

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA



TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN MECATRÓNICA

TEMA:

SISTEMA DE VISIÓN ARTIFICIAL PARA EL CONTROL DE CALIDAD DE
HOJAS DE CUERO PARA LA FÁBRICA GAMACUEROS

AUTOR:

MANUEL ALEJANDRO FUENTES ROJAS

DIRECTOR:

Ing. COSME MEJÍA

Ibarra, 2016

DECLARACIÓN


Yo, MANUEL ALEJANDRO FUENTES ROJAS, declaro que el trabajo aquí descrito es de mi autoría, no ha sido previamente presentado para ningún grado o calificación profesional y certifico la veracidad de las referencias bibliográficas que se incluyen en este documento.



.....
Manuel Alejandro Fuentes Rojas

CERTIFICACIÓN

En calidad de Director de Trabajo de Grado “Sistema de visión artificial para el control de calidad de hojas de cuero para la fábrica GAMACUEROS”, presentado por el señor Manuel Alejandro Fuentes Rojas, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.



.....

Ing. Cosme Mejía

DIRECTOR

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del norte dentro del proyecto Repositorio Digital institucional, determino la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

DATOS DEL CONTACTO		
CÉDULA DE IDENTIDAD:	DE	1721547675
APELLIDOS Y NOMBRES:	Y	MANUEL ALEJANDRO FUENTES ROJAS
DIRECCIÓN:		MALDONADO 2-11 Y PEÑAHERRERA
E-MAIL:		manulfuentes90.mf@hotmail.com
TELÉFONO MÓVIL:		0984626124

DATOS DE LA OBRA	
TÍTULO:	SISTEMA DE VISIÓN ARTIFICIAL PARA EL CONTROL DE CALIDAD DE HOJAS DE CUERO PARA LA FÁBRICA GAMACUEROS
AUTOR:	MANUEL ALEJANDRO FUENTES ROJAS
FECHA:	14 de septiembre de 2016
PROGRAMA:	PREGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA
DIRECTOR:	ING. COSME MEJÍA

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Manuel Alejandro Fuentes Rojas, con cédula de identidad Nro.172154767-5 en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior, Artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 14 días del mes de septiembre del 2016

EL AUTOR

(firma).....

Manuel Alejandro Fuentes Rojas

CI: 172154767-5

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL
NORTE**

Yo, Manuel Alejandro Fuentes Rojas, con cédula de identidad Nro.172154767-5 Manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, Artículos 4,5 y 6, en calidad de autor de la obra o trabajo de grado denominado “Sistema de visión artificial para el control de calidad de hojas de cuero para la fábrica GAMACUEROS”, que ha sido desarrollada para optar por el título de Ingeniero en Mecatrónica en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago la entrega del trabajo final en formato impreso y digital en la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 14 días del mes de septiembre del 2016

(firma) 

Nombre: Manuel Alejandro Fuentes Rojas

Cédula: 172154767-5

AGRADECIMIENTO

Deseo agradecer en primer lugar a Dios, por darme salud y fortaleza para lograr superar pequeños obstáculos que la vida me ha puesto.

Agradezco a mis padres German Fuentes y Lucia Rojas por haberme forjado como la persona que soy en la actualidad.

A mis hermanos y amigos por ser la motivación para conseguir una meta más en mi vida.

A Sr Juan Carlos Guerrero por brindar el conocimiento y los requerimientos que se necesitaron para la creación y culminación de este trabajo.

A los Sres: Guillermo Morillo, Pablo Cuasquer, Esteban Brusil por la colaboración brindada.

A los Ingenieros Gabriela Verdezoto y Cosme Mejía por ser los inspiradores para la realización y culminación de este trabajo.

DEDICATORIA

Este trabajo está dedicado primeramente a Dios por darme la fortaleza y sabiduría para salir adelante cada día y enseñarme que ante la adversidad debemos ser más fuertes y confiar totalmente en El.

A mis padres por ser el apoyo incondicional para culminar con éxito mis estudios Universitarios y llegar a ser un profesional de la Patria.

ÍNDICE GENERAL

DECLARACIÓN	I
CERTIFICACIÓN.....	III
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	IV
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE GRÁFICOS.....	XVI
ÍNDICE DE TABLAS.....	XVII
ÍNDICE DE ECUACIONES	XVIII
RESUMEN.....	XX
SUMMARY.....	XXI
PRESENTACIÓN.....	XXII
CAPITULO I	1
1. PROBLEMA DE LA INVESTIGACIÓN	1
1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. OBJETIVO DE LA INVESTIGACIÓN	2
1.3. JUSTIFICACIÓN.....	2
1.4. ALCANCE.....	3
CAPITULO II.....	5
2. MARCO TEÓRICO.....	5
2.1. ANTECEDENTES.....	5
2.1.1. TRABAJOS PREVIOS.....	5
2.1.2. MAQUINAS EXISTENTES.....	5
2.2. EL CUERO	6
2.2.1. HOJA O MEDIA PIEL DE CUERO	6
2.2.2. FALLAS DEL CUERO	7
2.2.3. VALOR DEL CUERO	10

2.3. VISIÓN ARTIFICIAL	11
2.3.1. SOFTWARE.....	13
2.3.2. FUENTES DE ILUMINACIÓN	19
2.3.3. LENTE.....	20
2.3.4. DISTORSIONES ÓPTICAS.....	22
2.3.5. PIXEL.....	22
2.4. INTELIGENCIA ARTIFICIAL	23
2.4.1. REDES NEURONALES ARTIFICIALES.....	23
CAPITULO III	26
3. MARCO METODOLÓGICO	26
3.1. METODOLOGÍA.....	26
3.1.1. OBJETIVO 1.....	26
3.1.2. OBJETIVO 2.....	26
3.1.3. OBJETIVO 3.....	26
3.1.4. OBJETIVO 4.....	27
3.1.5. OBJETIVO 5.....	27
3.1.6. OBJETIVO 6.....	27
3.1.7. OBJETIVO 7.....	28
3.1.8. OBJETIVO 8.....	28
3.2. CÁMARA.....	29
3.2.1. ALTERNATIVAS DE CÁMARAS	29
3.2.2. SELECCIÓN DE LA ALTERNATIVA ADECUADA	31
3.3. MÓDULO PARA LUMINARIA UBICADA EN DIRECCIÓN FRONTAL	33
3.3.1. SELECCIÓN DE TIPO DE ILUMINACIÓN PARA EL MÓDULO	33
3.3.2. DISTRIBUCIÓN DE LA LUMINARIA	35
3.3.3. ÉTAPA DE POTENCIA	36
3.4. MÓDULO PARA LUMINARIA UBICADA EN DIRECCIÓN TRASERA	39
3.4.1. SELECCIÓN DE TIPO DE ILUMINACIÓN PARA EL MÓDULO	39
3.4.2. ÉTAPA DE POTENCIA	41
3.5. MÓDULO DE RECEPCIÓN DE DATOS.....	42
3.5.1. PROCESAMIENTO DEL DATO.	42
3.6. TABLERO DE DISTRIBUCIÓN.....	46

3.6.1. ELEMENTOS PANEL DE CONTROL.....	47
3.7. DISEÑO ESTRUCTURAL.....	47
3.7.1. ELEMENTOS QUE COMPLEMENTAN A LA ESTRUCTURA INFERIOR.....	48
3.7.2. CALCULO DE LA ESTRUCTURA INFERIOR.....	50
3.7.3. ELEMENTOS QUE COMPLEMENTAN A LA ESTRUCTURA SUPERIOR.....	61
3.8. SOFTWARE PARA VISIÓN ARTIFICIAL.....	63
3.8.1. REQUERIMIENTOS FUNDAMENTALES DEL SISTEMA DE VISIÓN.....	63
3.8.2. IMPLEMENTACIÓN DEL SOFTWARE.....	63
CAPITULO IV.....	96
4. RESULTADOS.....	96
4.1. INTRODUCCIÓN.....	96
4.2. IMPLEMENTACIÓN DEL SISTEMA DE VISIÓN ARTIFICIAL PARA EL CONTROL DE CALIDAD DE HOJAS DE CUERO.....	96
4.2.1. DIAGRAMA DE CONSTRUCCIÓN.....	96
4.2.2. FUNCIONAMIENTO.....	96
4.2.3. CONSTRUCCIÓN DE LA MESA DE TRABAJO.....	97
4.2.4. CONSTRUCCIÓN DE LA ESTRUCTURA INFERIOR.....	97
4.2.5. CONSTRUCCIÓN DEL CAJÓN LUMINOSO.....	98
4.2.6. INSTALACIÓN DE LA CUBIERTA.....	99
4.2.7. CONSTRUCCIÓN DE LA ESTRUCTURA COLGANTE.....	100
4.2.8. CONSTRUCCIÓN DE LA TARJETA DE CONTROL DE LUMINARIA.....	102
4.2.9. CONSTRUCCIÓN DEL TABLERO DE DISTRIBUCIÓN.....	104
4.2.10. SOFTWARE.....	105
4.3. PRUEBA DE FUNCIONAMIENTO DE LA TARJETA DE CONTROL DE LA LUMINARIA ...	107
4.4. PRUEBA DE NIVEL DE ILUMINACIÓN DE LA LUMINARIA SUPERIOR.....	107
4.5. PRUEBA DE FUNCIONAMIENTO DE LA CÁMARA.....	108
4.6. PRUEBA DEL SOFTWARE.....	109
4.6.1. PROGRAMA PRINCIPAL.....	109
4.6.2. PROGRAMA BASEDEDATOS.....	109
4.6.3. PROGRAMA VERIFICONDEPARAMETROSINICIALES.....	110
4.6.4. PROGRAMA CLASE_AREA.....	111
4.6.5. EXACTITUD DE MEDIDA.....	113

4.6.6. REPETITIVIDAD DE RESULTADOS	115
4.6.7. ESQUEMA GENERAL DEL SISTEMA DE VISIÓN ARTIFICIAL	118
CAPITULO V	120
5. CONCLUSIONES Y RECOMENDACIONES.....	120
5.1. CONCLUSIONES.....	120
5.2. RECOMENDACIONES.....	121
LISTA DE SIGLAS	122
BIBLIOGRAFÍA	123
ANEXO 1 MANUAL DE USUARIO Y MANTENIMIENTO.....	126
ANEXO 2 CIRCUITO IMPRESO DE LA TARJETA DE LUMINOSIDAD..	135
ANEXO 3 DIAGRAMA ELÉCTRICO	137
ANEXO 4 PROGRAMACIÓN SOFTWARE	139
ANEXO 4.1 FUNCIÓN APAGARLUZINFERIOR.....	140
ANEXO 4.2 FUNCIÓN APAGARLUZSUPERIOR.....	140
ANEXO 4.3 FUNCIÓN AREAPIEZA	141
ANEXO 4.4 FUNCIÓN AUMENTARLUMINOSIDADSUPERIOR	142
ANEXO 4.5 FUNCIÓN CALCULOAREA.....	142
ANEXO 4.6 FUNCIÓN CALCULODEPARAMETROS	143
ANEXO 4.7 FUNCIÓN CLASIFICACION	145
ANEXO 4.8 FUNCIÓN CONTROLLUMINOSIDADSUPERIOR	149
ANEXO 4.9 FUNCIÓN DETECCIONPIEZA.....	150
ANEXO 4.10 FUNCIÓN DETECCIONPLANTILLA	151
ANEXO 4.11 FUNCIÓN DISMINUIRLUMINOSIDADSUPERIOR	152
ANEXO 4.12 FUNCIÓN ELIMINADISTORSION.....	152
ANEXO 4.13 FUNCIÓN ELIMINAMESA	153
ANEXO 4.14 FUNCIÓN ELIMINAMESAAUMENTADOCONTRASTE.....	153
ANEXO 4.15 FUNCIÓN ENCENDERLUZINFERIOR.....	154
ANEXO 4.16 FUNCIÓN ENCENDERLUZSUPERIOR.....	154
ANEXO 4.17 FUNCIÓN IMAGENCLASE	155
ANEXO 4.18 FUNCIÓN VIDEO.....	155
ANEXO 4.19 FUNCIÓN IMPRIMIR.....	156
ANEXO 4.20 FUNCIÓN RN_COMPROBACIONLUMINARIA.....	157

ANEXO 4.21 PROGRAMA BASEDEDATOS	158
ANEXO 4.22 PROGRAMA CLASE_AREA	161
ANEXO 4.23 PROGRAMA PRINCIPAL.....	171
ANEXO 4.24 PROGRAMA VERIFICACIONDEPARAMETROSINICIALES.....	173
ANEXO 5 PROGRAMACIÓN MICROCONTROLADOR.....	179
ANEXO 6 ESPECIFICACIONES COMPONENTES	182
ANEXO 6.1 DICROICOS HALÓGENOS SYLVANIA	183
ANEXO 6.2 LÁMPARA TUBULAR SYLVANIA.....	184
ANEXO 7 PLANOS.....	185
ANEXO 7.1 ESTRUCTURA SUPERIOR.....	186
ANEXO 7.2 ÁNGULO EXTERIOR.....	187
ANEXO 7.3 ÁNGULO INTERIOR	188
ANEXO 7.4 VIGA CENTRAL.....	189
ANEXO 7.5 VIGA EXTERIOR CORTA PARA ANCLAJE	190
ANEXO 7.6 VIGA EXTERIOR CORTA	191
ANEXO 7.7 VIGA EXTERIOR LARGA	192
ANEXO 7.8 VIGUETA	193
ANEXO 7.9 MESA	194
ANEXO 7.10 CUBIERTA.....	195
ANEXO 7.11 CAJÓN LUMINOSO.....	196
ANEXO 7.12 ESTRUCTURA INFERIOR.....	197
ANEXO 7.13 TABLERO DE DISTRIBUCIÓN	198
ANEXO 7.14 PLACA DE MONTAJE.....	199
ANEXO 8 CATÁLOGO TUBERÍA ESTRUCTURAL	200
ANEXO 8.1 TUBERÍA ESTRUCTURAL CUADRADA	201
ANEXO 8.2 TUBERÍA ESTRUCTURAL RECTANGULAR	202
ANEXO 8.3 CATÁLOGO CEDAL.....	203
ANEXO 9 SELECCIÓN DE VIDRIO PARA CUBIERTA	204
ANEXO 10 COMPONENTES ELECTRÓNICOS	206
ANEXO 10.1 DATASHEET BTA/BTB12	207
ANEXO 10.2 DATASHEET BTA10	208
ANEXO 10.3 DATASHEET MOC-3021.....	209

ANEXO 11 FOTOGRAFÍAS IMPLEMENTACIÓN DEL SISTEMA	212
ANEXO 11.1 CONSTRUCCIÓN ESTRUCTURA INFERIOR.....	213
ANEXO 11.2 CONSTRUCCIÓN CAJÓN LUMINOSO	213
ANEXO 11.3 CONSTRUCCIÓN ESTRUCTURA SUPERIOR	214
ANEXO 11.3 INSTALACIÓN ELÉCTRICA.....	215
ANEXO 11.4 ENSAMBLE DE LA TARJETA.....	215
ANEXO 11.5 ENSAMBLE DEL TABLERO DE DISTRIBUCIÓN.....	216
ANEXO 11.6 IMAGEN REPRESENTATIVA DE LA ETIQUETA.	216

ÍNDICE DE FIGURAS

Figura 2.1 Medidora de cueros MD ROLL.....	5
Figura 2.2 Medición artesanal con malla cuadrículada	6
Figura 2.3 Hoja de cuero.....	6
Figura 2.4 Marca por herrete en cuero.....	7
Figura 2.5 Agujero mordedura de maquina	8
Figura 2.6 Marca de una corneada en cuero.....	9
Figura 2.7 Cuero sin secar correctamente	10
Figura 2.8 Lesiones típicas y divisiones del cuero	11
Figura 2.9 Elementos del sistema de visión artificial	12
Figura 2.10 Tablero de ajedrez con distorsión y sin distorsión.....	15
Figura 2.11 GUI de aviso	19
Figura 2.12 GUI de aviso	19
Figura 2.13 Disposición de las fuentes de luz en visión artificial	20
Figura 2.14 Tipos de lentes	20
Figura 2.15 Distintos tipos de objetivos	21
Figura 2.16 Tipos de distorsiones ópticas.....	22
Figura 2.17 Representación del pixel en una imagen	23
Figura 2.18 Estructura jerárquica de un sistema basado en RNA	24
Figura 3.1 NI 1778 Cámara inteligente.....	29
Figura 3.2 WideCam F100.....	30
Figura 3.3 Fotometría MR16 Coolfit Home (Havells Sylvania, 2016)	34

Figura 3.4 Esquema de parámetros de la luminaria.....	35
Figura 3.5 Distribución lámparas dicroicos halógenas	36
Figura 3.6 Diagrama etapa de potencia lámparas halógenas	38
Figura 3.7 Formas del haz de luz.....	39
Figura 3.8 Distribución lámparas fluorescentes	40
Figura 3.9 Diagrama etapa de potencia lámparas fluorescentes	41
Figura 3.10 Diagrama de conexión eléctrica para lámparas fluorescentes	42
Figura 3.11 Conexión eléctrica Conversión de RS232 a TTL	42
Figura 3.12 Panel de control	47
Figura 3.13 Cajón Luminoso.....	50
Figura 3.14 Modelado de estructura inferior con sus cargas.....	51
Figura 3.15 Diagrama de cortes y momentos de la Viga AB y CD	52
Figura 3.16 Diagrama de cortes y momentos de la Viga BC y AD	53
Figura 3.17 Diagrama de cortes y momentos de la Viga EG y HJ.....	53
Figura 3.18 Diagrama de cortes y momentos de la Viga EG y HJ.....	54
Figura 3.19 Propiedades tubería estructural rectangular (IPAC)	55
Figura 3.20 Propiedades tubería estructural cuadrada (IPAC).....	57
Figura 3.21 Propiedades tubería estructural cuadrada (IPAC).....	58
Figura 3.22 Valor de K para diferentes tipos de conexión a los extremos.	59
Figura 3.23 Simulación factor de seguridad de la estructura inferior.....	61
Figura 3.24 Tubería cuadrada Cedal.....	62
Figura 3.25 Estructura superior.	62
Figura 3.26 Imágenes de referencia para entrada de redes neuronales.....	67
Figura 3.27 Aumento de contraste en imundgray	69
Figura 3.28 Imágenes representativas de la función eliminamesa	70
Figura 3.29 Realce de características.....	72
Figura 3.30 Discernición de píxeles.	72
Figura 3.31 Imágenes de flechas.	74
Figura 3.32 Sectores imagen para localización de plantilla.....	74
Figura 3.33 Bordes plantillas negras con pieza clara.....	77
Figura 3.34 Bordes plantillas blancas con pieza oscura.....	78
Figura 3.35 Segmentación de la pieza	82

Figura 3.36 Imagen representativa a funcion imagenclase	83
Figura 3.37 Interfaz de usuario de programa verificacion de parametros iniciales....	86
Figura 3.38 Interfaz de usuario de programa clase_area	89
Figura 3.39 Imagen representativa al presionar boton guardar	90
Figura 3.40 Interfaz de usuario de programa basededatos	94
Figura 3.41 Interfaz de usuario de programa principal.....	95
Figura 4.1 Diagrama de construcción	96
Figura 4.2 Diseño de la tarjeta.....	102
Figura 4.3 Carga de datos y archivos para ejecutable.....	106
Figura 4.4 Ubicación para la prueba del nivel de luminosidad	107
Figura 4.5 Prueba funcionamiento de la cámara	109
Figura 4.6 Esquema del sistema de vision artificial para el control de calidad de hojas de cuero	118

ÍNDICE DE GRÁFICOS

Gráfico 3.1 Diagrama de flujo del programa en el μ C	45
Gráfico 3.2 Malla referencial para selección de espesor del vidrio.....	49
Gráfico 3.3 Diagrama de flujo de la función apagarluzinferior	64
Gráfico 3.4 Diagrama de flujo de la función video	65
Gráfico 3.5 Diagrama de flujo de la función eliminadistorsion	66
Gráfico 3.6 Diagrama de flujo de la función rn_comprobacionluminaria.....	66
Gráfico 3.7 Diagrama de flujo de la función eliminamesa	68
Gráfico 3.8 Diagrama de flujo de la función eliminamesaaumentadocontraste.....	71
Gráfico 3.9 Diagrama de flujo de la función areapieza	73
Gráfico 3.10 Diagrama de flujo de la función calculodeparametros	76
Gráfico 3.11 Diagrama de flujo de la función detecciónpieza	77
Gráfico 3.12 Diagrama de flujo de la función detecciplantilla	79
Gráfico 3.13 Diagrama de flujo de la función calculoarea	79
Gráfico 3.14 Diagrama de flujo de la función clasificación	80
Gráfico 3.15 Diagrama de flujo de la función imagenclase.....	83
Gráfico 3.16 Diagrama de flujo de la función controlluminosidadsuperior	84
Gráfico 3.17 Diagrama de flujo de la función imprimir	85

Gráfico 3.18 Diagrama de flujo del botón verificar	87
Gráfico 3.19 Diagrama de flujo del botón guardar.....	90
Gráfico 3.20 Diagrama de flujo del botón medir área	91
Gráfico 3.21 Diagrama de flujo de la función control luminosidad superior	92
Gráfico 4.1 Grafica de porcentaje error medido (% de repetitividad).....	117

ÍNDICE DE TABLAS

Tabla 2.1 Daños mecánicos que afectan a los cueros y pieles	8
Tabla 2.2 Enfermedades e infecciones parasitarias	9
Tabla 2.3 Categorización del cuero	10
Tabla 2.4 Valor del cuero por clase	11
Tabla 2.5 Operaciones morfológicas para la función regionprops.....	17
Tabla 2.6 Operaciones morfológicas para la función bwmorph.....	17
Tabla 2.7 Clasificación de objetivos.....	21
Tabla 2.8 Tipos de distorsiones ópticas	22
Tabla 2.9 Principales tipos de funciones de activación o transferencia.....	25
Tabla 3.1 Resumen de especificaciones NI 1778 Smart Camera.....	29
Tabla 3.2 Resumen de especificaciones WideCam F100	30
Tabla 3.3 Evaluación del peso específico de cada criterio	32
Tabla 3.4 Evaluación del peso específico por criterio	32
Tabla 3.5 Conclusiones de la selección de cámara para el sistema.....	33
Tabla 3.6 Parámetros de funcionamiento BTA12-600C.....	37
Tabla 3.7 Características físicas de lámparas fluorescentes.....	40
Tabla 3.8 Dato a enviar según la acción de la luminaria	43
Tabla 3.9 Comportamiento de la luminaria.....	43
Tabla 3.10 Acción en el μC	44
Tabla 3.11 Lista de variables del programa en el μC	44
Tabla 3.12 Componentes del panel de control	47
Tabla 3.13 Características de materiales para la cubierta.	49
Tabla 3.14 Descripción de cargas en la estructura	50
Tabla 3.15 Factores de diseño para materiales dúctiles.....	51
Tabla 3.16 Elementos que complementan a la estructura superior	61

Tabla 3.17 Área de la plantilla según su forma.....	78
Tabla 3.18 Plantillas, lugar y cantidad por clase.....	81
Tabla 3.19 Plantillas, clasificación por área.....	81
Tabla 4.1 Material estructura.....	97
Tabla 4.2 Material cajón luminoso	98
Tabla 4.3 Material cubierta	99
Tabla 4.4 Material estructura superior	100
Tabla 4.5 Material instalación eléctrica	101
Tabla 4.6 Material tarjeta de control de luminaria.	103
Tabla 4.7 Material tablero de control.....	104
Tabla 4.8 Resultados de prueba de funcionamiento en la tarjeta de control de luminosidad.	107
Tabla 4.9 Nivel de iluminación en la luminaria superior.....	108
Tabla 4.10 Prueba de comportamiento del programa principal	109
Tabla 4.11 Prueba de comportamiento del programa basededatos	110
Tabla 4.12 Prueba de comportamiento de luminaria en el programa verificaciondeparametrosiniciales	110
Tabla 4.13 Prueba de comportamiento del programa verificaciondeparametrosiniciales	110
Tabla 4.14 Prueba de comportamiento de luminaria en el programa clase_area.....	111
Tabla 4.15 Prueba de comportamiento del programa clase_area.....	112
Tabla 4.16 Estudio de exactitud con una pieza patrón	114
Tabla 4.17 Criterio de aceptación.....	114
Tabla 4.18 Constante para expandir el rango promedio	116
Tabla 4.19 Prueba repetitividad y reproductibilidad de resultados	117
Tabla 4.20 Detalle de los componentes	119

ÍNDICE DE ECUACIONES

Ecuación 2.1 Valor monetario del cuero.....	11
Ecuación 2.2 Sumatoria de las entradas pesadas.....	24
Ecuación 3.1 Calculo de conclusiones del método ordinal corregido de criterios ponderados.	32

Ecuación 3.2	Calculo diámetro del haz de luz	35
Ecuación 3.3	Calculo de corriente para las 14 lámparas	36
Ecuación 3.4	Calculo de impedancia del transformador	36
Ecuación 3.5	Calculo de corriente para las 6 lámparas	41
Ecuación 3.6	Esfuerzo de diseño.....	54
Ecuación 3.7	Modulo de sección requerido	55
Ecuación 3.8	Esfuerzo de diseño.....	56
Ecuación 3.9	Modulo de sección requerido	56
Ecuación 3.10	Constante de la columna	58
Ecuación 3.11	Radio de giro de la sección transversal.....	58
Ecuación 3.12	Longitud efectiva.....	59
Ecuación 3.13	Relación de esbeltez.....	59
Ecuación 3.14	Carga critica de pandeo calculada	60
Ecuación 3.15	Carga admisible en la columna.....	60
Ecuación 3.16	Formula para cálculo del factor de conversión de pixeles a cm ²	75
Ecuación 4.1	Calculo tolerancia para medida de una pieza	113
Ecuación 4.2	Calculo promedio de medida.....	113
Ecuación 4.3	Calculo % de exactitud	114
Ecuación 4.4	Calculo de rangos	115
Ecuación 4.5	Calculo promedio de rangos.....	115
Ecuación 4.6	Calculo error máximo (repetitividad)	115
Ecuación 4.7	Desviación estándar del error medido	115
Ecuación 4.8	Tolerancia.....	116
Ecuación 4.9	Porcentaje del error medido (% de repetitividad).....	116

SISTEMA DE VISIÓN ARTIFICIAL PARA EL CONTROL DE CALIDAD DE HOJAS DE CUERO PARA LA FÁBRICA GAMACUEROS

Autor: Manuel Fuentes

Tutor: Ing. Cosme Mejía

RESUMEN

En el presente trabajo de investigación se presenta un sistema basado en visión artificial, éste realiza la medida del área y clasificación en hojas de cuero para la fábrica GAMACUEROS.

