0.000000e...	0.000000e+000	disturbance variable
27	72.0	out	LMN	REAL	0.000000e...	0.000000e+000	manipulated value

Figura 121: Señales asociadas al bloque PID.

Fuente: Step 7.

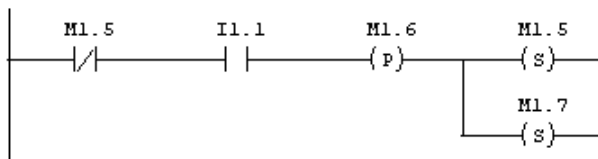
**FC8 – Pausa:** al dar un pulso al botón de pausa se activa este modo, deshabilitando todo movimiento hasta que se vuelva a presionar, entonces el proceso continúa de forma normal.

FC8 : Título:

Activacion de Pausa en Modo Clasificacion

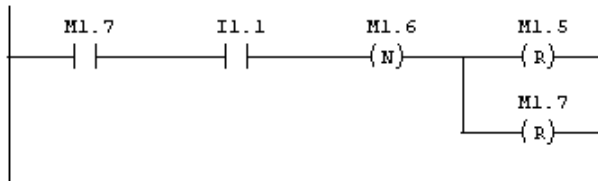
Segm. 1: Título:

Activacion pausa, pulso I1.1



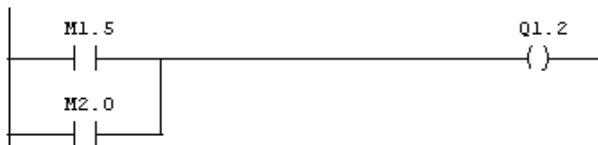
Segm. 2 : Título:

Desactivacion pausa, pulso I1.1



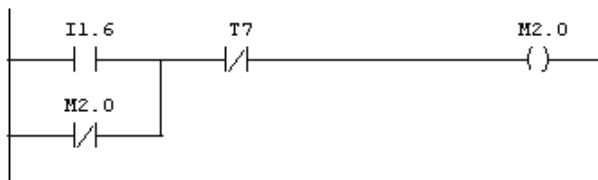
Segm. 3 : Título:

Activacion luz testigo Q1.2



Segm. 4 : Título:

Activacion pausa cuando no hay mas piezas para procesar, RB envia I1.6



Segm. 5 : Título:

Luego de colocar nuevas piezas y procesarlas el RB envia confirmacion de listos 4 bits, se inicia un timer de 1s antes de levantar la pausa, tiempo en el que se recibe la primera ubicacion del nuevo arreglo de piezas

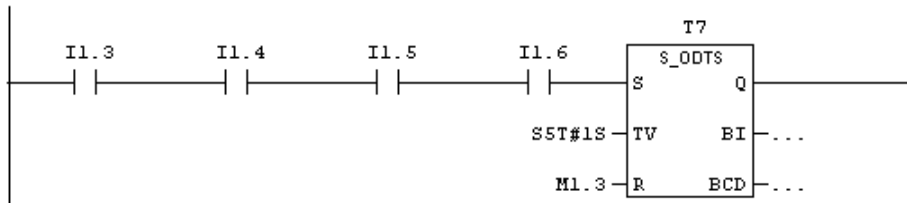


Figura 122: Programa de Pausa.

Fuente: Elaboración propia.

**FC9 – Stop:** detiene todo movimiento y se enclava hasta que se restablezcan condiciones iniciales.

FC9 : Título:

Programa Stop

Segm. 1: Título:

Una vez pulsado Stop se enclava memoria M2.1, esta solo se desenclava si se libera el pulsador stop y posteriormenete se regresa a condiciones iniciales (Switch I1.2 Off). Se envia señal a Raspberry.

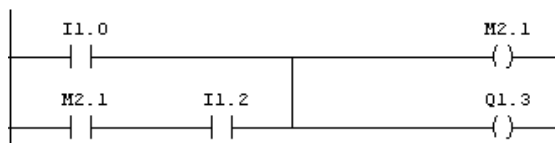


Figura 123: Programación del Stop.

Fuente: Elaboración propia.

Finalmente la estructura del programa registrada en las instancias del Step7:

Bloque (símbolo), DB de instancia (símbolo)	Datos lo	Lenguaje	Punto de aplicación
Programa S7			
OB1 (Prog. Principal) [máximo: 122]	[22]		
FC7 (Condiciones Iniciales)	[22]	KOP	Seg 1
FC6 (Referenciamiento/Pos.D)	[24]	KOP	Seg 2
FC1 (Clasificación)	[48]	KOP	Seg 4
FC5 (Calc. Setpoint)	[48]	KOP	Seg 1
FC8 (Pausa)	[48]	KOP	Seg 2
FC2 (Esc. Ent. Raspberry)	[76]	KOP	Seg 3
FC3 (Esc. Ent. Encoder)	[56]	KOP	Seg 4
FB41 (PID), DB41 (Datos PID)	[122]	KOP	Seg 5
FC4 (Esc. Sal. Driver)	[84]	KOP	Seg 6
FC9 (Stop)	[22]	KOP	Seg 5

Figura 124: Estructura del programa.

Fuente: Elaboración propia.

Seleccionar todos los programas y descargarlos al PLC con la misma configuración MPI que se utilizó para descargar el hardware, realizar pruebas de conectividad online entre PC – PLC (click en lentes de barra de herramientas) verificar en la barra de estado, inferior derecha, barra verde en movimiento sobre la palabra RUN.

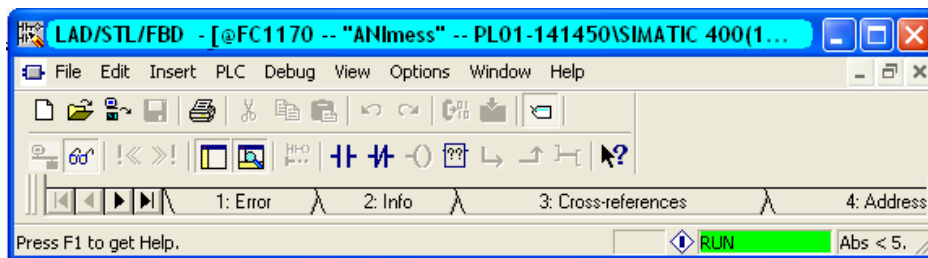


Figura 125: Ventana de estado.

Fuente: Step 7.

#### 4.5.2 Parametrización driver servomotor

Los elementos necesarios para la motricidad del proyecto fueron seleccionados del fabricante alemán, Sew EuroDrive, siendo los siguientes:

- Driver Movitrac LTPB0008-2B1-4-00 (0.75KW – 220VAC)
- Servomodule LTX-H1A
- Servomotor CMP40M/KY/AK0H/SM1 (4500 RPM -220VAC)
- Cable Mot. conector SM11/71/81
- Cable Enc. Sub-D 15 POL – 12 POL ASTA, con He
- Line Choke ND 020-013

Este driver puede funcionar en forma convencional variando la frecuencia para controlar la velocidad en motores sin un lazo de retroalimentación de su posición. Es posible sin embargo mediante el servo módulo LTX conectar el cable de encoder de diferentes Servomotores (Norma: EN 61984) y de esta forma ampliar las funciones del driver con lazos de control (dependiendo del modo de funcionamiento y/o interacción) para el uso de servomotores en diversas aplicaciones. El line Choke se usa para filtrar ruidos hacia las líneas de alimentación.

a) Definición de modo de funcionamiento/control

Los modos de funcionamiento y/o la forma en que el Driver se conecta a niveles de automatización superiores son los siguientes:

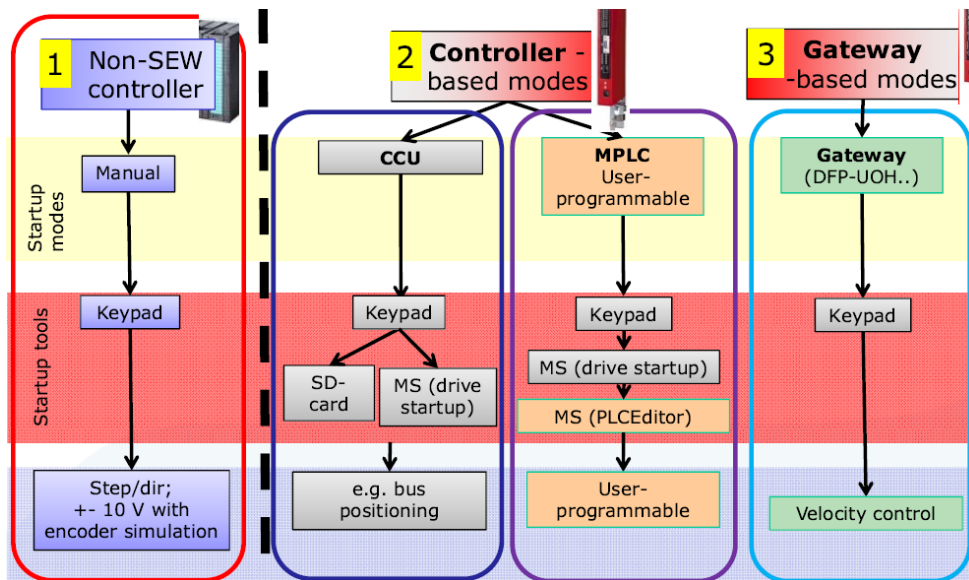
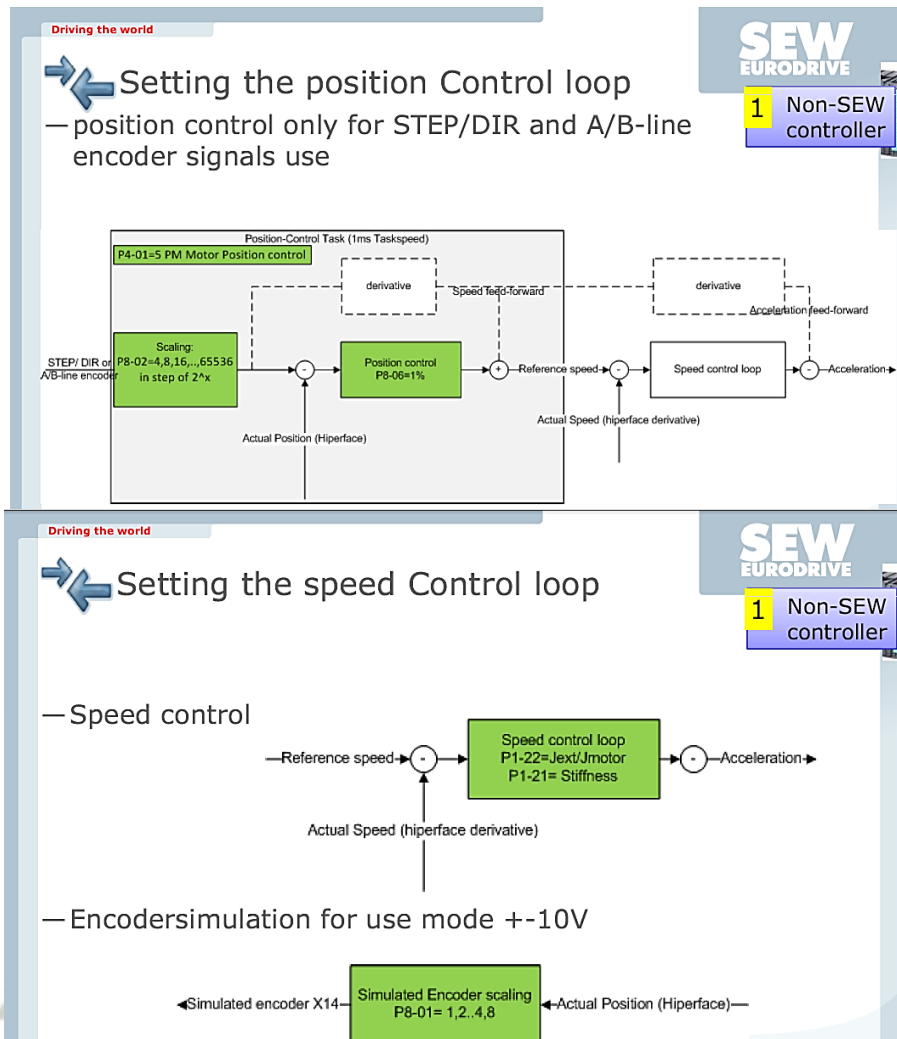


Figura 126: Modos de control del driver.

Fuente: Technical training Smart Servo Package.

El “modo 1” es a través de un controlador externo, PLC. En el “modo 2” se utilizan controladores dedicados del propio fabricante de esta forma se obtienen librerías y rutinas específicas para diversas aplicaciones. El “modo 3” implica el uso de Gateway para conectar uno o más drivers a la red, pudiendo controlar la velocidad de los mismos.

En el presente proyecto se trabajara en el “modo 1” ya que se cuenta con un controlador - PLC de marca VIPA, dentro del “modo 1” es posible implementar lazos de control de velocidad y de posición.



**Figura 127:** Lazos de control por posición y velocidad.

Fuente: Technical training Smart Servo Package.

El objetivo del proyecto es lograr posicionamiento, sin embargo el lazo de control de posición disponible en este modo es del tipo maestro – esclavo, es decir se conecta al Driver como Set Point la señal de un encoder (Step/Dir ó A-B Phase) que da la posición de otro sistema (Maestro) el Driver deriva las posiciones para tener un lazo de velocidad consiguiendo que el Servomotor (Esclavo) se mueva en función de la señal de posición recibida.

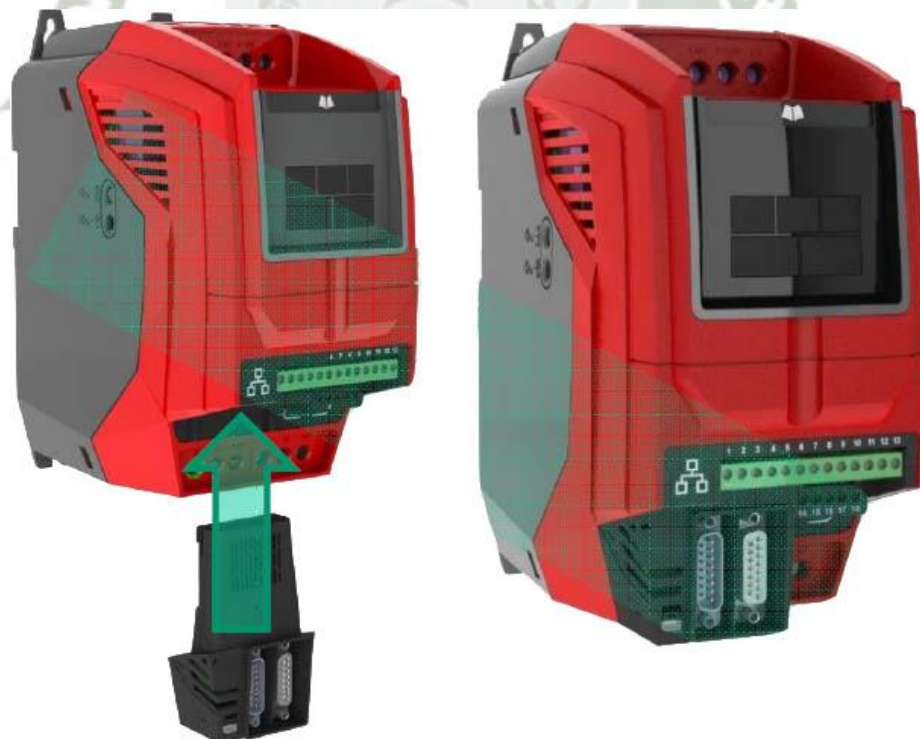
En los lazos de control de velocidad existen dos opciones, en la primera se ingresa una velocidad de referencia, el Driver deriva la posición (señal del encoder) en

el tiempo y compara ambas señales manteniendo la diferencia en 0. En la segunda, la señal física de encoder del servomotor ingresa al Driver allí es monitoreada y escalada, una señal de encoder simulado (A/B Phase) sale del Driver por el conector X14 hacia otro proceso, en esta opción se habilita entrada analógica en el Driver para ingreso de velocidad de referencia +/- 10VDC. Esta última opción es la que se utilizara ya que el PLC VIPA puede realizar externamente el lazo de control de posición utilizando como retroalimentación la señal de encoder simulado del Driver, además el PLC puede entregar un voltaje entre +/- 10VDC en función de la posición, el Set Point será establecido en función al proceso.

#### *b) Conexión de elementos Sew*

Se procede a realizar la conexión del Servomodule LTX, los cables entre Servomotor - Driver, cable de señal para procesos X14, Line Choke y cableado de acometida hacia el Driver.

- **Insertar Servomodule LTX**



**Figura 128:** Servomotor más Tarjeta LTX.

*Fuente: SEW Eurodrive.*

- Conectar cables de Encoder y Potencia

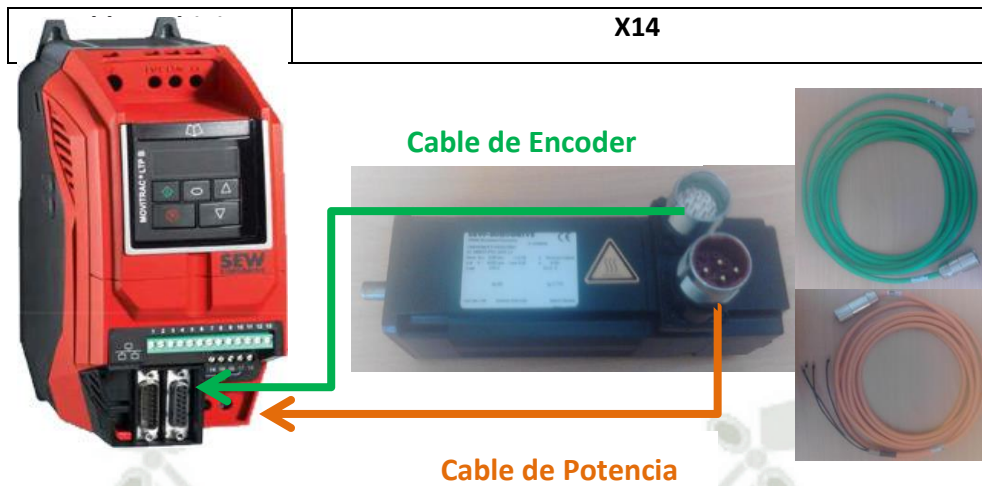


Figura 129: Diagrama de conexión Driver- Servomotor.

Fuente: SEW Eurodrive.

- Conectar cable X14

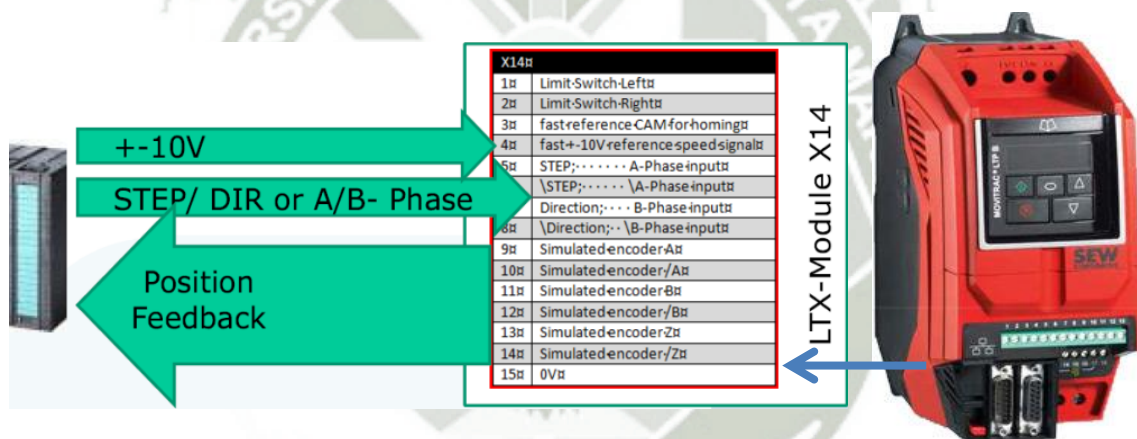


Figura 130: Conexión del conector X14.

Fuente: Elaboración propia.

Tabla 41:

Colores	Acción	Función
Blanco/Celeste	1	Limit Switch Izquierda
Blanco/Naranja	2	Limit Switch Derecha
Verde	3	Referencia Rápida Homing
Blanco/Verde	4	Señal de +-10V, referencia de velocidad
Rojo/rosado	5	Paso; Fase A
Negro	6	\Paso; Fase \A
Blanco/amarillo	7	Dirección; B
Marrón	8	\Dirección; \B
Blanco/negro	9	Simulación de encoder A
Blanco	10	Simulación de encoder \A
Anaranjado	11	Simulación de encoder B
Amarillo	12	Simulación de encoder \B
Azul	13	Simulación de encoder Z
Rojo	14	Simulación de encoder \Z
Plomo	15	0V

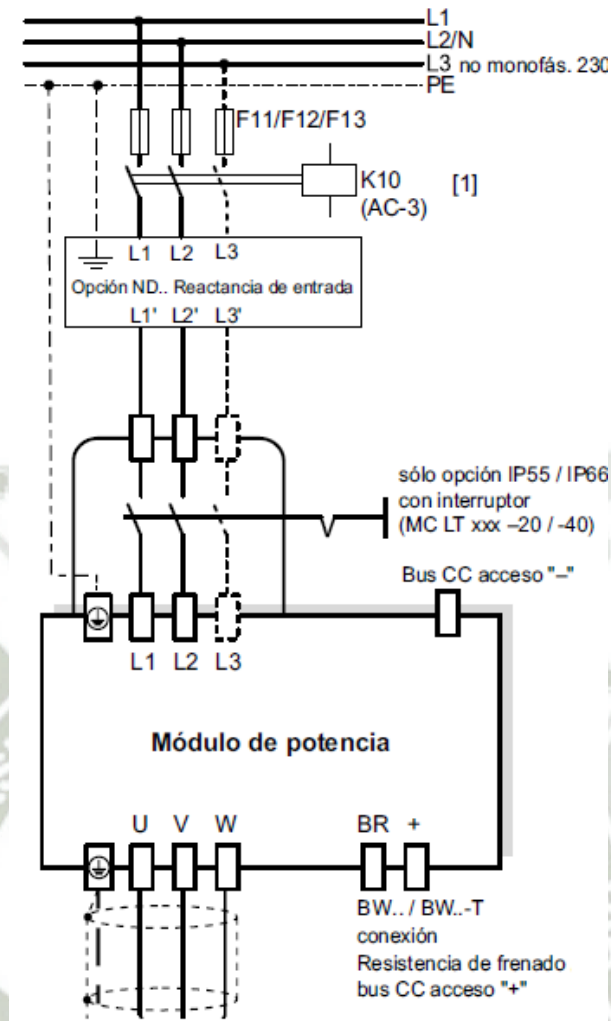
Conexión de cables X14.

- Conectar Line Choke y acometida



Figura 131:Line Choke, SEW.

Fuente: SEW Eurodrive.



**Figura 132:** Acometida eléctrica.

Fuente: Anexo a las instrucciones de funcionamiento, Sew.

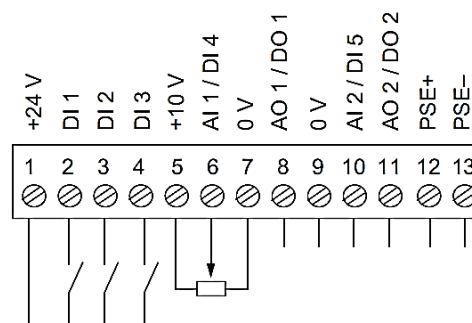
*c) Puesta en marcha / parametrización*

Luego de encender por primera vez el Driver del Servomotor es necesario parametrizar los valores de funcionamiento, en el modo de controlador externo esta parametrización se realiza de forma manual a través del teclado del Driver.

**Tabla 42:** Bloque de borneras del driver.

N° de borna	Señal	Conexión	Descripción
1	+24 V	Salida +24 V tensión de referencia	Ref. para activación de DI1 - DI3 (máx. 100 mA)
2	DI 1	Entrada binaria 1	Lógica positiva Rango de tensión de entrada "Lógico 1": 8 – 30 V <sub>CC</sub> Rango de tensión de entrada "Lógico 0": 0 – 2 V <sub>CC</sub> Compatible con demanda de PLC si está conectada 0 V a borna 7 o 9.
3	DI 2	Entrada binaria 2	
4	DI 3	Entrada binaria 3 / contacto de sonda	
5	+10 V	Salida +10 V tensión de referencia	10 V ref. para entrada analógica (alimentación de pot. +, 10mA máx., 1 kΩ mín.)
6	AI 1 / DI 4	Entrada analógica (12 bit) Entrada binaria 4	0 – 10 V, 0 – 20 mA, 4 – 20 mA Rango de tensión de entrada "Lógico 1": 8 – 30 V <sub>CC</sub>
7	0 V	Potencial de referencia 0 V	Potencial de referencia (alimentación de pot. –) 0 V
8	AO 1 / DO 1	Salida analógica (10 bit) Salida binaria 1	0 – 10 V, 20 mA analógica 24 V, 20 mA digital
9	0 V	Potencial de referencia 0 V	Potencial de referencia 0 V
10	AI 2 / DI 5	Entrada analógica 2 (12 bit) Entrada binaria 5	0 – 10 V, 0 – 20 mA, 4 – 20 mA Rango de tensión de entrada "Lógico 1": 8 – 30 V <sub>CC</sub>
11	AO 2 / DO 2	Salida analógica 2 (10 bit) Salida binaria 2	0 – 10 V, 20 mA analógica 24 V, 20 mA digital
12	PSE+	Habilitación de etapa final	+24 V debe estar conectada con PSE+
13	PSE–		GND debe estar conectada con PSE–

Los parámetros básicos a ingresar son los datos de placa del motor P1-01 a P1-10, velocidades máxima y mínima P1-01 y P1-02, rampas de aceleración y desaceleración P1-03 y P1-04. En P1-16 se puede elegir de una lista motores Sew en caso tener el Servomodulo al leer el encoder este parámetro se establece reconociendo automáticamente el Servomotor conectado. Parámetros/conexiones adicionales a tener en cuenta:



**Figura 133:** Borneras de driver.

Fuente: Anexo a las instrucciones de funcionamiento, Sew.

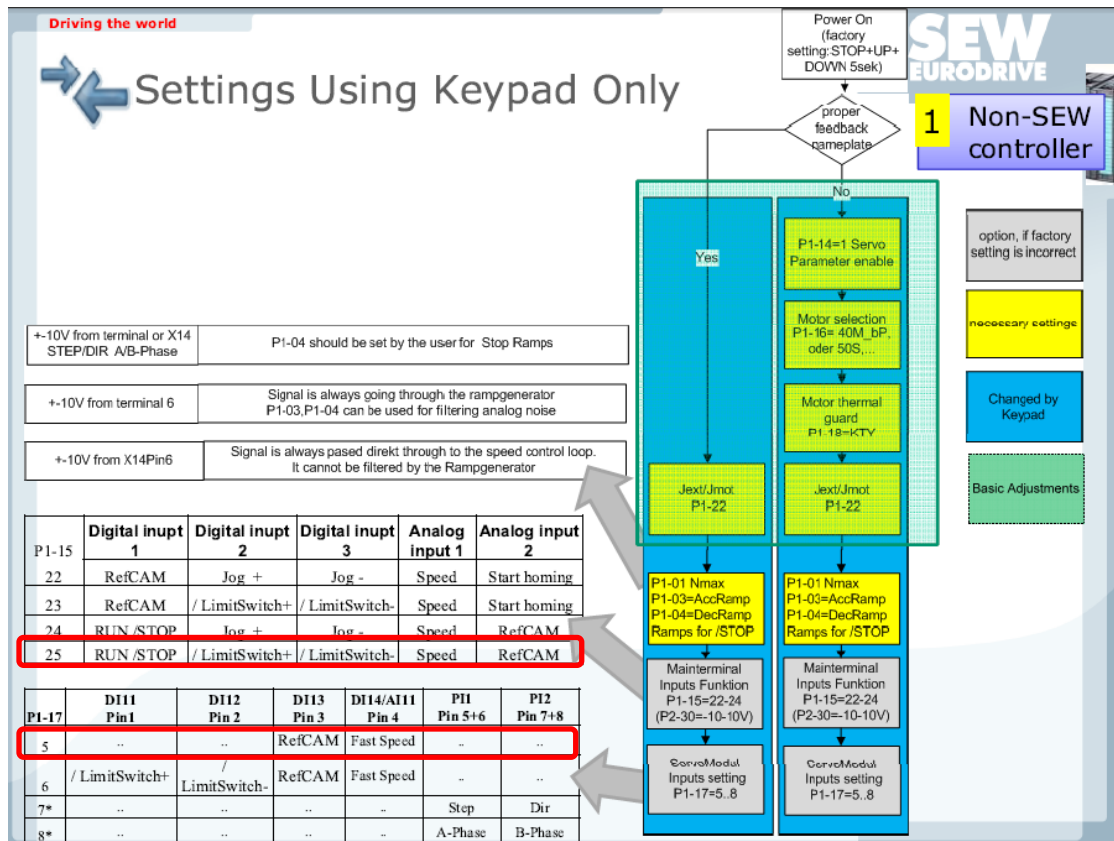


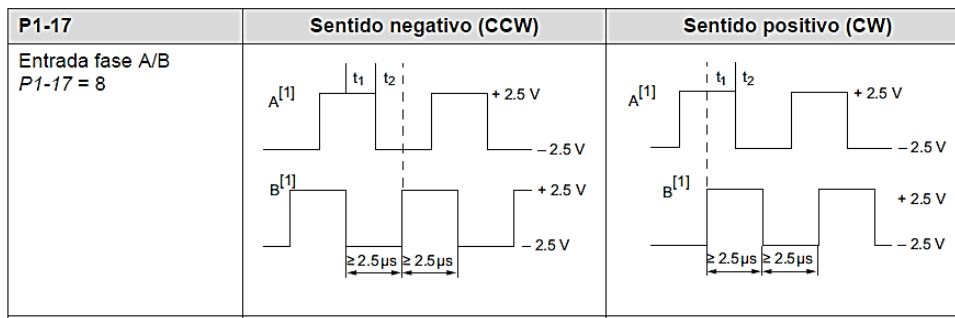
Figura 134: Configuración del modo 1.

Fuente: Technical training Smart Servo Package.

#### d) Conexión de señales con PLC

- Conectar salida analógica del PLC con los pines 4 (speed reference) y 15 (0V) del X14.
- Conectar pines 9 y 11 (encoder simulado A/B) del X14 hacia la entrada rápida del PLC.