El sistema se compone de software: programa para Calibración del sistema, Control de calidad y Base de datos; hardware que es el conjunto de: estructura inferior y superior, cámara, luminaria, tablero de control, ordenador e impresora.

Este software fue desarrollado en el entorno de lenguaje de ingeniería, éste permite procesar las imágenes obtenidas por la cámara situada en la estructura superior y realizar encendido, apagado y control de la intensidad de la luminaria. El programa de calibración comprueba el estado de la luminaria mediante el uso de redes neuronales artificiales en la imagen y calibra la medida usando la pieza de calibración. El programa de Control de calidad obtiene el valor de área en el cuero, realizando control de luminaria y procesamiento de la imagen; así mismo, clasifica de acuerdo con la norma de calidad que propone la corporación ganadera CORFOGA, usando plantillas situadas sobre los defectos que posee el cuero; los resultados se almacenan en un archivo Excel y se imprimen en una etiqueta. Adicional puede volver a visualizar o imprimir etiquetas en el programa Base de Datos, ingresando el código del cuero deseado.

Como resultado, tenemos valores de área con un error de $\pm 1\%$ y una clasificación correcta.

ARTIFICIAL VISION SYSTEM FOR QUALITY CONTROL OF LEATHER SHEETS FOR FACTORY GAMACUEROS

Author: Manuel Fuentes

Tutor: Ing. Cosme Mejía

SUMMARY

In the present research work it is presented a system based on artificial vision, this performs the measure of area and grading of leather sheets for GAMACUEROS factory.

The system is composed of software: Calibration system program, Quality Control and Database; hardware that is the set of: lower and upper structure, camera, luminary, control panel, computer and printer.

This software was developed in engineering environment, this allows to process the images obtained by the camera located on the upper structure and performs the on, off and intensity control of the luminary. The calibration program checks the status of the luminary by using artificial neural networks on the image and calibrates the measure using the calibration piece. The Quality Control program obtains the value of area in the leather, performing luminary control and image processing; likewise, classifies according to quality standard proposed by livestock CORFOGA corporation, using templates situated over the flaws that the leather possesses; the results are stored in an Excel file and printed on a label. Additionally, it can return to display or print labels in the database program, entering the code of the desired leather.

As a result, we have area values with an error of $\pm 1\%$ and a correct classification.

PRESENTACIÓN

Con este trabajo se brinda un sistema compacto y funcional enfocado en el control de calidad de hojas de cuero mediante visión artificial, consolidando los conocimientos adquiridos durante los semestres de estudio.

En el primer capítulo se detalla el problema, objetivos, justificación y alcance del trabajo

En el segundo capítulo se da a conocer el marco teórico necesario para la ejecución de este trabajo, herramientas y conceptos que son primordiales para la unión de cada módulo en un solo sistema.

En el tercer capítulo se describe la metodología usada para cumplir los objetivos. Asimismo, se fundamenta y desarrolla los cálculos correspondientes a los circuitos eléctricos y su conexionado, la estructura y selección de elementos que complementan al sistema. Igualmente se diseña el software para la adquisición de imágenes y procesamiento de datos. Además, se trata sobre implementación del sistema completo, su ensamble y puesta a punto.

El cuarto capítulo trata sobre comprobaciones eléctricas, verificación de posibles errores que puedan suscitarse durante su funcionamiento, análisis R&R de datos y rango de exactitud.

El quinto capítulo se presenta las conclusiones que se determinaron con la realización de este trabajo y las recomendaciones que ayudarían a en la mejora y el funcionamiento óptimo de este sistema.

CAPITULO I

1. PROBLEMA DE LA INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El adelanto tecnológico permite el uso de máquinas sofisticadas que logran facilitar el trabajo al ser humano, en este caso la visión artificial para el control de calidad.

La fábrica GAMACUEROS ubicada en el sector del parque industrial de la ciudad de Ibarra, brinda el servicio de tinturado, prensado y metraje en cueros; dotando de su producto al norte del país.

Actualmente, el control de calidad se realiza empíricamente por la experiencia del operador en tres fases: la primera donde se clasifica observando las imperfecciones como cortes, enfermedades, picaduras, marcas por hierros y arrugas, de qué tipo de cuero se trata (clase A, clase B, clase C); la segunda fase se lo realiza después del pintado donde el pintor a su criterio lo aprueba según la homogeneidad del acabado; la tercera fase es la medición del área del cuero, se mide mediante una plancha cuadrículada con dimensiones de 30 cm x 30 cm por cada cuadro que para la empresa representa $[1ft]^2$, esta se coloca sobre el cuero, posterior se calcula la cantidad de cuadros que cubre el cuero; los sobrantes se los va introduciendo por simple visualización en los cuadros que sobra espacio, generando que no se obtenga un área real del cuero, sino un estimado; consecuentemente esto provoca pérdidas para la empresa al momento de la venta.

El proceso de control de calidad se realiza por inspección visual de un operador, lo que incrementa el índice de error en este proceso.

El control de calidad permite certificar con normas establecidas de qué clase de cuero se trata y su medida con fiabilidad, proporcionando a la fábrica una herramienta para competir con el mercado nacional e internacional.

Todos los datos obtenidos son almacenados en una base de datos, además de imprimir una etiqueta con el área y su respectiva clase. Al implementar el sistema se reduce el margen de error en el total del área en alrededor de un $\pm 1\%$, esto debido a la calidad

de la cámara y el control de la luminosidad para obtener una correcta medición. Además, añade fiabilidad y calidad a los productos emitidos por la empresa.

Resulta conveniente añadir este sistema en la línea de producción teniendo en cuenta que el precio del producto tendría un valor justo y aumentaría la confianza para los compradores.

1.2. OBJETIVO DE LA INVESTIGACIÓN

Objetivo General

Garantizar la categorización de la calidad del producto final mediante visión artificial en la fábrica “GAMACUEROS”.

Objetivos Específicos

- Generar un algoritmo de adquisición y procesamiento digital de imágenes.
- Seleccionar las técnicas de visión adecuadas para el control de calidad.
- Desarrollar un programa que permita calcular la superficie del cuero.
- Desarrollar un programa que permita determinar la categorización del cuero de acuerdo a normas de calidad.
- Diseñar la estructura que contendrá el sistema de visión artificial.
- Implementar el sistema de control de calidad.
- Realizar pruebas y depuración de errores en el funcionamiento del sistema e interfaz gráfica.
- Elaborar manuales de usuario y mantenimiento del sistema.

1.3. JUSTIFICACIÓN

La promoción del desarrollo tecnológico en la investigación de las nuevas alternativas hacia una economía generadora de un alto valor agregado sobre la base del conocimiento y el talento humano, vinculada con la reducción de los costos y obtención de beneficios económicos; conlleva a la tecnificación de los procedimientos productivos en la generación de materias primas hacia nuevos emprendimientos, todo esto dentro del plan de desarrollo del buen vivir.

El control de calidad con el sistema de visión artificial garantiza una real clasificación del cuero de acuerdo a las normas sobre imperfecciones existentes y de un control en el acabado de la pintura a fin de darle un valor agregado importante al terminado del producto. Por tal razón la empresa debe insertarse en el cumplimiento de normas y

estándares de calidad debido a la implementación en el país de una nueva matriz productiva.

El uso de este sistema beneficia tanto al vendedor como al comprador ya que da un valor genuino en la medida y calidad, ofreciendo una carta de presentación de la empresa; además disminuye el tiempo que se desempeña en medir de la forma artesanal de 1 minuto a menos de 30 segundos por hoja de cuero, permitiendo una mejor utilización de tiempo de trabajo en el personal y una reducción en sus costos operativos.

La implementación de esta nueva tecnología en el proceso de control de calidad, hace que la empresa sea más competitiva y productiva de acuerdo con los tiempos actuales, consiguiendo un precio justo de acuerdo a la calidad y terminado del producto, siendo para la empresa importante mantener la confianza de sus clientes donde su lema se basa en ganar sin perjudicar a otros.

1.4. ALCANCE

Este proyecto tiene como alcance medir el área y dar una clase a la hoja de cuero. Entre los aspectos puntuales que comprende el diseño están la medición del área, clasificación por análisis de imperfecciones, almacenamiento en una base de datos e impresión de etiquetas con el área y la clase del cuero.

El desarrollo del sistema se lo realiza de la siguiente manera:

La cantidad de luz en el sistema se la regula con un micro controlador, este controla la cantidad de luz artificial dependiendo de la luz natural que exista en el entorno; con esto se asegura una buena calidad fotográfica para facilitar la identificación de imperfecciones tanto en la homogeneidad de pintura como en el análisis de desperfectos por marcas, huecos y cantidad de área aprovechable.

La superficie de la mesa de trabajo sobre la que se coloca el cuero es de color blanco, de esta forma ayuda con el contraste del cuero para facilitar su visualización, además, se usa una cámara con lente gran angular para disminuir la distancia entre el lente hasta el objeto a tomar.

El operario procede a la colocación del cuero verificando que no se encuentre con dobleces que puedan afectar al resultado, luego da la confirmación para proseguir con la toma de foto.

En el programa se procede al tratamiento de la imagen, identificación de patrones y al cálculo. Este valor es almacenado en una base de datos que tiene la imagen del cuero, su medida, y su clasificación, obteniendo así un reporte para la empresa.

Para la verificación se emite una etiqueta que contiene la clase asignada, la imagen del cuero y la medida, con esta información, el usuario verifica que a determinado cuero le pertenece dicha medida y categoría.

En este proyecto se utilizan cueros con terminado unicolor y medidas inferiores a 250cm de largo, 160cm de ancho debido al tamaño de la máquina y condiciones de enfoque de la cámara.

CAPITULO II

2. MARCO TEÓRICO

2.1. ANTECEDENTES

En este capítulo se analiza conceptos básicos necesarios para el desarrollo del proyecto

2.1.1. TRABAJOS PREVIOS

No existen trabajos relacionados al tema.

2.1.2. MAQUINAS EXISTENTES

Hoy en día los métodos para medir superficies son muy diversos, ya que pueden ser netamente artesanales o totalmente automatizados. El uso de maquinarias medidoras de superficie automáticas produce que el tiempo de medida y la precisión disminuyan, lo que contribuye a una ventaja en tiempos de producción sobre fábricas que son artesanales.

Como ejemplo se toma la maquina MD ROLL, *es una medidora que opera con rodillos ópticos, lo que permite medir con precisión las zonas de cueros suaves y sin perdidas debidas a la formación de arrugas de la piel.* (NBN Automação Industrial, 2014). En este tipo de maquinaria existe una serie de rodillos que al girar da la cantidad de área que posee el cuero, el mantenimiento de esta maquinaria es complejo ya que tiene demasiadas piezas móviles y existe el riesgo de que se pierda precisión en el resultado con el uso por desgaste.



Figura 2.1 Medidora de cueros MD ROLL

Fuente: NBN Automação Industrial; Sistemas de medição para cueros semi-acabado/acabado, 2014

La forma de medir la superficie artesanalmente provoca pérdidas en la medición por lo que no se obtiene una medida real sino un estimado. Para la medición se utiliza una malla con cuadros de 30.48 [cm] x 30.48 [cm] representando 1[ft²], se la coloca sobre el cuero y se cuenta la cantidad de cuadros que son cubiertos, los bordes que faltan se los va introduciendo proporcionalmente en los espacios donde falta de llenar.



Figura 2.2 Medición artesanal con malla cuadriculada

2.2. EL CUERO

El cuero, uno de los principales productos de comercialización en el país. Se producen aproximadamente 350 mil cueros y pieles al año a nivel nacional, la mayor parte es para el sector del calzado, marroquinería y confecciones.

La provincia de Imbabura tiene una gran participación en cuanto a producir artículos de cuero de alta calidad, su matriz se establece en el cantón de Cotacachi, reconocida como La Ciudad del Cuero.

2.2.1. HOJA O MEDIA PIEL DE CUERO

A las pieles de bovinos se las trabaja enteras o cortadas, debido al tamaño que posee. A la piel entera se la divide a la mitad, el corte iría desde las vértebras hasta las patas.

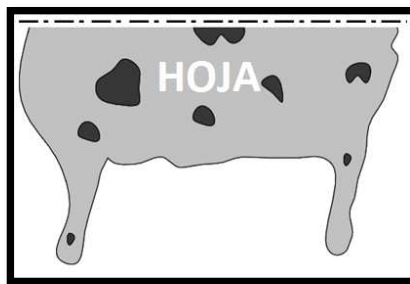


Figura 2.3 Hoja de cuero

Fuente: CUERONET; División Superficie de la piel, 2014

2.2.2. FALLAS DEL CUERO

La calidad del cuero y cantidad de imperfecciones se determina por la manera como se cría al animal, su modo de vida y alimentación.

El proceso de curtido tiende a acentuar las imperfecciones en los cueros de baja calidad, esto debido a que las pequeñas fallas por tórsalos o garrapatas pueden presentar agujeros, un pequeño corte puede agrandarse y llegar hasta a desgarrar la piel. Por tal razón, las empresas controlan la calidad y condiciones de las pieles desde el momento que se encuentra en el animal vivo, hasta el último instante antes de la venta del cuero terminado. Un control exhaustivo de la piel produce que se compre solo cueros de excelente calidad.

El curtidor ha de elegir su material con arreglo al tipo de cuero que desee producir. Por desgracia, no está en su mano remediar los diversos defectos que influyen en la calidad de los cueros y pieles en crudo.

Tales defectos se clasifican en dos grupos:

- a) defectos ante-mortem, o sea, daños sufridos por el animal vivo;*
- b) defectos post-mortem, es decir, daños causados durante el desuello, la conservación, el almacenamiento y el transporte (Mann, 1978).*

En los defectos ante-mortem se puede distinguir enfermedades, paracitos, vejez, corneadas no cicatrizadas, cortes o heridas en el traspaso al matadero, mordeduras, marcas por herretes (véase Figura 2.4); todos los problemas que haya obtenido el animal a lo largo de su vida.



Figura 2.4 Marca por herrete en cuero

Los defectos post-mortem son por lo general los que se refieren a la manipulación de la piel después del desuello, por ejemplo: conservación ineficaz, picaduras de insectos, mal secado, mordeduras de máquinas, des uniformidad en el tinturado, cortes y rayas de desuello.



Figura 2.5 Agujero mordedura de maquina

La mayoría de fallas que poseen las pieles son ocasionadas por el maltrato o poco cuidado que se le tuvo desde que se extrajo la piel del animal hasta q fue procesada, a esto se suma eventos que pudieron ocasionarse días antes que se sacrifique al animal que provoquen heridas abiertas.

Tabla 2.1 Daños mecánicos que afectan a los cueros y pieles

Tipo	Causa
MARCAS	Consecuencias del calor o frío extremos (marcado con hierro candente y con nieve carbónica)
ARAÑAZOS	Arbustos espinosos, alambre de espino y otros
CORNADAS	Luchas
CICATRICES	Luchas (mordeduras)
ABSCESOS	Inyecciones
MARCAS DE CAUTERIZACIÓN	Cerramiento de otras heridas mediante la aplicación de calor
CICATRICES PRODUCIDAS POR YUGOS Y ARREOS	Diseño o medidas inadecuados
IRRITACIONES PRODUCIDAS POR EXCREMENTOS	Suciedad y mugre general
DAÑOS PRODUCIDOS POR LA VEGETACIÓN	Penetración de semillas de malas hierbas en la superficie
CICATRICES PRODUCIDAS DURANTE LA ESQUILA	Técnicas inadecuadas y/o premura excesiva

Fuente: CUERONET; Defectos más habituales que afectan la calidad de los cueros y pieles,2014

Las fallas en cueros de animales enfermos son mucho más visibles ya que en el curtido no se absorben los productos químicos como en el resto de la piel (*véase Tabla 2.2*). La existencia de parásitos produce daños perjudiciales al cuero ya que tiende a formar agujeros en esas secciones por lo que disminuye la calidad de la misma .

Tabla 2.2 Enfermedades e infecciones parasitarias

Nombre	Causa
DERMATOMICOSIS	Hongo (<i>Trychophyton verucosum</i>)
DERMATITIS NODULAR	Herpesvirus
ESTREPTOTRICOSIS	Bacterias (<i>Dermatophilus congolensis</i>)
DEMODICOSIS	Ácaros parásitos (<i>Demodex bovis</i>)
BERNES	Mosca parásita (<i>Hypodermis bovis</i> , <i>H. Lineatum</i> y otras)
GARRAPATAS	<i>Boophilus micropilus</i> y otras
PIOJOS	Especies succionadoras (<i>lignognathus</i>) y mordedoras

Fuente: CUERONET; Defectos más habituales que afectan la calidad de los cueros y pieles,2014

Los procesos de prensado y pintado provocan que los defectos del cuero sumamente pequeños desaparezcan en un gran porcentaje, prácticamente elevando la categoría del cuero. Los defectos que son difíciles de disimular son los agujeros, cortes demasiado fuertes, corneadas (*véase Figura 2.6*) y marcas por herretes.



Figura 2.6 Marca de una corneada en cuero

Un factor importante en la determinación de la calidad es la homogeneidad de la pintura en cueros unicolor. Un cuero con problemas en la pintura da como resultado un gasto extra para la fábrica ya que tiene que regresar a ser lijado, fondeado y repintado. Estos errores se dan cuando no se seca bien la pintura (*véase Figura 2.7*), la mezcla no se preparó correctamente o el pintor no tuvo cuidado con la manipulación del cuero.



Figura 2.7 Cuero sin secar correctamente

2.2.3. VALOR DEL CUERO

El valor que se le da al cuero depende de la calidad y medida de la superficie que posee el cuero. La fábrica GAMACUEROS por el momento no posee un método matemático para calificar el cuero de acuerdo al área utilizable que posee, esta clasificación se la realiza empíricamente por decisión de los trabajadores según la cantidad de daños que cada cuero posee, dando a los cueros que no tienen marcas ni agujeros la categoría de A.

Para realizar la categorización del cuero, se va a tomar como referencia a la clasificación que nombra CORFOGA (2014): “... La metodología utilizada por Coopemontecillos donde en la clasificación se presentan tres categorías para los cueros que son: A, B y C.

Tabla 2.3 Categorización del cuero

CLASE	ÁREA ÚTIL [%]	CARACTERÍSTICAS	GRADO DE DAÑOS
A	90	No se permite cicatrices abiertas, mucha cicatriz cerrada solamente fuera de la banda central, no se permite tórsalos, garrapatas ni muchas arrugas, permite un solo daño no mayor de 5 cm de diámetro fuera de la banda central, no más de 2 marcas de fierro pequeñas, en un solo tercio cerradas y cercanas al borde del lomo.	1 y 2
B	60 - 90	No se permite defectos extendidos a más de 1/3 de la hoja, defectos mayores a 10 cm. de diámetro, tórsaleras muy grandes ni gran cantidad de garrapatas, permite no más de dos defectos en un mismo tercio, poca cicatriz abierta en un solo 1/3, no más de 3 marcas de fierro, cerradas y además cercanas al borde del lomo	3
C	1 - 60	Presentan gran cantidad de defectos	4 y 5

Fuente: CORFOGA; Clasificación de Cueros, 2014

En la figura 2.8 se presentan algunas de las lesiones típicas del cuero y su división.

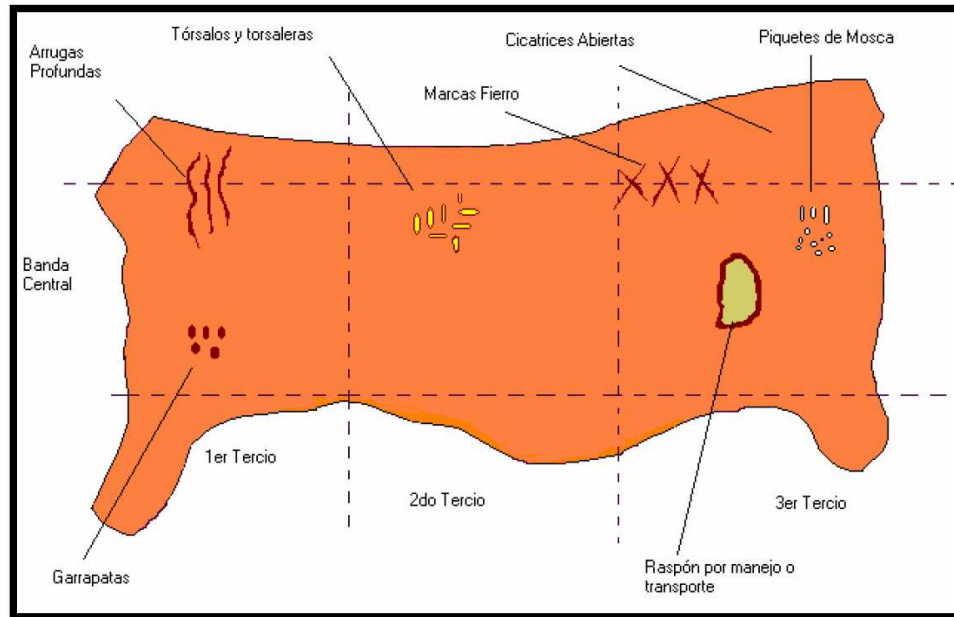


Figura 2.8 Lesiones típicas y divisiones del cuero

Fuente: CORFOGA; Clasificación de Cueros, 2014

El valor monetario final del cuero se lo obtiene con el área final de una hoja de cuero y la clase asignada, para cada clase de cuero se tiene un valor diferente por ft². Para el cálculo del precio del cuero se aplica la siguiente ecuación:

Ecuación 2.1 Valor monetario del cuero

$$\text{valor monetario} = (\text{precio correspondiente a la clase}) \times (\text{área final})$$

A continuación, se describe el precio final para algunos tipos de cuero:

Tabla 2.4 Valor del cuero por clase

CLASE	PRECIO [\$/FT ²]	ÁREA [FT ²]	VALOR [\$]
A	1.60	25	40
B	1.40	25	35
C	1.20	25	30

2.3. VISIÓN ARTIFICIAL

La visión artificial es un derivado de la Inteligencia Artificial, tiene como finalidad programar a un ordenador para que logre interpretar y dar significado a las imágenes por medio de una cámara. El uso de esta herramienta nos proporciona un sinnúmero

de aplicaciones, en algunas de estas se puede reemplazar al operador para reducir errores de verificación y aumentar la productividad.