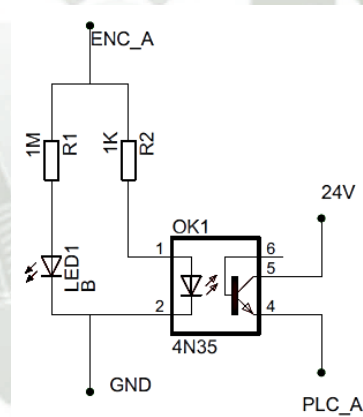
Para hacer posible esta última etapa es necesario considerar que la salida del simulador de encoder por los pines x14, tienen una salida de -2.5 V a 2.5 V. Pero las entradas digitales del PLC, en esta ocasión, las entradas rápidas del módulo 238 necesitan un mínimo de 20.4 V para tener un "1" lógico. Como se tiene tan solo 2.5 V, esta señal es imperceptible para el modulo, se debe amplificar para poder tener retroalimentación del movimiento del servomotor, por este motivo se elaboró una tarjeta que permite leer adecuadamente esta información.



**Figura 135:** Secuencia de señales de encoder.

Fuente: Anexo a las instrucciones de funcionamiento, Sew.

El circuito elaborado tiene como función tomar estos 2.5V, encender un Led y activar un fototransistor, este último es el elemento interno de un Optoacoplador, lo que nos permite conectar 24 VDC del otro lado, esta entrada es conectada al PLC y así logramos cerrar la retroalimentación del encoder hacia el controlador. De esta manera es posible realizar el control. El esquema completo del circuito se adjunta en los anexos.



**Figura 136:** Circuito de conexión del encoder mediante optoacoplador 4n35

Fuente: Elaboración propia.



**Figura 137:** Circuito de encoder ensamblado.

Fuente: Elaboración propia.

# Capítulo 5

## Pruebas y resultados

Finamente ya con todo el sistema ensamblado según requerimientos industriales, tenemos el modulo completo, los detalles se adjuntan en los planos eléctricos, anexos a este documento.



**Figura 138:**Módulo ensamblado.

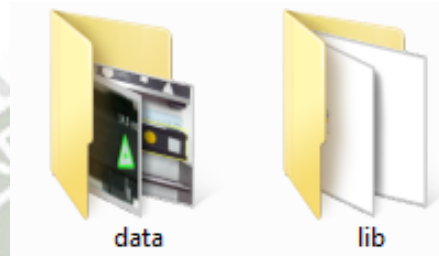
*Fuente: Elaboración propia.*

Se realizaron pruebas de movimiento del eje y procesamiento de las imágenes obteniendo óptimos resultados. La velocidad del motor la tenemos establecida en un valor entre 750 y 1000 RPM (de 4500 RPM que tiene el motor), lo cual le permite un movimiento de 2.8 m/s a 4 m/s, dentro de este rango el movimiento no presenta problemas. No es recomendable utilizarlo a mayor velocidad, pues el recorrido es corto y podrían presentarse problemas por la aceleración, desaceleración e inercia que se produce con la masa.

Para el procesamiento de imágenes se elaboró un interfaz gráfico basados en Pygame, se decidió utilizar esta plataforma, pues viene pre instalada en el Raspberry

Pi. Básicamente esta herramienta nos permite traslapar cuadros y mostrarlos en una ventana, con una adecuada programación, se implementó estas librerías con el procesamiento de imágenes y la comunicación con los puertos físicos GPIO. A continuación se explica el procedimiento de programación del interfaz, los detalles están en el mismo programa, que se encuentran adjuntos al documento.

El primero paso es el crear una carpeta donde se almacenaran todos los archivos. La carpeta se encuentra guardada con el nombre de “Prueba”, dentro de ella se tienen estas dos carpetas, “data” y “lib”.























**Figura 139:**Archivos en el Raspberry Pi.

*Fuente: Elaboración propia.*

Fuera de estas dos carpetas se tiene el programa “Sistema\_de\_clasificacion.py”, este programa es el único en utilizarse, a través de él se accede a los programas siguientes. Su contenido es el siguiente:

```
from lib import main  
main.main()
```

En la carpeta de **lib** se tienen los siguientes programas:

Nombre	Fecha de modifica...	Tipo	Tamaño
 _init_	07/09/2008 02:19 a...	Archivo PY	0 KB
 _init_	03/11/2014 10:43 a...	Compiled Python ...	1 KB
 calibrar	10/03/2015 06:54 ...	Archivo PY	16 KB
 calibrar	10/03/2015 06:54 ...	Compiled Python ...	8 KB
 clasificacion	10/03/2015 07:01 ...	Archivo PY	47 KB
 clasificacion	10/03/2015 07:01 ...	Compiled Python ...	18 KB
 cutsscenes	10/03/2015 01:05 ...	Archivo PY	6 KB
 cutsscenes	10/03/2015 01:05 ...	Compiled Python ...	3 KB
 data	10/03/2015 09:33 a...	Archivo PY	13 KB
 data	10/03/2015 09:33 a...	Compiled Python ...	10 KB
 entrenamiento	28/02/2015 11:55 ...	Archivo PY	20 KB
 entrenamiento	01/03/2015 10:26 ...	Compiled Python ...	8 KB
 ezmenu	03/02/2015 08:19 a...	Archivo PY	6 KB
 ezmenu	03/02/2015 08:20 a...	Compiled Python ...	4 KB
 main	10/03/2015 06:56 ...	Archivo PY	5 KB
 main	10/03/2015 06:57 ...	Compiled Python ...	2 KB
 menu	11/03/2015 10:39 a...	Archivo PY	14 KB
 menu	10/03/2015 06:57 ...	Compiled Python ...	6 KB
 seleccion	10/03/2015 06:58 ...	Archivo PY	19 KB
 seleccion	10/03/2015 06:58 ...	Compiled Python ...	8 KB

**Figura 140:** Subprogramas dentro de la carpeta “lib”.

*Fuente: Elaboración propia.*

El primer paso es cargar el programa “main” en el cual se establece las condiciones básicas para que el programa se desarrolle sin problemas. En este programa se inicia los puertos GPIO, la cámara y se establece las características de la ventana, la primera ventana se muestra con el siguiente comando:

```
menu.Menu(screen, vel, cap)
```

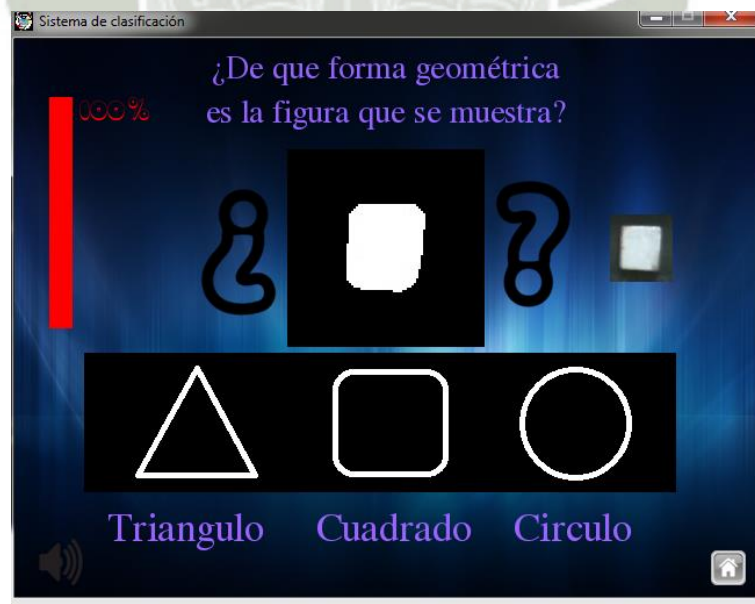
Después se abre la siguiente ventana de inicio. En esta ventana se da inicio al sistema, se tienen 4 opciones. La primera “Inicio” sirve para iniciar la clasificación, después de hacer click sobre ella se ingresa a otra ventana donde se elige como se espera clasificar las piezas. Solo se puede acceder a esta opción cuando se tienen las 3 características que se tiene a la derecha habilitadas, que el sistema tenga una base de datos al 100%, la cámara este calibrada y centrada, y finalmente que el actuador tenga el bloque móvil en posición inicial.



**Figura 141:** Ventana de Inicio.

*Fuente: Elaboración propia.*

La segunda opción accede a esta otra ventana, figura 136, esta es la opción de “aprendizaje”, aquí se le enseña al sistema que tipo de forma tiene cada “matriz”, para este sistema, lo probaremos con tres figuras estándar, cuadradas, círculo y triángulo. Este sistema se podría utilizar con cualquier otra forma geométrica.



**Figura 142:** Ventana de aprendizaje.

*Fuente: Elaboración propia.*

Al final del entrenamiento, se guarda la información adquirida de la siguiente forma:

```
filename=filepath("Memoria/generalsamples.data")
np.savetxt(filename,samples)
filename=filepath("Memoria/generalresponses.data")
np.savetxt(filename,responses)
```

Con estos comandos se generan dos archivos .txt, que contiene toda la información necesaria para poder clasificar las piezas más adelante. El archivo “generalsamples.data” contiene vectores con todos los valores, mientras que “generalresponses.data” tiene en números el tipo de figura que es. Internamente la clasificación se presenta así.

**Tabla 43:**Clasificación de piezas

<b>Forma: f=[1 2 3]</b>	<b>Tamaño: t=[1 2 3]</b>	<b>Color: c=[1 2 3]</b>
Triangulo: 1	Pequeño: 1	Rojo: 1
Cuadrado: 2	Mediano: 2	Plata: 2
Circulo: 3	Grande: 3	Amarillo: 3

Otra de las opciones que se tiene en el menú principal, es el de “calibración”. El objetivo principal es asegurarnos de que la cámara está enfocando la base donde se encuentran las piezas. Es posible que la cámara se mueva, si esto se da, el programa no podrá reconocer la base y por lo tanto se genera un error. Para este paso se recomienda que la cámara tenga a la base centrada como se muestra en la figura 143.



**Figura 143:**Imagen centrada en la pantalla.

*Fuente: Elaboración propia.*

También en el interior de esta ventana es posible variar 2 factores que necesita el sistema para procesar, el Umbral adaptativo tiene como factores los siguientes:

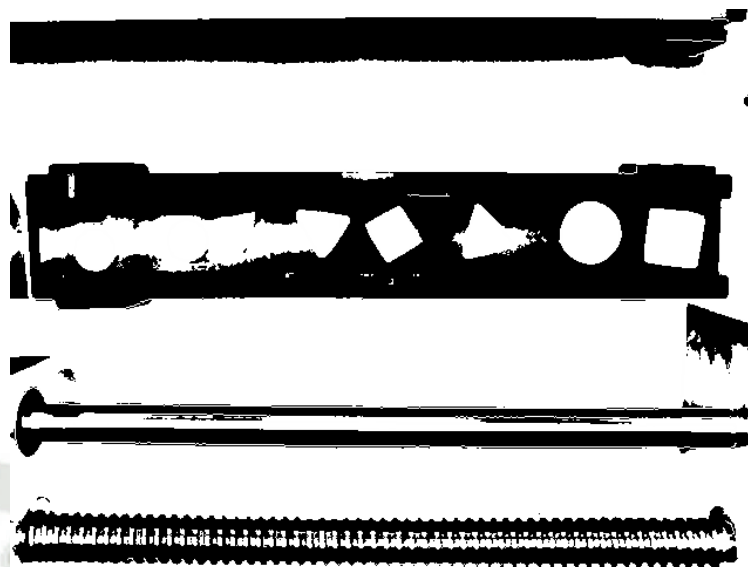
- bS:(blockSize) – Tamaño de un grupo de píxeles que se utiliza para calcular un valor de umbral para el píxel: 3, 5, 7, y así sucesivamente.
- Cs:(C) – Constante que resta de la media o media ponderada. Normalmente, es positivo, pero puede ser cero o negativo también.

```
thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, bS, Cs)
```

Para poder determinar la base se utiliza:

**Tabla 44:**Valores para determinar la base.

bS:(blockSize)	101
Cs:(C)	20



**Figura 144:**Resultado del Umbral adaptativo.

*Fuente: Elaboración propia.*

Para determinar las piezas en cambio, utilizamos otros valores, los cuales se encuentran determinados por defecto:

**Tabla 45:**Valores para determinar las piezas.

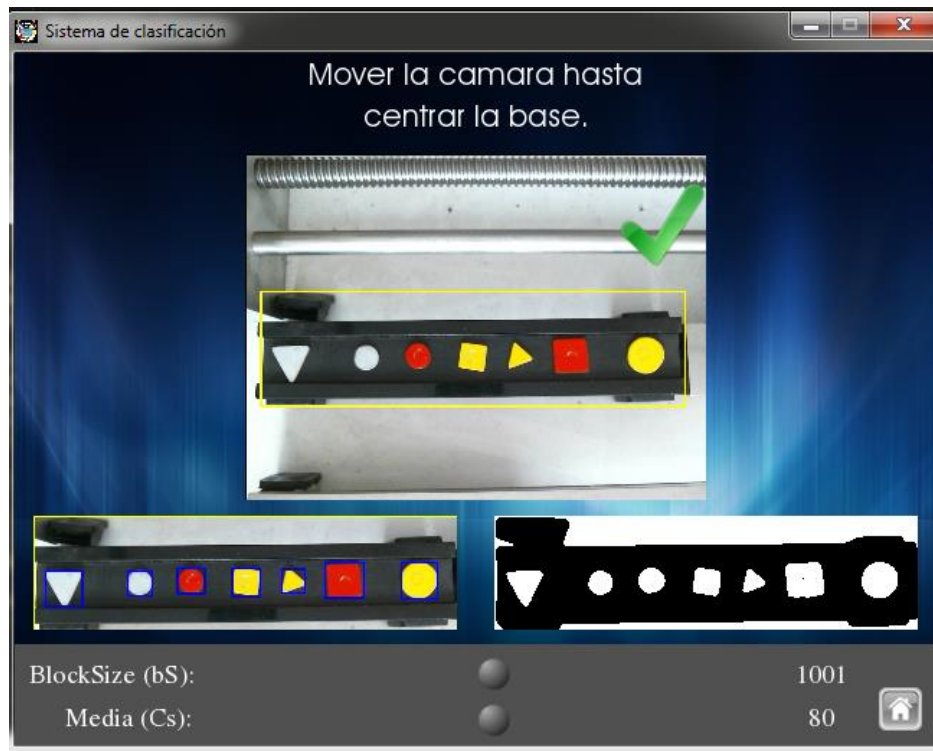
bS:(blockSize)	1001
Cs:(C)	80



**Figura 145:**Resultado de Umbral adaptativo.

*Fuente: Elaboración propia.*

Estos valores ayudan a compensar errores que se puedan producir por falta de iluminación, en la ventana de “calibración” permite variar estos parámetros y dejarlos así para iniciar la clasificación.



**Figura 146:** Vista de la opción “calibración”.

*Fuente: Elaboración propia.*

En esta imagen aparece un check cuando la imagen se encuentra centrada, de lo contrario aparece una “equis”, para esto es necesario mover la cámara hasta que se encuentra centrada la imagen, el sistema reconocerá que ya se encuentra en posición y se marcará con un check. Una vez que se ha tomado la imagen, en la parte inferior se pueden configurar los parámetros antes mencionados. Se recomienda hacerlo si sobre la figura situada al lado siquiera, no se marcan todos los objetos. Los valores seleccionados se almacenan en un archivo “.txt “ en la carpeta de “Memoria”, estos valores serán leídos en todo momento, para así tener un valor fijo en todo el proceso durante el procesamiento de imagen.

Nombre	Fecha de modifica...	Tipo	Tamaño
foto200	05/11/2014 10:17 a...	JPG File	176 KB
generalresponses	05/03/2015 01:55 ...	Archivo DATA	2 KB
generalcamper	05/03/2015 01:55 ...	Archivo DATA	110 KB
info	11/03/2015 05:42 ...	Archivo DATA	1 KB

**Figura 147:** Archivo .txt de procesamiento.

*Fuente: Elaboración propia.*

```
configuraciones = np.array([bs,cs], np.float32)
filename=filepath("Memoria/info.data")
np.savetxt(filename, configuraciones)
```

Retomando el tema del botón inicio, una vez que se ingresa a él se observa en la pantalla un menú, este nos permite elegir el tipo de característica a través del cual se desea hacer la clasificación. Se selecciona la opción y se inicia el proceso.



**Figura 148:** Selección de características.

*Fuente: Elaboración propia.*

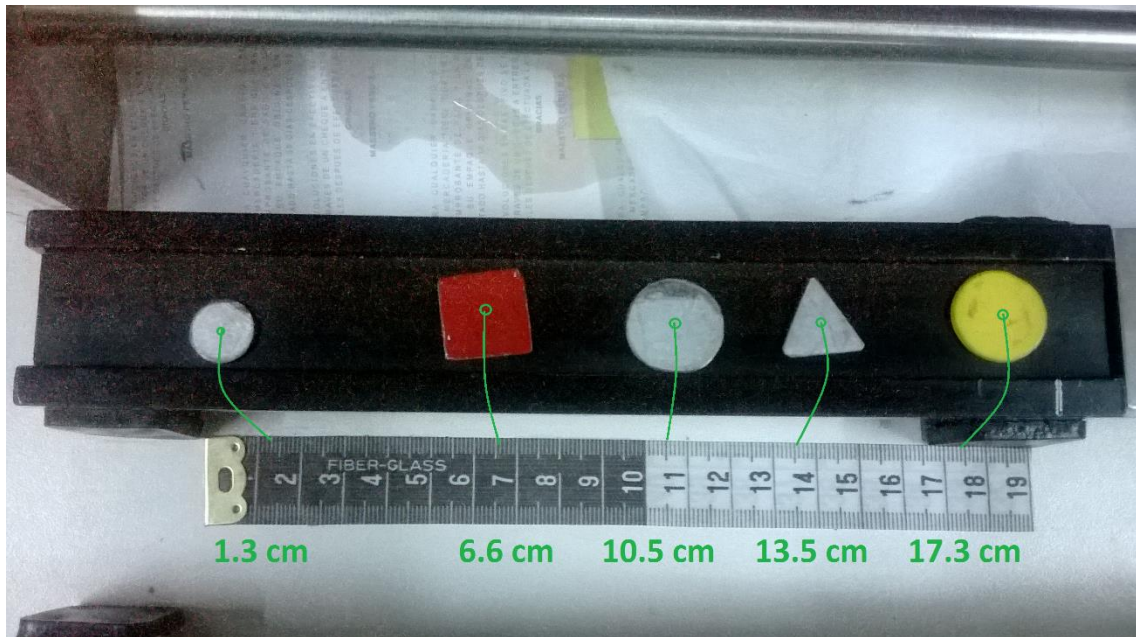
Una vez que ya se sabe cómo se desea clasificar las piezas, el sistema toma una foto, la cual será inspeccionada para poder hacer la clasificación, después procesada y se observa la imagen que se muestra en la figura 149, en ella se tienen todas las piezas, su forma, su tamaño y su color, del mismo modo se tiene la información de la distancia a la cual se encuentran cada una de ellas, después se procede a enviar esta información al PLC para que puedan ser separadas.



**Figura 149:** Piezas clasificadas.

*Fuente: Elaboración propia.*

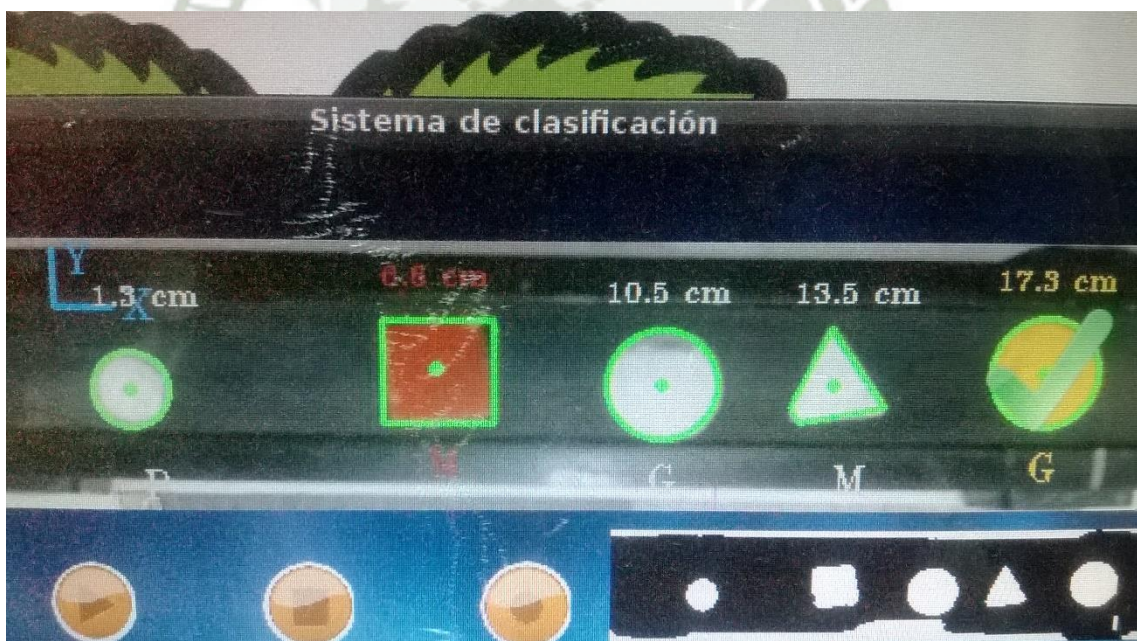
Para comprobar que los resultados fueron los mejores, se tomó algunas imágenes del sistema real y la respuesta que nos brinda el sistema. En el primero se puede ver las distancias reales que tienen las figuras respecto al punto de referencia, las lunas no se encuentran líneas pues por razones de óptica, las piezas y el “centímetro” se encuentran en diferentes planos, pero en la parte central es casi lineal.



**Figura 150:** Toma real de piezas.

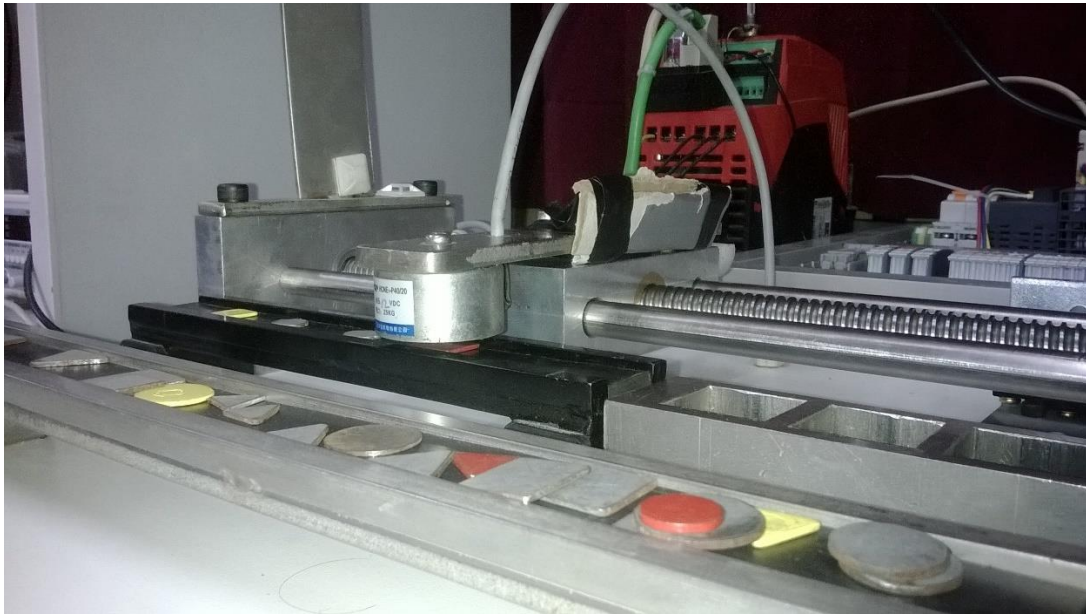
*Fuente: Elaboración propia.*

Estos valores son los mismos obtenidos por el sistema, se tiene una precisión de menos de 5 mm, lo cual es un buen resultado.



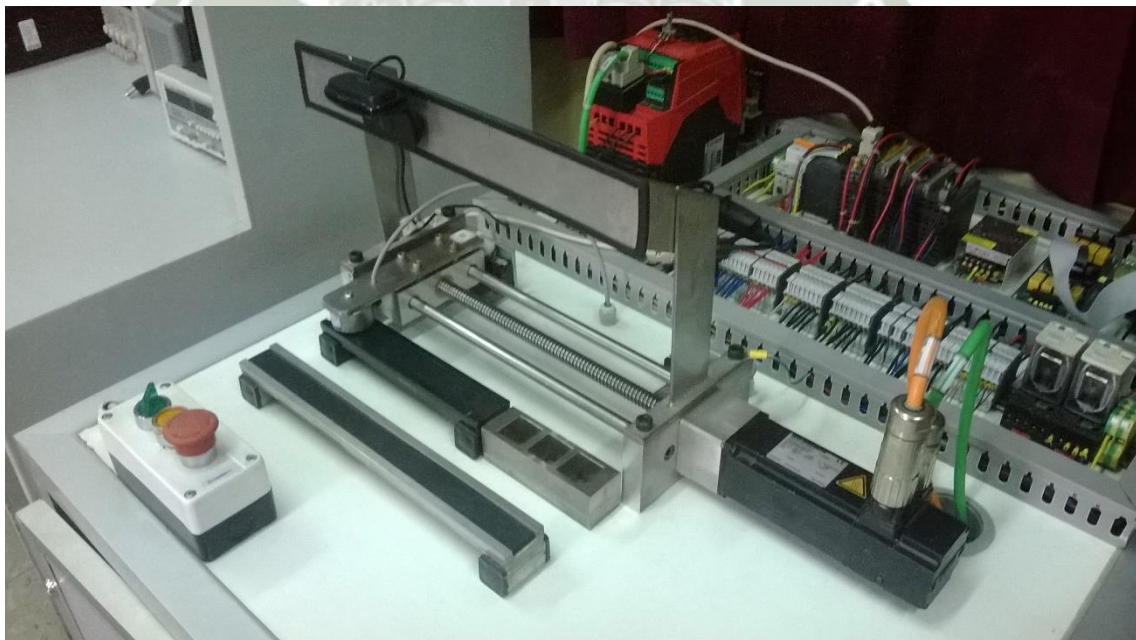
Finalmente el sistema separa las piezas, moviendo el husillo sin fin a través del servomotor y levantando cada una de ellas con el electroimán, el actuador se ubica sobre las piezas al recibir la señal analógica del Raspberry, como ya se ha mencionado antes, se presentaron algunos inconvenientes con el integrado MCP4151, pero fueron

superados, de todas maneras si se presentan algunas imperfecciones en el posicionamiento, se debe a pérdidas de tensión en el integrado. Recordar que las piezas deben moverse de izquierda a derecha, pues ninguna de ellas puede pasar encima de las otras, debido a la potencia del electroimán que cargaría hasta dos piezas.



**Figura 151:** Actuador clasificando piezas.

*Fuente: Elaboración propia.*



**Figura 152:** Sistema implementado.

*Fuente: Elaboración propia.*

# Conclusiones

- Se realizó el diseño y la implementación del sistema de clasificación, orientados por el procedimiento VDI 2221, determinamos los algoritmos de visión, elementos para el procesamiento de imagen, así como el control del proceso y validamos el sistema con un módulo funcional.
- Se diseñó y se implementó un sistema mecánico sencillo que nos permite clasificar las piezas. El husillo necesita estar adecuadamente fabricado, alineado y con los rodamientos necesarios para un movimiento suave. Fue necesario utilizar rodamientos lineales de billas para asegurar el desplazamiento adecuado del bloque móvil.
- Se ensambló adecuadamente el sistema, generando las listas de componentes así como el precio y los planos adecuados para su construcción. Se seleccionaron elementos de bajo costos y accesibles en el mercado local.
- El algoritmo K-Nearest Neighbord, mostró los mejores resultados, este método de aprendizaje de redes neuronales basado en memoria es eficiente para reconocer formas en un espacio de 2 dimensiones. El aprendizaje es importante para su adecuado funcionamiento.
- Es posible realizar la comunicación de elementos electrónicos de baja potencia, con elementos industriales de mayor potencia, mediante el desarrollo de tarjetas electrónicas y cálculos, adaptando así tecnología de bajo costo OpenSources a soluciones industriales.
- Implementar un proyecto es más eficaz si se sigue un procedimiento como el VDI 2221, esto reduce costos en manufactura y fabricación de la electrónica, pues se

minimiza al máximo las posibilidades de error. Se hicieron los cálculos necesarios y se obtuvo los planos de construcción de todo el sistema.

- En nuestra localidad, la creciente demanda de tecnología para productores agrícolas, hace de nuestro desarrollo un punto de partida para iniciar proyectos que solucionen estas deficiencias. Un poco de investigación sobre características de un producto en específico, más pruebas previas permitirían realizar nuevos módulos con soluciones completas.



# Recomendaciones

- La velocidad de procesamiento de la imagen es relativamente lenta, se utilizó un procesador a 700 MHZ, el procesamiento fue off-line. Es importante el elegir un adecuado procesador así como la cámara para tener éxito en el desarrollo y hacer procesamiento en tiempo real, que tiene mejores aplicaciones en la industria.
- El sistema puede ser aplicado en tiempo real si se utiliza un procesador embebido Jetson TK1 de NVidia, que es actualmente el procesador más rápido en el mercado a partir del Nov del 2015. En este procesador se puede realizar programación CUDA en paralelo que optimiza el tiempo de procesamiento.
- Utilizar un mejor actuador para separar las piezas, como una ventosa y generador de vacío o un electroimán más pequeño, de todas maneras el actuador depende del objeto o producto a clasificar.
- La programación en Python es muy amigable, pues los comandos de programación son sencillos e intuitivos, es posible mejorar los tiempos de procesamiento utilizando programación en C++.
- Cuando se inicie el sistema, tenerlo en un área cerrada con iluminación estándar, evitar sombras sobre el área de inspección.
- Durante la puesta en funcionamiento del sistema, se encuentran muchos problemas que al inicio del proceso de diseño no son predecibles, estos inconvenientes fueron resueltos, para así obtener un módulo funcional, estos problemas se presentan en todo proyecto y se debe ser paciente y persistente para obtener un buen resultado.

# Limitaciones

- El sistema no puede realizar el procesamiento en tiempo real, pues el embebido Raspberry es un procesador de 700 Mhz y no es lo suficientemente rápido para procesar las imágenes.
- Se requiere de periféricos para iniciar el proceso de clasificación del Raspberry, este inicio es manual, pues actualizaciones del sistema evitan hacer automático el programa, quizás más adelante sea posible.
- El movimiento lineal y la velocidad máxima alcanzada es la que el motor y el eje eléctrico lineal seleccionado nos permite, por imperfecciones en la fabricación del sistema mecánico, es recomendable que el motor no supere los 1000 RPM, teniendo un movimiento lineal máximo de 4 m/s.
- El actuador magnético, electroimán, por sus dimensiones nos impide colocar piezas a menos de 2 cm de distancia entre ellas y un máximo de 5 piezas en toda la base. Esto se debe al magnetismo remanente que se produce en la carcasa, ante esta situación no se llegó a establecer procedimientos de desmagnetización. En algunos casos esto impide que la pieza pueda caer en el depósito correspondiente.
- Para el sistema de clasificación, se requiere de piezas metálicas, esto a razón del actuador.
- Algunos de los elementos electrónicos implementados, como el MCP4151, varían su valor en cuanto a la temperatura, esto es debido a que estos elementos electrónicos no son robustos y están susceptibles a campos magnéticos y corrientes parasitas.

- Dentro del campo de la visión artificial, la iluminación es un tema importante, pero fue posible evitar el tener un área controlada de visión, mediante herramientas en el procesamiento como el umbral adaptativo, el número de píxeles de análisis considerados, nos permitieron obtener buenos resultados. Aun así el sistema puede funcionar bajo condiciones estándar de iluminación de una habitación cerrada, para otras condiciones será necesarios ajustes, de lo contrario puede que el sistema falle.
- El control del servomotor está limitado a un control de posición por medio de un PLC externo, esto a razón que para tener mayor manejo del driver, SEW te restringe a utilizar PLC's de la misma marca.
- El sistema solo puede reconocer cuadrados, triangulo y circulo, para que el sistema reconozca más formas requiere de una modificación de programa así como una foto inicial con todas las opciones.



# Bibliografía

- [1] J. Hodges y E. Fix, «An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951),» *International Statistical Review*, 1989.
- [2] S. Haykin, "Neural Networks, A comprehensive Foundation", Canada: Macmillan, 1994., 2005.
- [3] B. Barriga, Curso de Diseño Mecánico, Lima, 1985.
- [4] G. M. Benavides y E. R. Guerrer, «"Automated System for Classifying Hass Avocados Based on Image Processing Techniques",» *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, n° 9781479943401, pp. 1-6, 2014.
- [5] G. Feng y C. Qixin, «"Study on color image processing based intelligent fruit sorting system",» *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788)*, vol. 6, n° 0-7803-8273-0, pp. 4802-4805, 2004.
- [6] . H. Afrisal, M. Faris, G. Utomo P., . L. Grezelda, . I. Soesanti y M. Andri F., «"Portable smart sorting and grading machine for fruits using computer vision",» *International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, pp. 71-75, 2013.
- [7] M. Weyrich y M. Scharf, «"Quality assessment of row crop plants by using a machine vision system",» *CON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, n° 978-1-4799-0224-8, pp. 2466-2471, 2013.
- [8] S. Matuska, R. Hudec y . M. Benco, «"The comparison of CPU time consumption for image processing algorithm in Matlab and OpenCV",» n° 978-1-4673-1179-3, pp. 75-78, 2012.
- [9] K. Maza y R. Bustinsa, "Algoritmos de aprendizaje en visión artificial para la captura de objetos en un espacio tridimensional", Tesis BSc., Fac. Ing. Mec., Tesis Prof. Ing., Universidad Nacional de Ingeniería.: Lima, Perú, 2008.

- [10] E. Mejía, "Control por aprendizaje Neuro-fuzzy de un sistema servo-hidráulico de alta frecuencia", Tesis BSc., Fac. Ing. Mec, Univ. Nacional de Ingeniería.: Lima, Perú, 2011.
- [11] V. Sánchez, "Reconocimiento de patrones", Tesis BSc, Fac. Ing. Mec., Prof. Ing., Univ. Católica de Santa María: Aqp, Perú, 2013.
- [12] E. Cuevas, D. Zaldívar y M. Pérez, Procesamiento digital de imágenes con MATLAB y simulación, Madrid: Ra-Ma Editorial, 2010.
- [13] B. Martin and A. Sanz, "Redes neuronales y sistemas borrosos", México D.F.: RA-MA, 2001, 2007.
- [14] J. Hilera y V. Martinez, "Redes Neuronales Artificiales", España.
- [15] SEW EURODRIVE, "Anexo a las instrucciones de funcionamiento," Alemania, 2011.
- [16] SEW EURODRIVE, "Instruccion de funcionamiento," Alemania, 2010.
- [17] Wikipedia, «OpenCV,» [En línea]. Available: <http://es.wikipedia.org/wiki/OpenCV>. [Último acceso: 2015 02 26].
- [18] Wikipedia, «Pygame,» [En línea]. Available: <http://es.wikipedia.org/wiki/Pygame>. [Último acceso: 26 Febrero 2015].
- [19] Wikipedia, «Step7,» [En línea]. Available: [http://es.wikipedia.org/wiki/STEP\\_7](http://es.wikipedia.org/wiki/STEP_7). [Último acceso: 26 Febrero 2015].
- [20] Wikipedia, «Inteligencia Artificial,» [En línea]. Available: [http://es.wikipedia.org/wiki/Inteligencia\\_artificial](http://es.wikipedia.org/wiki/Inteligencia_artificial). [Último acceso: 12 Enero 2014].
- [21] Wikipedia, «Redes neuronales,» [En línea]. Available: [http://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](http://es.wikipedia.org/wiki/Red_neuronal_artificial). [Último acceso: 16 Enero 2014].
- [22] Wikipedia, «Controlador Logico Programable,» [En línea]. Available: [http://es.wikipedia.org/wiki/Controlador\\_l%C3%B3gico\\_programable..](http://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable..) [Último acceso: 12 Enero 2014].
- [23] Wikipedia, «Electroimán,» [En línea]. Available:

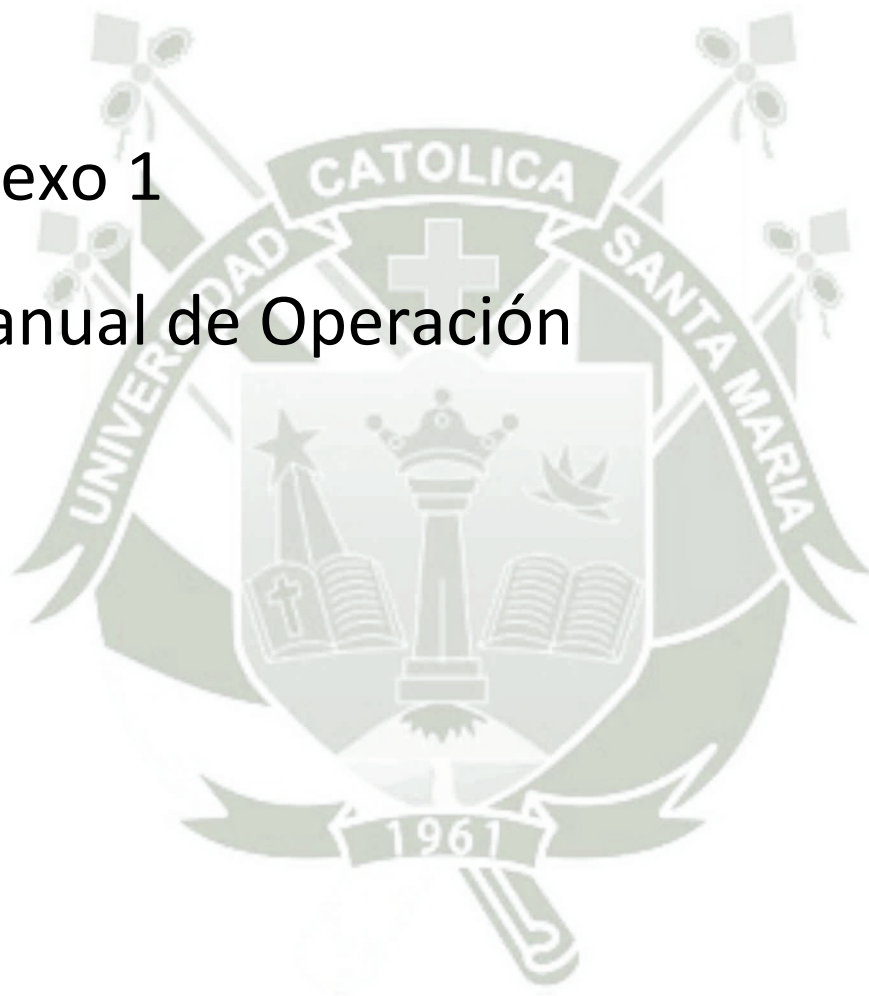
- <http://es.wikipedia.org/wiki/Electroim%C3%A1n>. [Último acceso: 26 Febrero 2015].
- [24] Compac, [En línea]. Available: <http://es.compacsort.com/wawcs0118327/In-18327.html>. [Último acceso: 22 Octubre 2014].
- [25] TOMRA, [En línea]. Available: <http://www.tomra.com/>. [Último acceso: 22 Octubre 2014].
- [26] Wikipedia, «Aplicaciones de la electricidad,» [En línea]. Available: [http://es.wikipedia.org/wiki/Aplicaciones\\_de\\_la\\_electricidad](http://es.wikipedia.org/wiki/Aplicaciones_de_la_electricidad). [Último acceso: 26 Febrero 2015].
- [27] Wikipedia, «Raspberry Pi,» [En línea]. Available: [http://es.wikipedia.org/wiki/Raspberry\\_Pi](http://es.wikipedia.org/wiki/Raspberry_Pi). [Último acceso: 16 Enero 2014].
- [28] Raspberry para torpes, pero para torpes, «Actualización a la nueva versión de Raspi-config,» [En línea]. Available: <http://rasberryparatorpes.net/tag/configuracion-inicial-raspberry/>. [Último acceso: 25 Junio 2015].
- [29] IDEASONBOARD, «Linux UVC driver and tools,» [En línea]. Available: <http://www.ideasonboard.org/uvc/>. [Último acceso: 25 Junio 2014].
- [30] OpenCV, «Welcome to OpenCV Documentation,» [En línea]. Available: <http://docs.opencv.org/>. [Último acceso: 22 Febrero 2015].
- [31] Detrás del Pinguino, «Linux y sus alrededores,» [En línea]. Available: <http://dplinux.net/guia-raspberry-pi/>. [Último acceso: 20 Febrero 2015].
- [32] OpenCV Documentation, «KNNearest,» [En línea]. Available: [http://docs.opencv.org/modules/ml/doc/k\\_nearest\\_neighbors.html#cvknearest-train](http://docs.opencv.org/modules/ml/doc/k_nearest_neighbors.html#cvknearest-train). [Último acceso: 12 Enero 2014].
- [33] Wikipedia, «Servomotor,» [En línea]. Available: <http://es.wikipedia.org/wiki/Servomotor..> [Último acceso: 12 Enero 2014].
- [34] Wikipedia, «OpenCV,» [En línea]. Available: <http://es.wikipedia.org/wiki/OpenCV>. [Último acceso: 16 Enero 2014].

# Anexos



# Anexo 1

## Manual de Operación



## MANUAL DE OPERACIÓN

### CONSIDERACIONES INICIALES

- 1) Inspeccionar el sistema en busca de cables sueltos u objetos en la línea de acción del eje mecánico y/o cualquier otra situación anómala.
- 2) Energizar el sistema mediante el cable de poder ubicado en la parte lateral izquierda del módulo, asegurarse que la toma cuente con puesta a tierra.
- 3) Ubicar el Switch General, parte lateral del módulo, en “1”.
- 4) Ubicar llave termomagnética en “1”.

### REFERENCIAMIENTO

- 1) Asegurarse que el pulsador de emergencia ubicado en la botonera frontal no se encuentre activado.
- 2) Para iniciar el referenciamiento el Switch de la botonera debe estar ubicado en posición derecha “On”, la luz testigo “Verde” se enciende.
- 3) El carro se mueve hasta la posición izquierda del eje, al alcanzar el Limit Switch de esta posición el sistema entra en pausa automáticamente, se enciende la luz testigo “Ambar”. En este instante el sistema se encuentra referenciado y se reconoce la posición como “0 mm”.
- 4) Pulsar el botón de pausa “Ambar” con esto el eje se mueve hasta alcanzar el Limit Switch de la posición derecha y entra en pausa nuevamente.

### PROCESAMIENTO DE IMAGENES

- 1) En este lapso de tiempo, mientras el sistema se encuentra en pausa, ubicar las piezas de trabajo sobre la zona de inspección.
- 2) Verificar condiciones iniciales para que el procesamiento de imágenes pueda iniciar (Aprendizaje, Calibración, En posición).
- 3) De ser necesario calibrar posición de la cámara y/o contraste para reconocer imágenes bajo ciertas condiciones de iluminación.
- 4) Ingresar a Iniciar, seleccionar el tipo de clasificación que se desea realizar por: Tamaño, Color, Forma o la combinación de estos y dar click en Start.
- 5) La cámara captura la imagen y realiza el procesamiento automáticamente.

### **CLASIFICACION**

- 1) La pausa es liberada cuando el procesamiento de las imágenes se ha completado entonces se envía confirmación al PLC y se inicia el proceso de clasificación de forma automática.
- 2) Si el sistema se queda sin piezas entra en pausa automáticamente hasta que se ubiquen más piezas y se procesen.

### **PAUSA**

- 1) La pausa solo puede ser activada durante el proceso de clasificación, al pulsar el botón ambar, se activa la luz testigo y se detienen todos los movimientos.
- 2) Al volver a pulsar el botón ambar se libera la condición de pausa, la luz testigo se apaga y el sistema sigue funcionando de forma normal.

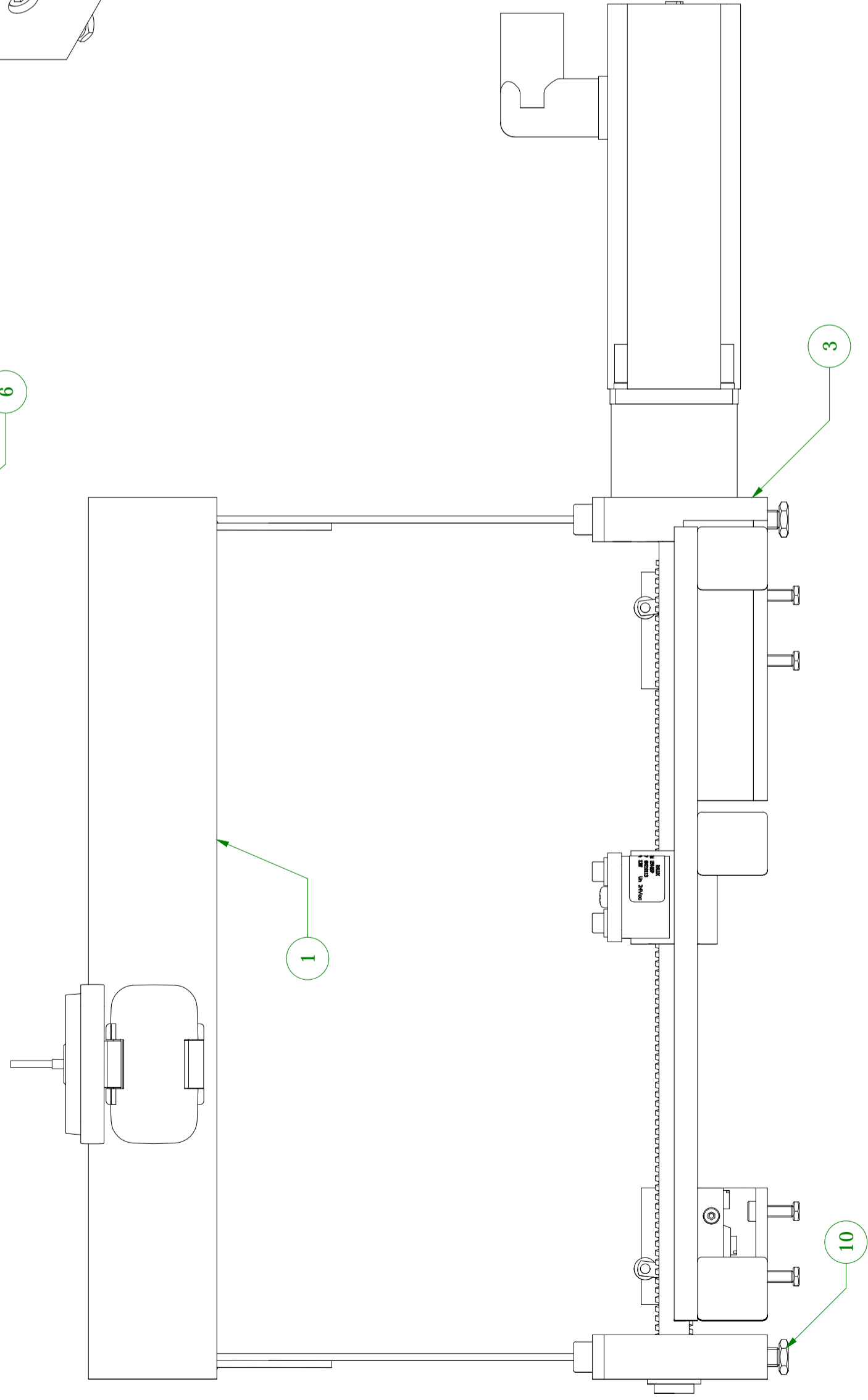
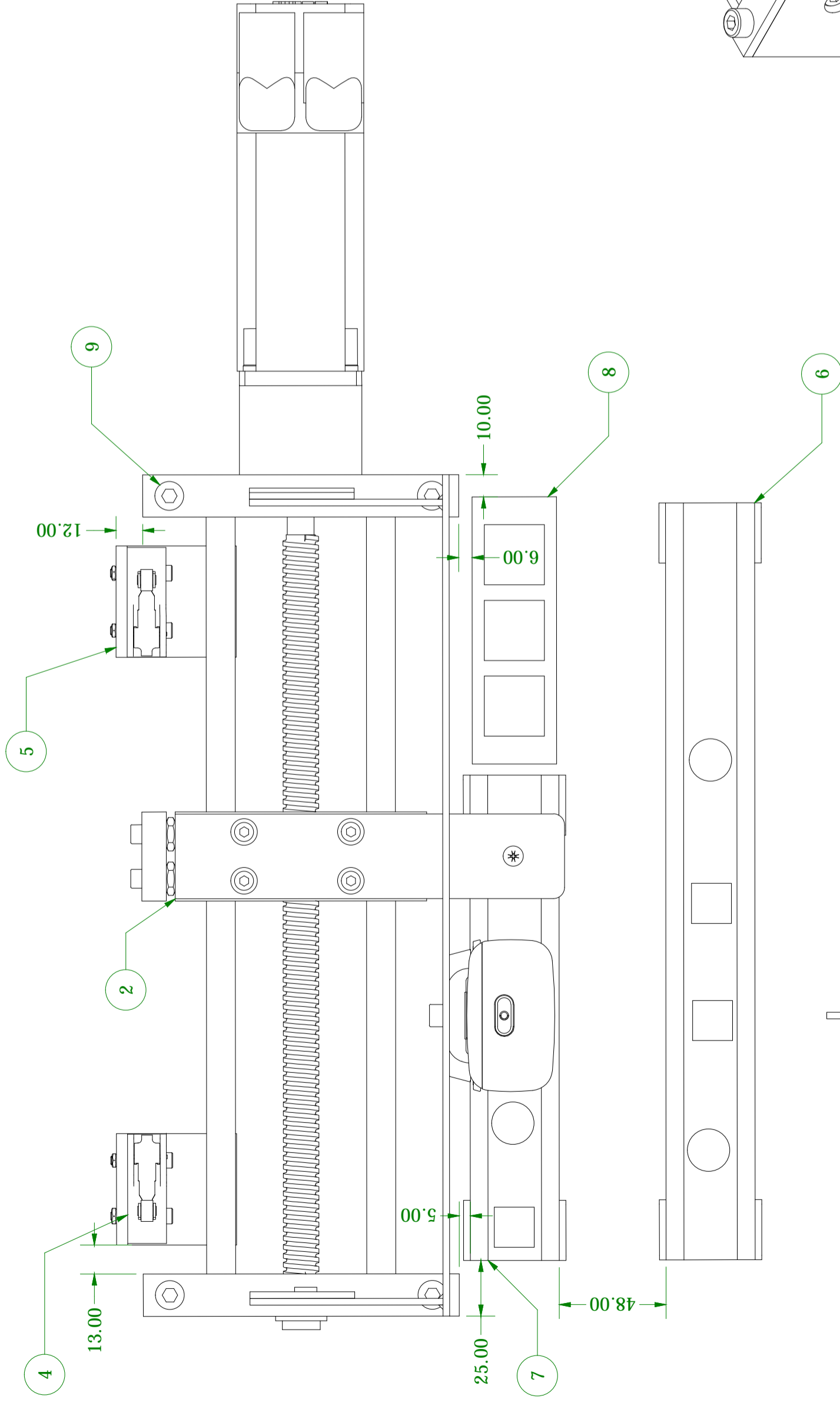
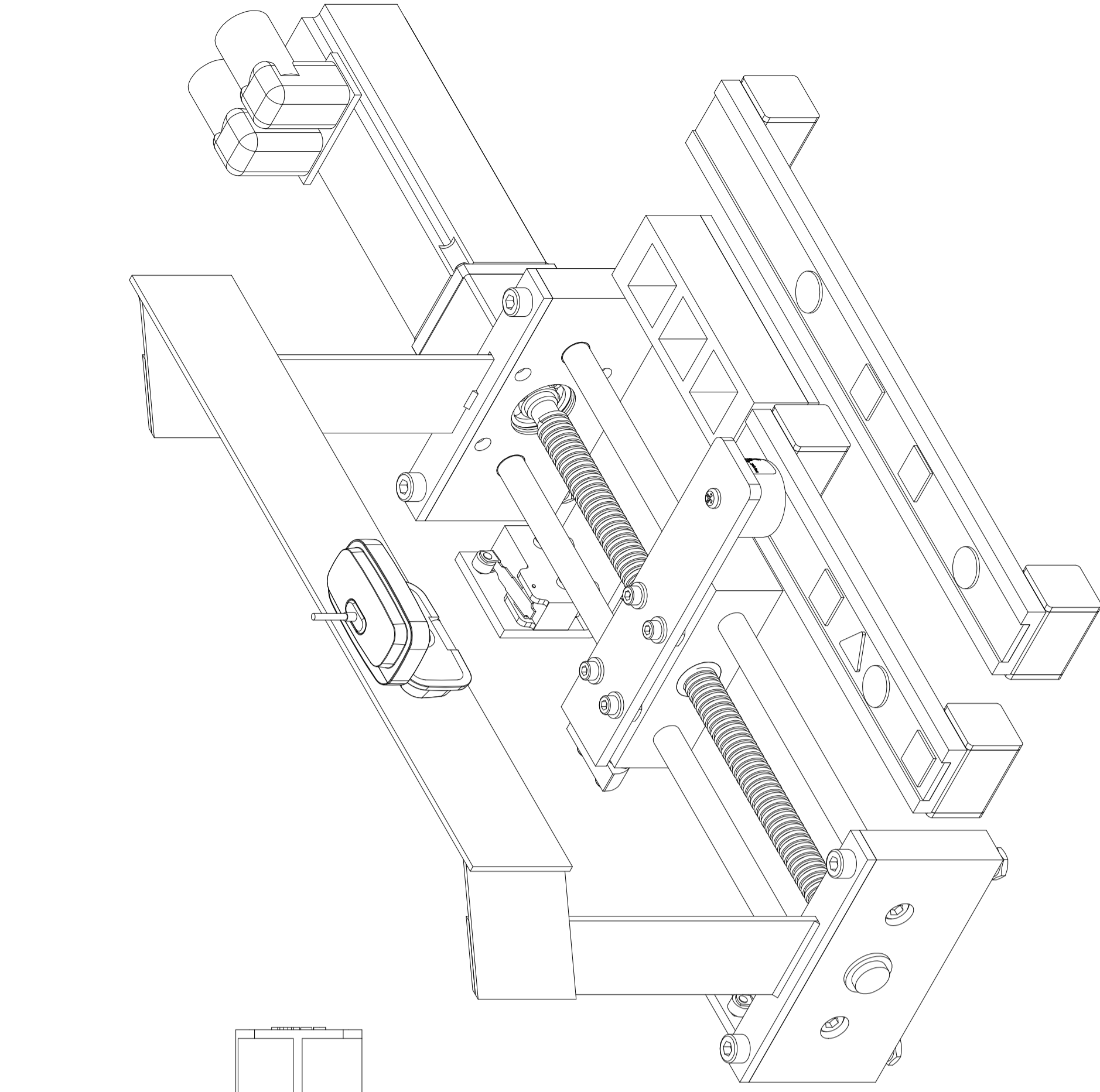
### **PARADA DE EMERGENCIA / STOP**

- 1) La parada de emergencia tiene prioridad de funcionamiento, se activa al presionar el pulsador de emergencia "Rojo" con enclavamiento, todo el sistema se detiene.
- 2) Para quitar la condición de parada de emergencia se debe desenclavar el pulsador girándolo levemente de forma horaria y ubicar el Switch de la botonera en la posición izquierda.
- 3) Para completar el proceso de eliminar la condición de parada de emergencia, es necesario volver a referenciar el sistema, ubicar Switch de botonera en posición derecha.

## Anexo 2

## Planos Mecánicos





IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Soporte de camara			
2	1	Carro transversal			
3	1	Soporte de husillo			
4	1	Limit switch izquierda			
5	1	Limit switch derecha			
6	1	Base externa			
7	1	Base para piezas			
8	1	Recipiente de recepcion			
9	4	Tor. allen Hexagonal M8 x 12	DIN 912	8.8	
10	4	Tuerca Hexagonal M8	DIN439B		

UNIVERSIDAD CATOLICA DE SANTA MARIA  
 FACULTAD DE CIENCIAS FISICAS Y FORMALES

PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA  
 SISTEMA DE CLASIFICACION  
 POR VISION ARTIFICIAL

**Sistema de clasificacion**

ZARATECM, RODRIGUEZSE



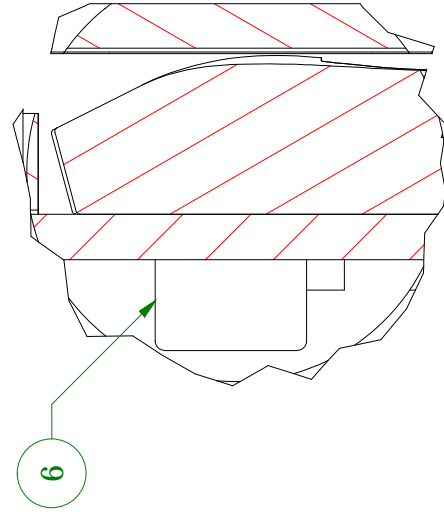
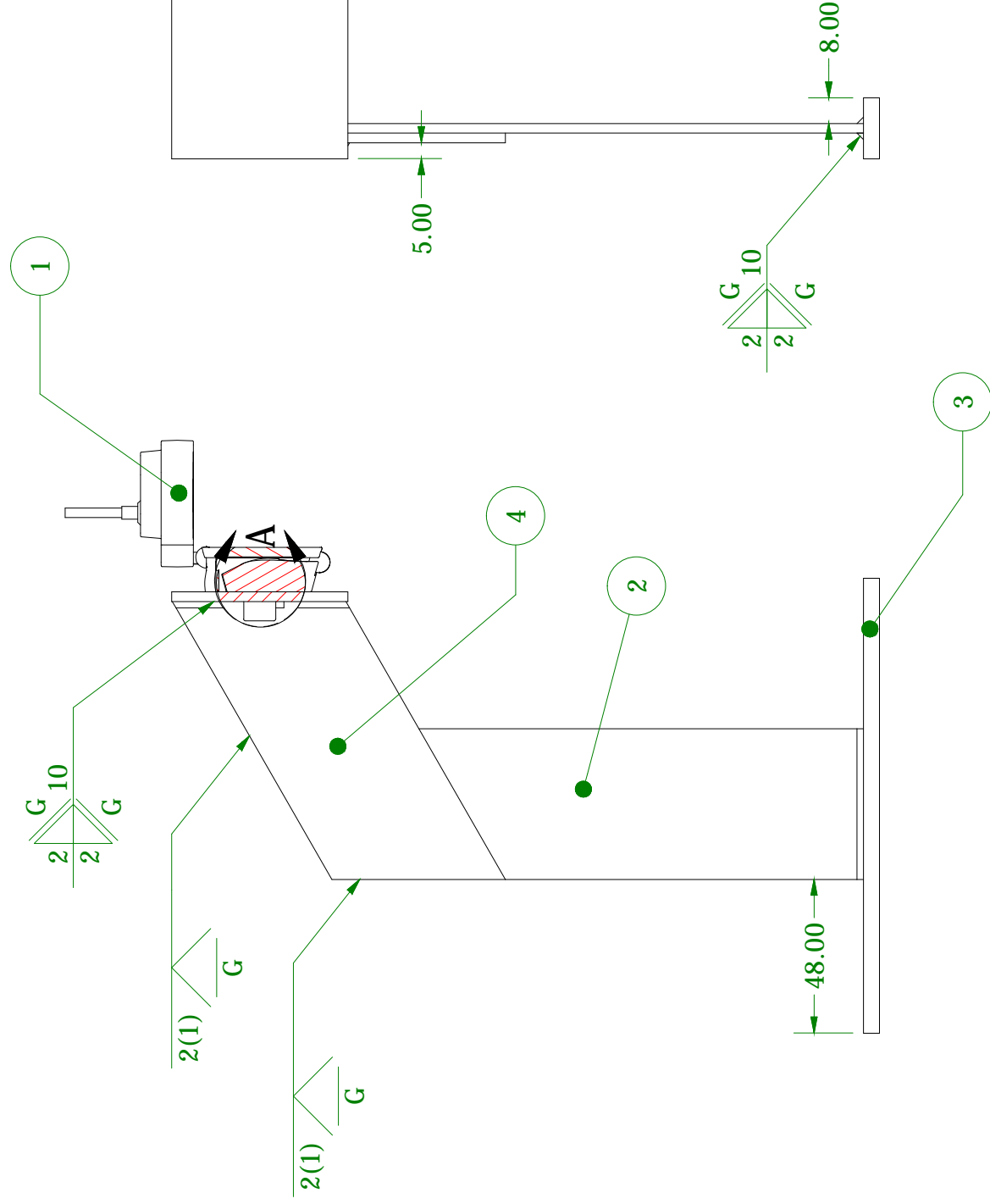
2008220181