Los elementos que comprenden un sistema de visión artificial son: el objeto a ser analizado, la fuente de iluminación, la cámara y el ordenador donde consta el programa para el que fue diseñado el sistema.



Figura 2.9 Elementos del sistema de visión artificial

Entre los procesos que componen la visión artificial para el análisis de una imagen se encuentran: la obtención de la imagen, el procesamiento e interpretación de resultados.

El uso de cámaras como sensores para captar información del entorno produce que se tome en cuenta diversos factores que pueden perturbar la calidad de la imagen.

... Las mayores dificultades surgen cuando los sistemas tienen que operar en condiciones de iluminación variable y no controlada, con sombras, o tienen que tratar con objetos complejos y difíciles de escribir, y con objetos ocultos, como ocurre en las escenas con objetos no rígidos. Algunos de estos problemas se pueden reducir si tratamos con entornos artificiales como el interior de los edificios y, por consiguiente, la visión por computadora ha sido aplicada en dichos entornos con mayor éxito... (Nilsson, 2001).

Estos problemas se los puede minimizar ya sea cambiando el espacio físico donde va a estar situado el sistema o controlando artificialmente la cantidad de luz al que va a estar expuesto.

2.3.1. SOFTWARE

Para facilitar la interacción entre el usuario y la máquina se ha desarrollado una extensa gama de programas que ayudan a entrelazar el ordenador con periféricos. Para la selección del software se debe tomar en cuenta la simplicidad de la programación y ayuda de complementos que tenga dicho instrumento.

El procesamiento digital de imágenes se remonta a la década de los 60 y está directamente ligada con el desarrollo y evolución de las computadoras. Su progreso ha ido de la mano con el desarrollo de las tecnologías de hardware, ya que requiere un alto poder y recursos computacionales para almacenar y procesar las imágenes. (García, 2008)

Un software de visión artificial se basa en el análisis e interpretación de píxeles para el seguimiento, detección o clasificación de objetos para dar sentido a una escena.

2.3.1.1. MATLAB R2015a

Matlab posee un entorno interactivo para la programación y simulación mediante cálculo vectorial, facilita el desarrollo de algoritmos para la creación de modelos y análisis de datos, por medio de la ayuda de sus toolbox y funciones matemáticas incorporadas.

MATLAB se puede utilizar en una gran variedad de aplicaciones, tales como procesamiento de señales y comunicaciones, procesamiento de imagen y vídeo, sistemas de control, pruebas y medidas, finanzas computacionales y biología computacional. Más de un millón de ingenieros y científicos de la industria y la educación utilizan MATLAB, el lenguaje del cálculo técnico. (Mathworks, 2014)

Matlab es un software que soporta operaciones con matrices y vectores, obteniendo resultados inmediatos. Permite la creación de Scripts y el uso de funciones para reutilizar y automatizar el trabajo. Con el uso de este software se añade la posibilidad de diseñar interfaces gráficas personalizadas que se pueden distribuir en forma de archivos ejecutables.

Los complementos que vienen con Matlab facilita la resolución de aplicaciones como: procesamiento de señales, procesamiento de video, diseño de sistemas de control, etc.

Matlab es utilizado frecuentemente para proyectos de visión artificial debido a su facilidad de integración con cámaras profesionales y webcams, aportando en el desarrollo de aplicaciones en tiempo real.

Para mejorar la experiencia del operador en el sistema de visión artificial se eligió Matlab R2015a, debido a que nos proporciona los toolbox de Computer Vision System e Image Processing; estos complementos facilitan el proceso de anular las distorsiones de la cámara proporcionándonos imágenes nítidas y bien distribuidas.

Requerimientos mínimos del Sistema: Intel o AMD x64 que soporte set de instrucciones SSE2, espacio de disco 1[GB] para MATLAB y de 3 a 4 para una instalación típica, 1024 [MB] de RAM.

2.3.1.1.1. Computer Vision System Toolbox

Esta herramienta facilita la simulación del procesamiento de video al proporcionar funciones y algoritmos para la detección y rastreo de objetos, además permite detectar y extraer características de una imagen.

Computer Vision System Toolbox ofrece una aplicación y funciones para realizar todas las tareas esenciales del flujo de trabajo para calibración de la cámara:

- *Detección y localización totalmente automático de la ubicación del patrón del tablero de ajedrez incluyendo detección de equinas con precisión subpixel.*
- *Estimación de parámetros intrínsecos y extrínsecos que incluye eje de inclinación.*
- *Cálculo de coeficientes de distorsión radial y tangencial de la lente.*
- *Corrección de distorsión óptica. (Mathworks, 2014)*

De este toolbox se utiliza Camera Calibrator App para la obtención de los parámetros intrínsecos y extrínsecos de la lente utilizando el patrón del tablero de ajedrez, estos datos ayudan a disminuir las distorsiones en la imagen para poder obtener un resultado preciso.

Los pasos para obtener los parámetros usando este toolbox se detalla a continuación:

1. Una vez iniciado el toolbox cargamos un mínimo de 20 fotos tomadas con la cámara al tablero de ajedrez en diferentes posiciones, ingresándolas al ingresar en el botón Add Images; para maximizar precisión la cámara debe estar estática en posición paralela al tablero de ajedrez.
2. Se ingresa la medida real de uno de los lados de las casillas del tablero de ajedrez, para este caso 50 [mm].
3. Una vez aceptadas las imágenes el software analiza la posición en el espacio de todos los vértices de los casilleros dentro del tablero, mostrando en la interfaz las coordenadas de origen y los ejes “X” y “Y”.

Adicional se escoge las características físicas de la cámara.

4. Se elige el uso de 3 coeficientes a causa de que el lente de la cámara es gran angular.
5. Se selecciona Compute Skew puesto que no existe la seguridad que los ejes “X” y “Y” estén perpendiculares, estos errores se deben a imperfecciones en la lente.
6. Por último, se selecciona Compute Tangential Distortion, con esta herramienta eliminamos la posibilidad de que exista un error debido a que el sensor no se encuentre paralelamente con la lente y seleccionamos Calibrate.
7. Al finalizar el proceso de calibración se exporta los datos en un archivo cameraParameters.mat para su posterior uso.

Para el posterior uso de los parámetros se emplea el siguiente código:

```
load('cameraParameters.mat', 'cameraParams');%carga de parámetros
im=imread('imagencondistorsion.jpg');%adquisición de imagen
imund=undistortImage(im,cameraParams); %elimina distorsión
```

El resultado de usar este toolbox se puede apreciar en la figura, donde la imagen “a” tiene distorsión y la imagen “b” tiene eliminada la distorsión.

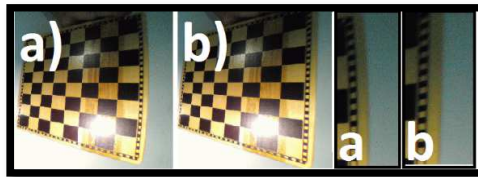


Figura 2.10 Tablero de ajedrez con distorsión y sin distorsión

2.3.1.1.2. Image Acquisition toolbox

Este toolbox proporciona algoritmos para el análisis de imágenes, reducción de ruido, registro de imágenes. Además, facilita las operaciones con imágenes dándonos la

posibilidad de manipular imágenes, ajustar contraste, unir imágenes, crear contornos, analizar formas y examinar regiones de píxeles.

Otras funciones que posee son:

- *Mejora y filtrado de imágenes y enfoque de imágenes borrosas.*
- *Análisis de imágenes, incluyendo segmentación, morfología, extracción de funciones y medición.*
- *Transformaciones geométricas y métodos de registro de imágenes basados en intensidad.*
- *Transformaciones de imágenes, incluyendo FFT, DCT, Radon y proyección de haz de rayos en abanico.*
- *Flujos de trabajo para procesar, visualizar y navegar por imágenes arbitrariamente grandes.*
- *Aplicaciones Visor de imágenes y Visor de vídeos.*
- *Importación y exportación de archivos DICOM. (Mathworks, 2014)*

Esta herramienta ayuda de forma interactiva a configurar la cámara que se va a utilizar, configurar la cantidad de imágenes a adquirir, delimita la región de interés y muestra una vista previa con su respectiva programación en lenguaje C.

Este toolbox es compatible con imágenes de tipo RGB, YUV o escala de grises, permitiendo convertir un formato a otro con solo utilizar una función de intercambio.

2.3.1.1.3. Función bwconncomp o logical

La función “bwconncomp” al igual que la función “logical” son propias de matlab, estas permiten etiquetar a los objetos conectados entre sí en una imagen binaria usando la sintaxis Etiqueta=logical(BW,n), donde n es el tipo de vecindad que puede ser de 4 u 8 y BW es la imagen en binario.

La función regionprops da a conocer las propiedades que pertenecen a cada grupo de objetos que entrega la función anterior, las cuales se enlistan en la tabla 2.5:

Tabla 2.5 Operaciones morfológicas para la función *regionprops*.

PROPIEDAD	DESCRIPCIÓN
Area	Calcula el área en pixeles cuadrados de la región
BoundingBox	Calcula la posición y dimensiones del mínimo rectángulo
Centroid	Posición del centroide (centro de masa) de la región
Eccentricity	Número escalar que da la excentricidad de la imagen
MayorAxisLenght	Longitud del eje de mayor longitud de la región (en pixeles)
MinorAxisLenght	Longitud del eje de menor longitud de la región
Orientation	El ángulo (en grados) del eje x y el eje de mayor longitud de la región
Image	Imagen binaria del mismo tamaño que la frontera (bounding box) de la región

Fuente: García; Visión Artificial y Procesamiento de Imágenes usando Matlab, 2008

2.3.1.1.4. Funcion *bwmorph*

La función *bwmorph* es nativa de matlab, realiza operaciones morfológicas con imágenes binarias, su sintaxis es: `BW2=bwmorph(BW,'operación',#repeticiones)`

Tabla 2.6 Operaciones morfológicas para la función *bwmorph*.

OPERACIÓN	DESCRIPCIÓN
'bothat'	Realiza la operación morfológica "bottom hat", resalta detalles brillantes.
'clean'	Elimina pixeles aislados(1's individuales que están rodeados por 0's)
'close'	Realiza la operación morfológica closing, dilatación seguida de erosión
'dilate'	Realiza la dilatación usando un elemento estructurante ones(3)
'erode'	Realiza la erosión usando un elemento estructurante ones(3)
'fill'	Llena pixeles aislados (0's individuales que están rodeados por 1's)
'open'	Realiza la operación morfológica opening (erosión seguida de dilatación)
'remove'	Remueve pixeles interiores
'skel'	Con n=Inf, elimina los pixeles en los límites de los objetos
'tophat'	Realiza la operación morfológica "top hat", resalta detalles oscuros

Fuente: García; Visión Artificial y Procesamiento de Imágenes usando Matlab, 2008

2.3.1.1.5. Interfaz gráfica para el usuario (gui).

Una GUI permite al desarrollador de programas crear un ambiente controlado para un usuario, por medio de botones, selectores, cuadros, texto, etc.; con esta herramienta un software es mucho más que código.

Se describe algunos de sus elementos gráficos:

- **Push Button:** Permite llamar a un evento al presionarse, cambia de estado mientras se mantenga presionado, puede usarse como pulsador para un contador.
- **Slider:** Representa valores comprendidos en un rango definido, mediante una barra deslizante.
- **Radio Button y Check Box:** Estos dos elementos permiten seleccionar o deseleccionar una opción, la cual convoca a un evento.
- **Static text:** Muestra como texto el resultado de la ejecución de una tarea o a su vez sirve de etiqueta, para usar un dato numérico y mostrarlo como texto se emplea la función `num2str(A)` donde A es la variable en la que se guardó el dato. El usuario no puede modificar el texto.
- **Edit text:** Permite el ingreso y visualización de texto que puede ser usado posteriormente como dato, por ejemplo, si se quiere usar un numero se debe ingresar la función `str2num(A)`, donde A es la variable en la que se guardó el dato. El usuario puede modificar el texto
- **Pop-up menu:** Permite seleccionar una opción de una lista desplegable que se muestra al hacer clic en la flecha.
- **List box:** Permite seleccionar una o más opciones de una lista deslizable.
- **Toggle Button:** Tiene la finalidad de un botón de dos posiciones on-off, puede ser usada para iniciar un evento con el primer clic y terminarlo con otro.
- **Table:** Crea una tabla que se carga posteriormente con datos.
- **Axes:** Permite visualizar imágenes o gráficos.
- **Panel:** Reúne objetos como axes, tablas permitiendo organizarlas de una forma más adecuada
- **Button Group:** Agrupa selectores y botones con la finalidad de dar una mejor presentación al usuario.

Para próximas referencias se debe considerar que se da a conocer el código de forma muy general, por ejemplo: si se tiene la función “close GUI” quiere decir que se reemplaza la palabra GUI por el nombre de la ventana que se desea cerrar para este ejemplo es PRINCIPAL, resultando `close PRINCIPAL`.

Igualmente si se desea llamar a un componente en especial se expresa de la siguiente manera, por ejemplo: se requiere quitar el check de la casilla de un checkbox llamado “ckeckdistorsion”, para lo cual usaremos la línea de código `set(handles.tag_checkbox,'value',0)`, en esta línea se reemplaza “tag_checkbox” por “ckeckdistorsion” resultando `set(handles.ckeckdistorsion,'value',0)`.

Añadido a estos complementos Matlab proporciona GUI's predefinidas. Se muestra unos ejemplos con ellas.

Por ejemplo: se requiere una ventana de Aviso, queda de la siguiente manera.

warndlg ('Asegúrese de haber colocado bien el cuero', 'Aviso');

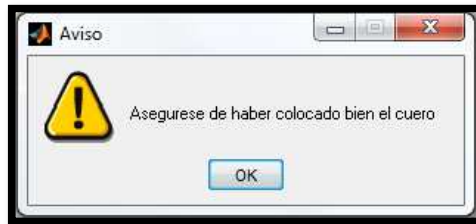


Figura 2.11 GUI de aviso

Para usar la ventana de pregunta se debe añadir las respuestas al código:

questdlg ('Se encuentra seguro de continuar', 'Pregunta', 'Si', 'No', 'No');

El ultimo “No”, sirve para resaltar el botón No.

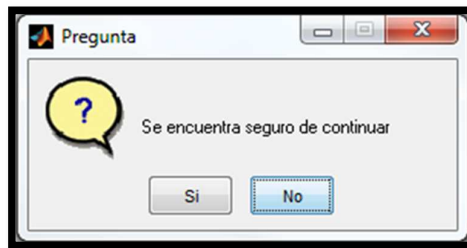


Figura 2.12 GUI de aviso

2.3.2. FUENTES DE ILUMINACIÓN

Cuando se habla de iluminación se debe tomar en cuenta que existen dos tipos de fuentes de luz: la natural (el sol), la artificial (un foco).

La fuente de luz artificial puede ser puntual o difusa dependiendo el área de incidencia con respecto al objeto que ilumina.

La luz difusa permite cubrir un área más extensa ya que incide sobre el cuerpo desde varias direcciones, creando un ambiente con iluminación uniforme y suave, dando claridad a todo un espacio sin originar sombras.

La luz puntual se origina en un punto reducido al objeto al que se desea iluminar, creando sombras en cuerpos sólidos y reflejos en objetos brillantes.

Con la finalidad de dar realce a ciertos rasgos característicos y minimizar información no valiosa de un objeto se puede controlar la dirección de la luz de la siguiente forma:

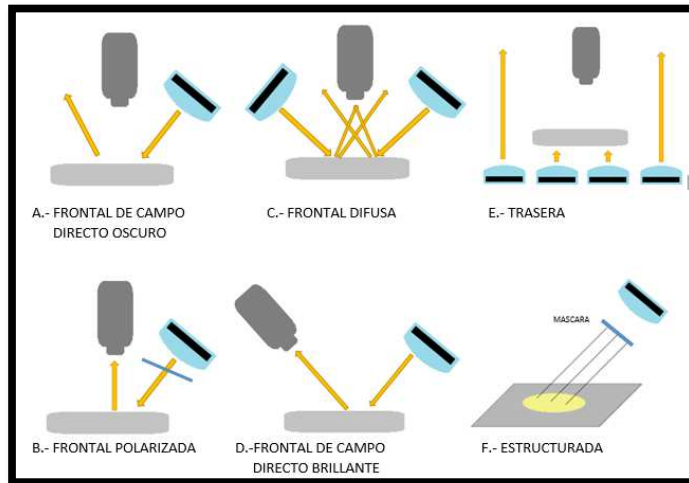


Figura 2.13 Disposición de las fuentes de luz en visión artificial

Fuente: Monsó, Julia; Sistema de identificación y control automáticos, 1994

- **Luz frontal:** Esta técnica se usa para realzar la textura, y características tridimensionales de un objeto.
- **Luz trasera:** Genera solamente la silueta del objeto, es usada para identificar la presencia o forma de este.
- **Luz estructurada:** Proyecta la imagen que se encuentra trazada en la máscara, es usada para realzar partes específicas de un objeto.

2.3.3. LENTE

Es el dispositivo u objeto transparente de forma circular, cuyo objetivo es enviar los rayos de luz paralelos al eje hacia al foco. Todo lente está restringido por dos superficies, en la que por lo menos una es curva. Clasificándose en:

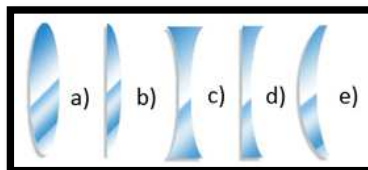


Figura 2.14 Tipos de lentes

Fuente: Cromer, Alan; Física en la ciencia y en la industria, 2006

- a) Biconvexas
- b) Planoconvexas
- c) Bicóncava
- d) Planocóncava
- e) Menisco

2.3.3.1. Objetivo

El objetivo es el conjunto de lentes que permite re-direccionar los rayos de luz para formar una imagen en el sensor.

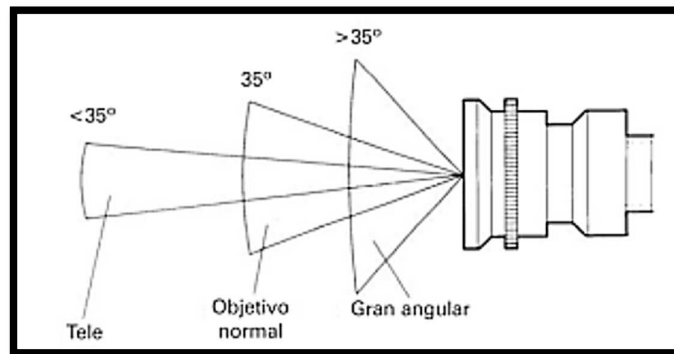
Se los clasifica según la distancia focal y el ángulo de visión en:

Tabla 2.7 Clasificación de objetivos

Tipo de Objetivo	Distancia focal [MM]	Angulo de visión [grados]
Gran angular	28 – 40	60 – 180
Normal	45 – 70	40 – 65
Teleobjetivos	70 – 300	< 40

- **Objetivo gran angular:** Presentan un gran rango de distorsión, es utilizado para obtener efectos especiales o abarcar áreas grandes.
- **Objetivo Normal:** La profundidad de campo la mantiene moderada y se asemeja a la visión del ojo humano.
- **Objetivo Teleobjetivo:** La profundidad de campo es reducida, capta imágenes que el ojo humano apenas percibiría y las hace tamaño normal, utilizado para fotografía de fauna silvestre.

Figura 2.15 Distintos tipos de objetivos



Fuente: Gormaz, Isidro; Técnicas y procesos en las instalaciones singulares en los edificios, 2010

2.3.4. DISTORSIONES ÓPTICAS

Las distorsiones son deformaciones ópticas causadas por el tipo de objetivo que usamos al momento de la foto, desfigurando la imagen capturada de la real.

Se puede definir la distorsión como la variación o alteración que sufre el aspecto del objeto registrado en una imagen. En la práctica, las mayores distorsiones se producen cuando se trabaja con distancias focales muy cortas. (Torres, Pomaras, & Gil, 2002)

La variación de la distancia focal en el sistema óptico provoca dos tipos de distorsiones conocidas como tipo barril y cojín.

Tabla 2.8 Tipos de distorsiones ópticas

DISTORSIÓN	FORMA	CAUSA
Barril	Las líneas ubicadas a los extremos de la imagen tienden a curvarse hacia el exterior (<i>véase Figura 2.16-c</i>).	Objetivo gran angulares
Cojín	Las líneas ubicadas a los extremos de la imagen tienden a curvarse hacia el interior (<i>véase Figura 2.16-b</i>).	Objetivo teleobjetivos

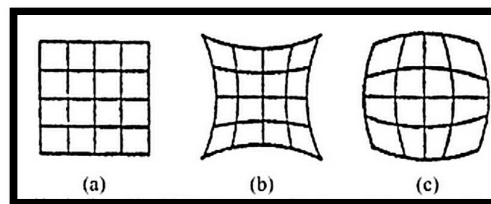


Figura 2.16 Tipos de distorsiones ópticas

Fuente: Bueno, Juan; Introducción a la óptica instrumental, 1999

2.3.5. PIXEL

El pixel es la menor unidad que forma una imagen digital, cada uno de estos representa una pequeña área, para lograr observarlo basta con hacer zoom en la imagen.



Figura 2.17 Representación del pixel en una imagen

Una imagen digital está formada por pixeles organizados en forma de una matriz, cada uno de estos representa un punto que contiene el valor de elemento de la matriz que identifica el nivel de luz en dicho punto.

Al valor del pixel se lo asigna según la resolución de la imagen, como estándar se usa 8 bits, obteniendo 256 niveles de gris, en donde el valor de 0 se le asigna al color negro y el 255 al color blanco.

Para obtener el número total de pixeles de una imagen basta con multiplicar el alto por el ancho de la imagen, asemejándose a la obtención del área del rectángulo; mientras mayor sea la cantidad de pixeles es mejor la resolución de la imagen.

2.4. INTELIGENCIA ARTIFICIAL

La inteligencia artificial tiene como fin el solucionar problemas de la vida diaria mediante la creación y diseño de sistemas computacionales basados en la inteligencia humana.

El comportamiento inteligente supone percibir, razonar, aprender, comunicarse y actuar en entornos complejos. Una de las metas a largo plazo de la Inteligencia Artificial es el desarrollo de máquinas que puedan hacer todas estas cosas igual, o quizá incluso mejor, que los humanos. (Nilsson, 2001)

Estas cualidades anteriormente descritas viabilizan a los desarrolladores de software la construcción de sistemas que logren reconocer tanto la escritura, el habla o patrones.

2.4.1. Redes Neuronales Artificiales

La inteligencia artificial emplea a las RNA asemejando el modelo biológico de las neuronas, imitando su comunicación, interconexión en paralelo y organización

jerárquica. La RNA debe de ser previamente entrenada ajustando su respuesta final en base a los ejemplos de la capa de entrada.

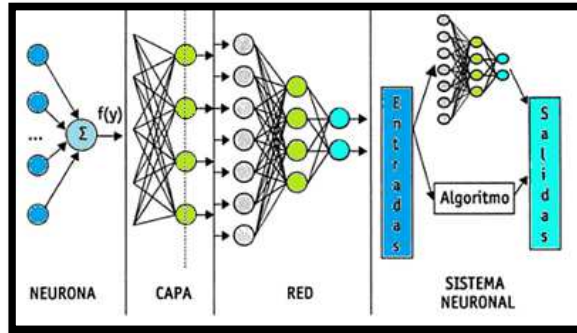


Figura 2.18 Estructura jerárquica de un sistema basado en RNA

Fuente: (Flórez López & Fernández, 2008)

Desglosando el Sistema Neuronal de la Figura 2.27 se consigue visualizar:

- **Entradas:** Estas tienen un peso correspondiente por cada una de ellas.
- **Algoritmo:** El cual contiene las funciones de entrada, funciones de activación y función de salida.
- **Salidas:** Contiene el valor de respuesta según la función de activación y las entradas asignadas.

2.4.1.1. Función de entrada

La función de entrada comúnmente usada es:

Ecuación 2.2 Sumatoria de las entradas pesadas

$$Red_j = \sum_{i=1}^N w_{ij} * X_i + \theta_j(t)$$

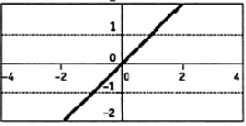
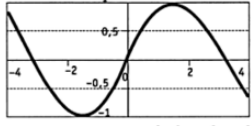
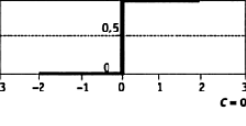
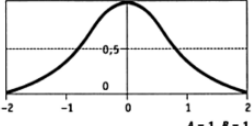
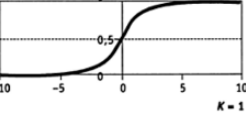
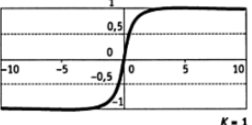
Fuente: (Flórez López & Fernández, 2008)

Donde:

w_{ij} : Pesos de entrada; X_i : Vector de entrada; θ_j : Umbral

2.4.1.2. Función de activación

Tabla 2.9 Principales tipos de funciones de activación o transferencia

<p>Función de transferencia identidad o lineal: $a_j = f(\text{Net}_j(t)) = \text{Net}_j$ Rango = $(-\infty, +\infty)$</p> 	<p>Función de transferencia sinusoidal: $a_j = f(\text{Net}_j) = A \cdot \text{sen}(\omega \cdot \text{Net}_j + \varphi)$ Rango = $(-1, 1)$</p>  <p style="text-align: right;">$A = 1, \omega = 1, \varphi = 1$</p>
<p>Función de transferencia escalón: $a_j = f(\text{Net}_j) = \begin{cases} 1 & \text{si } \text{Net}_j \geq c \\ 0 & \text{si } \text{Net}_j < c \end{cases}$ Rango = $\{0, 1\}$</p>  <p style="text-align: right;">$c = 0$</p>	<p>Función de transferencia gaussiana: $a_j = f(\text{Net}_j) = Ae^{-B \text{Net}_j ^k}$ Rango = $(0, 1)$ $A > 0, B > 0$</p>  <p style="text-align: right;">$A = 1, B = 1$</p>
<p>Función de transferencia sigmoide o logística: $a_j = f(\text{Net}_j) = \frac{1}{1 + e^{-k \text{Net}_j}}$ Rango = $(0, 1)$ $k > 0$</p>  <p style="text-align: right;">$k = 1$</p>	<p>Función de activación tangente, tangente hiperbólica o tangencial: $a_j = f(\text{Net}_j) = \frac{e^{k \text{Net}_j} - e^{-k \text{Net}_j}}{e^{k \text{Net}_j} + e^{-k \text{Net}_j}}$ Rango = $(-1, 1)$ $k > 0$</p>  <p style="text-align: right;">$k = 1$</p>

Fuente: (Flórez López & Fernández, 2008)

Esta función define el estado de la neurona, ya sea activa (1) o inactiva (0 o -1). Según el tipo de función que se use la salida tiene cualquier valor dentro del rango determinado simulando las neuronas biológicas.

2.4.1.3. Función de salida

De esta función se obtiene los valores de la capa de salida de la neurona, además de obtener el valor que se transmite a las neuronas asociadas.

Habitualmente, la función de salida coincide con la función identidad $F(x)=x$, por lo que el estado de activación de la neurona se considera su salida final. (Flórez López & Fernández, 2008).

Las funciones de salida normalmente usadas son:

- **Binaria:** Toma el valor de 1 si el valor de entrada es mayor al valor del umbral caso contrario toma el valor de 0.
- **Ninguna:** Si el valor de la salida es el mismo que la entrada, es decir la función identidad.

CAPITULO III

3. MARCO METODOLÓGICO

3.1. METODOLOGÍA

Una investigación se rige a métodos sobre el cual se da cumplimiento de los objetivos planteados para resolver un problema.

Para este caso se ha escogido usar el método cuantitativo debido a que tanto el procesamiento como en el resultado de este trabajo es mediante información numérica.

Con la metodología escogida se analiza cada objetivo indicando el tipo de diseño que se usa y los pasos que se realizan para su desarrollo y cumplimiento.

3.1.1. Objetivo 1

Generar un algoritmo de adquisición y procesamiento digital de imágenes.

3.1.1.1. *Tipo de diseño*

Diseño experimental.

3.1.1.1.1. Actividades

- Diseño de algoritmo para adquisición de imágenes en tiempo real.
- Diseño de algoritmo para eliminación de distorsión en el lente de la cámara
- Diseño de algoritmo que generalice imágenes tanto para cueros claros como oscuros.

3.1.2. Objetivo 2

Seleccionar las técnicas de visión adecuadas para el control de calidad.

3.1.2.1. *Tipo de diseño*

Diseño experimental.

3.1.2.1.1. Actividades

- Selección del tipo de luminaria.
- Selección de método para identificación de patrones.

3.1.3. Objetivo 3

Desarrollar un programa que permita calcular la superficie del cuero.

3.1.3.1. Tipo de diseño

Diseño experimental.

3.1.3.1.1. Actividades

- Diseño de algoritmo que permita el cálculo de área de una imagen.

3.1.4. Objetivo 4

Desarrollar un programa que permita determinar la categorización del cuero de acuerdo a normas de calidad.

3.1.4.1. Tipo de diseño

Diseño no experimental, de tipo transversales.

3.1.4.1.1. Actividades

- Análisis de normas.

3.1.4.2. Tipo de diseño

Diseño experimental.

3.1.4.2.1. Actividades

- Diseño de algoritmo que permita categorizar cueros de colores tanto claros como oscuros.

3.1.5. Objetivo 5

Diseñar la estructura que contendrá el sistema de visión artificial.

3.1.5.1. Tipo de diseño

Diseño experimental

3.1.5.1.1. Actividades

- Diseño de elementos que contiene la luminaria inferior
- Diseño de elementos que contiene la cámara y la luminaria superior

3.1.6. Objetivo 6

Implementar el sistema de control de calidad y medición de superficie.

3.1.6.1. Tipo de diseño

Diseño de campo

3.1.6.1.1. Actividades

- Construcción de elementos que contenga el sistema
- Creación de ejecutable para el software

3.1.7. Objetivo 7

Realizar pruebas y depuración de errores en el funcionamiento del sistema e interfaz gráfica.

3.1.7.1. *Tipo de diseño*

Diseño experimental, estadístico

3.1.7.1.1. Actividades

- Corrección de valores de variables para precisión de medida, detección de pieza de cuero y detección de patrones.
- Pruebas de funcionamiento del sistema
- Prueba de funcionamiento de la luminaria
- Prueba de funcionamiento del sistema eléctrico

3.1.8. Objetivo 8

Elaborar manuales de usuario y de calibración del sistema.

3.1.8.1. *Tipo de diseño*

Diseño no experimental, de tipo ex post facto

3.1.8.1.1. Actividades

- Descripción de los componentes del sistema
- Requisitos para el software
- Requisitos para el hardware
- Descripción del modo de uso para la correcta calibración del sistema
- Descripción del modo de uso del sistema en general
- Descripción del mantenimiento del sistema en general

3.2. CÁMARA

El sistema de visión artificial no comprende solo el uso de una cámara, sino la unión de pequeños módulos que lo complementan. Por ello se toma en cuenta diversos factores que pueden influir en el resultado que se desea.

3.2.1. Alternativas de cámaras

Para solucionar el problema presente, se plantean alternativas, cada una de ellas exhibe una solución para abarcar el área del cuero manteniendo una longitud considerable para el enfoque entre la cámara y el objeto. Para el efecto, se plantea dos alternativas de solución:

- Alternativa 1: NI 1778 Smart Camera (AI 1)
- Alternativa 2: Genius WideCam F100 (AI 2)

3.2.1.1. NI 1778 smart camera (al 1)

Esta cámara inteligente tiene procesador interno de 1.6 [Ghz] con un grado de protección IP67, de uso industrial. Posee sensor de imagen Sony CCD monocromático con resolución de 5 [Mp]. Integrado con E/S digitales de alto rendimiento para aplicaciones industriales en tiempo real.



Figura 3.1 NI 1778 Cámara inteligente

Fuente: National Instrument; Ni 1778 Smart Camera, 2014

Tabla 3.1 Resumen de especificaciones NI 1778 Smart Camera

Característica	Especificación
Enfoque	Manual
Conexión	Gigabit – Ethernet
Máximo rango de cuadros	15 [fps]

Tipo de sensor	CCD monocromática
Resolución de fotografía	5 [MP]
Salida de video	VGA
Angulo de visión	90°
Precio	1020 [\$]

Fuente: National Instrument; Ni 1778 Smart Camera, 2014

... El sistema operativo determinístico en tiempo real ofrece mayor fiabilidad para sistemas de inspección que requieren operación continua y a largo plazo. (National Instrument, 2014)

El uso de esta cámara está destinado a líneas de producción continuas en las que un sistema en tiempo real es de vital importancia.

3.2.1.2. *Genius widecam f100 (al 2)*

A diferencia de la anterior esta cámara depende de un ordenador para su funcionamiento. Posee un lente gran angular con un ángulo de visión de 120°, una resolución de 12[MP] y un lente de cristal lo que permite ver con más nitidez.



Figura 3.2 WideCam F100

Fuente: Genius, KYE Systems Corp; Cámara web Full HD gran angular, 2014

Tabla 3.2 Resumen de especificaciones WideCam F100

Característica	Especificación
Enfoque	Manual
Conexión	USB 2.0
Máximo rango de cuadros	30 [fps]
Tipo de sensor	CMOS
Resolución de fotografía	12 [MP]
Salida de video	1080p FHD
Angulo de visión	120°
Precio	97 [\$]

Fuente: Genius, KYE Systems Corp; Cámara web Full HD gran angular, 2014

Requerimientos mínimos del Sistema: Intel Core2 1.8 [GHz], 512[MB], 200[MB] de espacio de disco, CD-ROM, Conexión Internet, puerto USB 2.0

3.2.2. Selección de la alternativa adecuada

Para la selección de la mejor alternativa para nuestro proyecto se escoge el Método Ordinal Corregido De Criterios Ponderados para llegar a obtener el mejor resultado según nuestras necesidades. Donde se establece lo siguiente:

- *1 Si el criterio (o solución) de las filas es superior (o mejor; >) que el de las columnas.*
- *0,5 Si el criterio (o solución) de las filas es equivalente (=) al de las columnas.*
- *0 Si el criterio (o solución) de las filas es inferior (o peor; <) que el de las columnas.*

Luego, para cada criterio (o solución), se suman los valores asignados en relación a los restantes criterios (o soluciones) al que se le añade una unidad (para evitar que el criterio o solución menos favorable tenga una valoración nula); después, en otra columna se calculan los valores ponderados para cada criterio (o solución).

Finalmente, la evaluación total para cada solución resulta de la suma de productos de los pesos específicos de cada solución por el peso específico del respectivo criterio.

(Riba, 2002)

Los criterios de valoración que se toman en cuenta son:

- Costo moderado: El costo final del sistema debe resultar económico comparado con el costo de importación de una maquina similar.
- Resolución: Ya que el sistema va a medir áreas es de vital importancia tener la mayor resolución para disminuir el margen de error.
- Angulo de visión: Con el fin de minimizar la distancia entre el objeto y el lente se vio la necesidad utilizar cámaras gran angulares, tomando en cuenta las distorsiones que podrían provocar.
- Tipo de sensor: Define la calidad de la imagen
- Estética: De acuerdo a las exigencias del mercado debe ser atractiva a la vista del cliente.

El tipo de conexión no se toma en cuenta ya que se trata de un sistema de monitoreo periódico y no continuo. Si bien una conexión USB 2.0 transmite a 480 [Mbit/s] y una Gigabit Ethernet a 1 [Gbit/s] la demora de transmisión de la imagen es mínima. Por tal motivo la transmisión de datos en tiempo real no es de vital necesidad.

A partir de estos datos se realiza la siguiente evaluación de alternativas:

Tabla 3.3 Evaluación del peso específico de cada criterio

CRITERIO	COSTO	RESOLUCIÓN	ANGULO DE VISIÓN	TIPO DE SENSOR	ESTÉTICA	$\Sigma+1$	PONDERADO POR CRITERIO
COSTO		0.5	0	0.5	0.5	2.5	0.16
RESOLUCIÓN	0.5		1	0.5	1	4	0.26
ANGULO DE VISIÓN	1	0		0.5	1	3.5	0.23
TIPO DE SENSOR	0.5	0.5	0.5		1	3.5	0.23
ESTÉTICA	0.5	0	0	0		1.5	0.1
					suma	15	1

Evaluación de los pesos específicos de las alternativas para cada criterio:

Tabla 3.4 Evaluación del peso específico por criterio

CRITERIO	COSTO	AL 1	AL 2	PONDERADO POR ALTERNATIVA
Costo	AL 1		0	0.33
	AL 2	1		0.66
Resolución	AL 1		0	0.33
	AL 2	1		0.66
Angulo de visión	AL 1		0	0.33
	AL 2	1		0.66
Tipo de sensor	AL 1		0	0.33
	AL 2	1		0.66
Estética	AL 1		0	0.33
	AL 2	1		0.66

Calculo de la tabla de conclusiones:

Ecuación 3.1 Calculo de conclusiones del método ordinal corregido de criterios ponderados.

$Al1 = \text{Ponderado por criterio costo} \times \text{Ponderado por alternativa Al 1}$

Fuente: Riba, C; Diseño concurrente, 2002

Ejemplo, calculo Alternativa 1 con respecto al costo:

Reemplazando queda:

$$A1 = 0.16 \times 0.33 = 0.055$$

Tabla 3.5 Conclusiones de la selección de cámara para el sistema

	COSTO	RESOLUCIÓN	ANGULO DE VISION	TIPO DE SENSOR	ESTÉTICA	Σ	PRIORIDAD
AL1	0.055	0.088	0.077	0.15	0.06	0.44	2
AL2	0.11	0.17	0.15	0.07	0.033	0.55	1

Como resultado se obtuvo que la alternativa 2 sobresale a la alternativa 1 ya que presenta una mejor resolución y disminuye los costos de producción, por lo que se realiza el sistema de visión artificial con la cámara WideCam F100, debido a que cumple con los criterios requeridos.

3.3. MÓDULO PARA LUMINARIA UBICADA EN DIRECCIÓN FRONTAL

Como se mencionó en capítulos anteriores, una buena iluminación es un requisito indispensable cuando se trata de visión artificial.

El módulo que se presenta a continuación ayuda a encontrar fácilmente las piezas de clasificación, al mantener un rango estable de la cantidad de luz que influye en el objeto, agregando a esto la correcta distribución de la luminaria para reducir sombras y factores que puedan afectar la repetitividad de los resultados. En próximas referencias se la nombra como luminaria inferior.

3.3.1. Selección de tipo de iluminación para el módulo

La selección de la luminaria tiene una gran influencia en el resultado final de la imagen, por esa razón tomamos se debe seleccionar correctamente la clase de luz dependiendo de la función que va a realizar. Los requisitos son:

- Iluminación distribuida en la imagen.
- Posibilidad de ser Dimerizable.
- Cantidad de luxes necesarios para realizar la visualización de fallas por el operador y la cámara.
- Distancia entre lámpara y objetivo 120 [cm]

Se analiza los aspectos básicos que necesita el sistema para la eliminación de opciones.

La ubicación de la luminaria se la coloca en posición frontal con respecto a la pieza de cuero con la finalidad de resaltar las piezas que se sitúan sobre las imperfecciones.

La opción de controlar la intensidad de luz ayuda a minimizar el consumo eléctrico del sistema, además de mantener repetitividad en los datos. El uso de lámparas que producen luz por medio de calentamiento de filamentos facilita el uso de dimmer, muy por el contrario, las lámparas que usan gases.

Es prudente señalar que las lámparas dicróicas minimizan la fuga de luz al focalizar el haz de luz emitido en un espacio reducido, lo cual no ocurre con los tubos o bombillas de uso común.

Según (Ministerio del Trabajo, 2016), *establece la cantidad mínima de 500 luxes para el área de trabajo en los que sea indispensable una fina distinción de detalles, bajo condiciones de contraste, tales como: corrección de pruebas, fresado y torneado, dibujo.*

Con las consideraciones anteriormente planteadas se opta utilizar lámparas dicróicas halógenas.

Sylvania plantea un estudio de fotometría de acuerdo a la altura que exista entre el objetivo y la lámpara con respecto a la potencia de la lámpara.

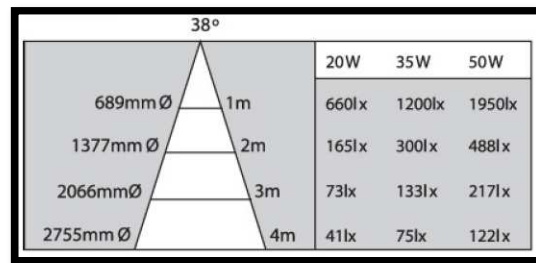


Figura 3.3 Fotometría MRI16 Coolfit Home (Havells Sylvania, 2016)

De acuerdo a la fotometría se tendría un aproximado a 660 [lx] a una distancia de 120 [cm] con una lámpara de 20 [W]. Cumpliendo con la cantidad de luxes sugerida por el ministerio de trabajo.

Se emplea lámparas dicróicas halógenas dicróicas 20 [W] y 12 [V~].

3.3.2. Distribución de la luminaria

Una vez establecido el tipo de luminaria se procede a la distribución para lograr cubrir un área que tiene la mesa de trabajo a una altura de 120 [cm], tomando en cuenta que en el punto central se encuentra la cámara.

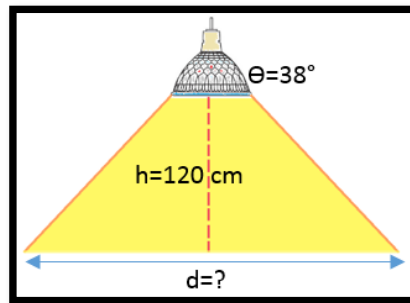


Figura 3.4 Esquema de parámetros de la luminaria

Para conseguir el diámetro que va a cubrir el haz de luz de las lámparas dicroicas tomamos el ángulo del Anexo 6.1 y reemplazamos en la ecuación.

Ecuación 3.2 Calculo diámetro del haz de luz

$$d = 2h \tan \frac{\theta}{2}; d = 2(120) \tan \frac{38}{2}; d = 82.63 [cm]$$

Donde:

d: diámetro de la circunferencia que forma el haz de luz [cm]; h: altura a la que se encuentra [cm]; Θ : ángulo del haz de luz [grados]

Como resultado se obtiene que colocando las lámparas a la altura de 120 [cm] misma a la que se coloca la cámara esta tiene un diámetro de aproximadamente 80 [cm] para el haz de luz.

Para ello se coloca círculos de 80 [cm] de diámetro sobre una base rectangular de dimensiones 254 x 163 [cm]. Se los distribuye de tal forma que logre cubrir el área interna del rectángulo, tomando en cuenta que no existe un círculo con centro en el punto medio. Dando como resultado la figura 3.4, que usa 14 lámparas dicroicas distribuidas de forma simétrica tanto en el eje X como en el eje Y, esta distribución se usa como base para el diseño de la estructura.

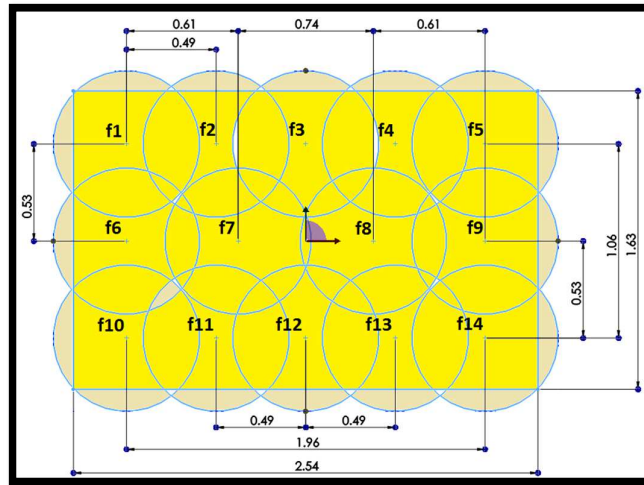


Figura 3.5 Distribución lámparas dicróicas halógenas

Debido a que se trata de haz de luz dirigidos, las zonas en las que se intersecan los círculos poseen la misma cantidad de luxes que en las zonas internas de los círculos donde no hay intersección.

3.3.3. Etapa de potencia

La variación en cuanto a intensidad de luminosidad se efectúa controlando la potencia entregada a la carga. Para ello se analiza el amperaje que soportar el circuito a una intensidad de brillo del 100%.

Ecuación 3.3 Calculo de corriente para las 14 lámparas

$$14 * P_{ts} = V_{ts} * I_{ts}; 14 * 20 = 12 * I_{ts}; I_{ts} = 23.24 [A]$$

Donde:

P_{ts} : Potencia total en el lado secundario del transformador; V_{ts} : Tensión total en el lado secundario del transformador; I_{ts} : Impedancia total en el lado secundario del transformador

Ecuación 3.4 Calculo de impedancia del transformador

$$\frac{V_{tp}}{V_{ts}} = \frac{I_{ts}}{I_{tp}}; \frac{110}{12} = \frac{23.24}{I_{tp}}; I_{tp} = 2.53 [A]$$

Fuente: Boylestad; Introducción al análisis de circuitos, 2011

Donde:

V_{tp} : Tensión total en el lado primario del transformador; V_{ts} = Tensión total en el lado secundario del transformador; I_{tp} = Impedancia total en el lado primario del transformador; I_{ts} = Impedancia total en el lado secundario del transformador

Como resultado se tiene el amperaje máximo que debe soportar el triac, para este proyecto se ha seleccionado el BTA12-600C debido a que soporta un amperaje máximo de 12 [A] como lo indica la tabla 3.10, con esto se verifica que amperaje calculado se encuentra en dicho rango.

Tabla 3.6 Parámetros de funcionamiento BTA12-600C

Símbolo	Parámetro	Valor	Unidad
$I_{T(RMS)}$	Corriente promedio on-state	12	A
$V_{DRM}/$ V_{RRM}	Tensión de pico repetitiva en adelanto e inversa off-state	600 and 800	V
$I_{GT(Q1)}$	Corriente de activación de la puerta	2.5min -50max	mA
V_{Gt}	Tensión de la puerta activada	1.3 max	V
V_{GD}	Tensión de la puerta desactivada	0.2 min	V

Fuente: STMicroelectronics; Datasheet BTA12-600C

La posibilidad de dimerizar la cantidad de luz emitida por las lámparas se logra colocando resistencias en serie, conectadas cada una en paralelo a un optoacoplador; el efecto que produce este tipo de conexión es: al activar el optoacoplador la corriente circula internamente por este hasta los pines de salida, al desactivarlo circularía por la resistencia.

El valor de cada resistencia tiene un valor aproximado a la mitad del valor anterior, simulando el peso que posee un dato de 8 bits; de esta manera se obtiene un total de 256 valores distintos en la resistencia final, asemejándose a un potenciómetro.

Para el cálculo del valor mínimo de la resistencia que debe ir conectada entre la puerta y el ánodo 1 del triac se lo hace tomando el valor de la corriente máxima de la puerta ($I_{gt} = 50[mA]$), el valor de la tensión máxima de la puerta ($V_{gt} = 1.3[V]$), y el valor del voltaje al que está sometido el circuito ($V_{ac} \approx 110[V]$).

$$R_g \approx \frac{V_{ac}-V_{gt}}{I_{gt}} ; R_g \approx 2.2K\Omega$$

Resultando un valor mínimo para la resistencia de $2,4\text{ K}\Omega$, con este valor se asegura el disparo del triac. Del mismo modo se calcula el valor máximo de la resistencia que del circuito, con el valor de la corriente mínima de la puerta ($I_{gt} = 2.5[mA]$), y el voltaje de la puerta desactivada $V_{GD} = 0.2[V]$.

$$R_g \approx \frac{V_{ac}-V_{GD}}{I_{gt}} ; R_g \approx 43.9\text{ K}\Omega$$

Con los límites máximo y mínimo establecido se puede fijar el valor que le corresponde a cada resistencia, como se mencionó anteriormente simula un dato de 8 bits; se adiciona una resistencia mayor al límite mínimo calculado para asegurar la activación de disparo en el triac.

Para este caso usaremos transformadores electrónicos de $110[V]$ a $12[V]$ AC dimerizables para las lámparas halógenas, con un $\cos\phi = 0.99$ lo que la hace una carga fuertemente resistiva.

Se une con el circuito de potencia mediante el optoacoplador MOC-3010, debido a su versatilidad de trabajo con corriente alterna. Este chip viene acompañado de diagramas dependiendo del tipo de carga, dando como resultado el diagrama de la figura 3.6.