ESCALA

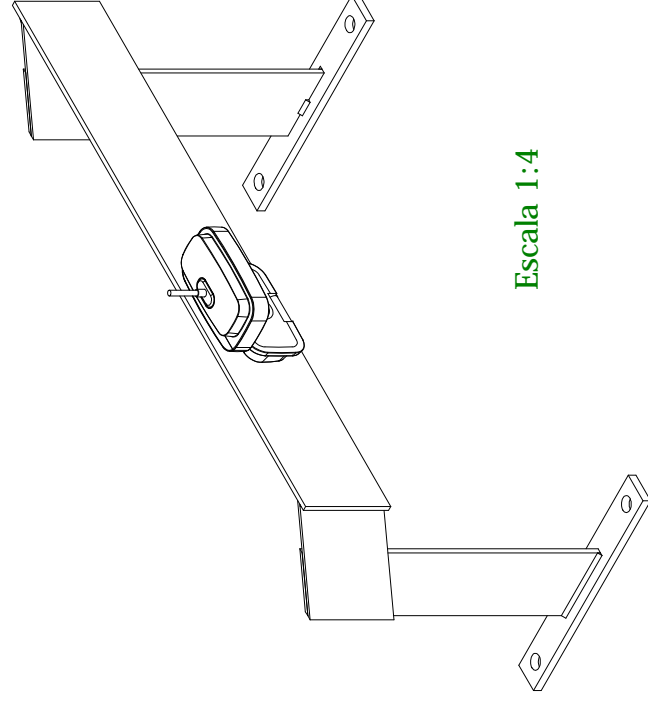
1:2

FECHA:  
25/02/2015

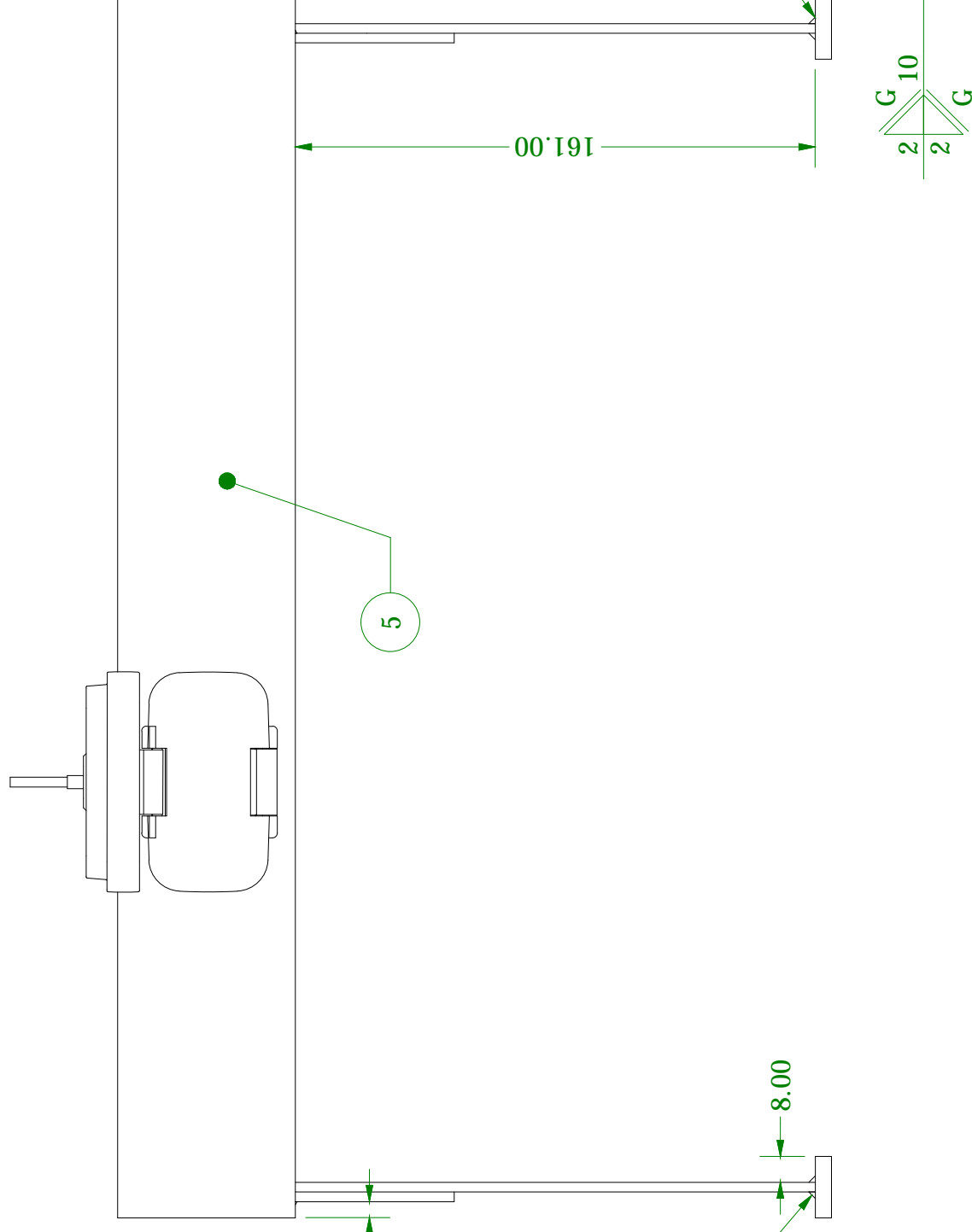
LAMINA: A2  
1 of 32



Detalle A  
Escala 2 : 1



Escala 1:4



IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Cámara Digital HD Webcam			Logitec 615
2	2	Apoyo lateral			
3	2	Base de apoyo			
4	2	SopORTE superior			
5	1	Base dela cámara			
6	1	Tor. de cabeza hexagonal M6X10	DIN 933		

UNIVERSIDAD CATOLICA DE SANTA MARIA  
FACULTAD DE CIENCIAS FISICAS Y FORMALES

PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA  
SISTEMA DE CLASIFICACION  
POR VISION ARTIFICIAL

**SopORTE de cámara**

ZARATECM, RODRIGUEZE

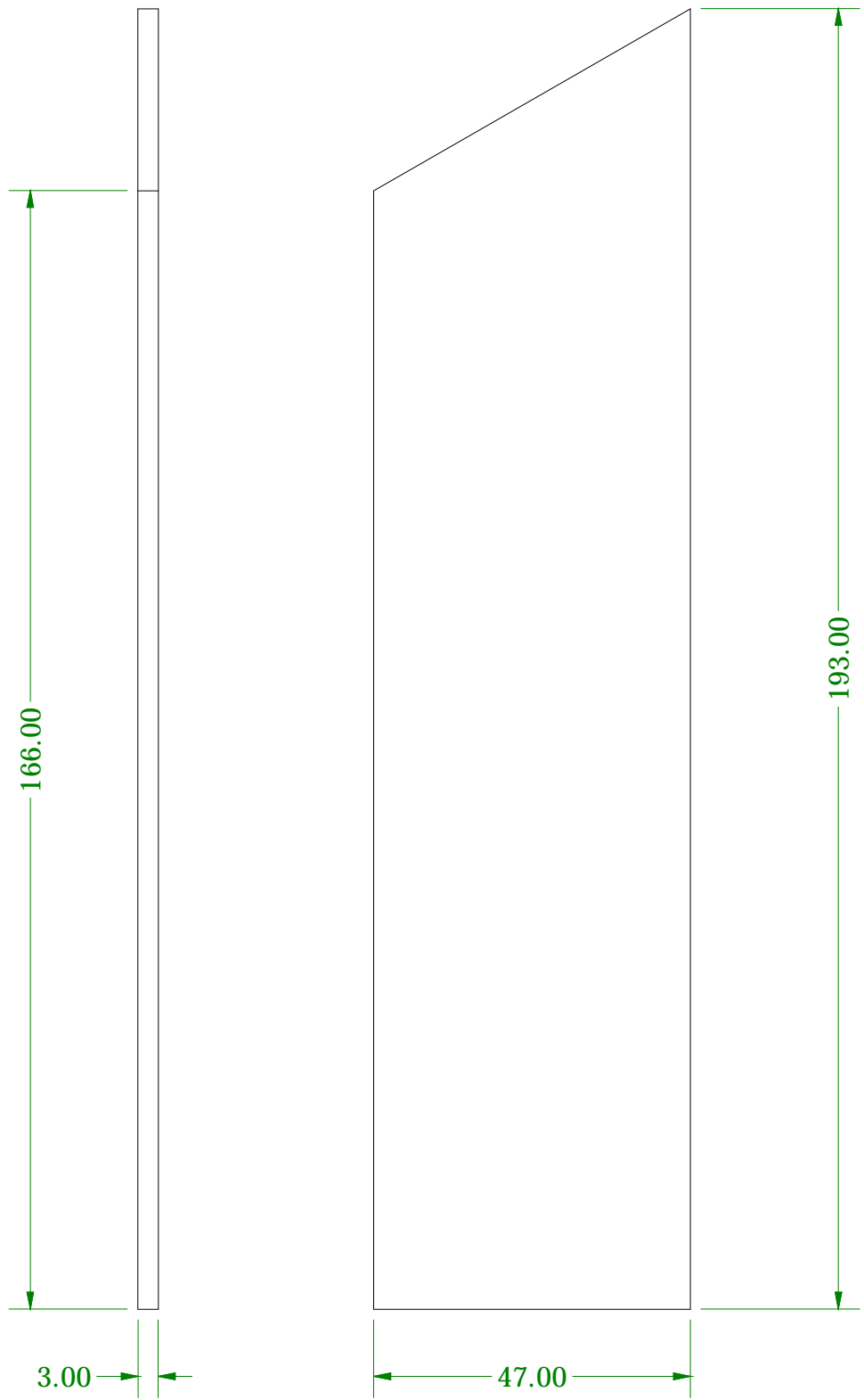
2008220181

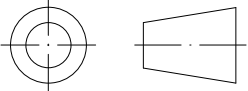
ESCALA

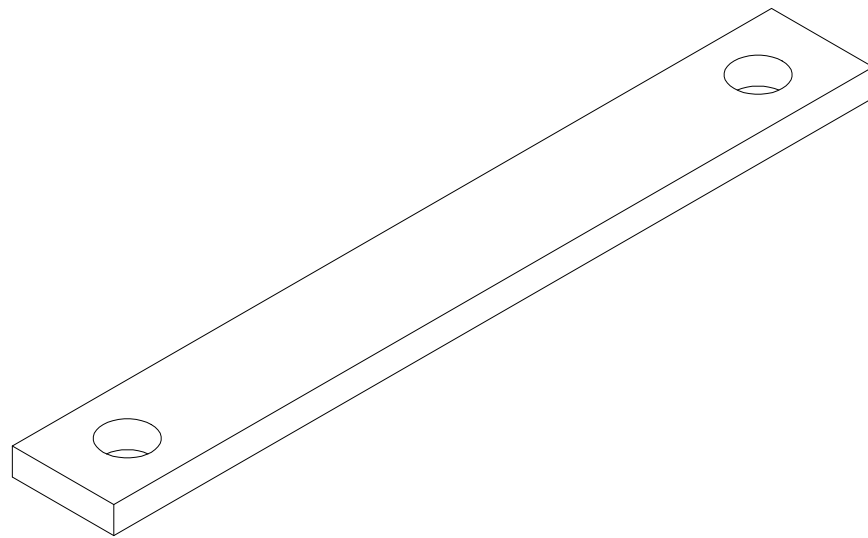
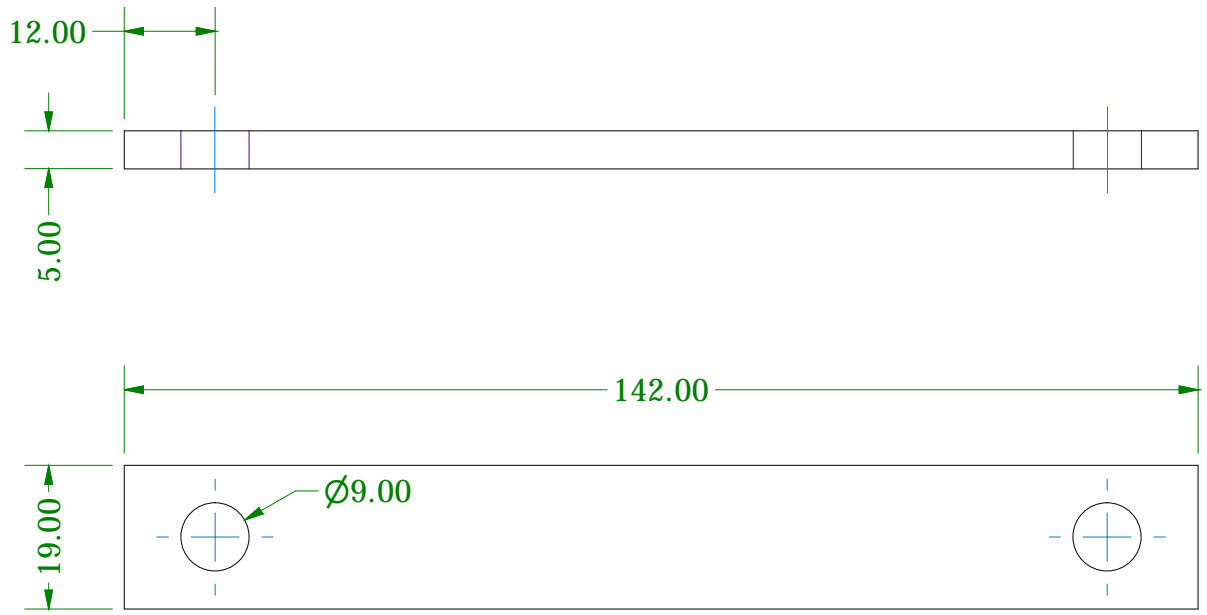
1:2

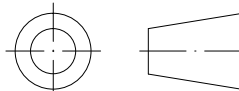
FECHA:  
25/02/2015

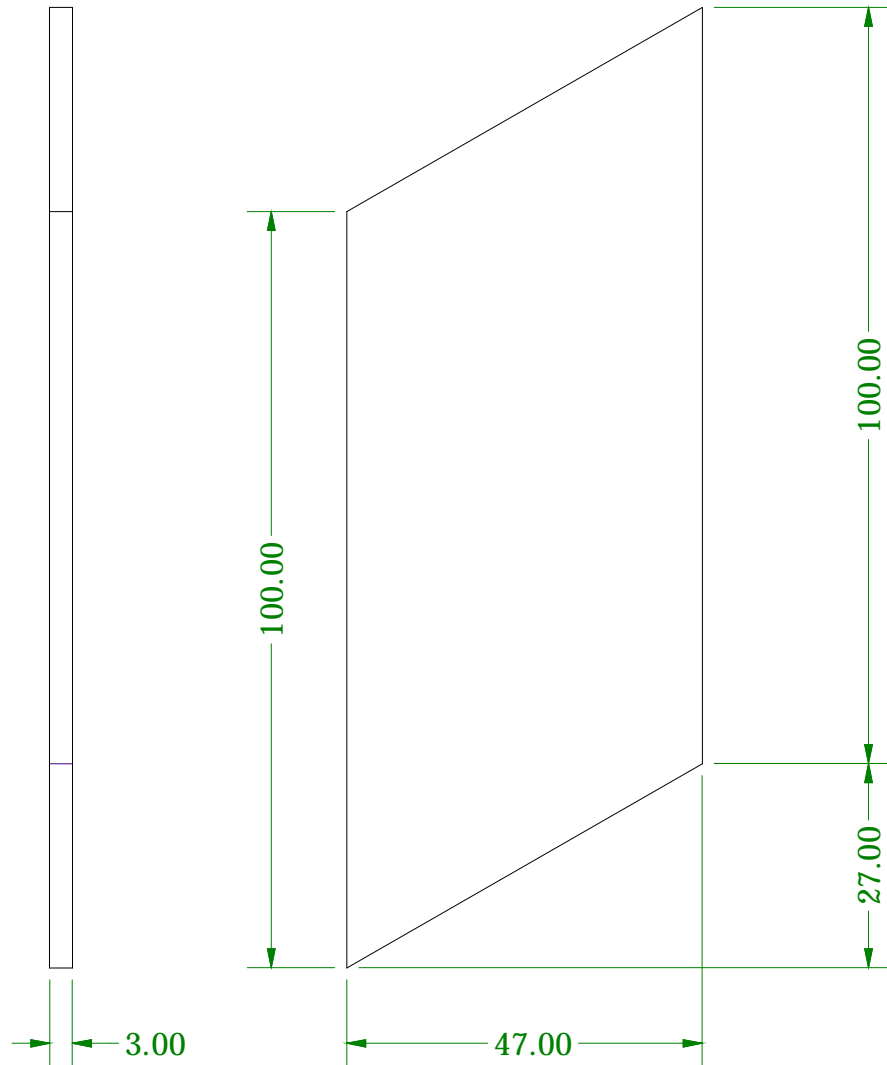
LAMINA: A3  
2



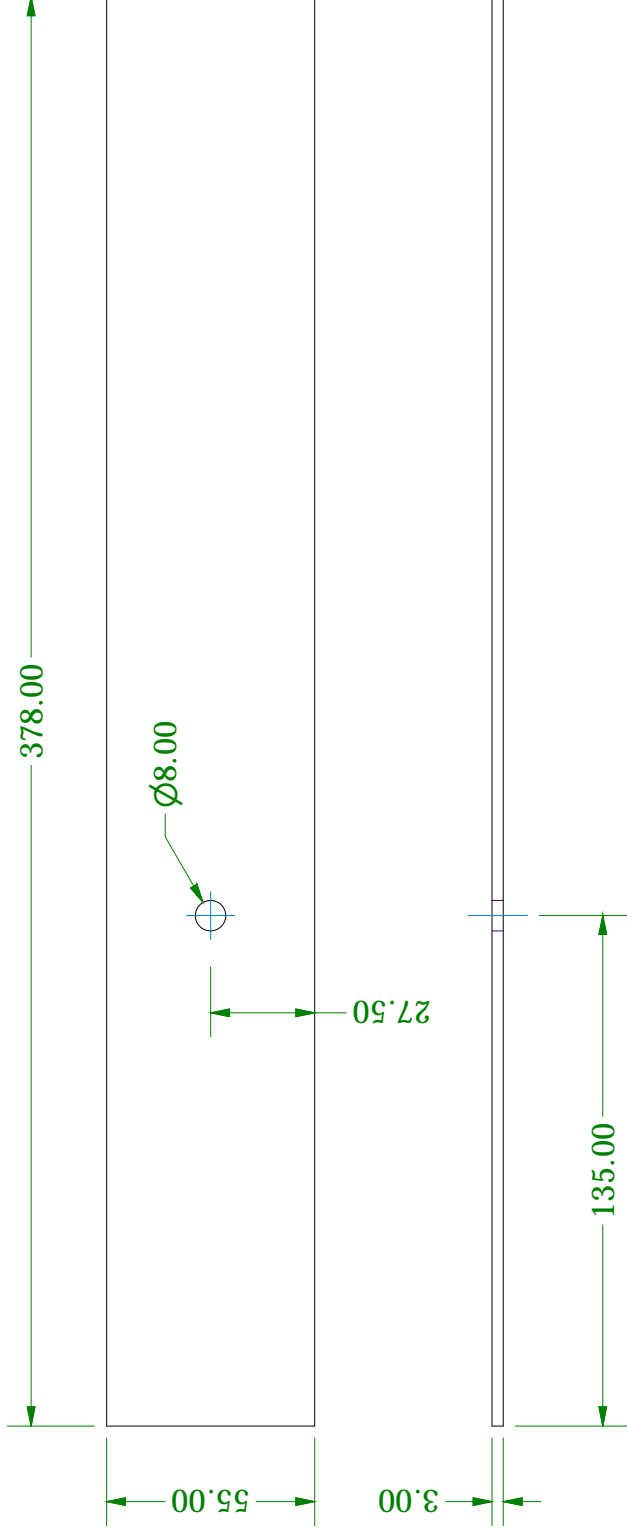
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Apoyo lateral</b>	ESCALA <b>1:1</b>
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 3 of 32

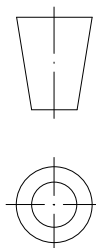


ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Base de apoyo</b>	ESCALA  1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 4 of 32

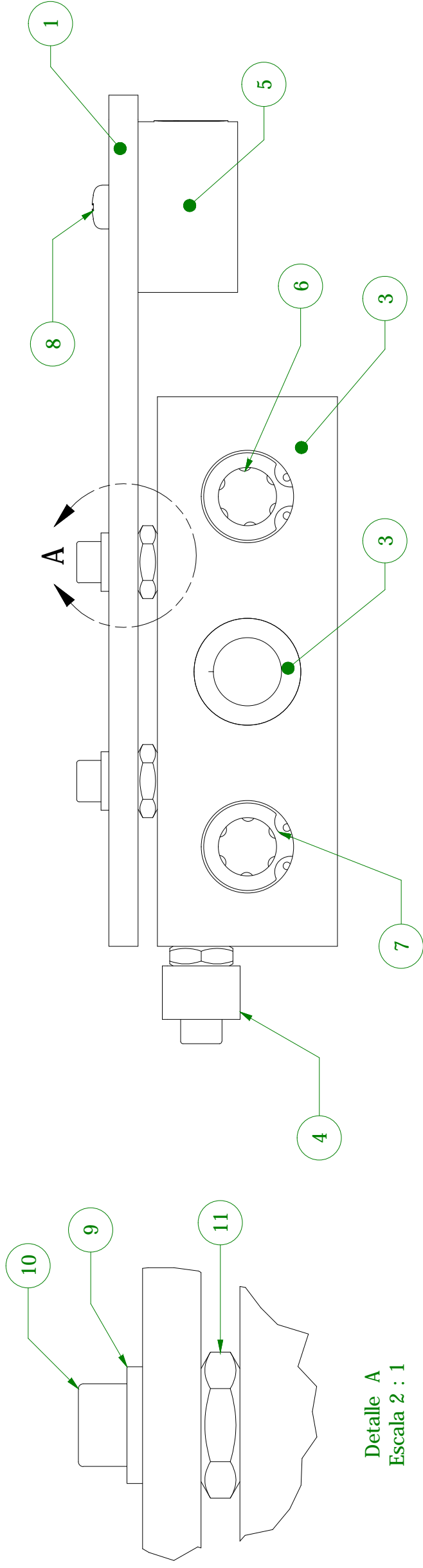


ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Soporte superior</b>	ESCALA <b>1:1</b>
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 5 of 32

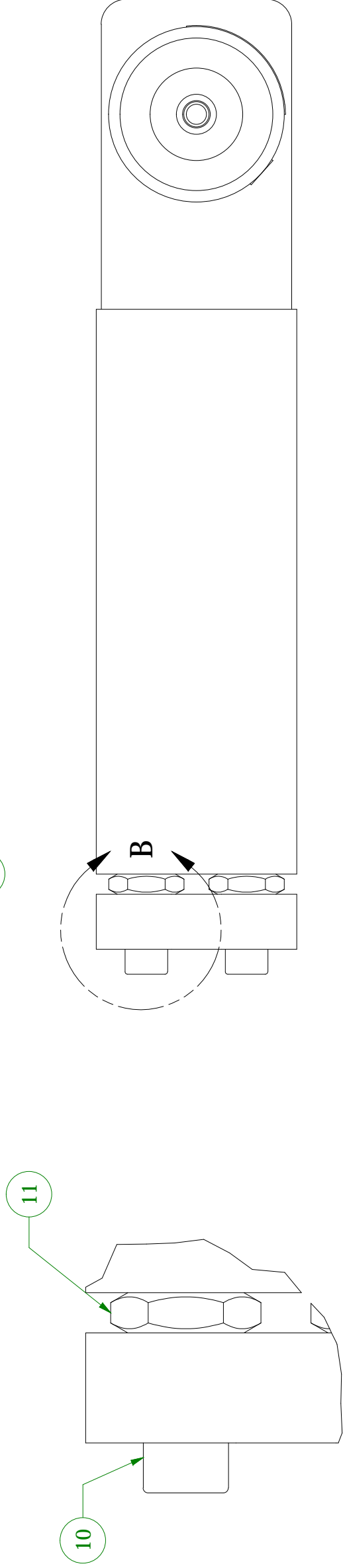


ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Base dela cámara</b>	ESCALA 1:2
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 6 of 32

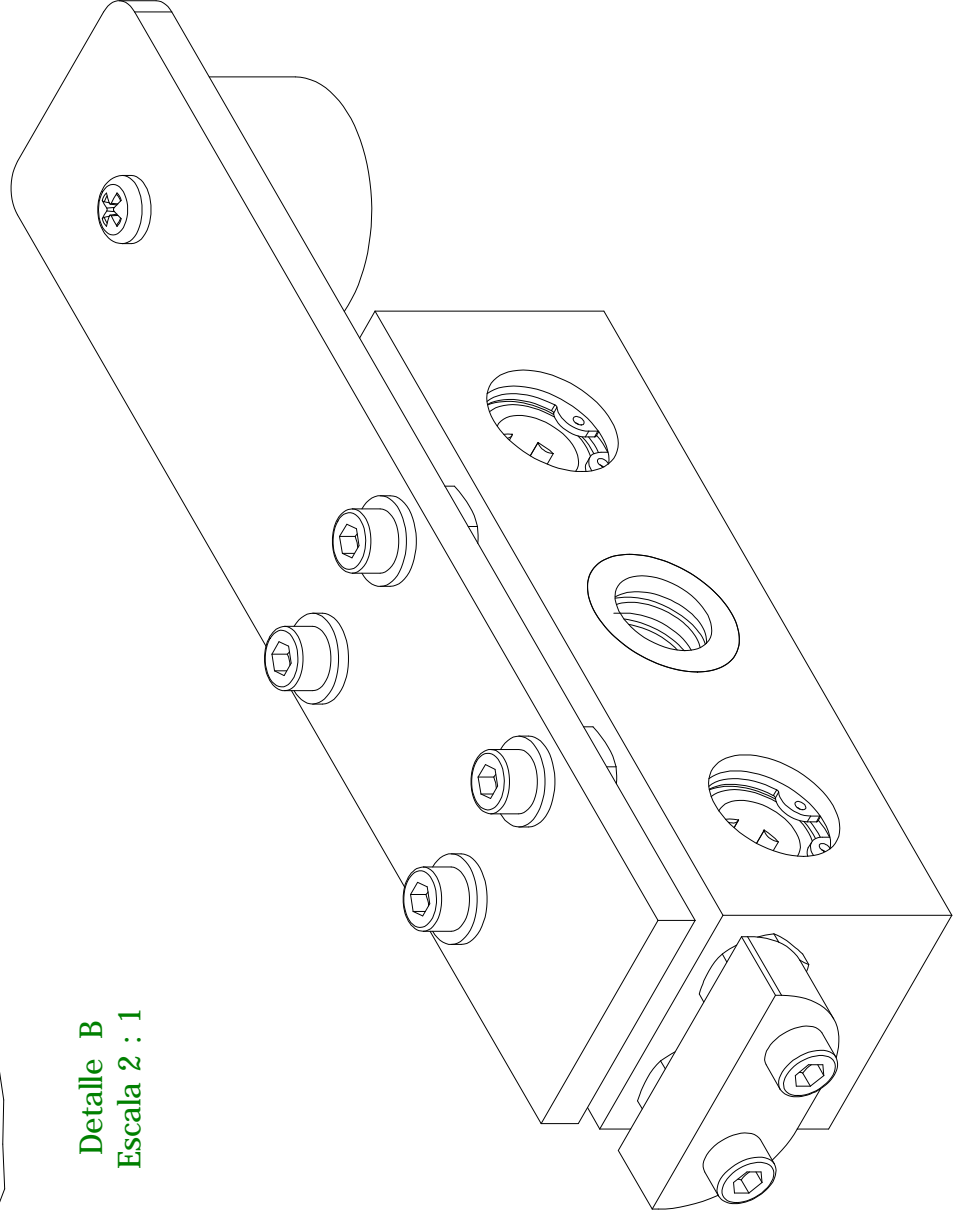




Detalle A  
Escala 2 : 1



Detalle B  
Escala 2 : 1

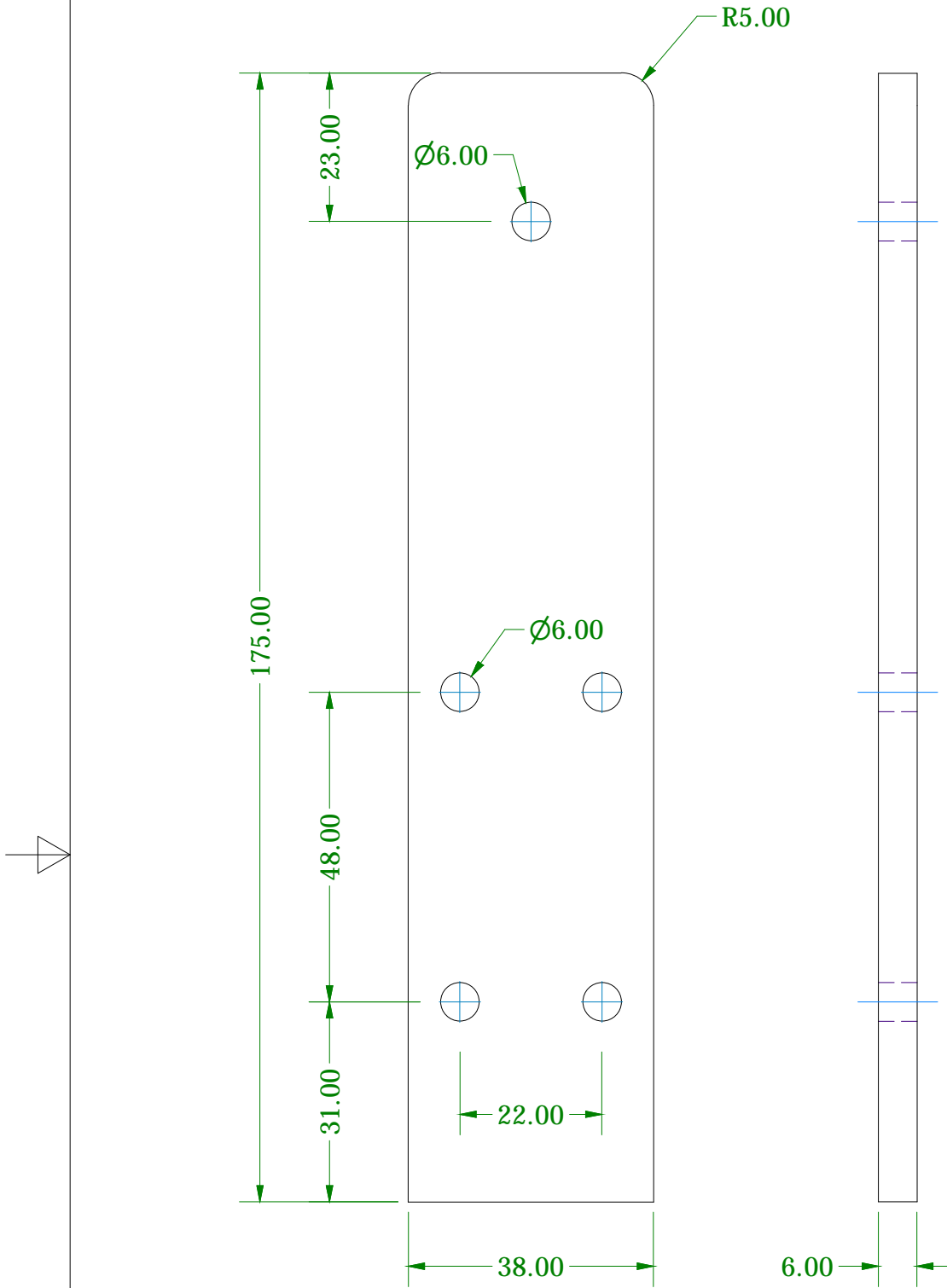


IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Base de Electroimán			
2	1	Guía de Husillo			
3	1	Carrito movil			
4	1	Muesca de control			
5	1	Electroimán de 12 VDC, 25 Kg			HCN- P40/20
6	2	Rodamiento lineal de billas			Ø19
7	2	Anillo de Carcasa Ø22		SAE 1060	
8	1	Tor. allen Hexagonal M5 x 10	DIN 912	8.8	
9	4	Arandela plana M5	DIN 988	5	
10	6	Tor. allen Hexagonal M5 x 20	DIN 912	8.8	
11	6	Tuerca hexagonal M6	DIN 439B	5	