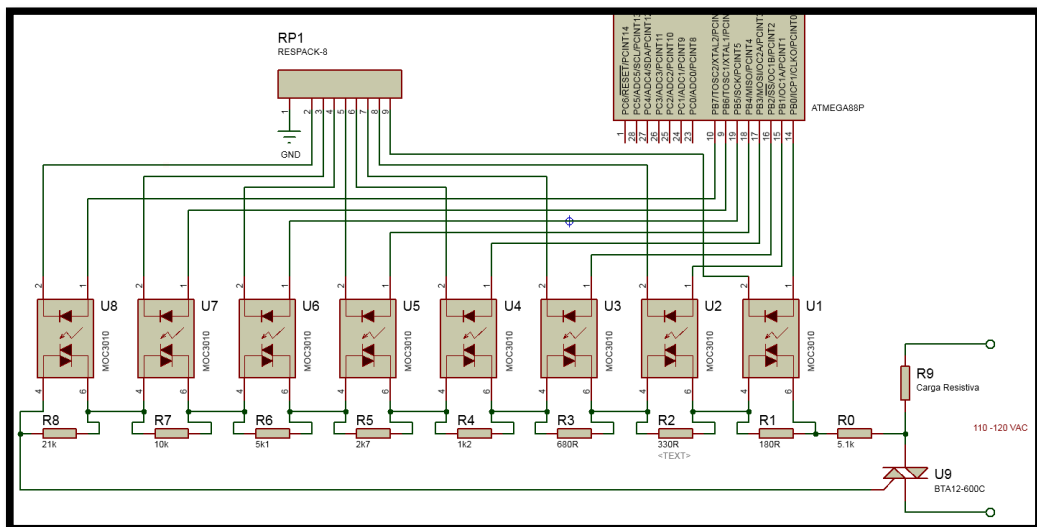


Figura 3.6 Diagrama etapa de potencia lámparas halógenas

Esta configuración hace que varíe el ángulo de disparo del triac dependiendo de la cantidad de corriente que pase por el arreglo de resistencias hacia la puerta.

El transformador electrónico dimerizable que se usa es marca Sylvania, posee las siguientes características técnicas:

- PRIMARIO: 110-120~; 60Hz; 0.55A; $\cos\varphi = 0.99$
- SECUNDARIO: 11.5V~rms; Max.5A; 50-60W; 12V

3.4. MÓDULO PARA LUMINARIA UBICADA EN DIRECCIÓN TRASERA

Según se ha citado en el capítulo anterior, se puede cambiar la posición de luminaria con el fin de resaltar rasgos específicos, en este caso lo que se requiere resaltar en la imagen es la forma que tiene el cuero y no las características de textura como se lo hizo anteriormente, es por esta razón que se posiciona en dirección trasera a la pieza de cuero. En próximas referencias se la nombrara como luminaria superior.

3.4.1. Selección de tipo de iluminación para el módulo

Se analiza las necesidades que intervienen en este módulo para la discriminación de posibilidades.

- La mayor parte de la superficie de la mesa esta cubierta por la pieza de cuero, dejando el área cercana a los bordes descubierta.
- No tiene que haber ningún objeto que pueda mostrar bordes, ni una sobre iluminación puntual en el área interna donde va a estar la pieza; ya que esto interfiere con los resultados finales.
- La luz que emita debe ser luz blanca y potente, con esto se consigue un contraste, haciendo siempre oscuro al cuero y claro a la mesa.

El tipo de luminaria se lo examinara por la forma del haz de luz que emite, una lámpara de bombilla dejaría un espectro de forma circular, muy por lo contrario de las lámparas tubulares fluorescentes como se muestra en la figura 3.7.

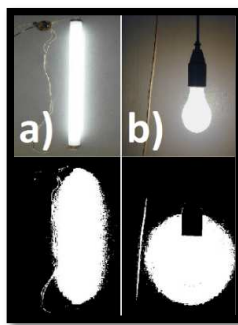


Figura 3.7 Formas del haz de luz.

- a) Lámpara tubular
- b) Lámpara de bombilla

Considerando lo anterior dicho se coloca lámparas fluorescentes en el perímetro de la mesa. Para distribuir las se sabe que la mesa es de forma rectangular con medidas de 250x160 [cm], se debe usar la menor cantidad de lámparas y debe de mantener un tamaño invariable para obtener un brillo constante en los laterales.

Tomamos el valor del lado mayor y lo dividimos a la mitad quedando un tamaño máximo de 125 [cm]. Se tienen las medidas de lámparas fluorescentes cercanas al tamaño calculado en el catálogo de Sylvania que se encuentra en el anexo 6.2:

Tabla 3.7 Características físicas de lámparas fluorescentes

TAMAÑO	POTENCIA [W]	LONGITUD [MM]
T5	28W	1163.2
T8	38W	1061.2
T12	40W	1213.6
T5	54W	1163.2

FUENTE: Sylvania; Catalogo Lámparas 2011-2012, 2011

Buscamos el valor inferior más cercano y su potencia, obteniendo como resultado la lámpara Fluorescente F40WT12 con un valor de longitud de 121 [cm]. El diagrama de distribución es la figura 3.7, se usa para el diseño de la estructura.

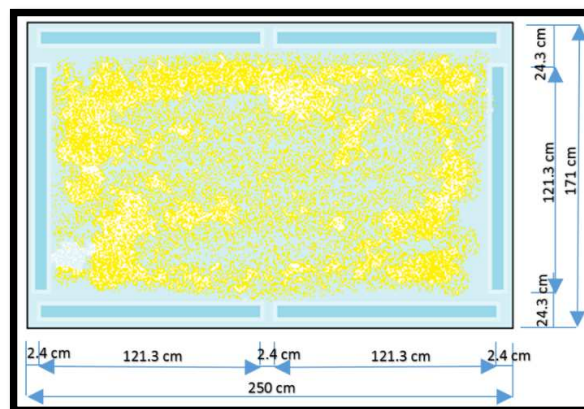


Figura 3.8 Distribución lámparas fluorescentes

3.4.2. Etapa de potencia

Con referencia a lo anterior se toma el valor de la potencia y la cantidad de lámparas a usarse, para poder escoger el triac necesario para este circuito, este valor se reemplaza en la ecuación 3.5:

Ecuación 3.5 Calculo de corriente para las 6 lámparas

$$6 * P = V * I; 6 * 40[W] = 110[V] * I; I = 2.18 [A]$$

Donde:

P: Potencia [W]; V: Tensión [V]; I: Impedancia [A]

Con esto que comprobado que el amperaje es mucho menor de los 10 [A] que tolera el triac BTA10-600B. La unión del circuito de potencia con el digital se realiza por medio del opto acoplador MOC-3010, el circuito a usarse es el indicado en su datasheet para cargas inductivas.

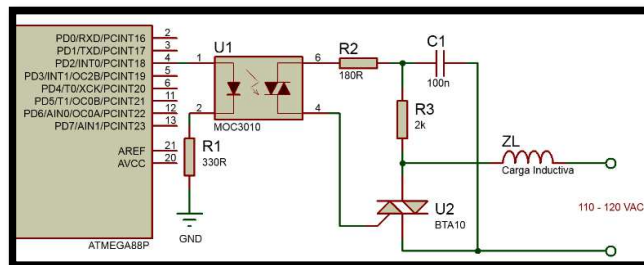


Figura 3.9 Diagrama etapa de potencia lámparas fluorescentes

El balastro que a usarse es marca Electrocontrol, posee las siguientes características técnicas:

- 120V~; 0.7A; 60Hz; Distorsión de armónicos <15%
- Factor de cresta < 1.7; Factor de potencia >96%

La conexión de las lámparas fluorescentes con los balastos y el circuito de control es la indicada en la figura 3.10:

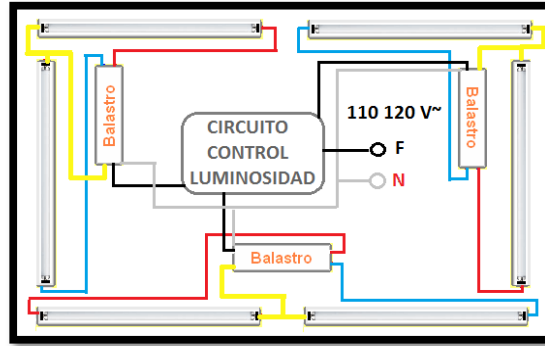


Figura 3.10 Diagrama de conexión eléctrica para lámparas fluorescentes

3.5. MÓDULO DE RECEPCIÓN DE DATOS.

Una vez procesada la necesidad del tipo de luminaria a encender o controlar se envía un valor mediante cable USB-Serial hacia el μC . Para lograrlo primero se convierte la señal del puerto serial RS-232 a niveles TTL con un C.I. Max232, para que el micro-controlador pueda interpretar los datos.

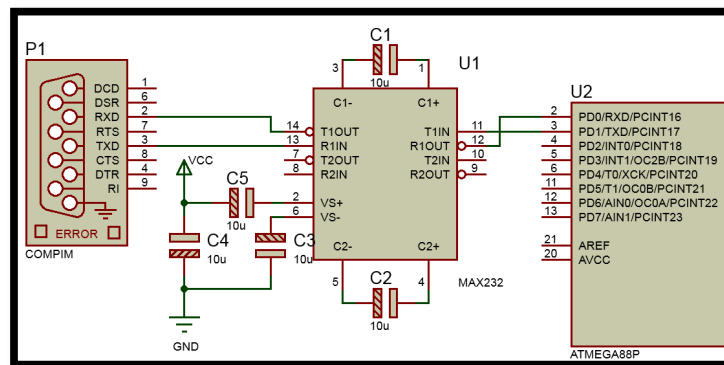


Figura 3.11 Conexión eléctrica Conversión de RS232 a TTL

Esta conversión se la realiza debido a que el μC recibe solo valores TTL que son (5v el valor lógico 1 y 0v el valor 0) a diferencia de RS-232 (-10v el valor lógico 1 y 10v el valor 0). El valor del pixel es enviado al μC en código ASCII.

3.5.1. Procesamiento del dato.

Los datos a enviarse son para el encendido o apagado tanto de la luminaria superior como de la inferior y la graduación de intensidad de luminosidad en la luminaria superior.

Tabla 3.8 Dato a enviar según la acción de la luminaria

Acción	Dato
Subir brillo luminaria superior	1
Bajar brillo luminaria superior	2
Encender luminaria superior	3
Apagar luminaria superior	4
Encender luminaria inferior	5
Apagar luminaria inferior	6

Para conseguir variar la intensidad del brillo del circuito de luminaria superior se controla la cantidad de pines activados en el puerto B del μC , cada pin controla el accionamiento de su respectivo optoacoplador; para ello se define límites: límite de encendido, valor de activación de la luminaria, valor de desactivación de la luminaria, límite de apagado.

Esto se realiza en base al comportamiento de la luminaria mediante pruebas. Para ello se simuló la activación de los 8 pines de salida del puerto B del μC , desde todos los pines desactivados hasta todos activados y viceversa, simulando un conteo en binario. En la tabla 3.9 se declara el valor en binario 1 representado 5v y 0 a 0v.

Tabla 3.9 Comportamiento de la luminaria

Prueba N°	Valor en binario según el pin del puerto B								Valor en decimal	Comportamiento
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0		
1	1	1	1	1	1	1	1	1	255	Encendido total
2	1	0	1	0	0	1	0	0	164	Encendido parcial
3	1	0	0	1	0	0	0	0	144	Apagado parcial
4	1	0	1	0	0	0	0	1	161	Encendido parcial
5	1	0	0	1	0	0	0	0	144	Apagado parcial
6	1	0	0	1	1	1	0	1	157	Encendido parcial
7	1	0	0	0	1	1	0	1	141	Apagado parcial
8	1	0	0	1	1	1	1	1	159	Encendido parcial
9	0	1	0	1	1	1	1	0	110	Apagado parcial
10	0	0	0	0	0	0	0	0	0	Apagado total

Con estos resultados se establece los siguientes valores de referencia: el límite de encendido en 255, valor de activación de la luminaria 160, valor de desactivación de la luminaria 110, límite de apagado 0.

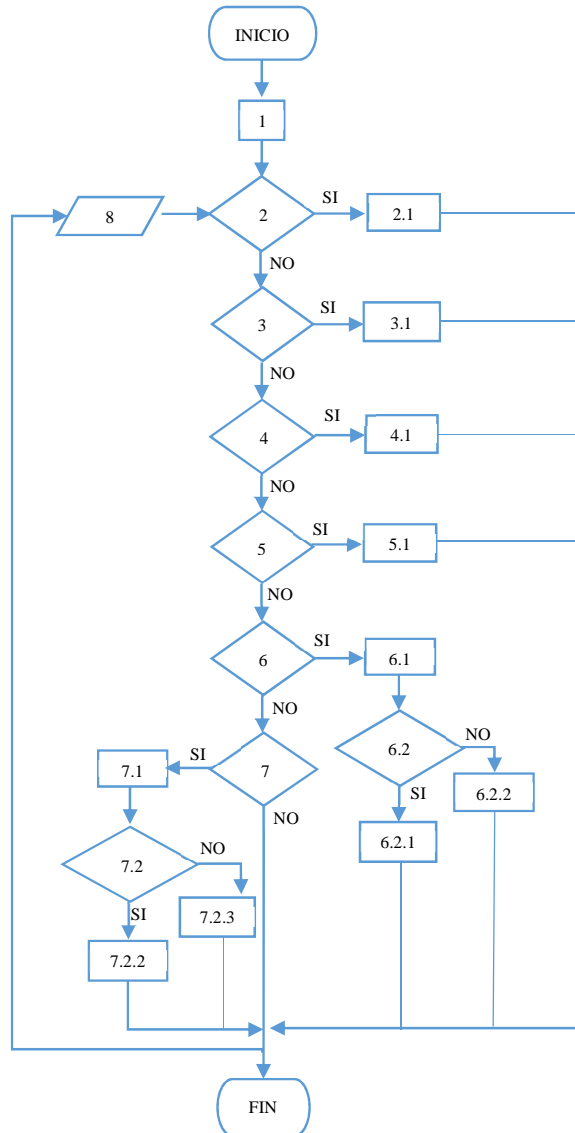
Tabla 3.10 Acción en el μC

Código	Acción
1	Inicialización de variables
2	Comparación (dato serial) =enlusu
2.1	Activa puerto correspondiente a la luminaria superior
3	Comparación (dato serial) =apalusu
3.1	Desactiva puerto correspondiente a la luminaria superior
4	Comparación (dato serial) =enluinf
4.1	Activa pin correspondiente a luminaria inferior
5	Comparación (dato serial) =apalusu
5.1	Desactiva pin correspondiente a luminaria inferior
6	Comparación (dato serial)=subrilusu
6.1	Aumenta en 1 la variable val
6.2	Comparación val=limen
6.2.1	<ul style="list-style-type: none"> • Carga valor de la variable limen en variable val • Carga valor de variable val en el puerto correspondiente a la luminaria superior
6.2.2	Carga valor de variable val en el puerto correspondiente a la luminaria superior
7	Comparación (dato serial)=babrilusu
7.1	Disminuye en 1 la variable val
7.2	Comparación val=limapa
7.2.1	<ul style="list-style-type: none"> • Carga con el valor 160 en variable val • Desactiva puerto correspondiente a la luminaria superior
7.2.2	Carga valor de variable val en el puerto correspondiente a la luminaria superior
8	Adquisición (dato serial)

Tabla 3.11 Lista de variables del programa en el μC

Variable	Descripción	Valor inicial de la Variable
ilum	Dato de la iluminación que envía desde Matlab.	-
val	Valor de activación de la luminaria, 8 bits	200
subrilusu	Dato que indica subir el brillo de luminaria superior.	1
babrilusu	Dato que indica bajar el brillo de luminaria superior.	2
enlusu	Dato de encendido de luminaria superior.	3
apalusu	Dato de apagado de luminaria superior.	4
enluinf	Dato de encendido de luminaria inferior.	5
apaluinf	Dato de apagado de luminaria inferior.	6
limen	Valor máximo al que la iluminación está al 100%	255
limapa	Valor mínimo al que la iluminación está al 0%	110

Gráfico 3.1 Diagrama de flujo del programa en el μC



Para controlar el brillo se inicia desde el valor de activación de luminaria, cada vez que el μC reciba el dato 2 resta 1 del valor anterior, hasta llegar al valor de desactivación de luminaria e inmediatamente se carga el valor 0 en el puerto B; si recibe el dato 1 suma 1 al valor anterior, en el caso de llegar al límite superior se toma el valor del límite.

Este valor final se transforma a binario e indica los pines que se activan o desactivan del puerto B, por ejemplo, si el valor final fuese 170, el valor en binario sería

10101010, desactivando los pines PB0, PB2, PB4, PB6 y activando los pines PB1, PB3, PB5, PB7.

El encendido total de luminaria superior se acciona cuando el μC reciba el dato 3, indicando que se active todos los puertos y por lo tanto la corriente circula a través del opto acoplador, disparando el triac. El apagado total con el dato 4 induciendo la desactivación de todos los pines, conduciendo la corriente a través de las resistencias lo que hace que se desactive el triac.

La luminaria inferior solo tiene la función de encendido con el dato 5 y apagado con el dato 6 y es comandada únicamente por el puerto PD2. Si este es activado cierra circuito de la etapa de potencia accionando el opto acoplador encendiendo la luminaria, y si se desactiva se abre el circuito de potencia provocando el apagado de la luminaria.

3.6. TABLERO DE DISTRIBUCIÓN.

Siendo uno de los principales componentes debido a que da protección cabida a la mayoría de los elementos eléctricos que conforman este sistema.

La selección de cada uno de estos componentes se lo realiza tomando en cuenta la potencia requerida por cada una de las cargas ala que van a estar conectados, para ello se las separa dependiendo del tipo de corriente de funcionamiento.

Cargas con corriente alterna AC:

- Intensidad luminaria superior: 2.18 [A]
- Intensidad luminaria inferior: 2.53 [A]
- Intensidad periféricos (computador, impresora): 5 [A]
- Intensidad fuente de voltaje 110 [V ac] a 5[V dc]: 0.15 [A]

Cargas de corriente continua o directa DC:

- Modulo control de luminaria: 0.5 [A]

Tomando en cuenta los valores referenciales se selecciona los componentes para protección eléctrica para el usuario, como el interruptor diferencial e interruptor termoelectrico. Estos soportan una impedancia mínima de 11 [A].

3.6.1. Elementos panel de control.

En el panel de control se requiere selectores tanto para encender y apagar la luminaria inferior, del mismo modo para regular la intensidad de luminosidad en la luminaria superior. Conjuntamente un selector para que el usuario pueda controlar si desea encender la luminaria de una forma manual o automática. Los elementos se describen a continuación:

Tabla 3.12 Componentes del panel de control

Elemento	Función
Luz piloto verde	Indica que el tablero esta energizado
Luz piloto verde 2	Indica que el tablero y la luminaria esta energizada
Luz piloto blanco 1	Indica que se ha seleccionado la activación de luminaria automática
Luz piloto blanco 2	Indica que se ha seleccionado la activación de luminaria manual
Luz piloto blanco 3	Indica luminaria inferior encendida
Luz piloto blanco 4	Indica luminaria superior encendida
Conector hembra DB9	Permite conexión para comunicación de la tarjeta con el computador
Selector 3 posiciones	Permite seleccionar el tipo de encendido para la luminaria
Selector 2 posiciones	Permite encender o apagar la luminaria inferior
Dimmer	Permite encender, regular o apagar la luminaria superior
Toma corriente de 2 entradas	Permite la conexión de 2 periféricos
Salida cable USB	Cable USB perteneciente a la cámara



Figura 3.12 Panel de control

3.7. DISEÑO ESTRUCTURAL.

Una necesidad básica de todo sistema de visión artificial es el cuerpo o estructura que soporta y contiene los elementos necesarios para que este cumpla su función. En este caso la medición de área e identificación de patrones.

Como se dijo previamente los elementos mínimos que se tiene es la cámara y la iluminación, por esta razón se divide la estructura en dos segmentos:

- Estructura para la luminaria inferior
- Estructura para la luminaria superior y cámara.

Esta estructura además de contener la luminaria, debe dar soporte a los diferentes elementos que la complementan. Los parámetros que se consideran para la estructura de la luminaria inferior son:

- Delimitación del perímetro para colocación del cuero. 254 x 163 [cm]
- Permitir el paso del rayo de luz emitido por la lámpara fluorescente.
- Distancias entre lámparas en referencia con la Figura 3.8.

Así mismo la estructura para la luminaria superior debe mantener ciertos parámetros que se establecieron anteriormente como:

- Distancia entre la luminaria superior y la superficie de sobre la que se coloca en cuero de 120[cm].
- Distancias entre lámparas en referencia con la Figura 3.5.
- Contiene la cámara en el centro de la estructura.
- Disipar cualquier tipo de vibraciones producidas al colocar o extraer las piezas de cuero.

3.7.1. Elementos que complementan a la estructura inferior.

Se considera el uso de tres elementos que se adhieren a la estructura:

- Cubierta: Superficie sobre la que se coloca el cuero.
- Cajón luminoso: Contiene la luminaria.
- Tablero de distribución: Contiene los elementos eléctricos.

La cubierta debe poseer ciertas características extras por el trabajo que se realiza en la fábrica, como, por ejemplo:

- Protección de agentes químicos como disolventes que se usan para el tinturado.
- Resistencia a la abrasión, esto por la forma de ingreso que va a tener el cuero a este sistema, y debe servir de apoyo para otros oficios como cortes de bordes, etc.
- Rigidez, no presentar curvatura al ejercer presión en la superficie.
- Seguridad para el usuario.

Por lo antes descrito se analiza diferentes tipos de materiales:

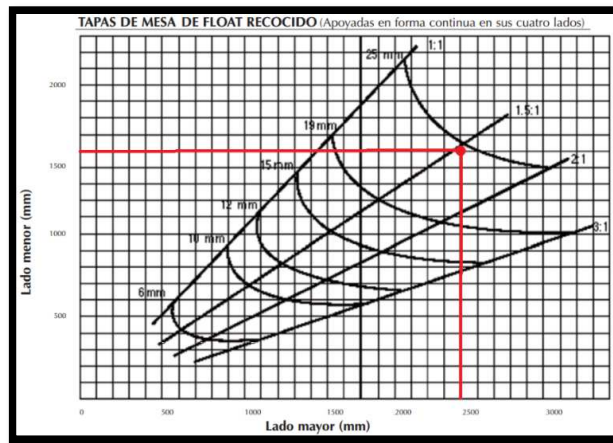
Tabla 3.13 Características de materiales para la cubierta.

Material	Protección contra agentes químicos	Dureza unidades (escala de Mohs)	Módulo de elasticidad [kMpa]	Seguridad
Policarbonato	No	5.5	2.3	Si
Vidrio templado	Si	6	70	Si

Fuente: Elaboración propia en base a Pearson, CAVIPLAN; Manual del vidrio plano E industrias JQ; Datos técnicos policarbonato, 2016

Como se puede apreciar el mejor material para que ocupar en la cubierta es el vidrio templado, esto a la presencia de rigidez y es resistente a la abrasión y ralladuras. Su alteración estructural debido al templado hace que se desintegre en gránulos redondeados cuando se rompa, eliminando el riesgo de cortes. La selección de espesor del vidrio se toma basado en el gráfico 3.2 tomado del anexo 9.

Gráfico 3.2 Malla referencial para selección de espesor del vidrio



Fuente: VASA; Vidrios para tapas de mesa y estantes, 2016

Para analizar este gráfico se toma los valores de la distancia que hay entre los apoyos para el vidrio, 2540 [mm] para el lado mayor y 1630 [mm] para el lado menor. Uniendo estos dos puntos nos da como resultado un vidrio de aproximadamente 25 [mm] de espesor. Pero, es importante enfatizar que “El Float Templado es un vidrio 4 a 5 veces más resistente que el Float recocido” (VASA, 2016). Por esta razón se divide el espesor encontrado para 4 obteniendo un espesor de 6.25 [mm], por seguridad se le

incrementa un factor de seguridad de 1.5 obteniendo un vidrio de espesor aproximado a 10 [mm].

El cajón Luminoso es de MDF, con revestimiento de color blanco y con un espesor de 15 [mm], redirigiendo el haz de luz hacia la cara que no se encuentra cerrada, maximizando el brillo y evitando la fuga de luz. Con tamaño 253 [cm] de largo, 174 [cm] de ancho y 9.5 [cm] de alto. Resultando la figura 3.13.

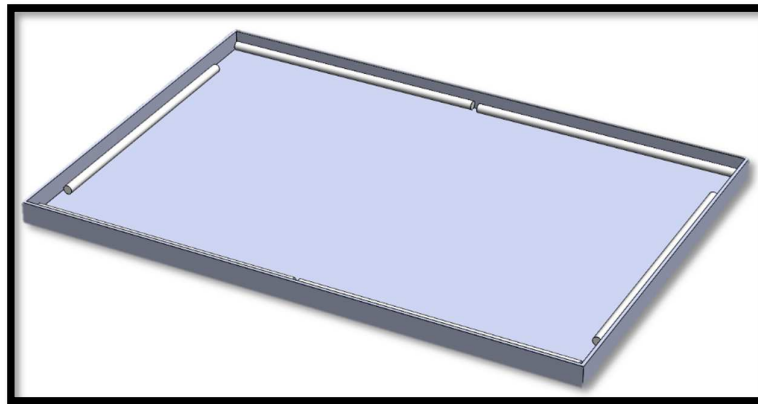


Figura 3.13 Cajón Luminoso

3.7.2. Calculo de la estructura inferior.

3.7.2.1. *Parámetros de diseño*

Esta estructura debe soportar algunas cargas muertas, las cuales se detallan a continuación:

Tabla 3.14 Descripción de cargas en la estructura

ELEMENTO	CANTIDAD [UNIDAD]	PESO [KG]	CARGA [NEWTON]
VIDRIO TEMPLADO	1	170	1667.7
CAJÓN LUMINOSO	1	45	441.45
TABLERO DE DISTRIBUCIÓN	1	3	29.43
HOJA DE CUERO	1	1	9.81
PERSONA	2	160	1569.6
Carga total		379	3717.99

3.7.2.2. Selección del factor de diseño

Este factor añade confianza y seguridad en el diseño de acuerdo a las condiciones a las que va a ser sometida.

Tabla 3.15 Factores de diseño para materiales dúctiles

Factor de diseño para materiales Dúctiles	Característica
1.25 a 2.0	El diseño de estructuras bajo cargas estáticas, para las que haya un alto grado de confianza en todos los datos del diseño
2.0 a 2.5	Diseño de elementos de maquina bajo cargas dinámicas con una confianza en todos los datos de diseño
2.5 a 4.0	Diseño de estructuras estáticas o elementos de maquina bajo cargas dinámicas con incertidumbre acerca de las cargas, propiedades de los materiales, análisis de esfuerzos o el ambiente
4.0 o más	Diseño de estructuras estáticas o elementos de máquinas bajo cargas dinámicas, con incertidumbre en cuanto a alguna combinación de cargas, propiedades del material, análisis de esfuerzos o el ambiente.

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Tomando en cuenta la tabla se selecciona $N=2$, ya que va a estar sometida a cargas dinámicas.

3.7.2.3. Diseño preliminar de la estructura inferior

Se traza un bosquejo previo, simulando la estructura y los puntos donde se coloca las cargas, según la siguiente figura.

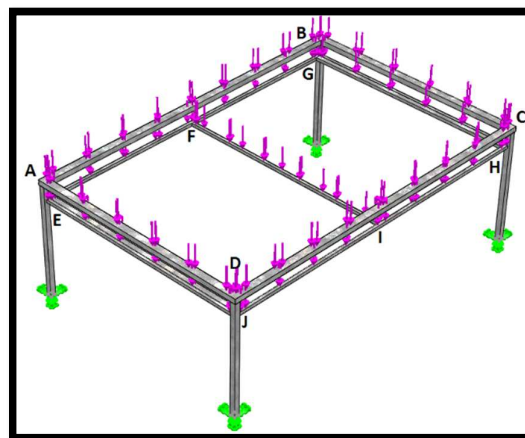


Figura 3.14 Modelado de estructura inferior con sus cargas.

Cargas en vigas AB, BC, CD, AD:

- VT= Carga distribuida del vidrio templado = 1667.7 [N]
- P= Carga distribuida de dos personas = 1569.6 [N]

Cargas en vigas EG, GH, HJ, JE, FI:

- CL= Carga distribuida del cajón luminoso = 441.45 [N]

No se toma en cuenta la carga producida por el tablero de distribución y la hoja de cuero, ya que son cargas pequeñas con relación al resto.

La adición de las dos personas es en caso del ensamblaje y mantenimiento de la estructura para la luminaria superior, debido a su tamaño.

3.7.2.4. Análisis de cargas en las vigas usando MDSolids

- **Viga AB y CD**

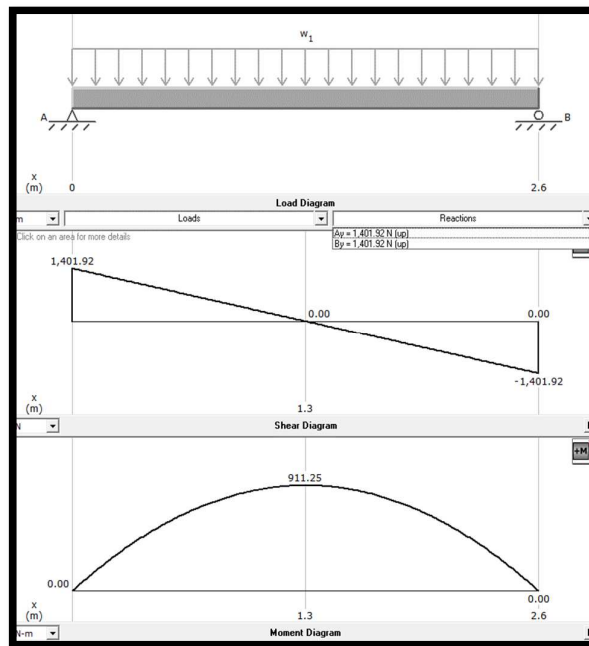


Figura 3.15 Diagrama de cortes y momentos de la Viga AB y CD

Reacciones y Momento

$$R_A=R_B=R_C=R_D= 1401.92 \text{ [N]}; MM^*= 911.25 \text{ [Nm]}$$

Donde:

MM=Momento máximo sobre la viga.

- **Viga BC y AD**

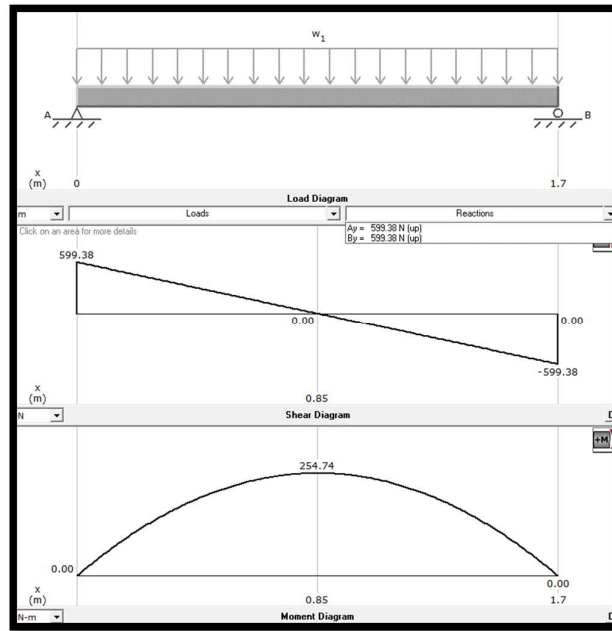


Figura 3.16 Diagrama de cortes y momentos de la Viga BC y AD

Reacciones y Momento

$R_B=R_C=R_A=R_D= 599.38$ [N]; $MM= 254.74$ [Nm]

- **Viga EG y HJ**

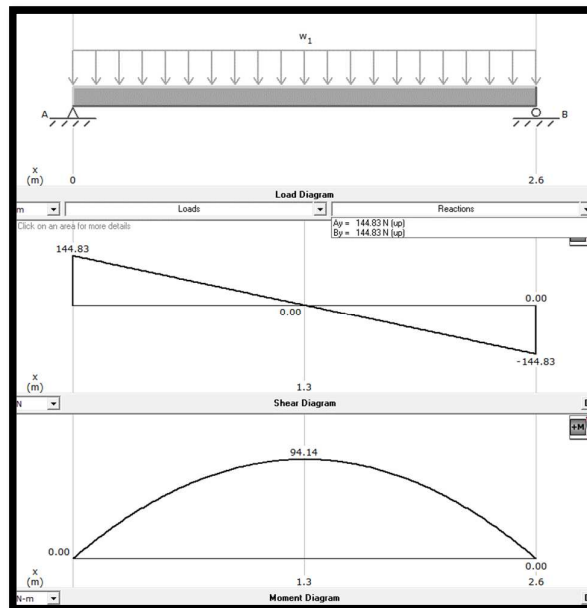


Figura 3.17 Diagrama de cortes y momentos de la Viga EG y HJ

Reacciones y Momento

$R_E=R_G=R_H=R_J= 144.83$ [N]; $MM= 94.14$ [Nm]

- **Viga GH, FI y EJ**

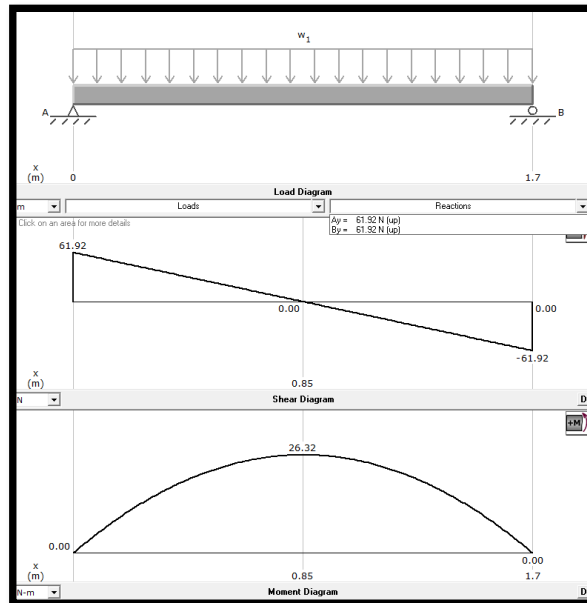


Figura 3.18 Diagrama de cortes y momentos de la Viga EG y HJ

Reacciones y Momento

RG=RH=RF=RI=RE=RJ= 61.92 [N]; MM= 26.32 [Nm]

3.7.2.5. *Calculo de la estructura para las vigas AB, BC, CD y DA*

Se toma valores de las reacciones y momentos máximos más altas en las vigas analizadas para encontrar el perfil adecuado.

- **Momento máximo.**
MM= 911.25 [Nm] = 8.06 [klbin]
- **Reacción máxima.**
R= 1401.92 [N]
- **Características mecánicas del material.**
Acero ASTM 500
Sy=2400 [Kgf/cm2] = 34.136 [Kpsi]
E=2100000 [Kgf/cm2] =29869.04 [Kpsi]

- **Esfuerzo de diseño**

Ecuación 3.6 Esfuerzo de diseño

$$\sigma_d = \frac{S_y}{N} = \frac{34.136 [kpsi]}{2} = 17.068 [kpsi]$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

σ_d : Esfuerzo de diseño; S_y : Resistencia de fluencia; N : Factor de diseño

- **Módulo de sección requerido**

Ecuación 3.7 Modulo de sección requerido

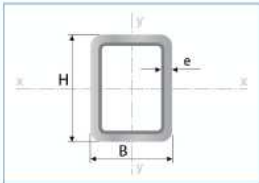
$$S = \frac{M}{\sigma_d} = \frac{8.06 \text{ [klbin]}}{17.068 \text{ [kpsi]}} = 0.47 \text{ [in}^3\text{]} = 7.7 \text{ [cm}^3\text{]}$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

S ; Modulo de sección requerido; M : Momento flexionante

Se selecciona de la siguiente tabla el perfil que tenga como sección un valor idéntico o superior a $S=7.7 \text{ [cm}^3\text{]}$. Un tubo rectangular permite que el vidrio tenga una mayor superficie de apoyo.



Designaciones	Área	Peso	Propiedades Estáticas								
			Eje x-x				Eje y-y				
			B	H	e	A	P	I	W	i	I
	mm	mm	mm	cm ²	kg/m	cm ⁴	cm ³	cm	cm ⁴	cm ³	cm
20	40	1,50	1,65	1,30	3,26	1,63	1,40	1,09	1,09	0,81	
		2,00	2,14	1,68	4,04	2,02	1,37	1,33	1,33	0,79	
25	50	1,50	2,10	1,65	7,65	3,02	0,82	2,50	2,02	1,05	
		2,00	2,74	2,15	8,37	3,35	1,75	2,80	2,24	1,01	
		3,00	3,91	3,07	12,90	5,08	1,75	4,12	3,26	0,99	
30	50	1,50	2,25	1,77	7,53	3,01	1,83	3,41	2,27	1,23	
		2,00	2,94	2,31	9,52	3,81	1,80	4,28	2,85	1,21	
		3,00	4,21	3,30	12,78	5,11	1,74	5,66	3,77	1,16	
40	60	1,50	2,85	2,24	14,40	4,79	2,26	7,71	3,85	1,65	
		2,00	3,74	2,93	18,39	6,13	2,22	9,81	4,90	1,62	
		3,00	5,41	4,25	25,31	8,44	2,16	13,37	6,69	1,57	
30	70	1,50	2,85	2,24	18,08	5,17	2,49	4,76	3,17	1,28	
		2,00	3,74	2,93	22,20	6,34	2,44	5,85	3,90	1,25	
		3,00	5,41	4,25	30,50	8,71	2,37	7,84	5,23	1,20	
40	80	2,00	4,54	3,56	27,32	8,22	2,87	12,70	6,25	1,67	

Largo Normal:
6 metros
Recubrimiento:
Negro o Galvanizado
Norma de Calidad:
ASTM A 500 Gr. A, B ó C
Norma de Fabricación:
NTE INEN 2415
Espesores:
Desde 1,50 a 6,00 mm

Figura 3.19 Propiedades tubería estructural rectangular (IPAC)

Fuente: IPAC; Tubería Estructural Rectangular, (2016)

Se tomó el tubo rectangular de 4 x 6 x 3 [mm] con $S = W = 8.44 \text{ [cm}^3\text{]} = 0.515 \text{ [in}^3\text{]}$

- **Comprobación:**

$$\sigma_d = \frac{M}{S} = \frac{8.06 \text{ [klbin]}}{0.515 \text{ [in}^3\text{]}} = 15.65 \text{ [kpsi]}$$

$$N = \frac{S_y}{\sigma_d} = \frac{34.136 \text{ [kpsi]}}{15.65 \text{ [kpsi]}} = 2.18$$

Como resultado se obtiene un factor de seguridad de 2.18 el cual es mayor al seleccionado, asegurando una estructura totalmente confiable.

3.7.2.6. *Calculo de la estructura para las vigas EG, GH, HJ, JE y FI*

Se toma valores de las reacciones y momentos máximos más altas en las vigas analizadas para encontrar el perfil adecuado.

- **Momento máximo.**
MM= 94.14 [Nm] = 0.8332 [klbin]
- **Reacción máxima.**
R= 144.83 [N]
- **Características mecánicas del material.**
Acero ASTM 500
Sy=2400 [Kgf/cm2] = 34.136 [Kpsi]
E=2100000 [Kgf/cm2] =29869.04 [Kpsi]
- **Esfuerzo de diseño**

Ecuación 3.8 Esfuerzo de diseño

$$\sigma_d = \frac{S_y}{N} = \frac{34.136 \text{ [kpsi]}}{2} = 17.068 \text{ [kpsi]}$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

σ_d : Esfuerzo de diseño; Sy: Resistencia de fluencia; N: Factor de diseño

- **Módulo de sección requerido**

Ecuación 3.9 Modulo de sección requerido

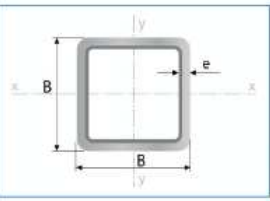
$$S = \frac{M}{\sigma_d} = \frac{0.8332 \text{ [klbin]}}{17.068 \text{ [kpsi]}} = 0.0488 \text{ [in}^3\text{]} = 0.799 \text{ [cm}^3\text{]}$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

S: Modulo de sección requerido; M: Momento flexionante

Se selecciona de la siguiente tabla el perfil que tenga como sección un valor idéntico o superior a $S=0.799 [cm^3]$.



Designaciones	Área	Peso	Propiedades Estáticas			
			Eje x-x = y-y			
B	e	A	P	I	W	i
mm	mm	cm ²	kg/m	cm ⁴	cm ³	cm
20	1,50	1,05	0,83	0,58	0,58	0,74
	2,00	1,34	1,05	0,69	0,69	0,72
25	1,50	1,35	1,06	1,41	1,11	0,97
	2,00	1,74	1,36	1,48	1,18	0,92
30	1,50	1,65	1,30	2,19	1,46	1,15
	2,00	2,14	1,68	2,71	1,81	1,12
	3,00	3,01	2,36	3,50	2,34	1,08
40	1,50	2,25	1,77	5,48	2,74	1,56
	2,00	2,94	2,31	6,92	3,46	1,53
	3,00	4,21	3,30	9,28	4,64	1,48
50	1,50	2,85	2,24	12,56	4,92	2,01
	2,00	3,74	2,93	14,13	5,65	1,94
	3,00	5,41	4,25	19,40	7,76	1,89
60	1,50	3,45	2,71	18,94	6,31	2,43

Largo Normal: 6 metros
 Recubrimiento: Negro o Galvanizado
 Norma de Calidad: ASTM A 500 Gr. A, B ó C
 Norma de Fabricación: NTE INEN 2415
 Espesores: Desde 1,50 a 6,00 mm
 Observaciones: Otras dimensiones, espesores y largos previa consulta

Figura 3.20 Propiedades tubería estructural cuadrada (IPAC)

Fuente: IPAC; Tubería Estructural Cuadrada, (2016)

Se tomó el tubo cuadrado de 25 x 25 x 1.5 [mm] con $S = W = 1.18 [cm^3] = 0.07 [in^3]$

• **Comprobación:**

$$\sigma_d = \frac{M}{S} = \frac{0.08332 [klbin]}{0.07 [in^3]} = 11.57 [kpsi]$$

$$N = \frac{Sy}{\sigma_d} = \frac{34.136 [kpsi]}{11.57 [kpsi]} = 2.95$$

Como resultado se obtiene un factor de seguridad de 2.95 el cual es mayor al seleccionado, asegurando una estructura totalmente confiable.

3.7.2.7. *Análisis de pandeo en los parantes de la estructura*

Este análisis se basa en el criterio de esbeltez, de esta forma se asegura la estabilidad estructural al aplicar cargas.

• **Características mecánicas del material.**

Acero ASTM 500
 $Sy=2400 [Kgf/cm^2] = 34.136 [Kpsi]$

$$E=2100000 \text{ [Kgf/cm}^2\text{]} =29869.04 \text{ [Kpsi]}$$

- **Reacción máxima.**

$$R= 1401.92 \text{ [N]}$$

- **Constante de la columna.**

Ecuación 3.10 Constante de la columna

$$C_c = \sqrt{\frac{2\pi^2 E}{S_y}} = \sqrt{\frac{2\pi^2 2100000 \text{ [kpsi]}}{2400 \text{ [kpsi]}}} = 131.2$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

C_c : Constante de la columna; E: Modulo de elasticidad de la columna;

S_y : Resistencia de fluencia.

Se selecciona un tubo cuadrado de 40 x 40 x 2 [mm], de este se toma los valores del área de la sección transversal y el momento de inercia.



Designaciones	Area	Peso	Eje X-X' = Y-Y'			
			Momento de inercia	Módulo de resistencia	Radio de giro	
B	e	A	P	I	W	r
mm	mm	cm ²	kg/m	cm ⁴	cm ³	cm
20	1,50	1,05	0,83	0,58	0,58	0,74
	2,00	1,34	1,05	0,69	0,69	0,72
25	1,50	1,35	1,06	1,41	1,11	0,97
	2,00	1,74	1,36	1,48	1,18	0,92
30	1,50	1,65	1,30	2,19	1,46	1,15
	2,00	2,14	1,68	2,71	1,81	1,12
40	3,00	3,01	2,36	3,50	2,34	1,08
	1,50	2,25	1,77	5,48	2,74	1,56
40	2,00	2,94	2,31	6,92	3,46	1,53
	3,00	4,21	3,30	9,28	4,64	1,48
50	1,50	2,85	2,24	12,56	4,92	2,01

Largo Normal:
6 metros
Recubrimiento:
Negro o Galvanizado
Norma de Galvanizado:
ASTM A 500 Gr. A, B ó C
Norma de Fabricación:
NTE INEN 2415

Figura 3.21 Propiedades tubería estructural cuadrada (IPAC)

Fuente: IPAC; Tubería Estructural Cuadrada, (2016)

- **Radio de giro de la sección transversal del perfil seleccionado.**

Ecuación 3.11 Radio de giro de la sección transversal

$$r = \sqrt{\frac{I}{A}} = \sqrt{\frac{6.92 \text{ [cm}^4\text{]}}{2.94 \text{ [cm}^2\text{]}}} = 1.53 \text{ [cm]}$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

r : Radio de giro de la sección transversal; I : Momento de inercia de la sección transversal; A : Área de la sección transversal.

- **Longitud efectiva de la columna.**

Para este cálculo se necesita el valor de la constante, correspondiente al tipo de conexión en los extremos de la columna. Se toma el valor de $k=0.65$ correspondiente a la columna empotrada - empotrada.

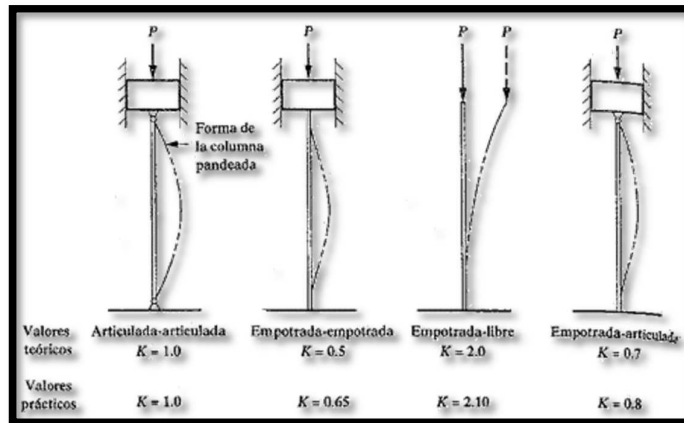


Figura 3.22 Valor de K para diferentes tipos de conexión a los extremos.

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Ecuación 3.12 Longitud efectiva

$$Le = KL = 0.65 \times 90 [cm] = 58.5 [cm]$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

Le : Longitud efectiva; K : Constante correspondiente al extremo fijo;

L : Longitud real de la columna entre los soportes.

- **Relación de esbeltez.**

Ecuación 3.13 Relación de esbeltez

$$relacion\ de\ esbeltez = \frac{Le}{r} = \frac{58.5 [cm]}{1.53 [cm]} = 38.23$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

Le : Longitud efectiva; r : radio de giro.

Se compara la relación de esbeltez con respecto a la constante de la columna, para la selección de la fórmula de la carga crítica de pandeo.

$$\text{relacion de esbeltez} < \text{Constante de la columna} \\ 38.23 < 131.2$$

Por esta razón se usa la fórmula de B. Johnson para columnas cortas.

- **Carga crítica de pandeo calculada**

Ecuación 3.14 Carga crítica de pandeo calculada

$$P_{cr} = A S_y \left(1 - \frac{S_y \left(\frac{L_e}{r} \right)^2}{4\pi^2 E} \right) \\ P_{cr} = 2.94 [cm^2] * 2400 [kpsi] \left(1 - \frac{2400 [kpsi] x 38.23^2}{4\pi^2 x 2100000 [kpsi]} \right) \\ P_{cr} = 66268.04 [N]$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

P_{cr}: Carga crítica de pandeo calculada.

- **Carga admisible en la columna**

Ecuación 3.15 Carga admisible en la columna

$$P_a = \frac{P_{cr}}{N} = \frac{66268.04 [N]}{2} = 33134.02 [N]$$

Fuente: Mott, Robert; Diseño de elementos de máquinas, 2006.

Donde:

P_a: Carga admisible en la columna; N: factor de diseño.

- **Análisis**

$$P_a > R \\ 33134.02 > 1401.92 [N]$$

La carga admisible de la columna es mayor a la carga máxima aplicada a la columna, por esta razón es seguro su diseño.