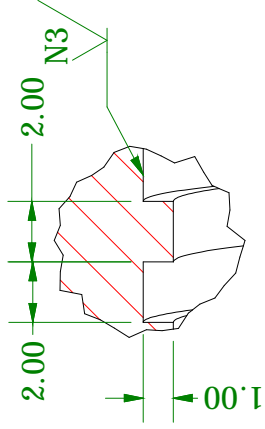
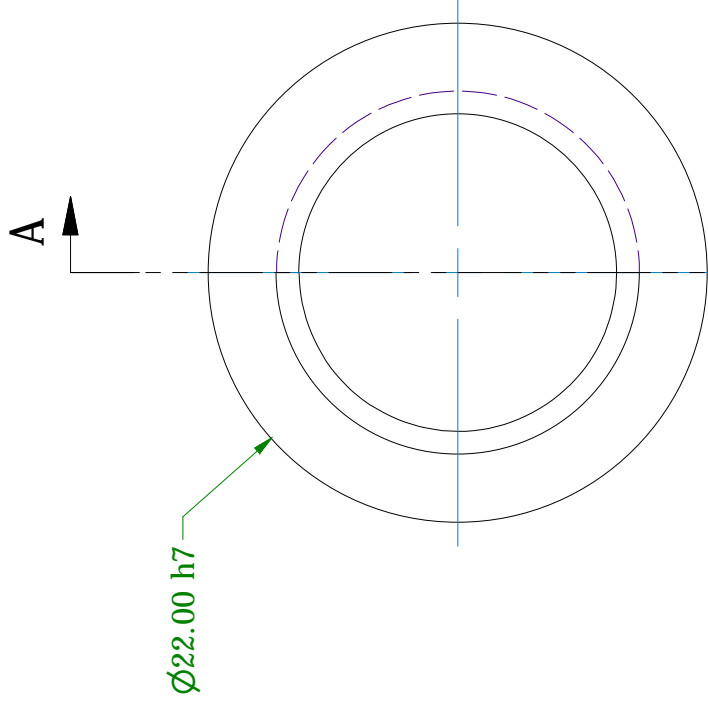
UNIVERSIDAD CATOLICA DE SANTA MARIA  
FACULTAD DE CIENCIAS FISICAS Y FORMALES

METODO DE PROYECCION	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA
	<b>Carro transversal</b>	1:1

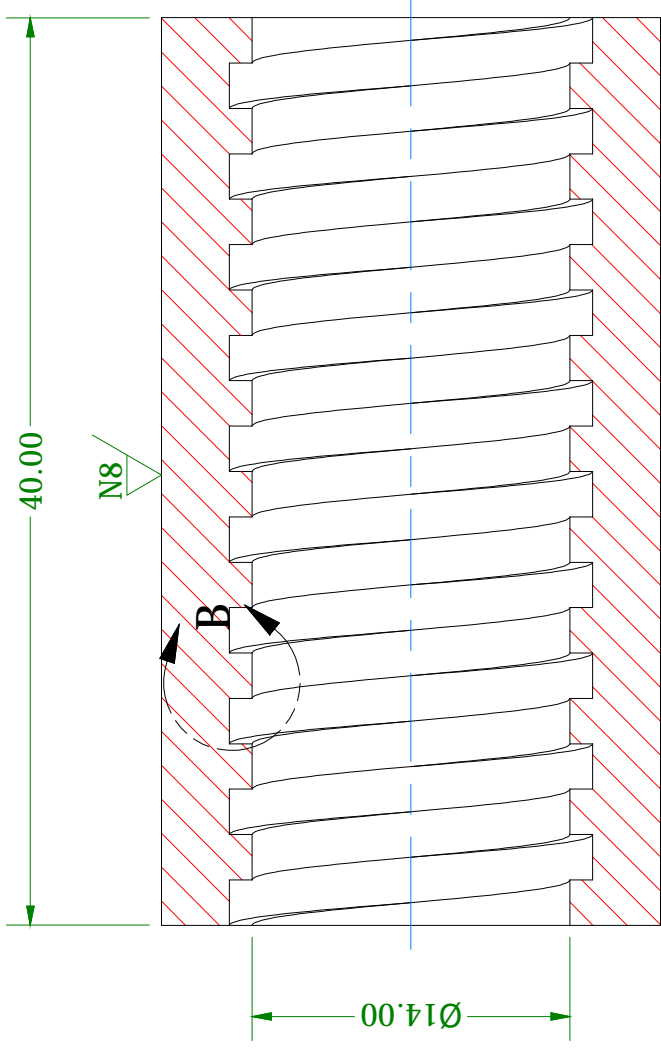
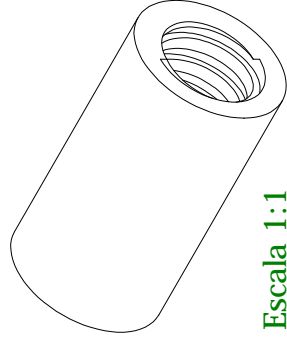
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A3 7 of 32



ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Base de Electroimán</b>	ESCALA  1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 8 of 32

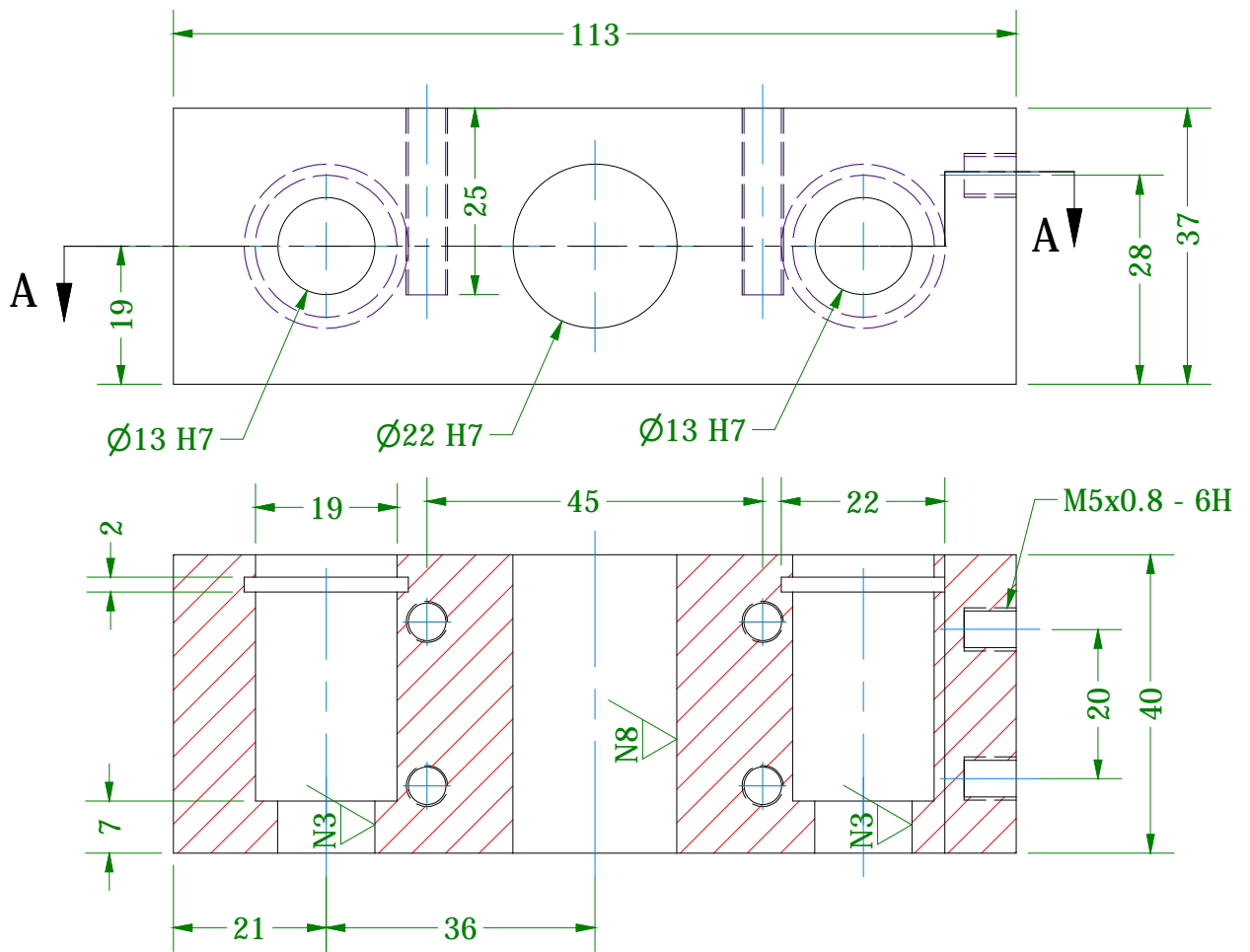


Detalle B  
Escala 4 : 1

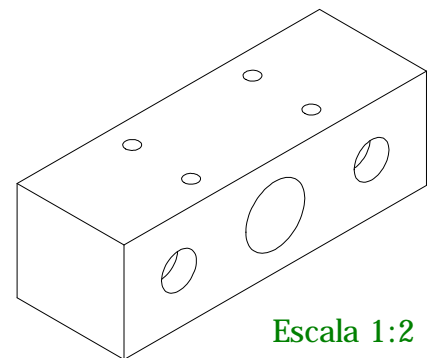


Sección A-A  
Escala 3: 1

ACABADO SUPERFICIAL N9	TOLERANCIA GENERAL DIN 768 TG-MEDIO	MATERIAL: BRONCE SAE 62
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA 3:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 9 of 32

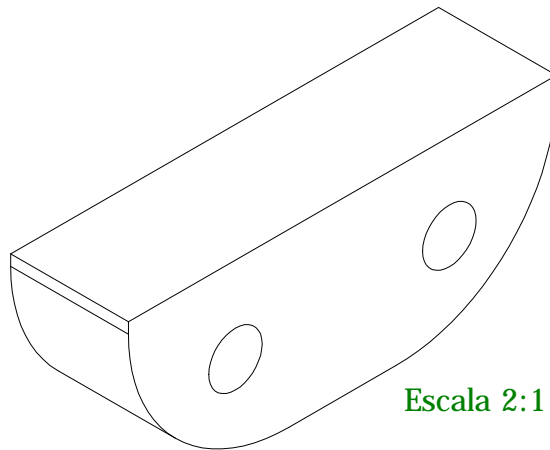
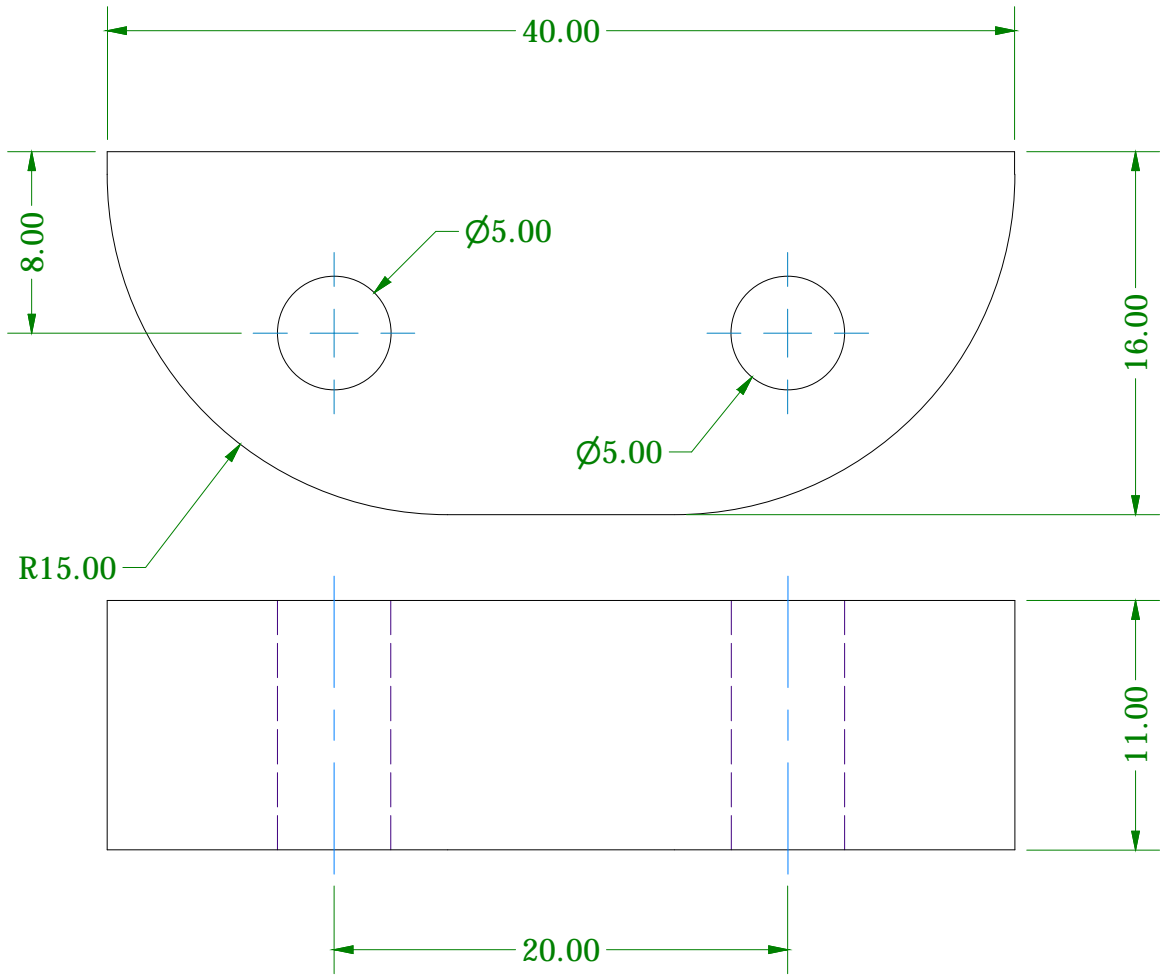


Seccion A-A  
Escala 1 : 1



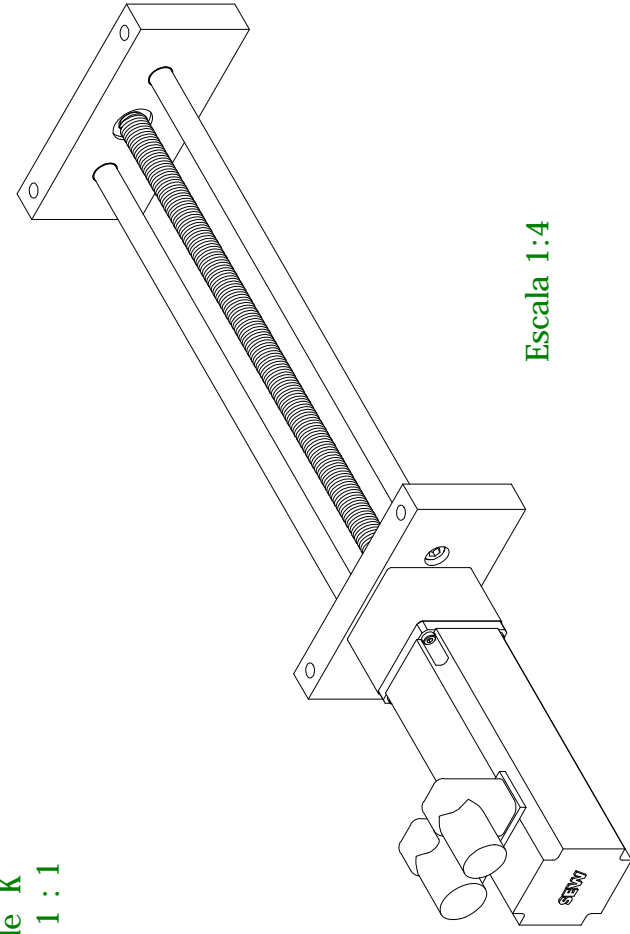
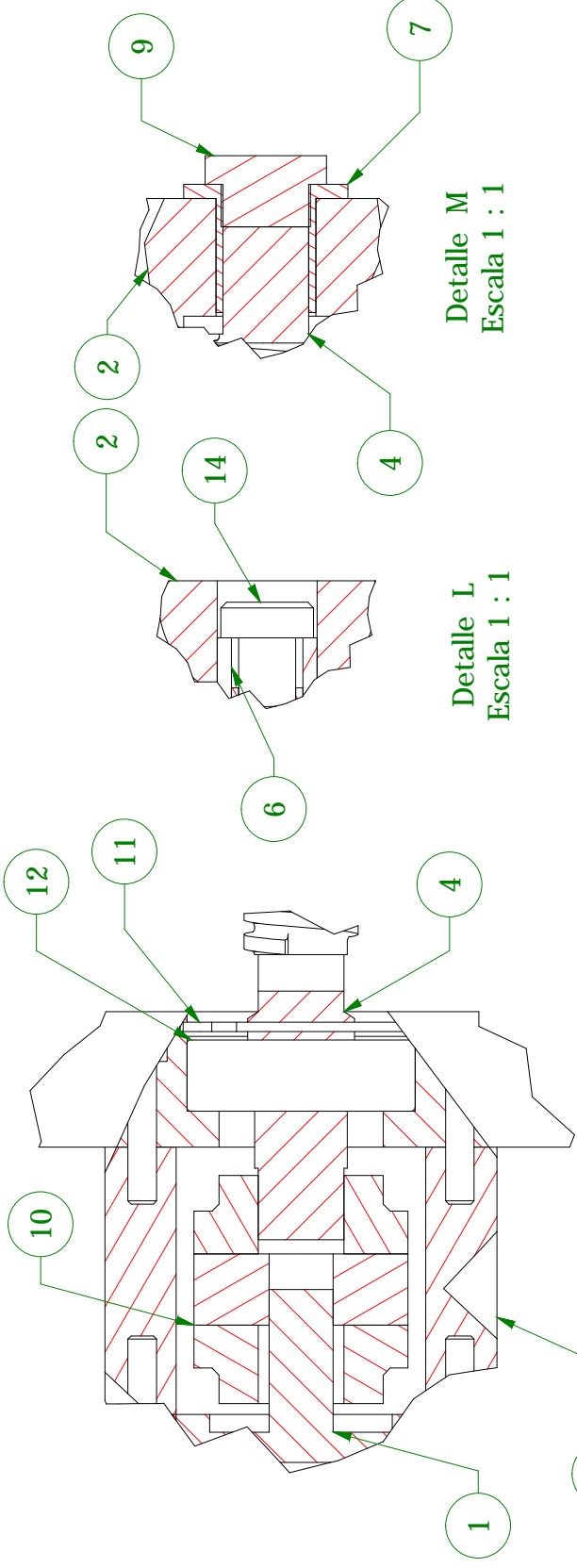
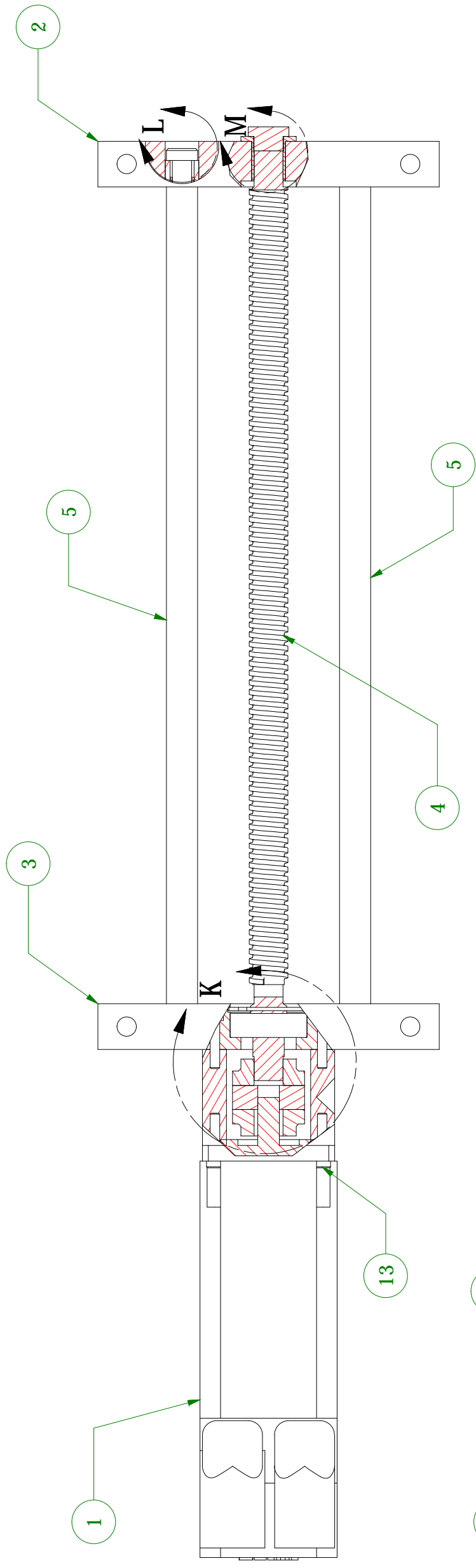
Escala 1:2

ACABADO SUPERFICIAL N9	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM 36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Carrito móvil</b>	ESCALA  1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 10 of 32



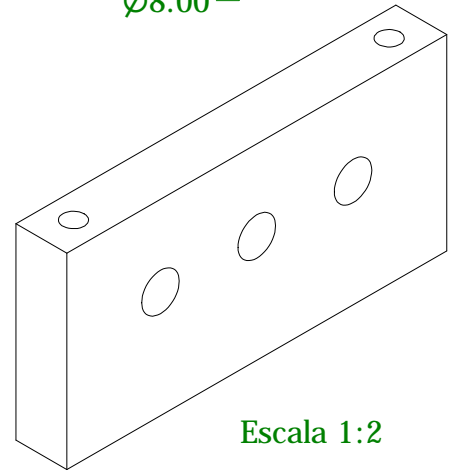
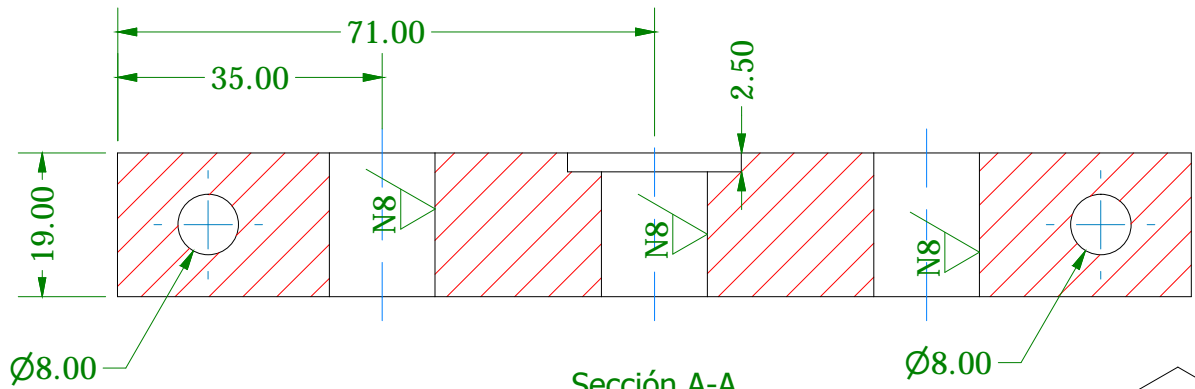
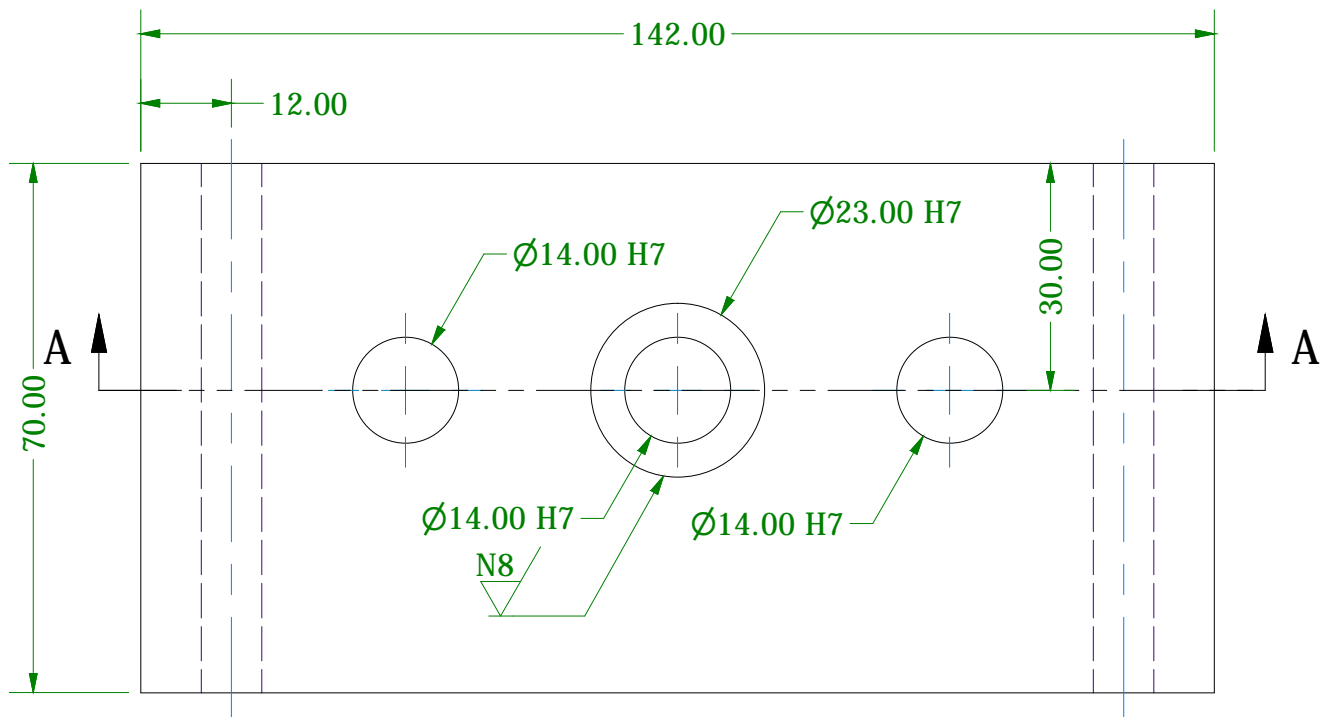
Escala 2:1

ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: NYLON
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA 3:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 11 of 32



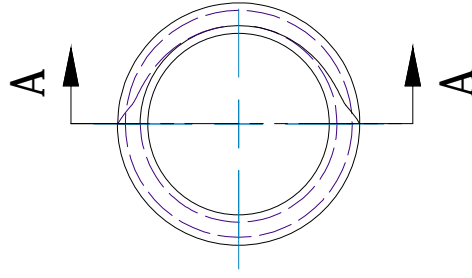
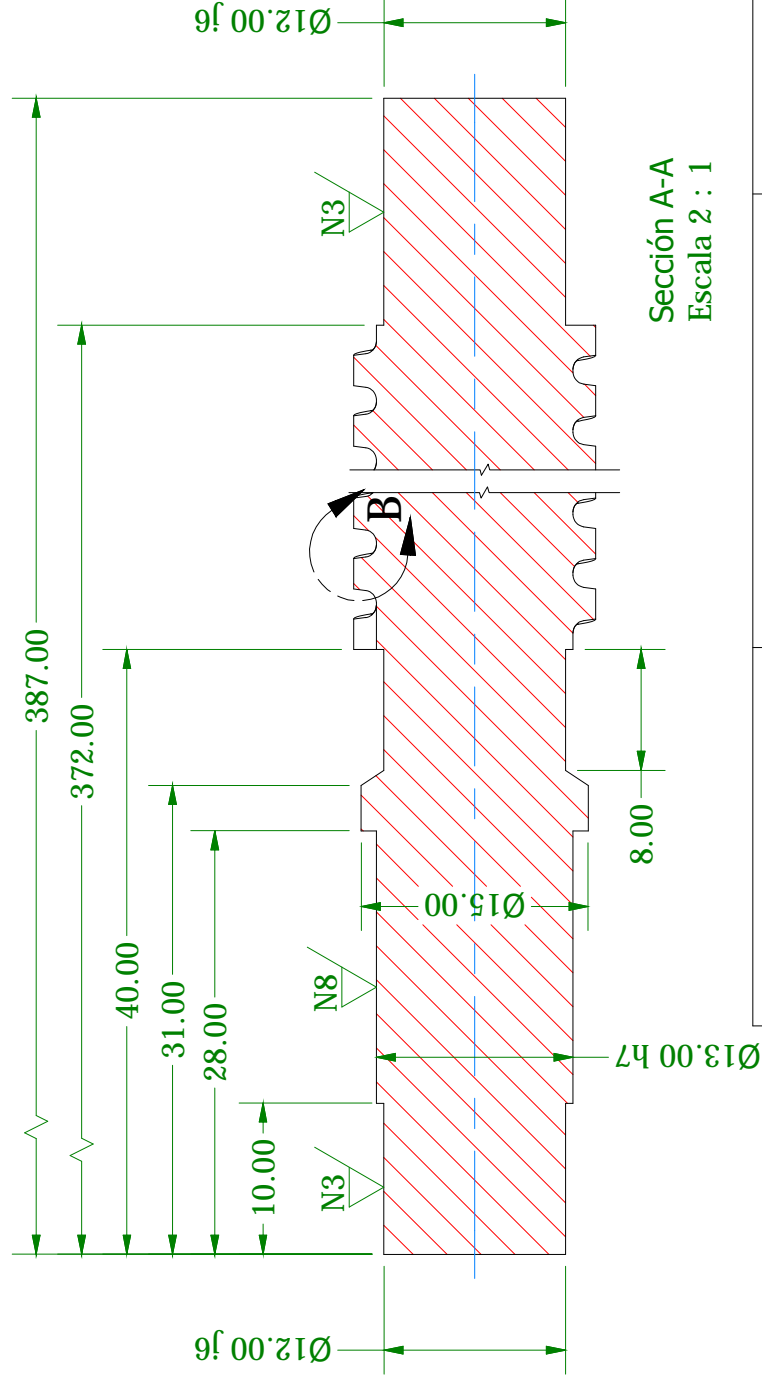
IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Servomotor Brushless 0.8 Nm			SEW CMP40M
2	1	Soporte lateral libre			
3	1	Soporte lado del motor			
4	1	Husillo sin fin			
5	2	Guia lateral			
6	4	Buje de guias			
7	1	Buje de husillo			
9	1	Tapon Lateral			
8	1	Tapa de motor			
10	1	Acoplamiento de Ø9-Ø12			
11	1	Anillo de Carcasa Ø34		SAE 1060	
12	1	Rodamiento de bolas Ø30			SKF 6201-RS1
13	8	Tor. allen Hexagonal M4 x 20	DIN 912	8.8	
14	4	Tor. allen Hexagonal M8 x 16	DIN 912	8.8	

ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL:
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA 1:2
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A3 12 of 32

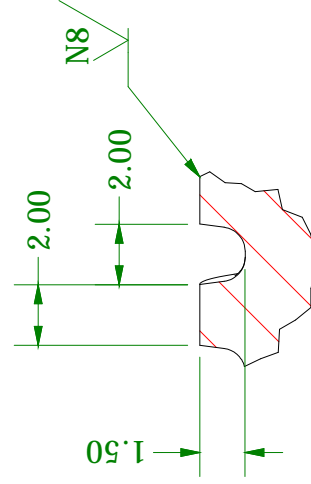


ACABADO SUPERFICIAL N9	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Soporte lateral libre</b>	ESCALA  1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 13 of 32





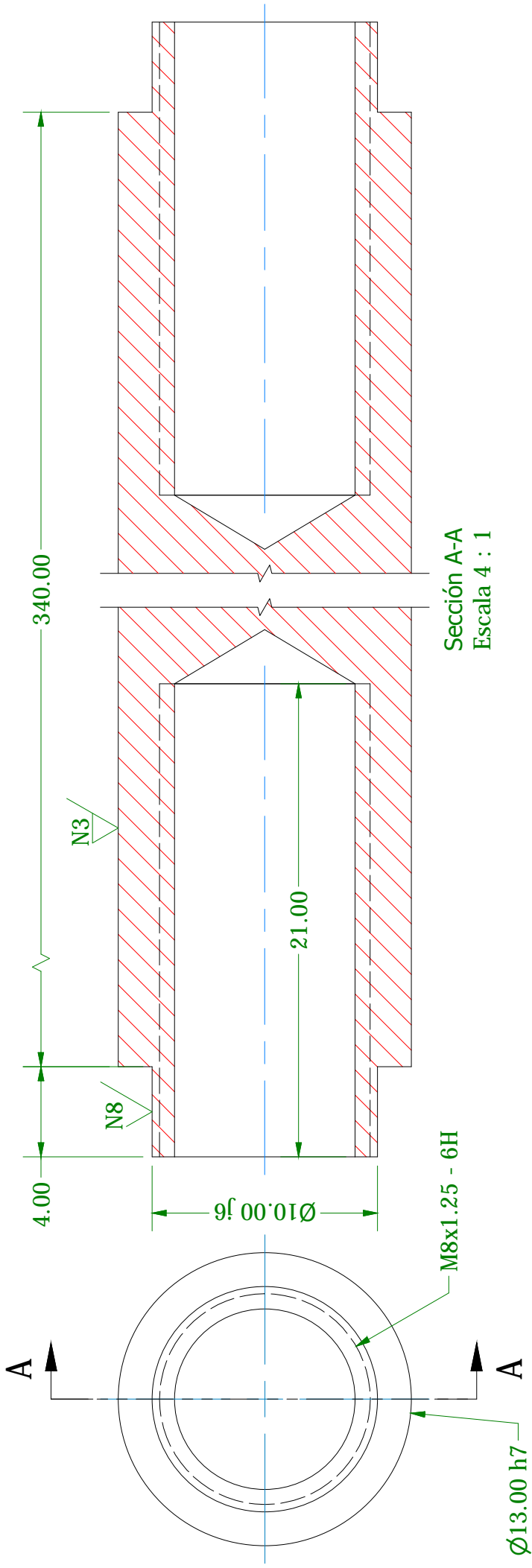
Sección A-A  
Escala 2 : 1



Detalle B  
Escala 4 : 1

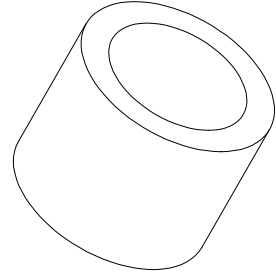
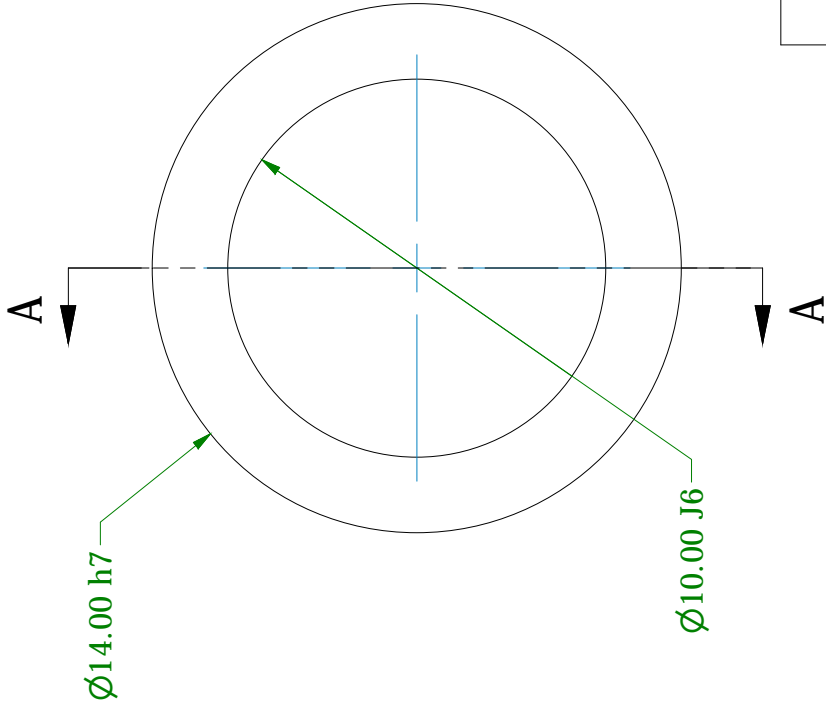
ACABADO SUPERFICIAL N9	TOLERANCIA GENERAL	MATERIAL: Ck45
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Husillo sin fin</b>	ESCALA 2:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 15 of 32



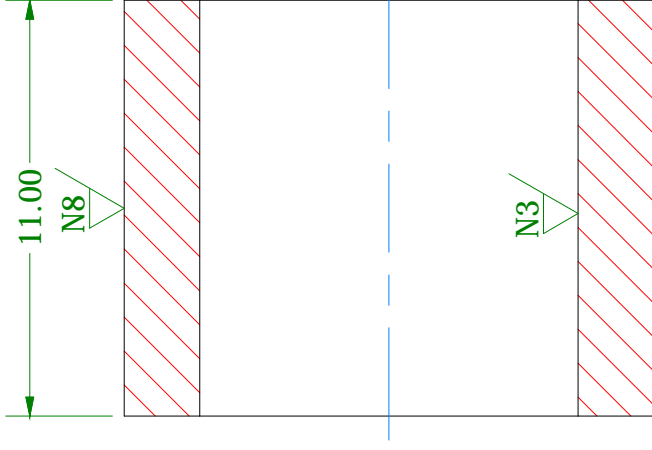


Sección A-A  
Escala 4 : 1

ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL:
		Ck45
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA
	<b>Guía lateral</b>	4:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 16 of 32

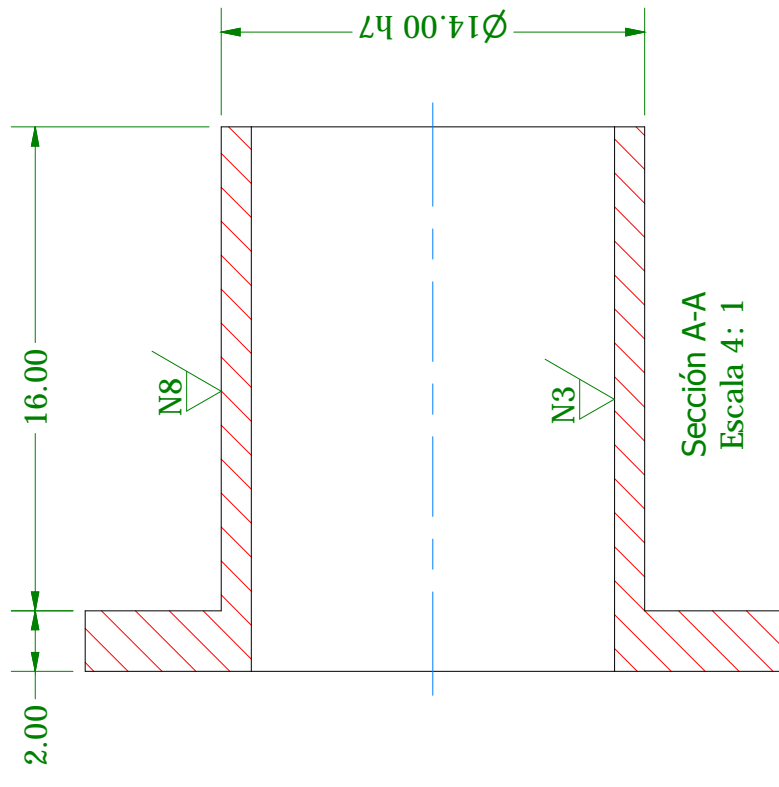
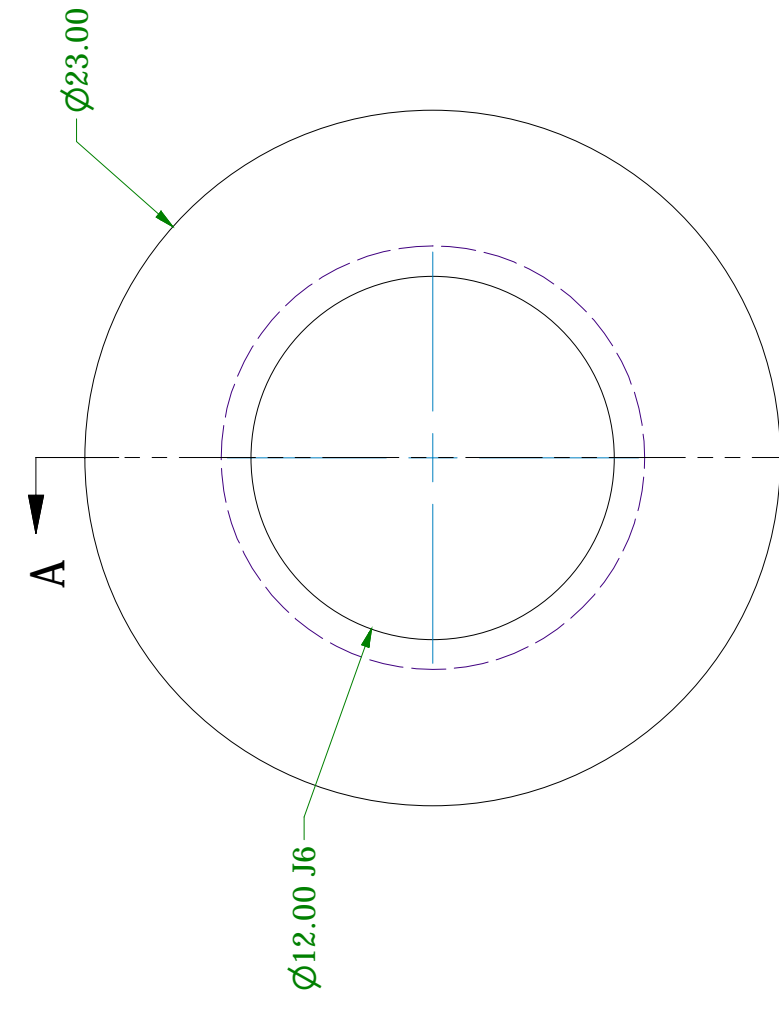


Escala 2:1

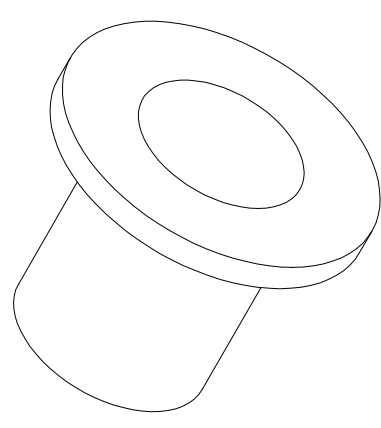


Sección A-A  
Escala 5:1

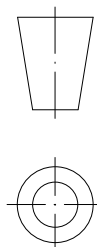
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL:
		BRONCE SAE 62
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA
	<b>Buje de guias</b>	5:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 17 of 32

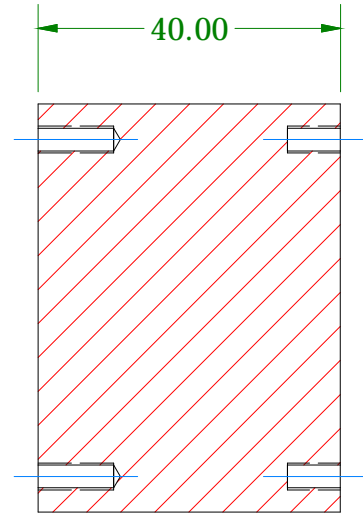
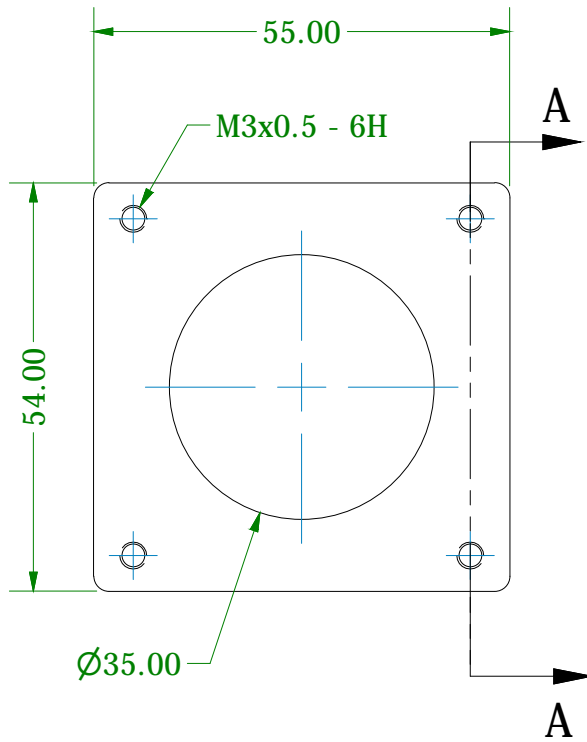


Sección A-A  
Escala 4: 1

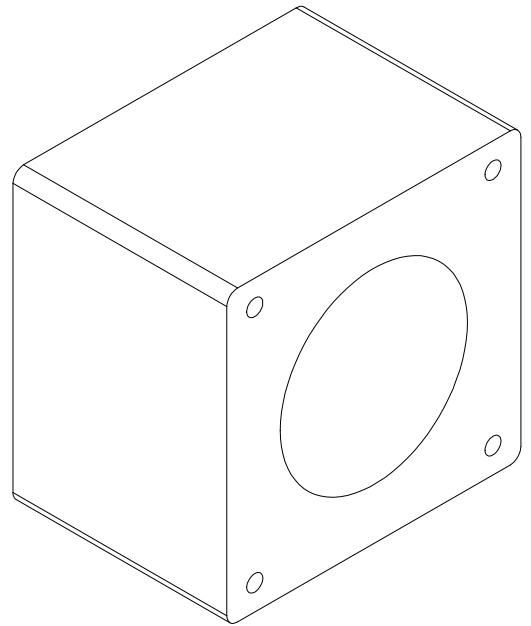


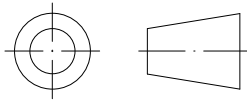
Escala 2:1

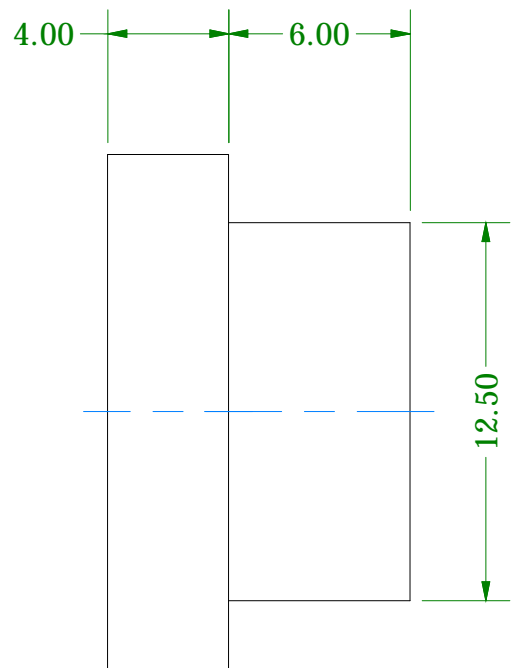
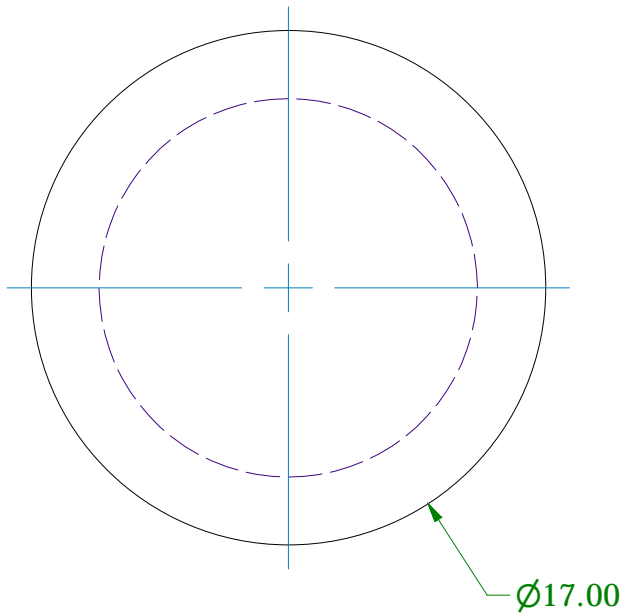
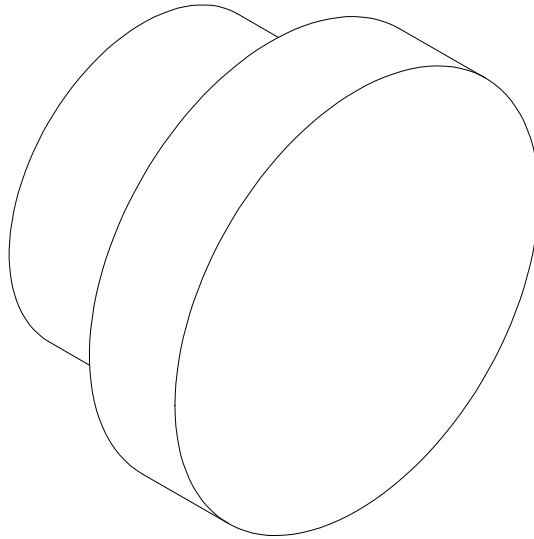
ACABADO SUPERFICIAL N9	TOLERANCIA GENERAL	MATERIAL: BRONCE SAE 62
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA 4:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
<b>Buje de husillo</b>		
PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL		
LAMINA: A4 18 of 32		



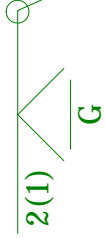
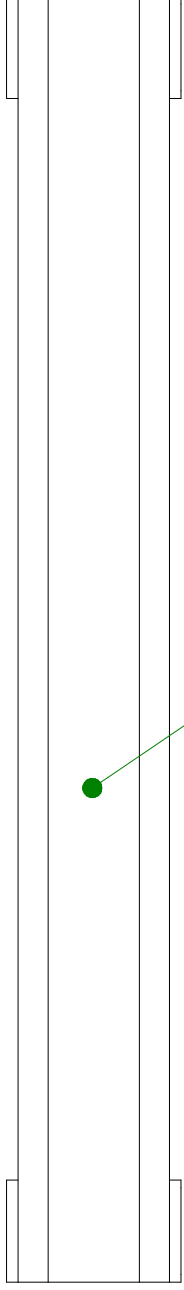
Sección A-A  
Escala 1 : 1



ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Tapa de motor</b>	ESCALA  1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 19 of 32



ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: NYLON
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <h2 style="text-align: center;">Tapon Lateral</h2>	ESCALA  <b>4:1</b>
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 20 of 32

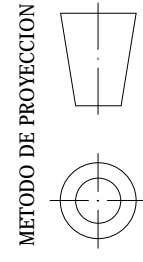


IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Base de piezas			
2	2	Perfil de apoyo			
3	1	Cobertor negro de base			
4	4	Tapon de base			

**UNIVERSIDAD CATOLICA DE SANTA MARIA**  
 FACULTAD DE CIENCIAS FISICAS Y FORMALES

PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA  
 SISTEMA DE CLASIFICACION  
 POR VISION ARTIFICIAL

## Base externa



ESCALA

1:2

2008220181

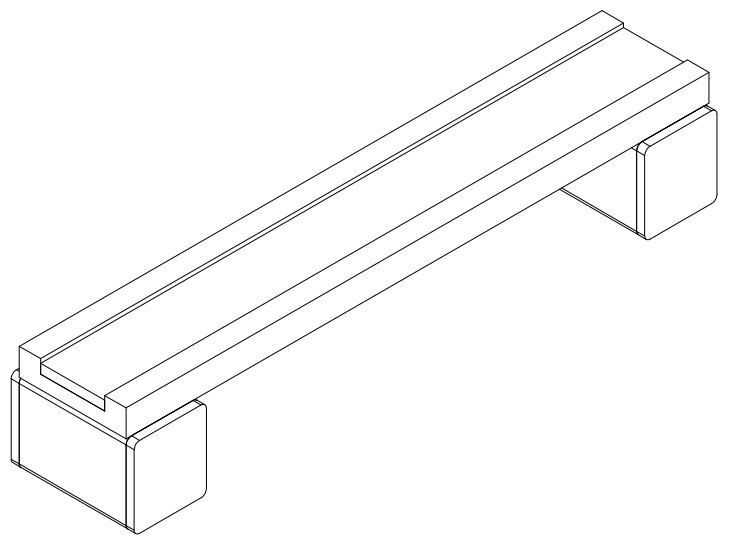
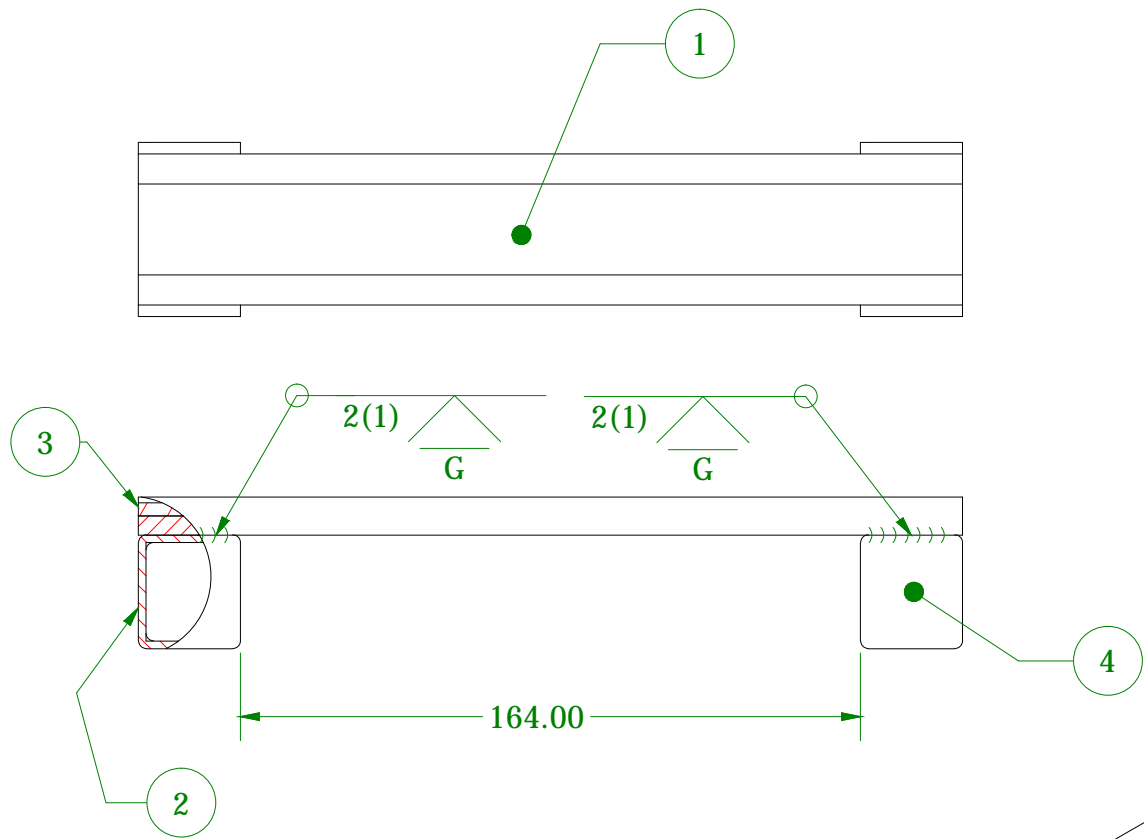
ZARATECM, RODRIGUEZSE

FECHA:  
25/02/2015

LAMINA: A4  
21

**Escala 1:4**

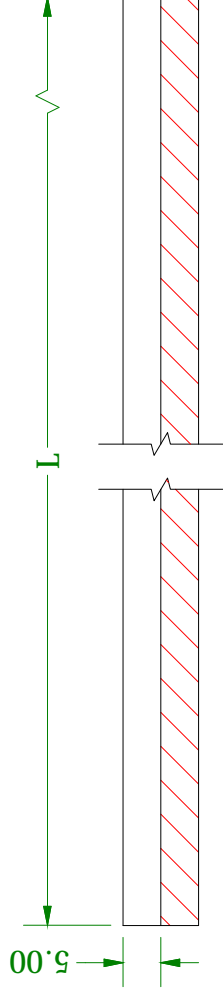
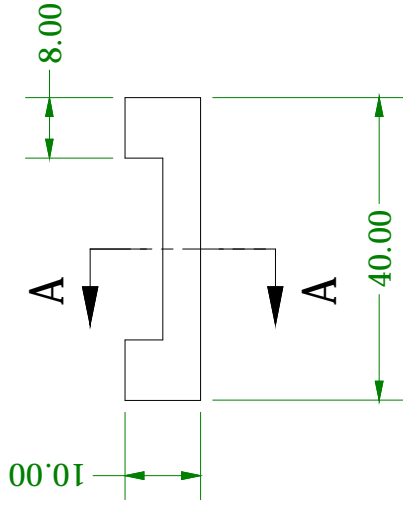




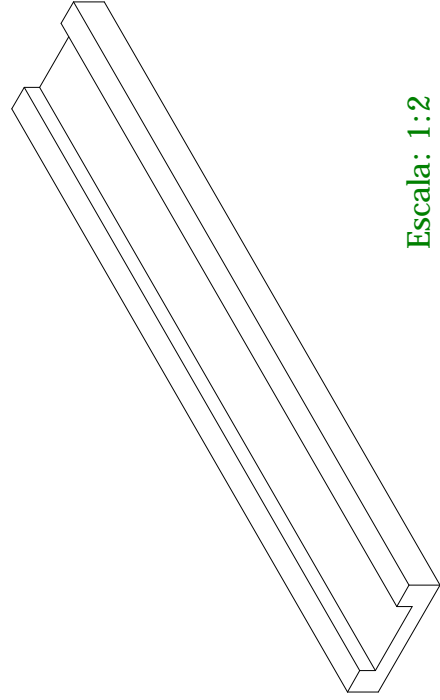
IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Base			
2	2	Perfil de apoyo			
3	1	Cobertor negro de base			
4	4	Tapon de base			

**UNIVERSIDAD CATOLICA DE SANTA MARIA**  
FACULTAD DE CIENCIAS FISICAS Y FORMALES

METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Base para piezas</b>	ESCALA  1:2
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 22

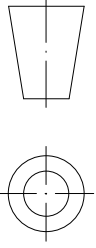


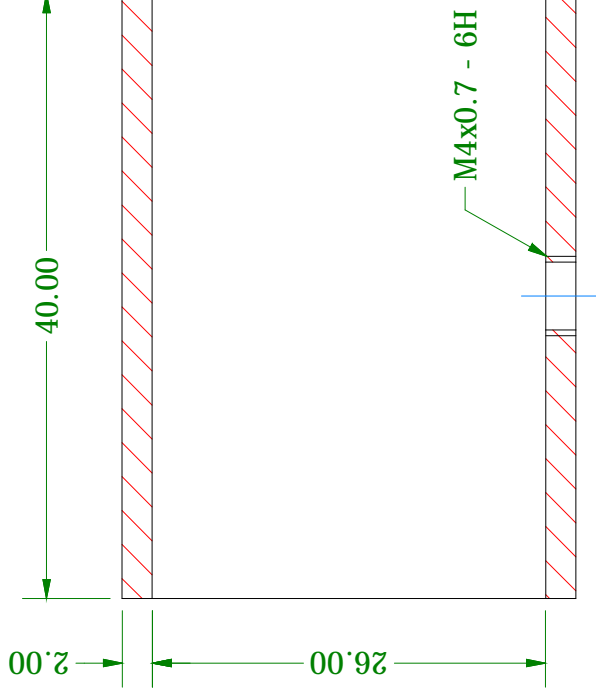
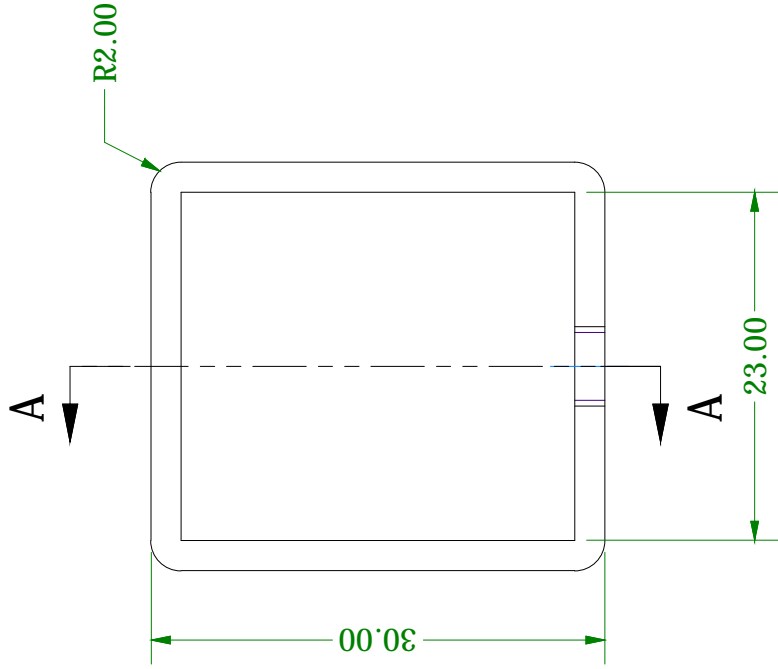
Sección A-A  
Escala 1 : 1



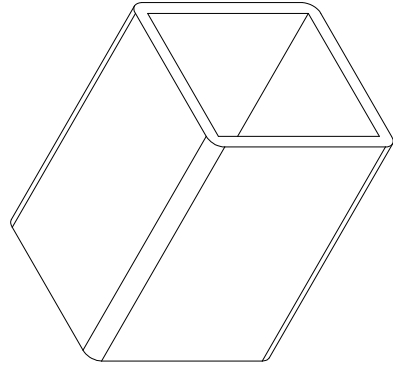
Escala: 1:2

Nota: Base para piezas: L= 218mm  
Base externa: L= 340mm

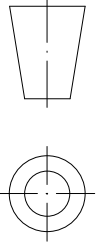
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Base</b>	ESCALA 1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015 LAMINA: A4 23 of 32