3.7.2.8. Factor de seguridad simulado en SolidWorks

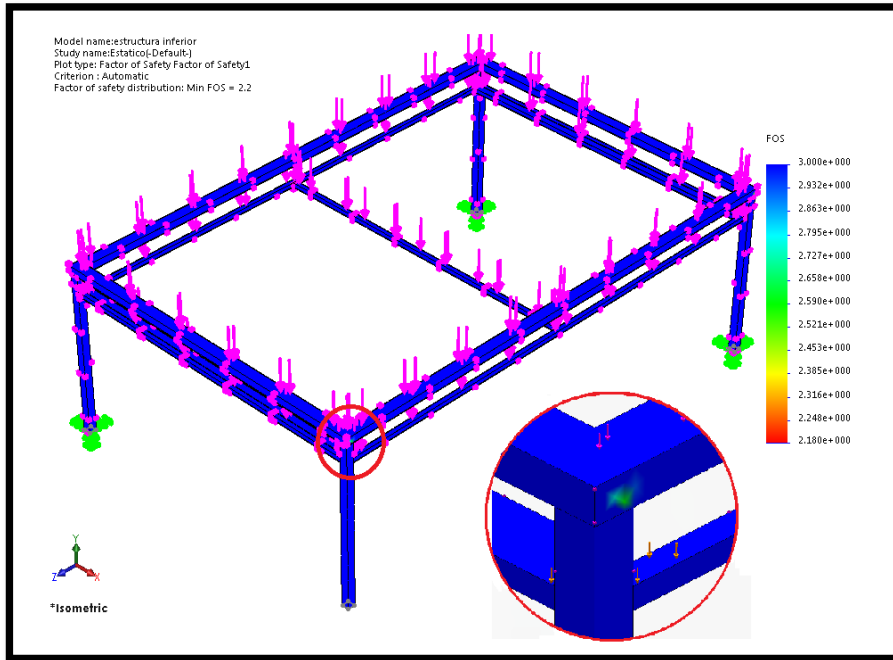


Figura 3.23 Simulación factor de seguridad de la estructura inferior

Se puede comprobar que el factor de seguridad más bajo simulado es de 2.2 y se encuentra en las vigas que soportan al vidrio templado, y se aproxima al valor calculado que es de 1.8.

3.7.3. Elementos que complementan a la estructura superior

Se considera el peso de los elementos que se detallan a continuación:

Tabla 3.16 Elementos que complementan a la estructura superior

Elementos	Cantidad	Ubicación en la estructura	Propiedades		
			Peso unidad	Peso total	Dimensiones externas Largo x ancho x alto
Lámparas halógenas	14 [u]	Colgaran de la estructura por medio de sócalos	25 [g]	0.35 [kg]	51 x 51 x 40 [mm]
Cableado	10 [m]	Interior de la estructura	30 [g]	0.3 [kg]	5.88 x 2.73 [mm]
Cable gemelo N° 18					
Transformador	5 [u]	Interior de la estructura	50 [g]	0.25 [kg]	92 x 34 x 20 [mm]
Cámara	1 [u]	Interior de la estructura (centro)	250 [g]	0.25 [kg]	150 x 33 x 30 [mm]
			Total	1.15 [kg]	

3.7.3.1. *Diseño de la estructura superior*

Tomando en consideración las medidas de los elementos que van en su interior se tiene que el valor mínimo considerable es en relación al transformador. Además, se puede apreciar que no existen pesos considerables ya que ninguno supera los 0.35 [kg].

Como se indicó anteriormente debe estar separada de la estructura inferior, con el fin de eliminar vibraciones que puedan afectar al resultado, es por ello que se anclara directamente al techo en forma de lámpara colgante. El material a usarse es tubo estructural de aluminio 6061.

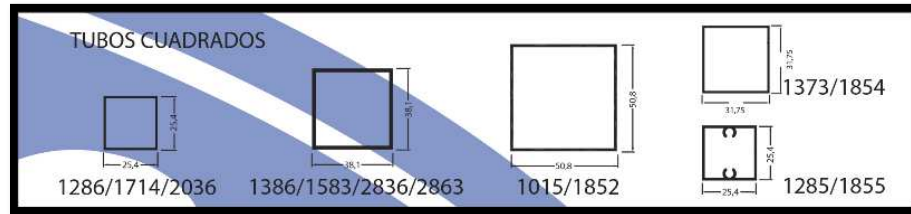


Figura 3.24 Tubería cuadrada Cedal

Fuente: Cedal; Perfiles de aluminio arquitectónico y estructural, 2016

El tamaño del exterior del tubo se lo escoge tomando en cuenta las dimensiones del elemento más grande a lo ancho y alto, en este caso el transformador con dimensiones 34 x 20 [mm], dándonos como apropiado el tubo con referencia 1386/1583/2836/2863 tomado del anexo 8.3.

La forma de la estructura se basa en la Figura 3.5.

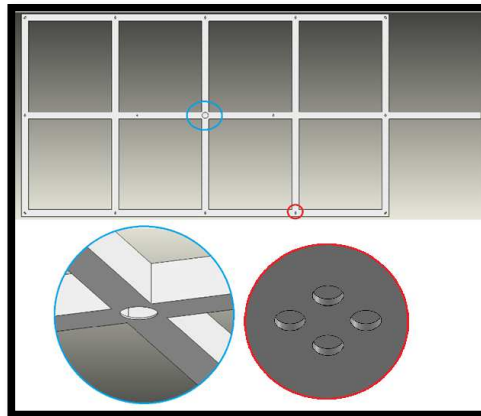


Figura 3.25 Estructura superior.

En la ampliación azul se visualiza el agujero por el cual pasara el lente de la cámara, en la ampliación roja se muestra los agujeros para el cableado de los zócalos y dos agujeros extras para los tornillos que sostienen a los mismos.

Los planos con medidas y formas de la estructura superior e inferior se encuentran en los anexos: 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.8.

3.8. SOFTWARE PARA VISIÓN ARTIFICIAL

Como se mencionó anteriormente se usa una cámara con un objetivo gran angular para abarcar el área de trabajo, este tipo de objetivo produce distorsiones tipo barril. Por lo que se tiene la necesidad de trabajar con software que procese la imagen y atenúe los errores en la adquisición.

3.8.1. Requerimientos fundamentales del sistema de visión

Se presenta la necesidad de automatizar el proceso de control de calidad en un entorno de producción intermitente, mediante un sistema de visión artificial. Se obtiene un grupo de hojas de cuero unicolores, se requiere medir el área real del objeto y clasificarlo según la cantidad de imperfecciones que presente. El cuero es ingresado al sistema de forma manual por el operador, el mismo que da la confirmación de que el cuero se encuentra correctamente posicionado dentro del sistema.

El sistema tiene que ser capaz de:

- Controlar la intensidad de la luz
- Captar la imagen del cuero
- Procesar la imagen
- Medir el área con su clasificación.
- Guardar la información en una base de datos.

Con el resultado de la inspección, se emite una etiqueta que contiene la imagen, el área y la clase de cada cuero.

3.8.2. Implementación del software

EL programa se divide en tres etapas:

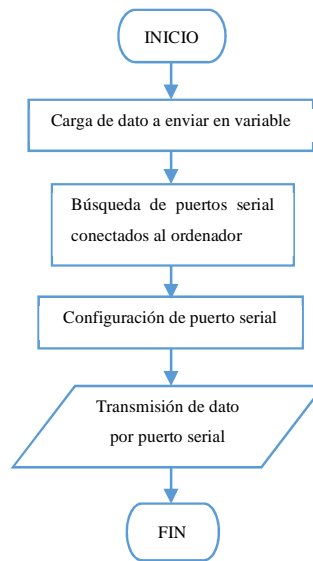
- Etapa de calibración de parámetros
- Etapa de medición y clasificación
- Etapa de visualización de imágenes almacenadas.

A continuación, se detalla los segmentos del código fuente.

3.8.2.1. *Función apagarluzinferior*

La función transmite el dato 6 hacia la tarjeta que controla la luminaria, provocando el apagado de la luminaria inferior.

Gráfico 3.3 Diagrama de flujo de la función apagarluzinferior



Para ello primero se debe verificar si existe algún puerto serial activo. En el caso de no existir puertos se obtiene el valor de 1 ya a que la variable serialdisponible está vacía, de lo contrario se obtiene el nombre del puerto y la respuesta es 0.

Seguido a esto se cargan los parámetros de funcionamiento del puerto como: la frecuencia de transmisión, los bits por caracteres, etc., esto inicia solo en el caso de que exista puerto serial. Por último, se transmite el dato precargado para ello se abre el puerto serial, envía el dato y cierra el puerto serial.

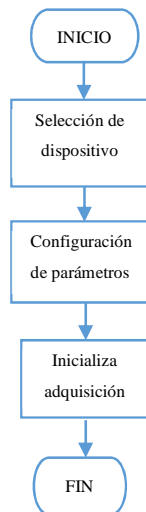
La programación correspondiente a esta función se encuentra en el anexo 4.1

Las funciones apagarluzsuperior, encenderluzinferior, encenderluzsuperior, aumentarluminosidadsuperior, disminuirLuminosidadsuperior mantienen el procedimiento de esta función cambiando únicamente el dato a transmitir siendo: 4, 5, 3, 1, 2 respectivamente.

3.8.2.2. *Función video*

La función selecciona la cámara, carga los parámetros de configuración e inicializar la toma de imágenes.

Gráfico 3.4 Diagrama de flujo de la función video



Para la selección del dispositivo se describe el formato de la imagen que se desea adquirir y la localidad de la cámara la cámara. Se configura los parámetros de está optando por un número infinito de fotos por disparo y cargar el tipo de color rgb. Por último, se inicializa la adquisición.

La programación correspondiente a esta función se encuentra en el anexo 4.18

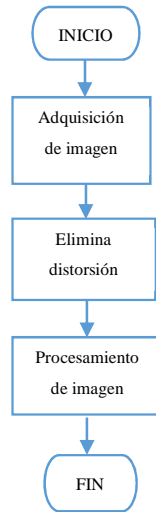
3.8.2.3. *Función eliminadistorsion*

De esta función se van a obtener dos imágenes resultantes, una de ellas a escala de grises y otra en binario.

Primero se obtiene una captura de la imagen cargada en la variable “vid” y se almacena en la variable “im”. Para eliminar la distorsión propia de la cámara se carga el archivo que contiene los parámetros de la cámara obtenidos anteriormente y se usa la sintaxis `undistortImage(im,cameraParams)`

Una vez se tiene la imagen sin distorsión se transforma a escala de grises, y se transforma a blanco y negro. Por último, se carga el valor 1 en la variable `elimdist`, indicando que se ha completado el proceso de eliminar la distorsión.

Gráfico 3.5 Diagrama de flujo de la función eliminadistorsion

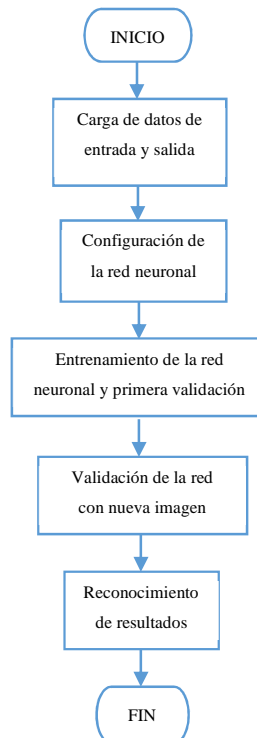


La programación correspondiente a esta función se encuentra en el anexo 4.12

3.8.2.4. Función *rn_comprobacionluminaria*

En esta función se da uso de redes neuronales artificiales para identificar si existe alguna luminaria en mal estado. Se efectúa un reconocimiento de imágenes.

Gráfico 3.6 Diagrama de flujo de la función *rn_comprobacionluminaria*



En la primera etapa se ingresa las 4 imágenes de referencia de tamaño 20 x 11 pixeles en el vector entrada. A las imágenes se las transforma a un vector de 220 filas y 1 columna, resultando un vector de entrada de 220 filas y 4 columnas.



Figura 3.26 Imágenes de referencia para entrada de redes neuronales.

Al vector de salida se lo carga con un vector de 2 filas y 4 columnas, la primera fila representa la luminaria inferior y la segunda fila la luminaria superior, se lo carga con las respuestas correspondientes a cada imagen, tiene el valor de 1 si está encendida o 0 si se encuentra apagada.

Se configura a la red neuronal multicapa con retro propagación con: una capa oculta de 50 neuronas, se carga tanto la capa oculta como la capa de salida con función de transferencia logística, se define un número máximo de iteraciones de 200; para evitar respuestas aleatorias se anula la división de datos y la función de pre procesamiento de datos tanto para la entrada como para la salida.

Ya creada la red neuronal se entrena con los vectores de entrada y salida de referencia, y se valida con el vector de entrada de referencia.

Ya establecidos los parámetros de referencia se obtiene un nuevo valor de salida simulando nuevamente la red, pero con la imagen recién adquirida, esta imagen previamente se la dilata para resaltar las características necesarias y se la reduce el tamaño a una escala de 1:100. El valor de esta simulación se guarda en la variable “validacion1”.

Para encontrar el lugar del fallo se toma los 2 valores que se encuentran en la columna 4 del vector resultante de la primera validación, correspondientes a la alternativa de luminarias inferior y superior encendidas; se los compara con el valor de la variable “validacion1” dándonos el tipo de fallo.

Por ejemplo:

$$validacion = \begin{bmatrix} 0 & 0.95 & 0 & 0.95 \\ 0 & 0 & 0.97 & 0.97 \end{bmatrix}$$

$$validacion1 = \begin{bmatrix} 0.02 \\ 0.97 \end{bmatrix}$$

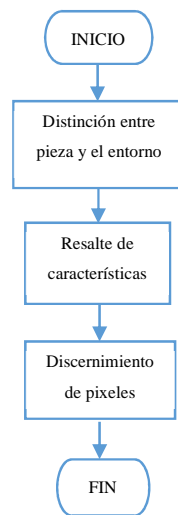
Se extrae de validación los valores de la última columna y se compara con validación 1. Comparando con los casos de la figura 3.26 se cargaría la variable iluminf con 0 e ilumsup con 1.

La programación correspondiente a esta función se encuentra en el anexo 4.20

3.8.2.5. *Función eliminamesa*

La finalidad de esta función es eliminar todo lo que no pertenezca a la pieza de cuero o guía, por ejemplo: manos, lámparas, mesa.

Gráfico 3.7 Diagrama de flujo de la función eliminamesa



Para ello se procesa a la imagen guardada en la variable imundgray usando la función imadjust, distribuyendo el histograma de la imagen en el rango especificado, provocando un aumento de contraste, como se puede ver en la siguiente imagen.

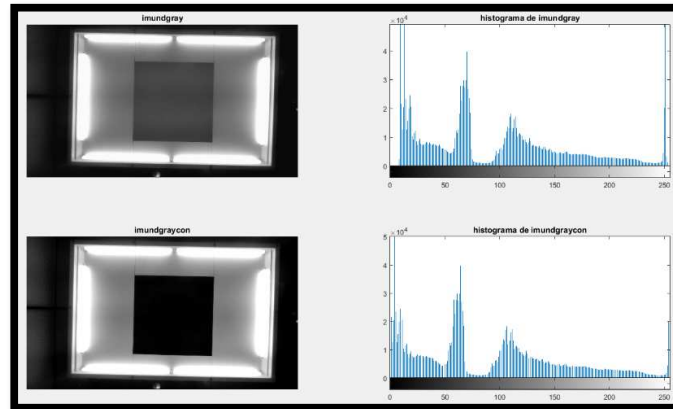


Figura 3.27 Aumento de contraste en imundgray

La nueva imagen pasa por el filtro mediana para de esta manera eliminar cualquier ruido que haya quedado en la imagen, esto lo hace tomando los valores de los pixeles vecinos que por lo general son parecidos y cambiando el valor que se dista del resto.

A esta imagen se la debe transformar a binaria, para eso se calcula el umbral promedio, este valor se encuentra en el rango del 0 al 1, y permite separar el objeto de interés del entorno; con este valor se transforma a binario, se resta al umbral un valor de 0.2 por condiciones externas de luminosidad.

Con la imagen en binario, se procede resaltar los elementos, como juntas y bordes usando la función `bwmorph`, se realiza dilatación quedando la mesa de color blanco y el cuero totalmente negro, se usa `skel` para perfilar el contorno de la pieza de cuero, se invierte los valores de la imagen y se guarda en variable "imbn".

Nuevamente se dilata la imagen del cuero usando la función `bwmorph`, con la finalidad de aumentar el área que conforma el marco de la mesa, esto ayuda cuando se superpongan las imágenes.

Ahora se debe segmentar una imagen en pequeños grupos de objetos. Con las piezas del marco y del cuero correctamente separadas una de la otra se procede a seleccionarla usando las funciones anteriormente mencionadas. Como se requiere el primer objeto que sería el marco, primero se etiqueta los objetos para luego seleccionar la propiedad que se necesita usando la función `regionprops`, resultando: `seccion=regionprops(etiquetaobj,'Image')`.

Como se sabe el primer objeto en la imagen de izquierda a derecha es el marco por ello se lo selecciona con la sintaxis: marco=seccion(1).Image, quedando aislado el marco de la mesa por completo. A continuación sumamos la imagen “imb” con la imagen que contiene solo el marco imcor=or(imb,marco) y dilatamos por última vez, para eliminar cualquier espacio sin rellenar que haya quedado.

Se indica las imágenes resultantes más importantes de este proceso con su respectiva variable a la que fueron asignadas.

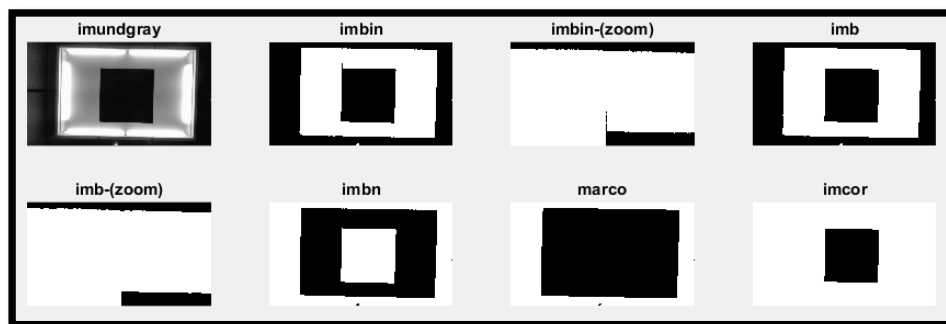


Figura 3.28 Imágenes representativas de la función eliminamesa

La programación correspondiente a esta función se encuentra en el anexo 4.13

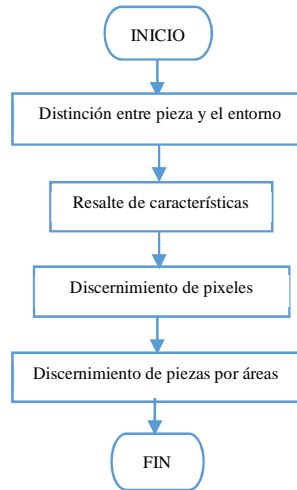
3.8.2.6. Función eliminamesaumentadocontraste

La finalidad de esta función es la misma que en la función eliminamesa, la diferencia se encuentra en el método para aumento de contraste a la imagen en escala de grises. En este caso se lo realiza por medio de filtro por enfoque, este detecta los cambios bruscos entre valores de pixel y resalta los bordes aumentando la diferencia que existe entre estos, con esto nos ayudamos a perfilar de mejor forma la pieza de cuero y ayuda posteriormente en la función areaplantilla para el resalte de bordes.

La sección de resalte de características y el discernimiento de pixeles son las mismas que se usaron en la función eliminamesa. A esta función también se añade un método de discernimiento extra, para escoger únicamente las secciones con áreas mayores a 207036 pixeles, que representan el 10% del área total de la imagen capturada por la cámara.

En la variable `imselecta` se tiene la nueva imagen únicamente con la pieza de cuero en blanco y el resto en negro.

Gráfico 3.8 Diagrama de flujo de la función *eliminamesaaumentadocontraste*



La programación correspondiente a esta función se encuentra en el anexo 4.14

3.8.2.7. Función *areapieza*

La función entrega el valor del área de la plantilla medida en píxeles a partir de la imagen en escala de grises guardada en la variable `"imundgraycon"`.

A esta imagen se la debe tratar para realzar los contornos. Por medio de la función `edge`, para aumentar la su eficacia se complementa con la función `eliminamesaaumentadocontraste`, la cual tiene un filtro por enfoque.

Se usa la función `"edge"` con el método `"sobel"`. De esta forma únicamente obtendremos los bordes en color blanco y el resto de color negro.

Ahora se debe discernir la información necesaria del resto de la imagen. Para ello primero se debe realzar las características que se necesita, uniendo los píxeles sueltos mediante la funcione `bwmorph`, con las operaciones `"dilate"`, `"bridge"` y `"diagonal"`, esta nueva imagen es almacenada en `imcontmorf`.

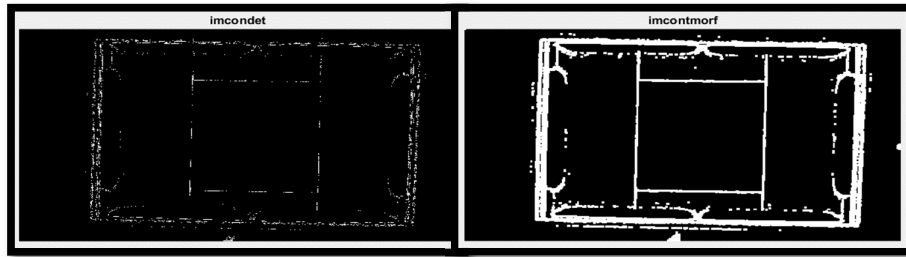


Figura 3.29 Realce de características.

Seguido a esto se descarta la información de la imagen que no se necesita, usando como plantilla la imagen guardada en la variable “imselec” con la guardada en “imcontmorf”. A estas dos imágenes se las conecta usando la función “and” resultando visibles solamente los píxeles de color blanco que concuerden en la misma ubicación. La imagen resultante corresponde a la pieza y se almacena en “impieza”.

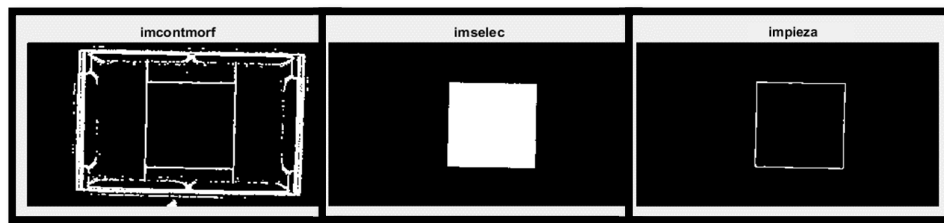
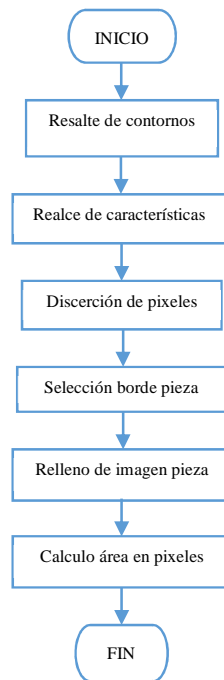


Figura 3.30 Discernición de píxeles.

Una vez obtenida la imagen de la pieza se conectan los píxeles de la última imagen con la finalidad de obtener los parámetros correspondientes al perímetro de cada región. Se analiza la imagen buscando las regiones que poseen un perímetro mayor a 2000 píxeles, de esta forma se elimina cualquier residuo de píxeles sueltos.

Con la imagen limpia, sin excedente de píxeles, se rellena el interior de la pieza con la sintaxis `imfill(imselecta,'holes')`. Como resultado se obtiene la imagen de la pieza de color blanco. Para terminar el cálculo del área se lo realiza conectando los píxeles de la última imagen y obteniendo la propiedad área, centroide, cuadro delimitador de la región analizada, este valor se guarda en la variable “area1”.

Gráfico 3.9 Diagrama de flujo de la función areapieza



La programación correspondiente a esta función se encuentra en el anexo 4.3

3.8.2.8. *Función calculodeparametros*

De esta función se obtiene como resultado el valor correspondiente al factor de conversión. Este valor sirve para transformar la unidad de medida de [pixel] a [cm²].

Primero se inicializa un ciclo while, dentro de este se elimina la distorsión de la cámara y se obtiene la imagen de la plantilla a partir de la función “eliminamesa”. A esta imagen se la considera dos probabilidades, la primera es que no existan pixeles de color negro en la imagen, a lo que muestra una imagen en blanco de dimensiones 1080 x 1920 [pixeles]; la segunda es si existen pixeles negros en la imagen, en este caso se filtra pixeles sueltos tomando en cuenta que la imagen de la pieza tiene un área mayor al 10% del área de la imagen en [pixeles].