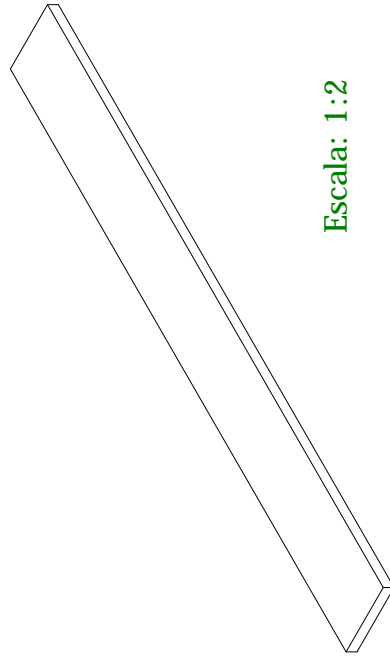
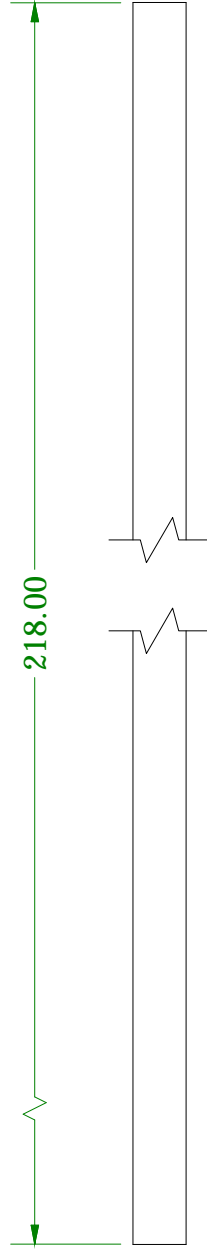
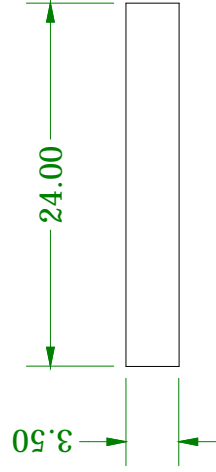


Sección A-A  
Escala 2 : 1



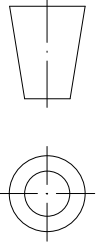
Escala 1:1

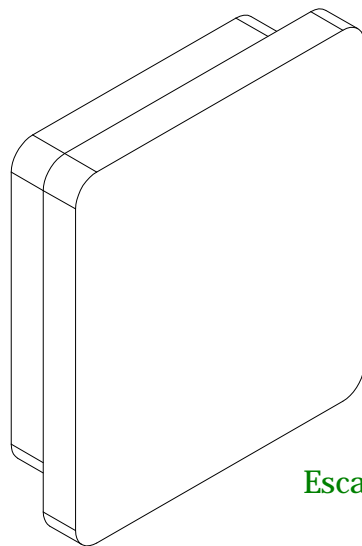
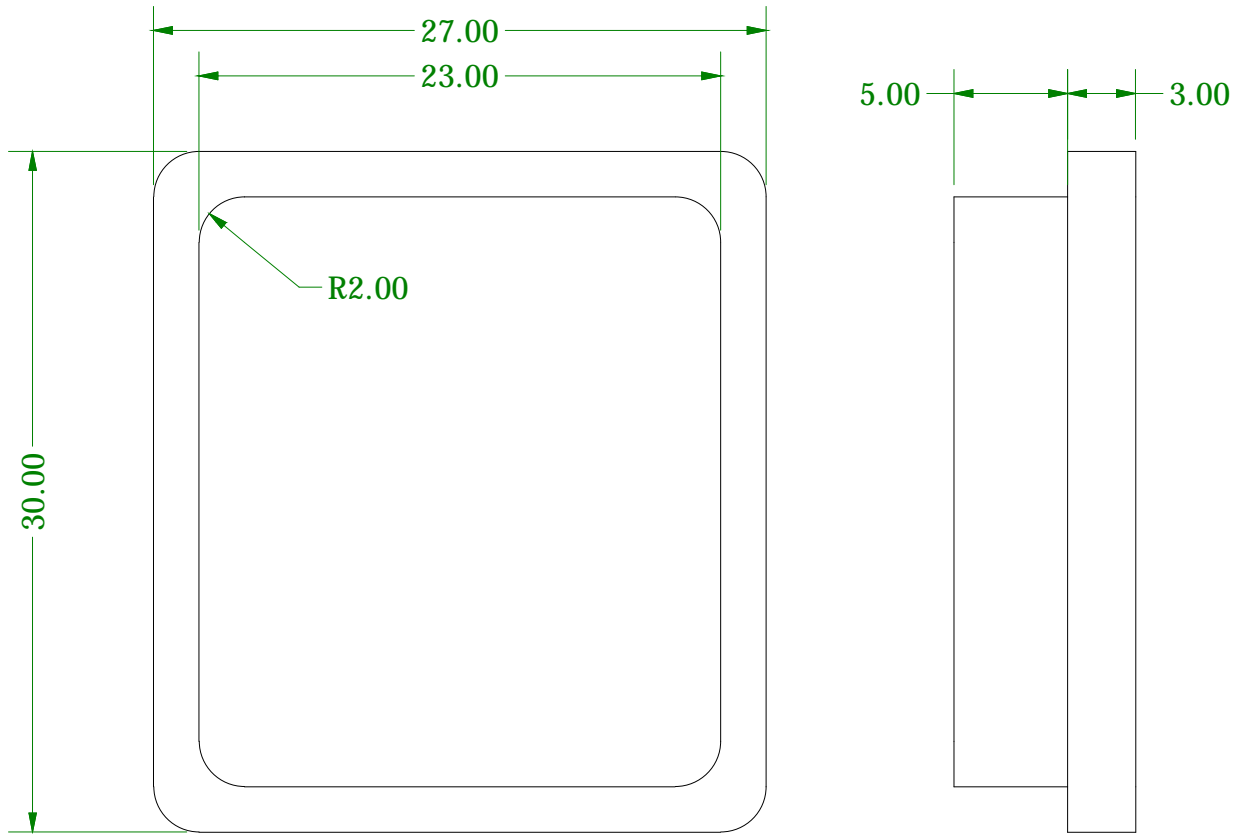
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Perfil de apoyo</b>	ESCALA 2:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 24 of 32



**Escala: 1:2**

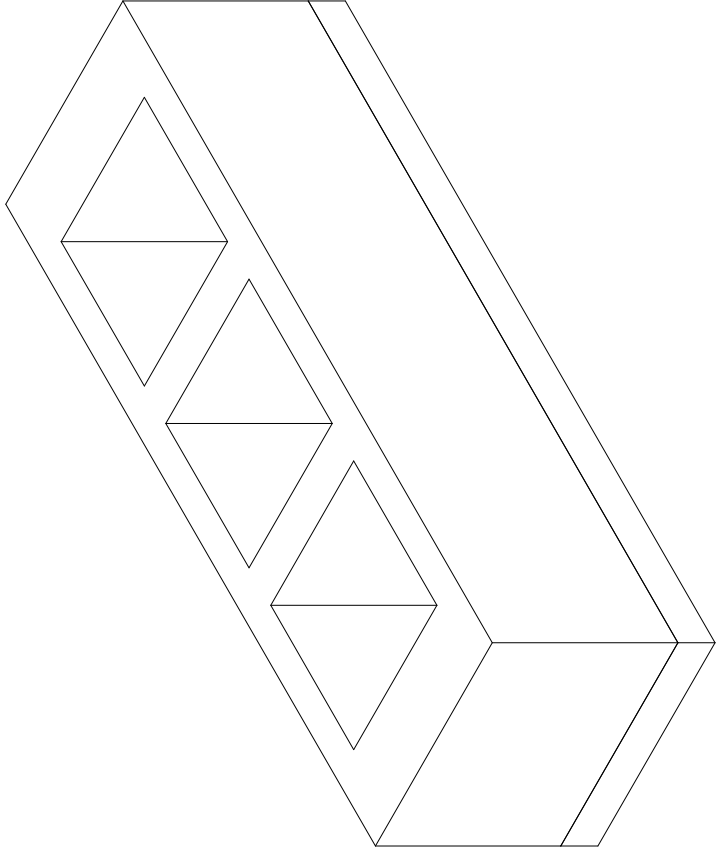
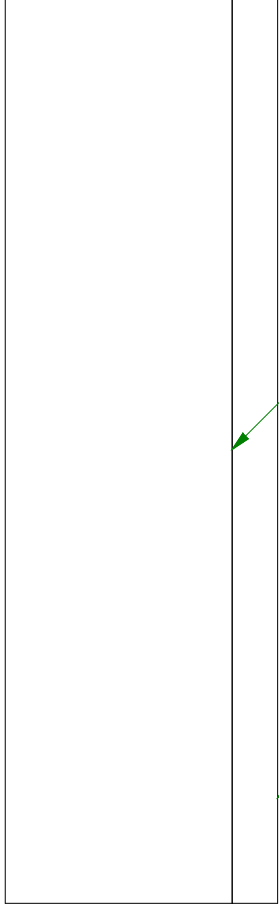
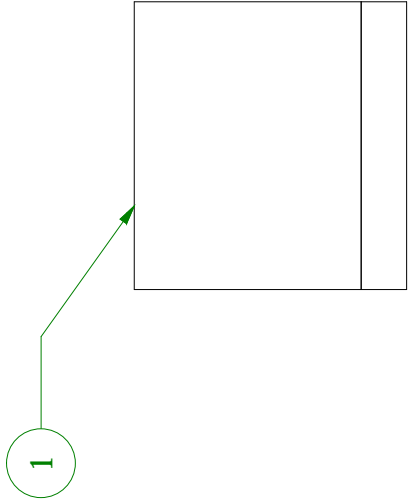
**Nota: Base de piezas: L = 218 mm**  
**Base externa: L = 340 mm**

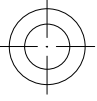
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: POLIMERO
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA 1:1
2008220181	<b>Cobertor negro de base</b> ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015 LAMINA: A4 25 of 32

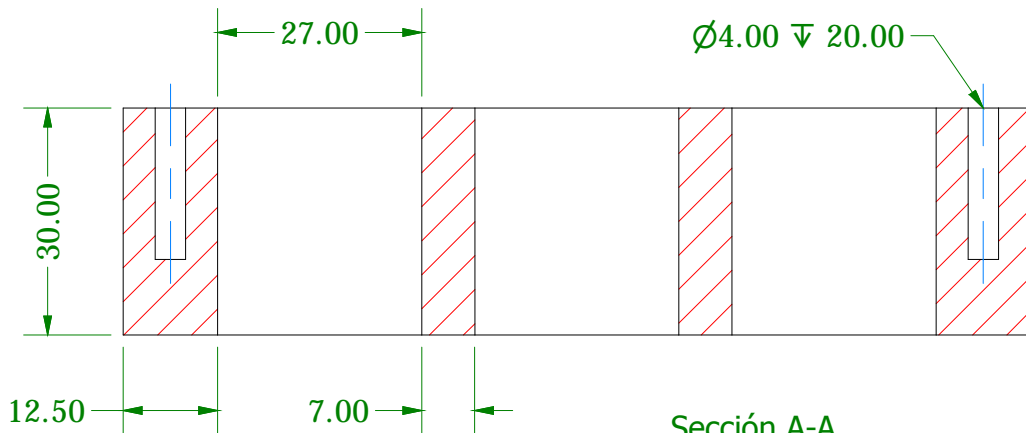
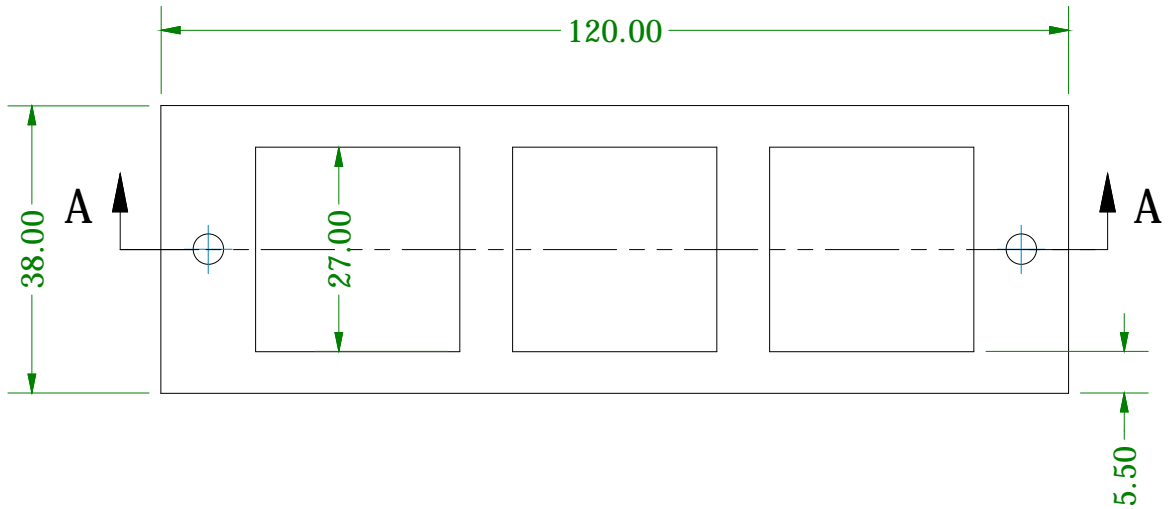


Escala 2:1

ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: NYLON
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <h2 style="text-align: center;">Tapon de base</h2>	ESCALA  <h3 style="text-align: center;">1:3</h3>
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 26 of 32

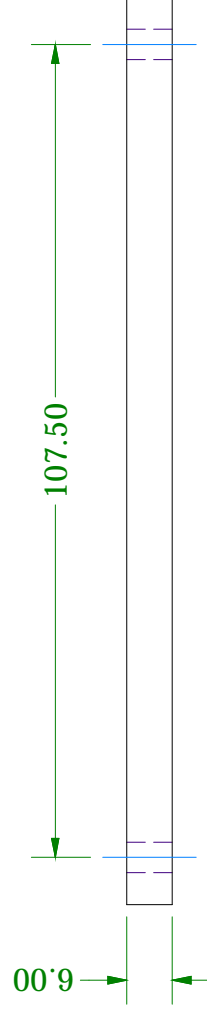
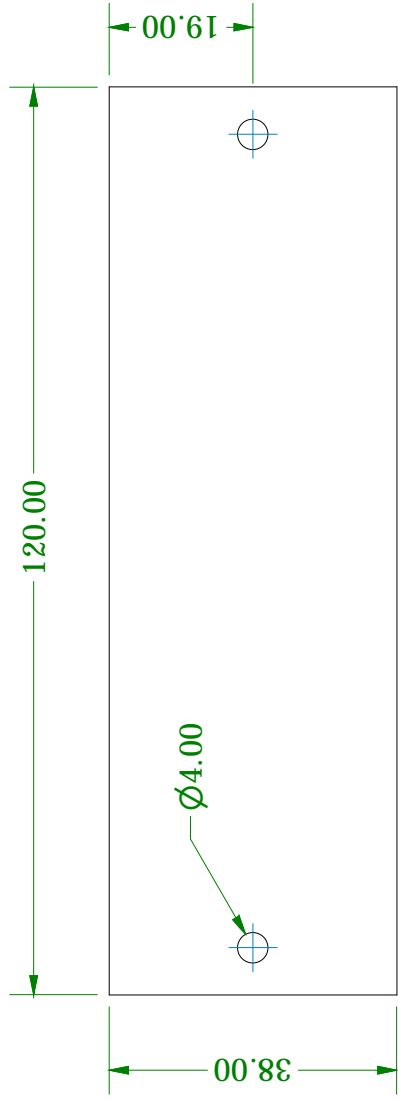


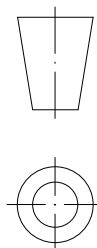
IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Bloque de recepción			
2	1	Tapa inferior			
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES					
METODO DE PROYECCION 		PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL		ESCALA 1:1	
2008220181		<b>Recipiente de recepción</b> ZARATECM, RODRIGUEZSE		FECHA: 25/02/2015 LAMINA: A4 27	

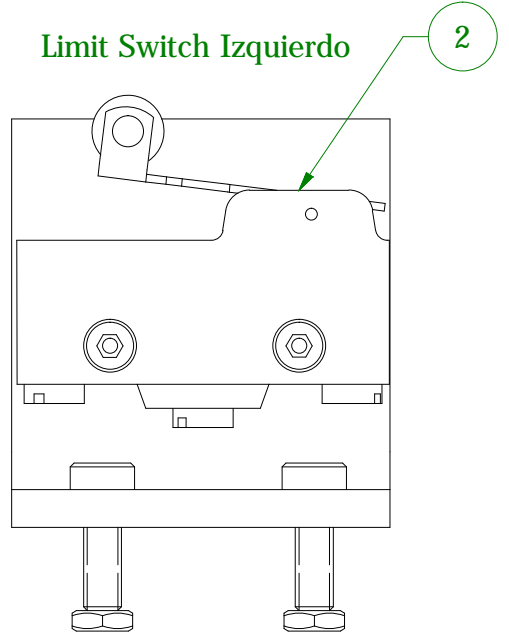
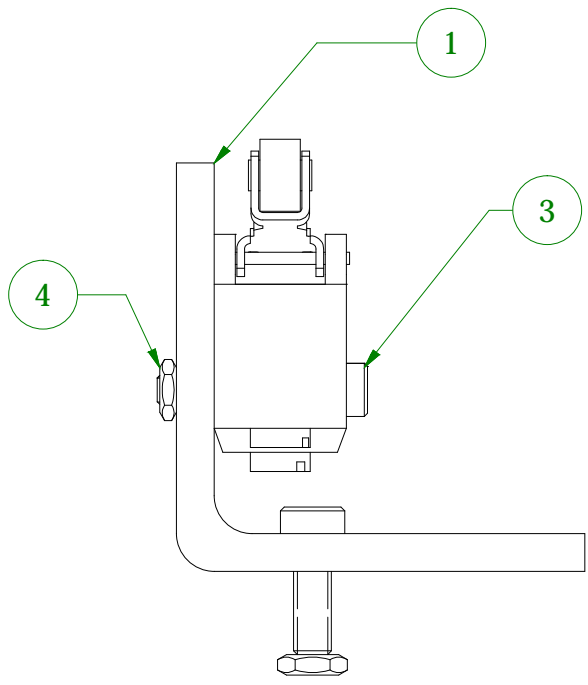


Sección A-A  
Escala 1 : 1

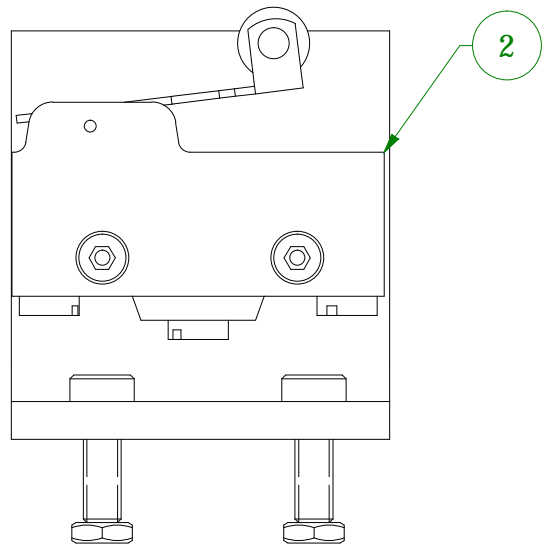
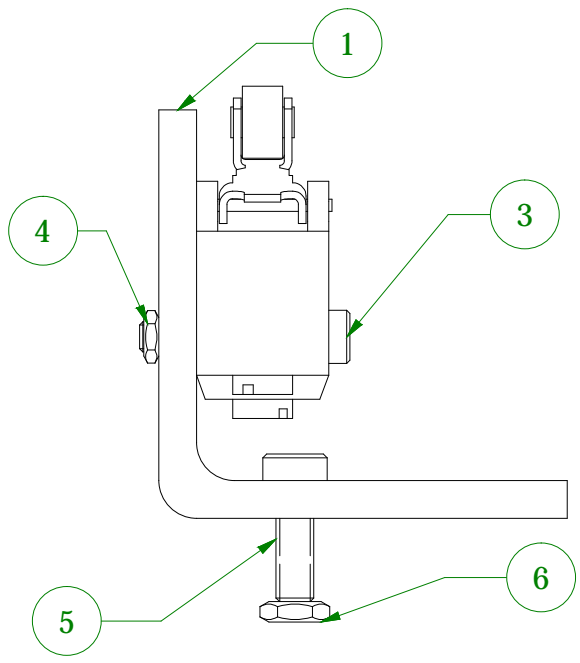
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <h2 style="text-align: center;">Bloque de recepción</h2>	ESCALA  <b>1:1</b>
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 28 of 32



ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Tapa inferior</b>	ESCALA 1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 29 of 32



Limit Switch Izquierdo

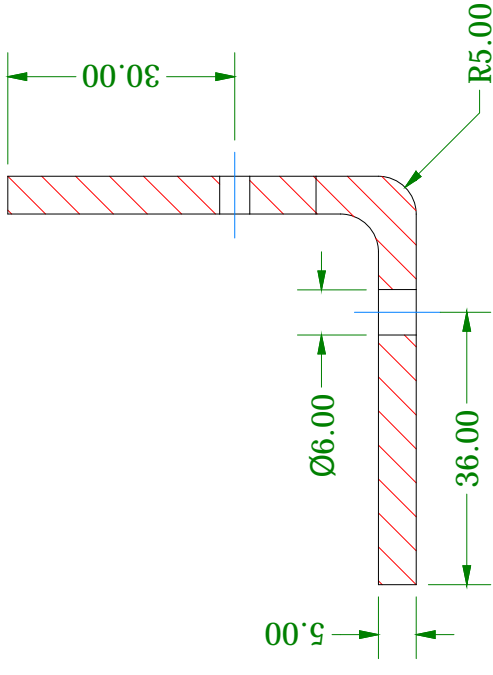
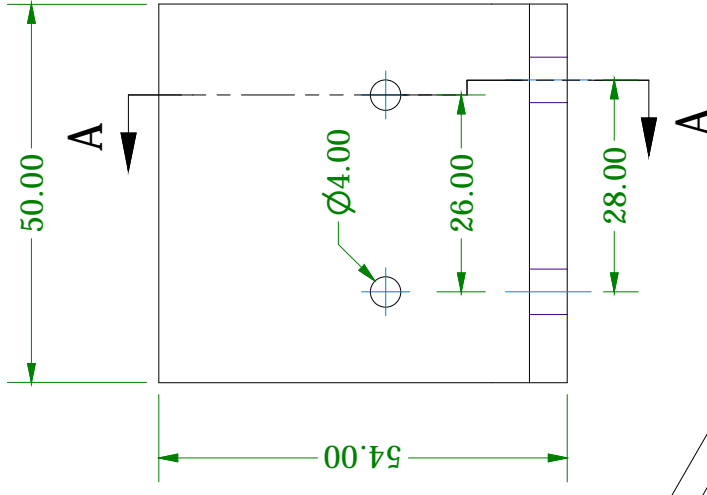


Limit Switch Derecho

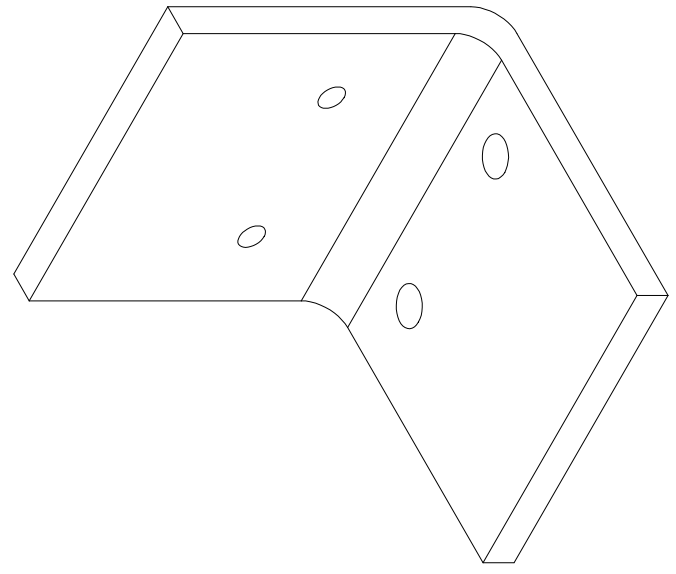
IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	2	Base para sensor			
2	2	Sensor de final de carrera			De palanca
4	4	Tor. allen Hexagonal M3 x 25	DIN 912	8.8	
3	4	Tuerca Hexagonal M3	DIN 439B		
5	4	Tor. allen Hexagonal M5 x 16	DIN 912		
6	4	Tuerca Hexagonal M5	DIN 439B		

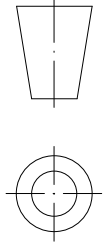
**UNIVERSIDAD CATOLICA DE SANTA MARIA**  
FACULTAD DE CIENCIAS FISICAS Y FORMALES

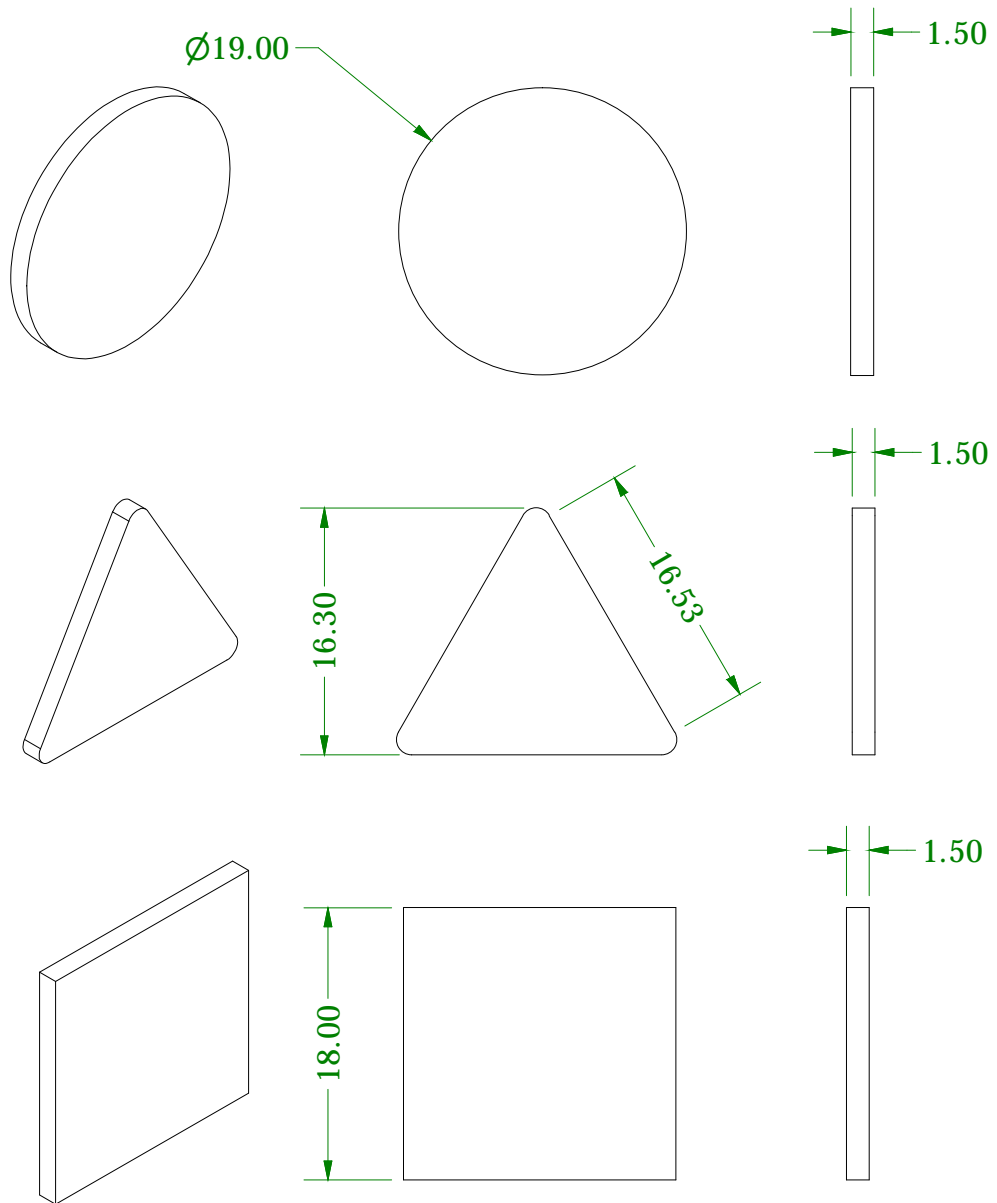
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Limit switch izquierda</b>	ESCALA  1:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 30



Sección A-A  
Escala 1 : 1



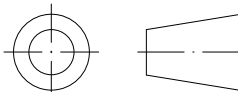
ACABADO SUPERFICIAL	TOLERANCIA GENERAL DIN 768 TOLERANCIA MEDIA	MATERIAL: ASTM A36
<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> FACULTAD DE CIENCIAS FISICAS Y FORMALES		
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL	ESCALA 1:1
2008220181	<b>Base para sensor</b> ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015 LAMINA: A4 31 of 32



Nota: Hay variaciones en tamaño en cada tipo de pieza.

IT	QT	DESCRIPCION	NORMA	MATERIAL	OBSERVACIONES
1	1	Circulo		ASTM A36	
2	1	Triangulo		ASTM A36	
3	1	Cuadrado		ASTM A36	

**UNIVERSIDAD CATOLICA DE SANTA MARIA**  
FACULTAD DE CIENCIAS FISICAS Y FORMALES

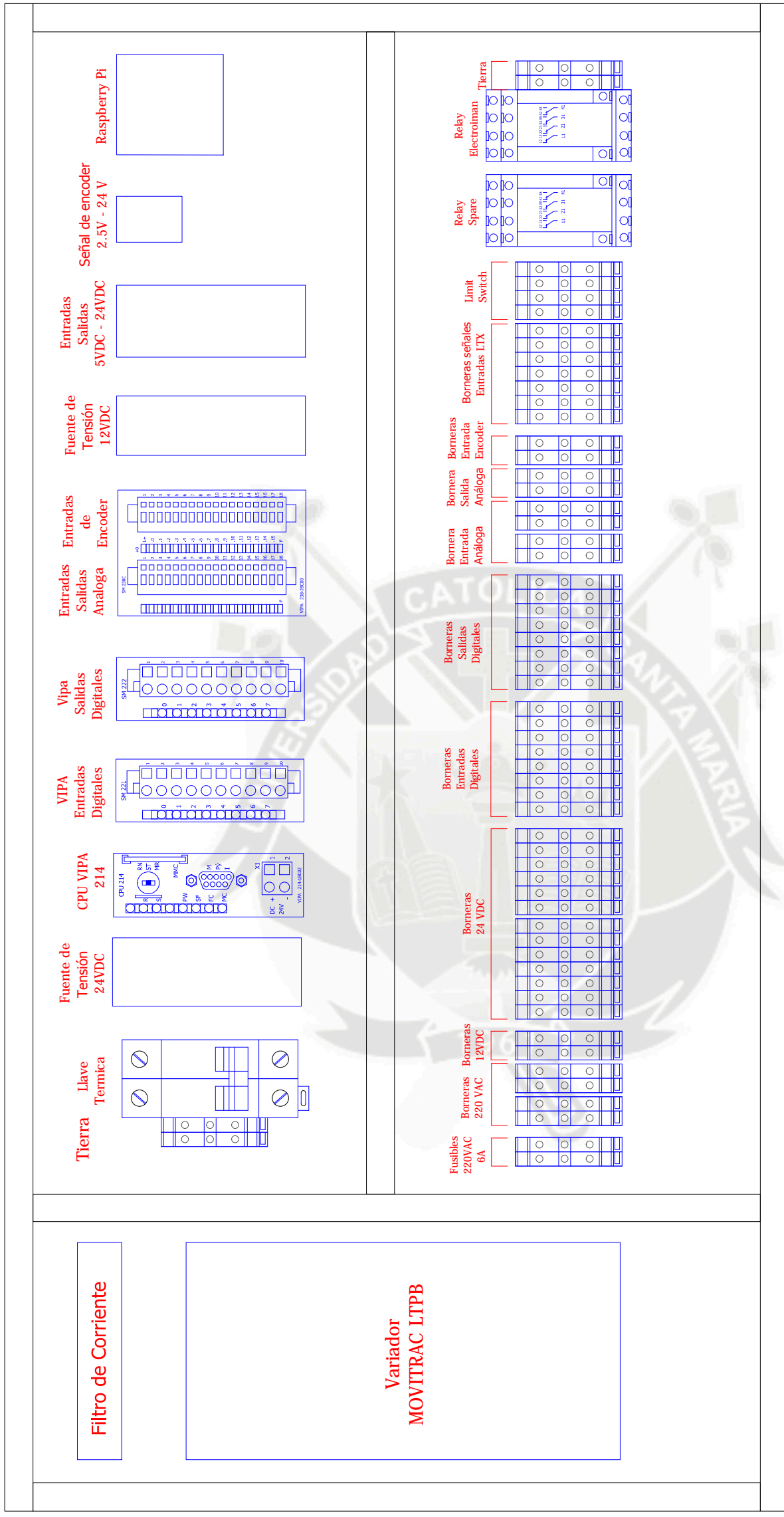
METODO DE PROYECCION 	PROGRAMA PROFESIONAL DE INGENIERIA MECATRONICA SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL <b>Piezas de prueba</b>	ESCALA  2:1
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 25/02/2015
		LAMINA: A4 32

## Anexo 3

## Planos Eléctricos

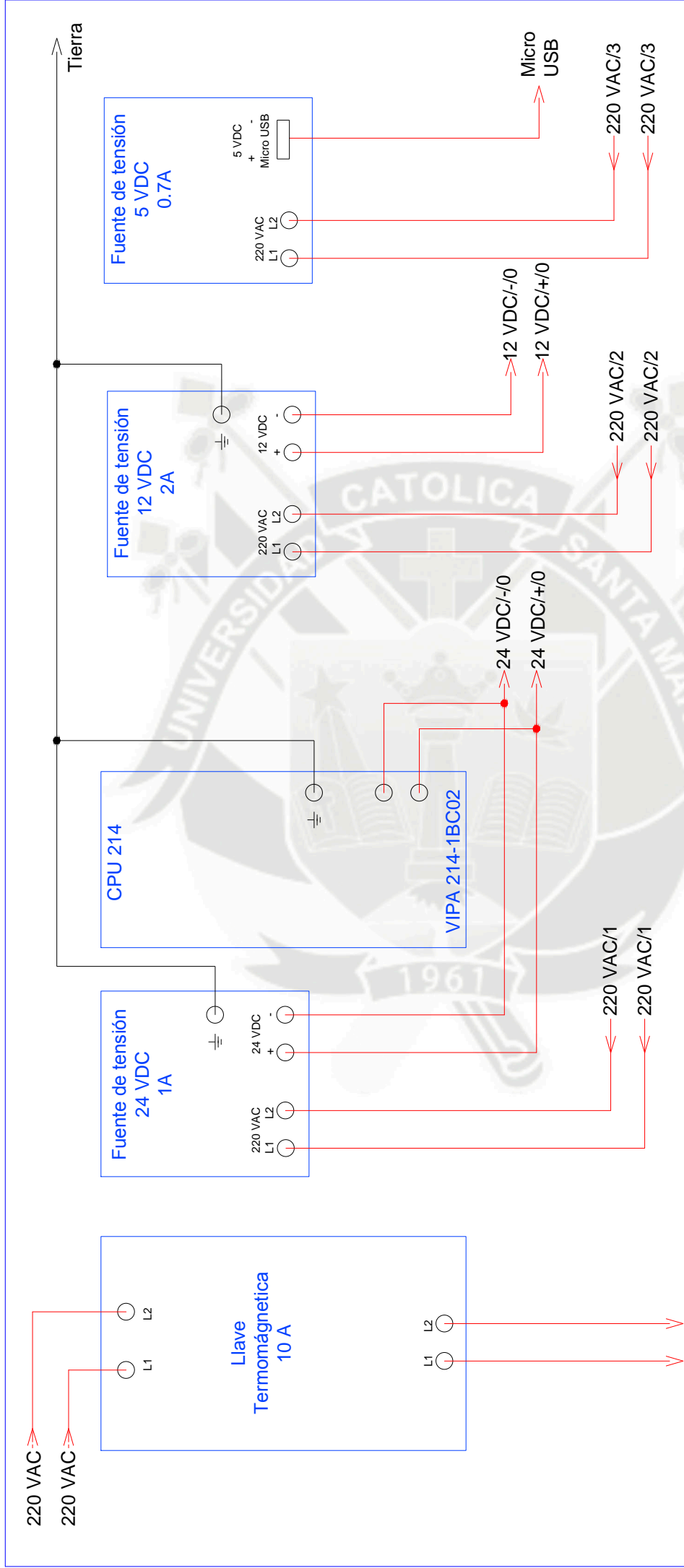


# DISTRIBUCION GENERAL

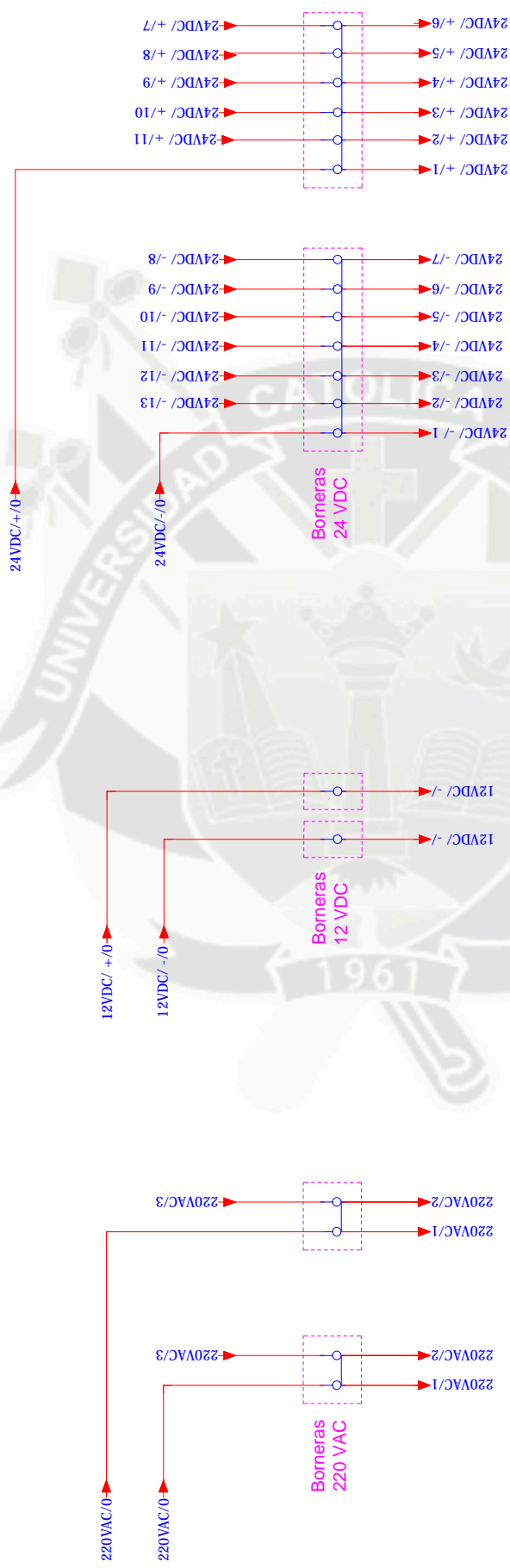


UNIVERSIDAD CATOLICA DE SANTA MARIA  
ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL DISTRIBUCIÓN GENERAL	ESCALA — — —
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015 LAMINA: A3 1/11

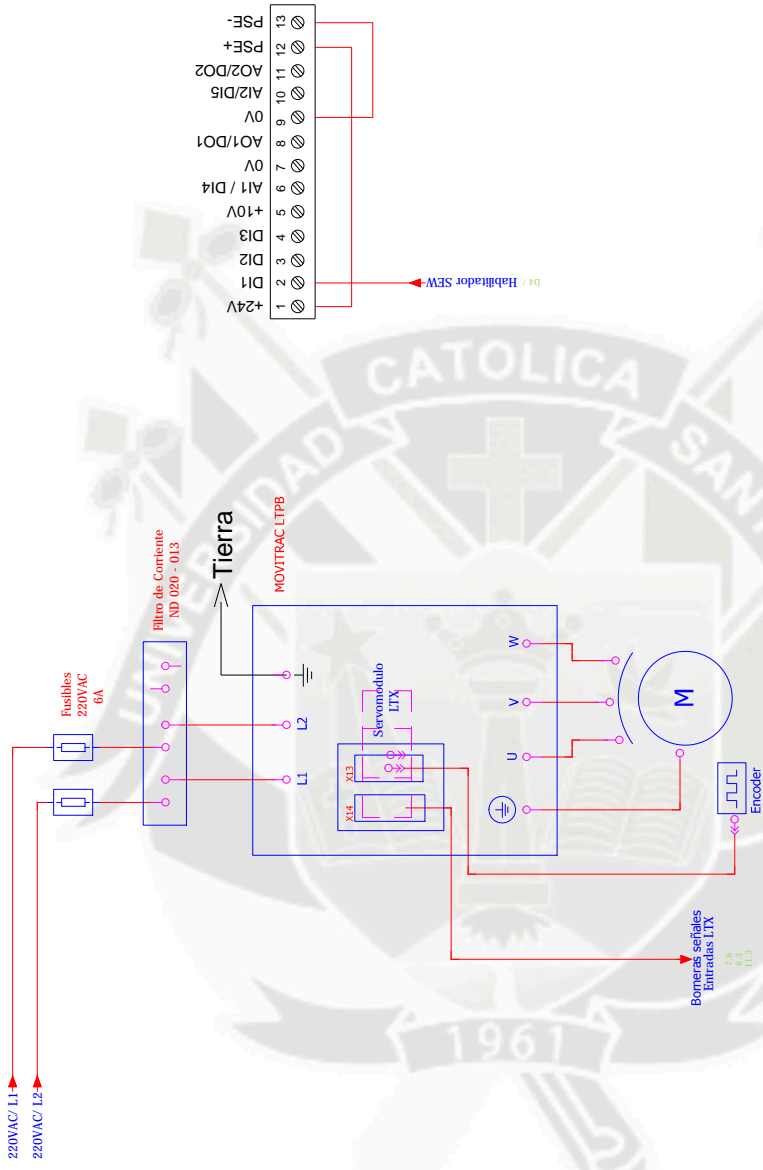


<b>UNIVERSIDAD CATOLICA DE SANTA MARIA</b> ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.	
PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL CONEXIONES DE ENERGIA
2008220181	ZARATECM, RODRIGUEZE
	ESCALA ---
	FECHA: 08/03/2015
	LAMINA: A4 2/11



UNIVERSIDAD CATOLICA DE SANTA MARIA  
 ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

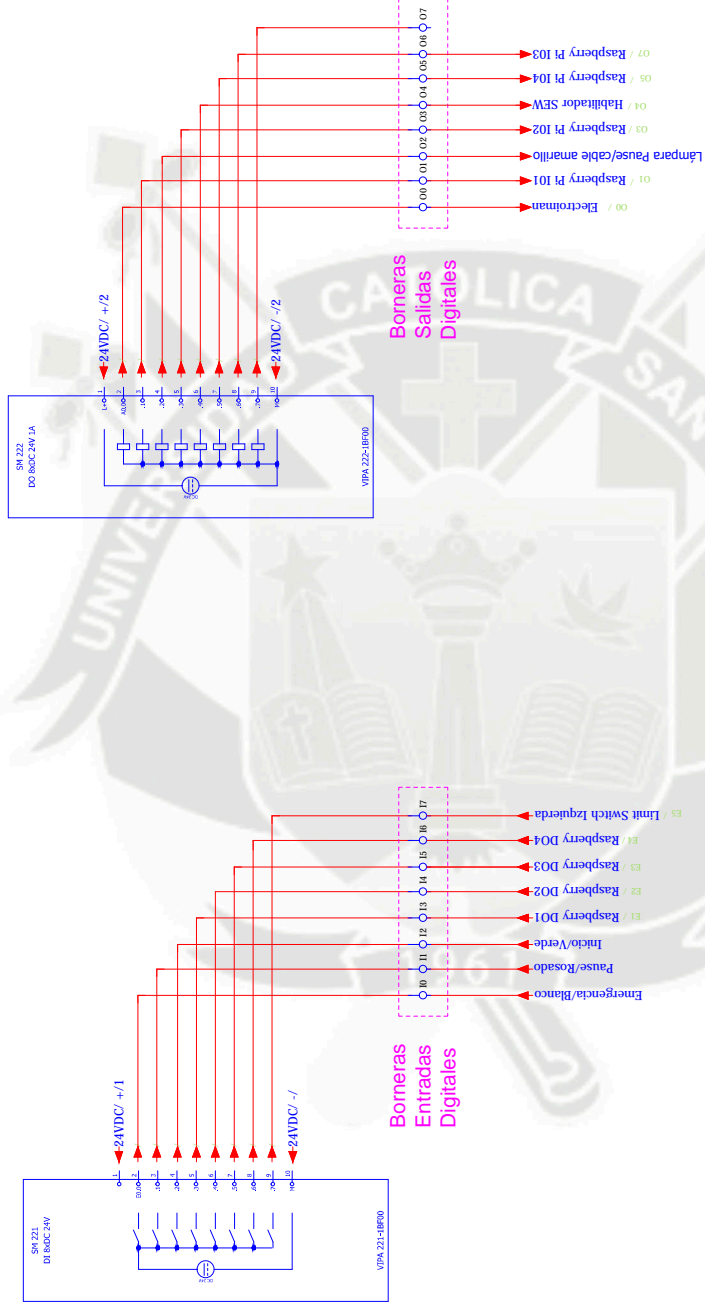
PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL DISTRIBUCION DE BORNERAS 12VDC / 24VDC	ESCALA — — —
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015
		LAMINA: A4 3/11



UNIVERSIDAD CATOLICA DE SANTA MARIA  
 ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL DISTRIBUCION DE VARIADOR Y SERVOMOTOR	ESCALA — — —
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015
		LAMINA: A4 4/11

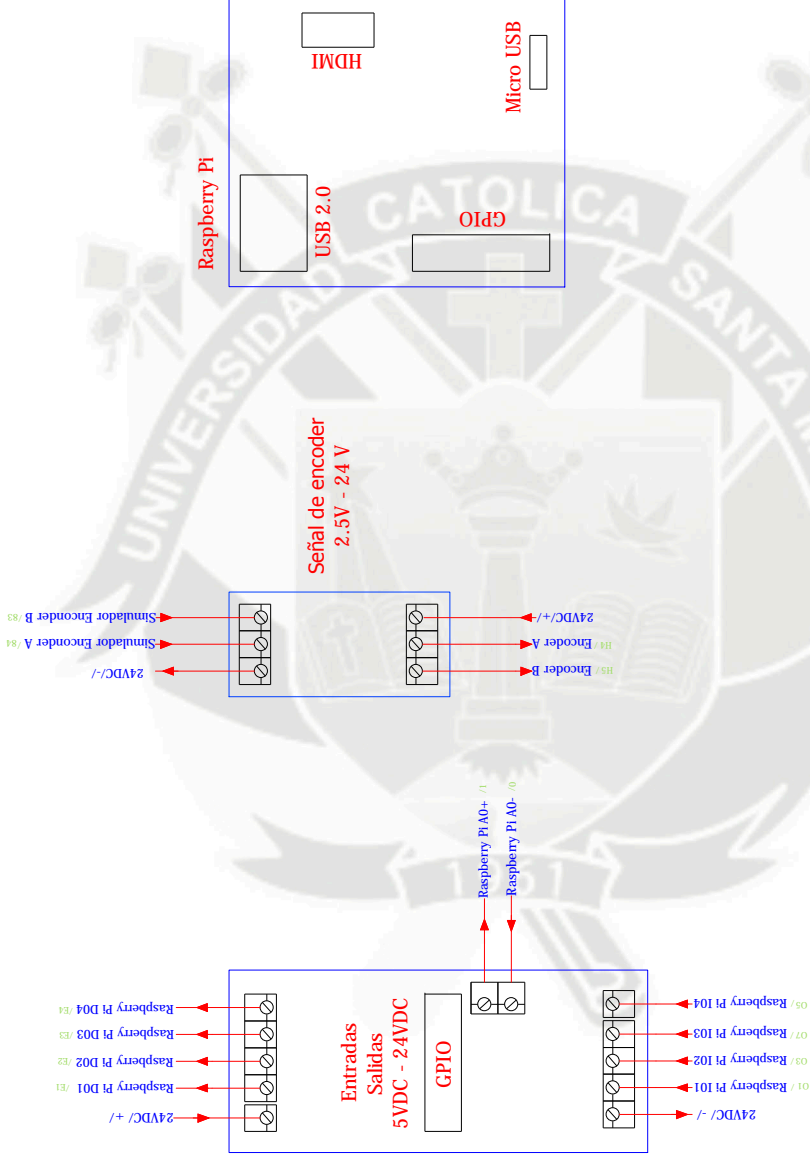
## DISTRIBUCION DE ENTRADAS Y SALIDAS



**UNIVERSIDAD CATOLICA DE SANTA MARIA**  
**ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.**

<b>PLANO DE CONEXIONADO ELECTRICO</b>	<b>SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL</b> <b>DISTRIBUCION DE ENTRADAS Y SALIDAS</b>	<b>ESCALA</b> ———
2008220181	ZARATECM, RODRIGUEZSE	<b>FECHA:</b> 08/03/2015 <b>LAMINA:</b> A4 5/11

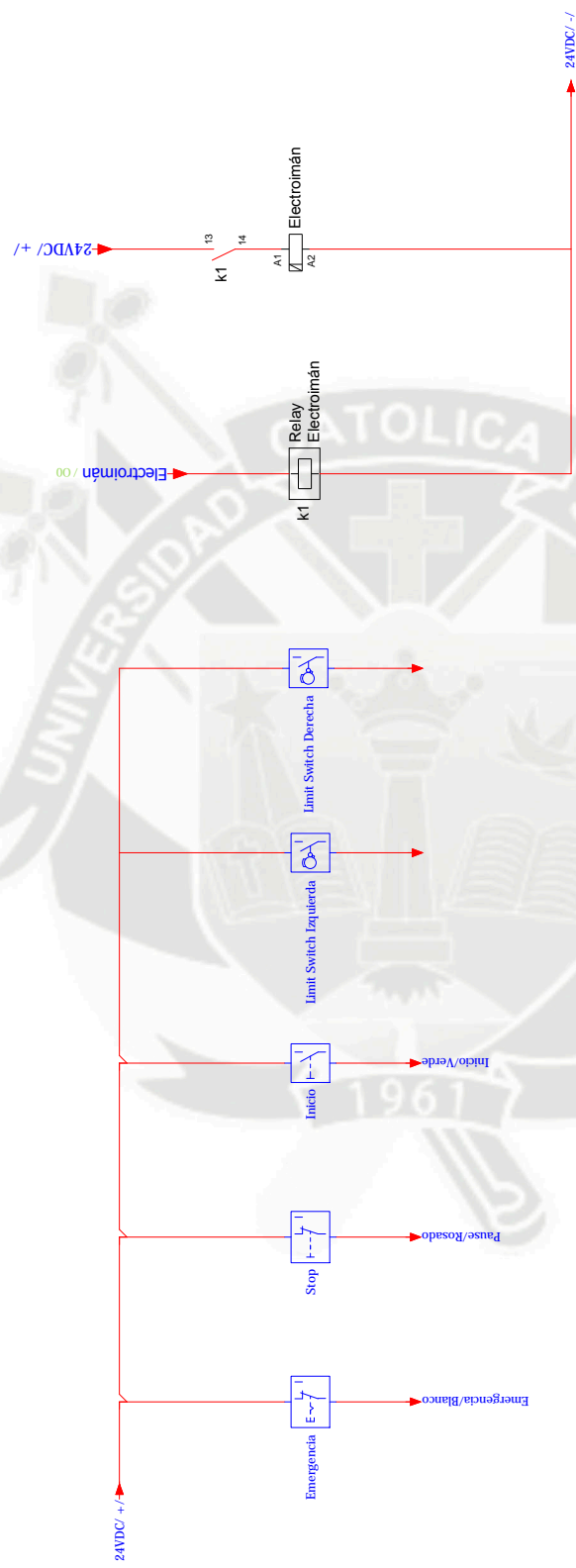




UNIVERSIDAD CATOLICA DE SANTA MARIA  
 ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

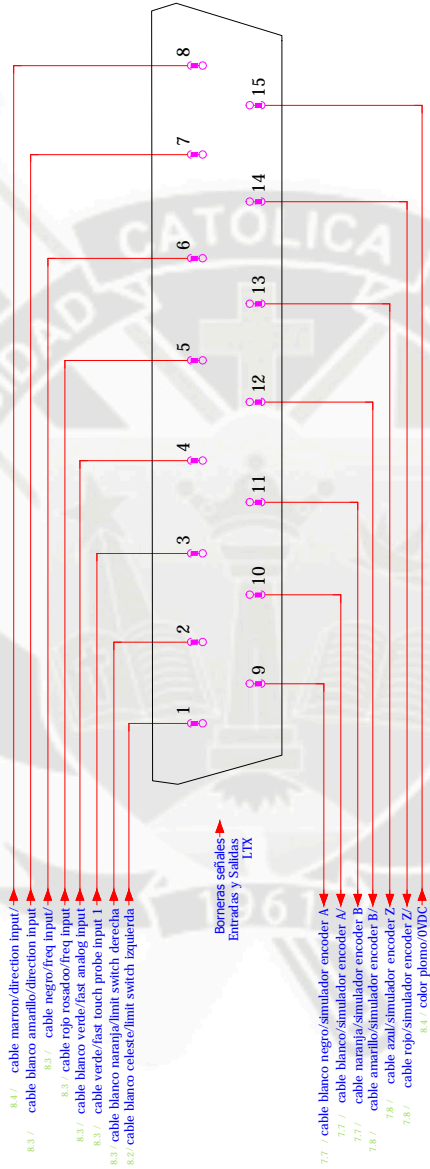
PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL DISTRIBUCION DE SEÑALES LTX Y BORNERAS SPARE	ESCALA — — —
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015
		LAMINA: A4 7/11

## DISTRIBUCION DE PULSADORES Y SENSORES



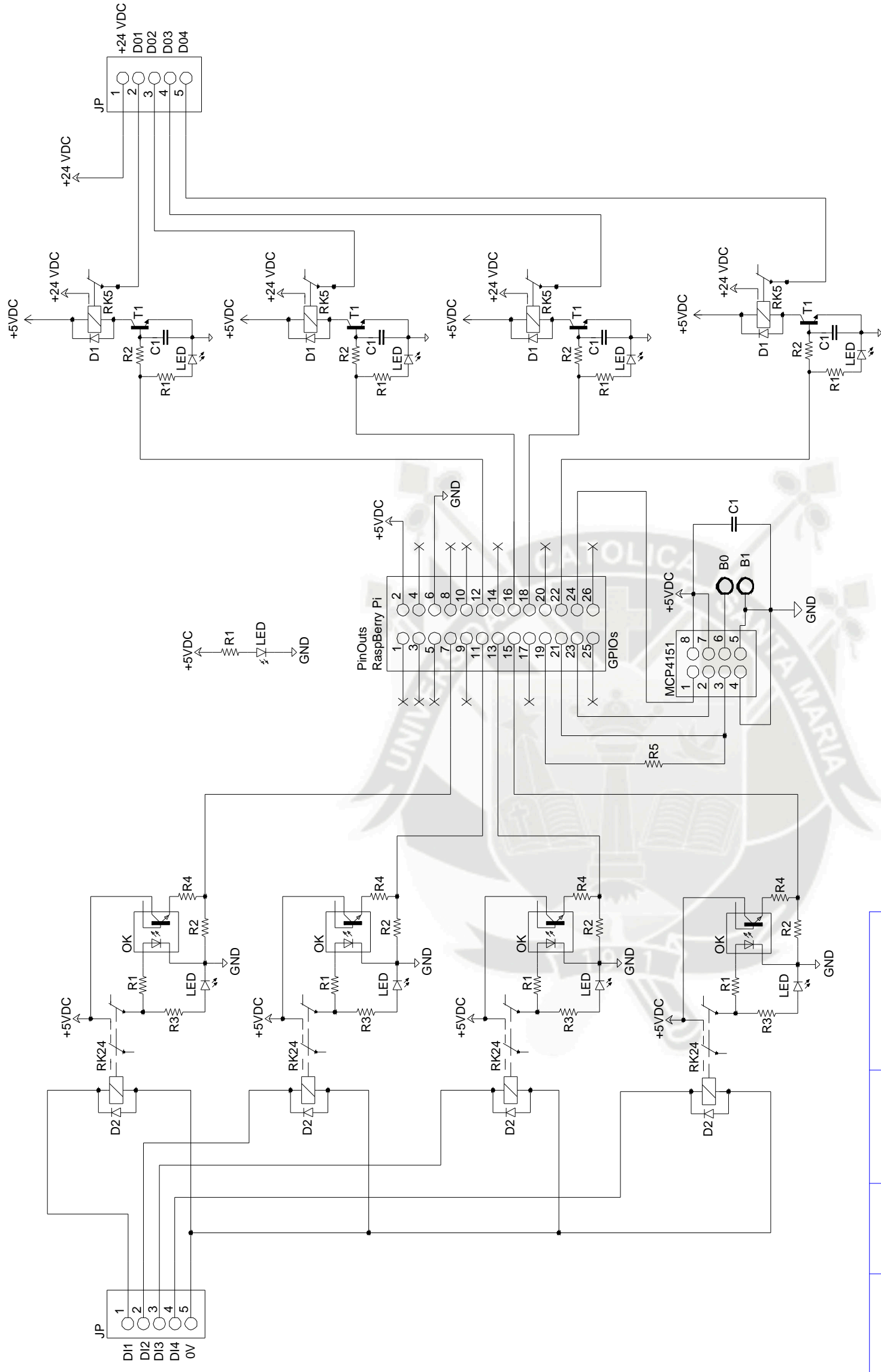
**UNIVERSIDAD CATOLICA DE SANTA MARIA**  
 ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL DISTRIBUCION DE SEÑALES LTX Y BORNERS SPARE	ESCALA — — —
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015
		LAMINA: A4 8/11



UNIVERSIDAD CATOLICA DE SANTA MARIA  
 ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL DISTRIBUCION DE X14 ENTRADAS Y SALIDAS LTX	ESCALA — — —
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015
		LAMINA: A4 9/11



11	8	Diodos 400 v, 1 A	--	--	[Dx] 1N400x
10	4	Réle de 24VDC, 1A, 450 mW, 2 contactos	--	--	[RK24] HRS2H-DC24V
9	4	Réle de 5VDC, 1A, 450 mW, 2 contactos	--	--	[RK5] HRS2H-DC5V
8	4	Optoacoplador 1.5 kV, 3A	--	--	[OK] Txs Instrument 4N35
7	6	Transistor NPN, 10V	--	--	[T1] 2N2222
6	4	Capacitor cerámico de 22 pF	--	--	[C1]
5	9	LED de 5VDC	--	--	[LED]
4	4	Resistencias de 560Ω de 0.25 W, tolerancia 1%	--	--	[R4]
3	4	Resistencias de 1KΩ de 0.25 W, tolerancia 1%	--	--	[R3]
2	8	Resistencias de 10KΩ de 0.25 W, tolerancia 1%	--	--	[R2]
1	9	Resistencias de 330Ω, 0.25 W, tolerancia 1%	--	--	[R1]

UNIVERSIDAD CATOLICA DE SANTA MARIA  
 ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

PLANO DE CONEXIONADO ELECTRICO

SISTEMA DE CLASIFICACION  
 POR VISION ARTIFICIAL  
 TARJETA ELECTRONICA  
 5-24VDC

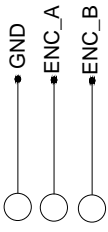
ESCALA  
 ---

2008220181

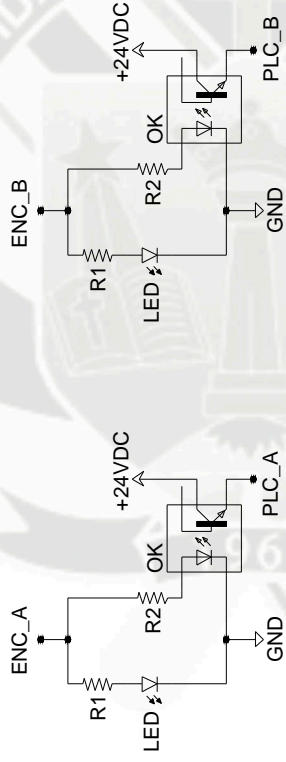
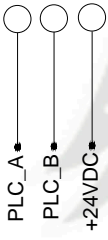
ZARATECM, RODRIGUEZSE

FECHA:  
 08/03/2015  
 LAMINA: A3  
 10/11

Inputs



Outputs



4	2		--	--	[OK] Tx Instrument 4N35
3	2		--	--	[LED]
2	2	Resistencias de 1KΩ de 0.25 W, tolerancia 1%	--	--	[R2]
1	2	Resistencias de 1MΩ, 0.25 W, tolerancia 1%	--	--	[R1]

UNIVERSIDAD CATOLICA DE SANTA MARIA  
ESCUELA PROFESIONAL DE MEC., MEC. ELEC. Y MECATRONICA.

PLANO DE CONEXIONADO ELECTRICO	SISTEMA DE CLASIFICACION POR VISION ARTIFICIAL SEÑAL DE ENCODER 2.5-24 VDC	ESCALA ---
2008220181	ZARATECM, RODRIGUEZSE	FECHA: 08/03/2015
		LAMINA: A4 11/11