De este filtrado de pixeles se obtiene una nueva imagen a la que se la debe de considerar las posibles dos condiciones, si no existen pixeles negros se muestra una imagen en blanco de tamaño 1080 x 1920 [pixeles], caso contrario analiza la posición en la que se encuentra la plantilla.

La posición se determina convirtiendo los píxeles de la imagen en regiones, para luego obtener el punto central de la región, que para este caso sería el punto medio de la plantilla.

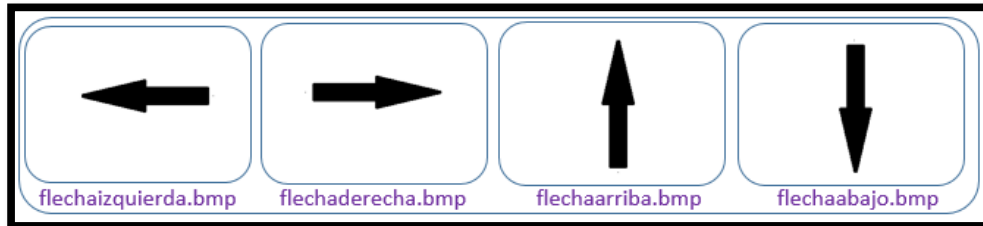


Figura 3.31 Imágenes de flechas.

Dependiendo esta posición se muestran flechas que indican hacia donde se debe mover la pieza para que el centro de la pieza coincida con el centro de la cámara. Para ello se divide la imagen en 5 sectores.

La imagen a continuación ejemplifica a una fotografía obtenida por la cámara de tamaño 1080 de alto y 1920 de ancho.

- Sector verde: zona izquierda, comprendido entre el vector 1 al 959 de la dimensión ancho.
- Sector azul: zona derecha, comprendido entre el vector 961 al 1920 de la dimensión ancho.
- Sector azul y verde oscuro: zona superior, comprendido entre el vector 1 al 539 de la dimensión alto.
- Sector azul y verde claro: zona inferior, comprendido entre el vector 541 al 1080 de la dimensión alto.
- Sector rojo: zona central, comprendido entre el vector 940 al 980 de la dimensión ancho y el vector 520 al 560 de la dimensión alto.

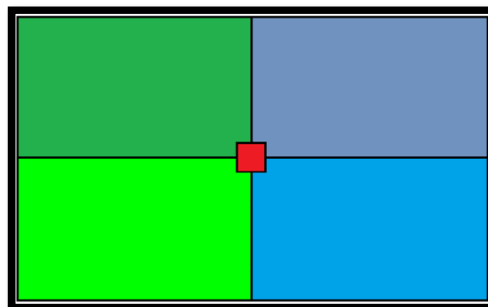


Figura 3.32 Sectores imagen para localización de plantilla.

Por ejemplo, si el punto central de la pieza nos da la posición [300 620], se interpreta que se encuentra en el vector 300 tomado a lo ancho y 620 a lo alto de la imagen, esto la localiza en el sector verde claro; por lo tanto, se visualiza las flechas: derecha y arriba, indicando que se debe trasladar la pieza en ese sentido para que se encuentre centrada con respecto al sector rojo, además de mostrar la imagen de la plantilla y un punto rojo situado en el centro de la pieza.

En el caso que se encuentre el punto central de la plantilla en el sector rojo, se muestra la imagen de la pieza y un punto de color verde situado en el centro de la pieza.

Una vez se tenga posicionada la pieza se detiene la adquisición de imágenes, se llama a la función areapieza y calcula el valor del factor de conversión de acuerdo a la siguiente formula:

Ecuación 3.16 Formula para cálculo del factor de conversión de pixeles a cm²

$$fc = \frac{14400 [cm^2]}{aipg [pixel]}$$

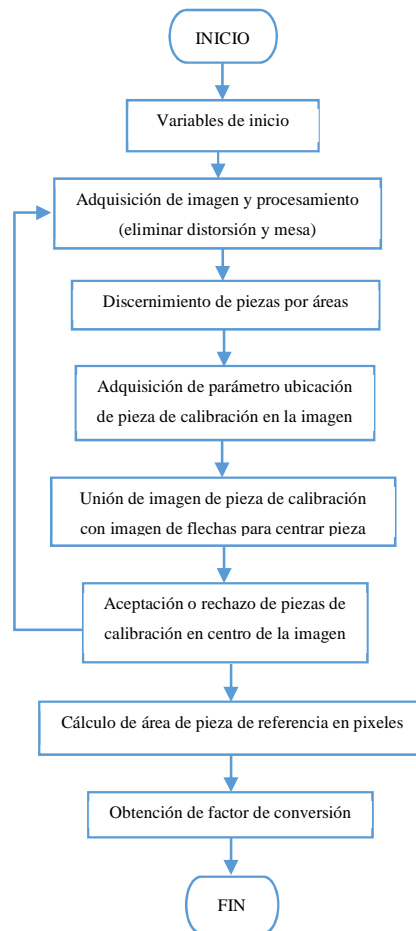
Donde:

fc= factor de conversión [$\frac{cm^2}{pixel}$]; aipg= Área en la imagen de la pieza guía [pixel]

El valor de 14400 corresponde a la medida real del área de la pieza de forma cuadrada que tiene una dimensión por lado de 120 cm.

Los resultados obtenidos en fc y aipg se los guarda en un archivo de formato .mat para su posterior uso.

Gráfico 3.10 Diagrama de flujo de la función calculodeparametros



La programación correspondiente a esta función se encuentra en el anexo 4.6

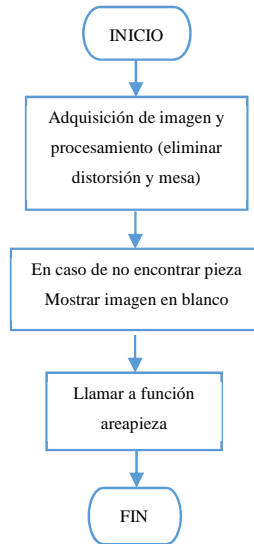
3.8.2.9. *Función deteccionpieza*

Entrega las propiedades de la pieza de cuero, como el área, las coordenadas del punto medio y el cuadro que delimita los bordes.

Para que esta función de inicio previamente se debe llamar a la función video y obtener imágenes de la cámara.

Dentro de esta función se llama a eliminadistorsion y eliminamesa, y se abre una condición en el caso de que la imagen se encuentre en blanco ya sea porque se detectó manos o la pieza fuera del borde termina el proceso, caso contrario llama a función areapieza.

Gráfico 3.11 Diagrama de flujo de la función *detecciónpieza*



La programación correspondiente a esta función se encuentra en el anexo 4.9

3.8.2.10. *Función deteccionplantilla*

La función permite extraer las propiedades como el área y el centroide de las plantillas, estos datos sirven para clasificar el cuero dependiendo de la plantilla usada y la ubicación que esta tenga.

Para ello se resalta los contornos, realiza las características y se discierne pixeles de la misma manera que se realizó en la función *areapieza*. En la sección de discernición de pixeles se tienen previamente una imagen en la variable *~imcor*, esta imagen al ser unida mediante la función *and* deja visibles únicamente los pixeles que se encuentran dentro de la pieza.

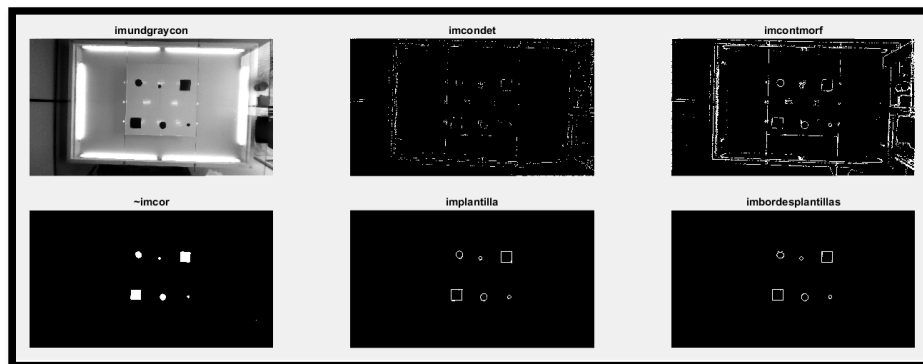


Figura 3.33 Bordes plantillas negras con pieza clara

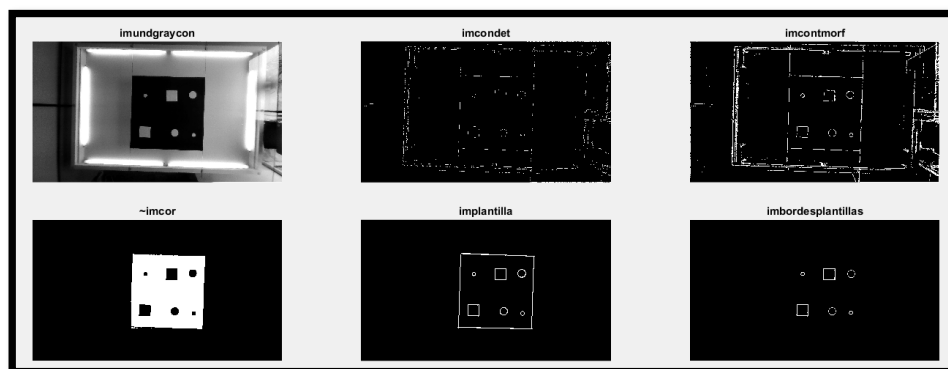


Figura 3.34 Bordes plantillas blancas con pieza oscura

Como se puede apreciar en la Figura 3.34 en la imagen con título *implantilla* si se la compara con la imagen del mismo título que se encuentra en la Figura 3.32, se puede observar que existe el borde que forma la pieza. Es por ello que se crea un filtro adicional, en el cual solo se escoja regiones de pixeles que no superen un perímetro de 600 o lo que es igual al 10% del perímetro de la imagen total. El resultado se guarda en la variable *imbordesplantillas*.

Con las plantillas identificadas se rellena el interior de las plantillas y se elimina los bordes de las mismas. De esta nueva imagen se adquiere la propiedad del área, de cada una de las piezas, para de esta forma clasificarla.

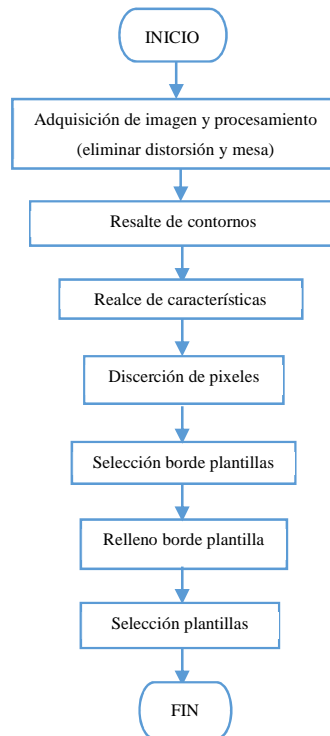
Tabla 3.17 Área de la plantilla según su forma

Prueba N°	Color plantilla	Forma de plantilla		
		Circular 5 [cm] de diámetro	Circular 10 [cm] de diámetro	Cuadrada 15 [cm] de lado
1	Blancas	430 [pixeles]	2283 [pixeles]	6457 [pixeles]
		528 [pixeles]	2380 [pixeles]	6717 [pixeles]
	Negras	374 [pixeles]	2162 [pixeles]	6491 [pixeles]
		359 [pixeles]	2048 [pixeles]	6437 [pixeles]
2	Blancas	498 [pixeles]	2273 [pixeles]	-
	Negras	362 [pixeles]	2007 [pixeles]	-

En base a la tabla anterior se delimita como valor mínimo de área 300 [pixeles], de las nuevas regiones se extrae el área y el centroide, los datos se guardan en la variable

datosregiones2. Este valor coincide únicamente cuando se encuentre la cámara a la distancia correcta.

Gráfico 3.12 Diagrama de flujo de la función deteccióplantilla

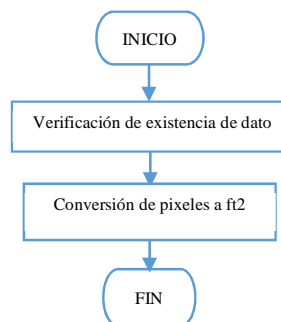


La programación correspondiente a esta función se encuentra en el anexo 4.10

3.8.2.11. Función calculoarea

De esta función se obtiene la medida del área tanto en $[cm^2]$ como en $[ft^2]$.

Gráfico 3.13 Diagrama de flujo de la función calculoarea



Para ello se extrae el valor del área a partir de la variable area1 obtenida en la función deteccionpieza, se carga el archivo que contiene el factor de conversión. Con estos dos

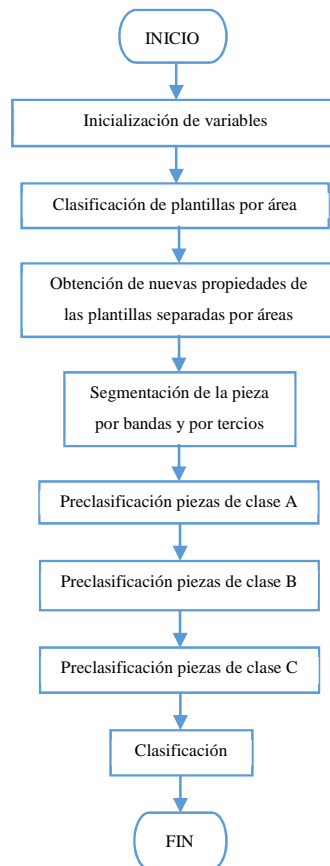
datos se convierte de [píxeles] a [cm^2]. Seguido a esto se convierte de [cm^2] a [ft^2]. Estos dos datos se transforman a formato string para su posterior uso.

La programación correspondiente a esta función se encuentra en el anexo 4.5

3.8.2.12. *Función clasificación*

El objetivo principal de esta función es dar una categoría a la pieza de acuerdo a las fallas y la ubicación sobre el mismo.

Gráfico 3.14 Diagrama de flujo de la función clasificación



Para ello se usa plantillas, cada una de estas representa un área afectada y un tipo de falla. A continuación, se detalla cada una en base a la clasificación de la tabla 2.3.

Tabla 3.18 Plantillas, lugar y cantidad por clase

Plantilla	Área afectada	Falla	Clase A		Clase B		Clase C	
			Cantidad aceptable	Lugar	Cantidad aceptable	Lugar	Cantidad aceptable	Lugar
Círculo de 5 [cm] de diámetro	5 [cm] de diámetro	Cicatriz cerrada, Arrugas, Pintura	1	En 1 tercio, fuera de la banda central				
Círculo de 10 [cm] de diámetro	10 [cm] de diámetro	Tórsaleras, Garrapatas, Cicatriz abierta, Cicatriz cerrada, Arrugas, Pintura	0		2	En cada tercio	3 o más	En cualquier lugar
Cuadrado de 15 [cm] por lado	No específica	Marcas de fierro.	2	En la banda del lomo	3	En la banda del lomo	4 o más	En cualquier lugar
			Área útil 90%		Área útil 60-90%		Área útil 1-60%	

Se conoce previamente las propiedades como el área y centroide de las plantillas, a estos datos se los separa con base a los datos obtenidos en la tabla 3.18 Obteniendo como resultado:

Tabla 3.19 Plantillas, clasificación por área

Plantilla circular 5 cm de diámetro		Plantilla circular 10 cm de diámetro		Plantilla cuadrada 15 cm de lado	
Área menor	Área mayor	Área menor	Área mayor	Área menor	Área mayor
300	1500	1501	5000	5001	8000
[pixeles]	[pixeles]	[pixeles]	[pixeles]	[pixeles]	[pixeles]

Una vez separadas las regiones por áreas, se obtiene nuevamente datos acerca de la posición de cada plantilla y del cuadro delimitador de las regiones analizadas.

La segmentación de la pieza se realiza adquiriendo el valor del boundingbox que se encuentra en la variable propimcor3. Tomando en cuenta la posición en la que debe ser introducida la pieza de cuero. Para ello se tiene señalado en la mesa de vidrio el lado al que debe ser ubicado el sector perteneciente al lomo.

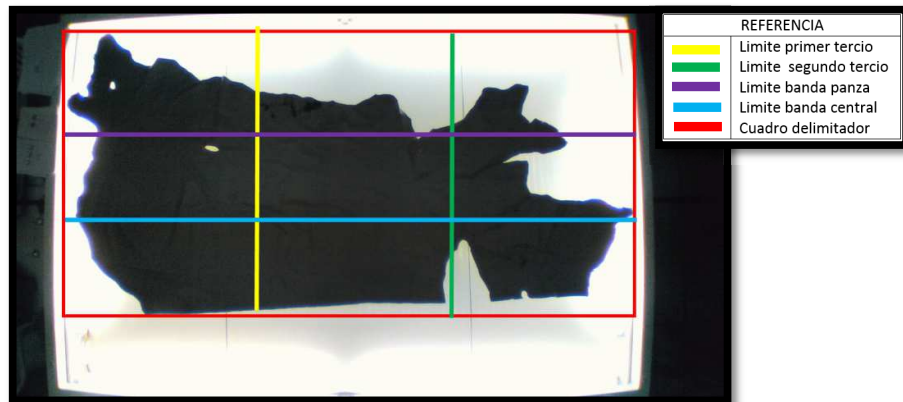


Figura 3.35 Segmentacion de la pieza

Para obtener el valor de los límites se lo hace de la siguiente manera

Con los datos obtenidos de las plantillas tanto por su tamaño y ubicación, ya se puede dar una clasificación. Para ello se toma como referencia la tabla 3.19

Donde se aceptan como cueros de clase “A” solamente si tiene un círculo de 5 [cm] de diámetro fuera de la banda del lomo, como máximo 2 cuadrados de 15 [cm] en la banda del lomo y que se encuentren en un solo tercio, un área útil mayor al 90%.

Para que una pieza de cuero sea aceptada como cueros de clase “B” solamente tiene dos círculos en un mismo tercio, como máximo 3 cuadrados de 15 [cm] en la banda del lomo, un área útil mayor al 60%.

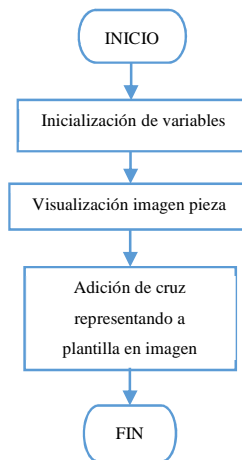
Para que una pieza de cuero sea aceptada como cueros de clase “C” debe tener más de 6 círculos o más de 2 círculos en un mismo tercio, más de 3 cuadrados de 15 [cm], un área útil inferior al 60%.

La programación correspondiente a esta función se encuentra en el anexo 4.7

3.8.2.13. Función imagenclase

Permite unificar los datos obtenidos del centroide de las plantillas con la imagen de la pieza.

Gráfico 3.15 Diagrama de flujo de la función imagenclase



Para lograrlo primero se muestra la imagen guardada en imrellena invertido colores. Se activa la función hold, para que mantenga la imagen actual mientras se muestran nuevas. De esta forma tomamos la ubicación del centroide de cada una de las plantillas y en su lugar se inserta un signo “+” de color blanco. Resultando como la siguiente imagen.

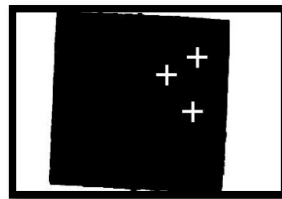


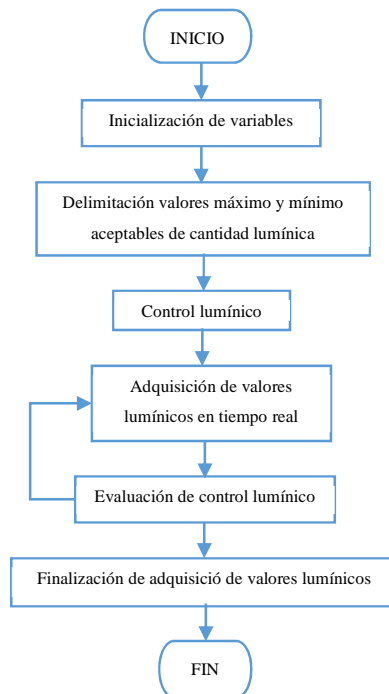
Figura 3.36 Imagen representativa a función imagenclase

La programación correspondiente a esta función se encuentra en el anexo 4.17

3.8.2.14. **Función controluminosidadsuperior**

La función proporciona un correcto nivel de luminosidad de acuerdo a las condiciones existentes en el ambiente.

Gráfico 3.16 Diagrama de flujo de la función control luminosidad superior



Lo que quiere decir compara el nivel de luminosidad del ambiente con el nivel de luminosidad que proporciona la luminaria superior. Para ello captura dos imágenes. La primera cuando la luminaria superior se encuentra apagada, de esta obtenemos el nivel de luminosidad del ambiente; la segunda con la luminaria superior encendida al máximo, en esta imagen obtenemos el nivel de luminosidad de la luminaria.

Con los datos obtenidos se tiene dos posibilidades:

- Si el nivel de luminosidad de la luminaria superior es menor al nivel de luminosidad del ambiente: En este caso no sería necesario encender la luminaria.
- Si el nivel de luminosidad de la luminaria superior es mayor al nivel de luminosidad del ambiente: Aumenta progresivamente la luminosidad hasta tener una luminosidad aceptable.

El nivel de luminosidad aceptable se encuentra en un rango de -2% y +20% del valor de luminosidad de la luminaria superior encendida. Se considera un rango de 2% cuando la luz es inferior ya que si no se tiene la suficiente cantidad de luz la imagen no va a tener las características apropiadas, y el rango de 20% es ya que no afecta si hay un poco más de luminosidad, pero se regula el excedente.

Para asegurar el que no se quede enclavado subiendo el nivel de luminosidad; se establece el caso en el que el valor actual de luminosidad en la imagen nueva es menor al límite inferior del nivel de luminosidad aceptable y además haya aumentado 150 veces, en este caso sale automáticamente del ciclo.

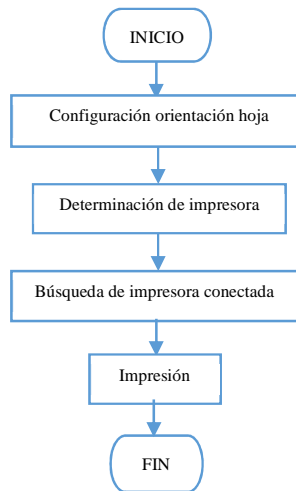
Así mismo existe la posibilidad de que supere el límite superior del nivel de luminosidad aceptable, en este caso envía a disminuir a el nivel de luminosidad progresivamente hasta un rango estable para que posteriormente aumente hasta estar en un rango estable.

La programación correspondiente a esta función se encuentra en el anexo 4.8

3.8.2.15. Función imprimir

La función permite imprimir la imagen que se tiene previamente cargada en una ventana grafica oculta.

Gráfico 3.17 Diagrama de flujo de la función imprimir



Para ello se configura la orientación de la hoja en horizontal, y se carga el nombre de la impresora.

A continuación, se realiza una búsqueda de todas las impresoras conectadas al equipo y se comprueba si coincide con el nombre de la impresora preseleccionada.

En el caso de que coincida se realiza la impresión, caso contrario no.

La programación correspondiente a esta función se encuentra en el anexo 4.19

3.8.2.16. Programa de verificación de parámetros iniciales.

La finalidad de este programa es la de verificar que se encuentre en buen estado la luminaria inferior y superior, y encontrar el factor de conversión exacta de [pixel] a [cm] para usos posteriores.

Para llamarlo se lo hace con el nombre “verificaciondeparametrosiniciales”, de esta forma se llama a la función .m y su respectiva Gui.

La Gui consta de:

- Un botón de inicio llamado VERIFICAR
- Dos Axes que contienen las imágenes del proceso y el logo de la universidad
- Un static text, para texto de indicaciones
- Cuatro Check List para la verificación visual de los siguientes parámetros: eliminación de distorsión, iluminación inferior en buen estado, iluminación superior en buen estado, área del objetivo.



Figura 3.37 Interfaz de usuario de programa verificación de parámetros iniciales

La programación correspondiente a esta función se encuentra en el anexo 4.24

3.8.2.16.1. Rutina verificaciondeparametrosiniciales OpeningFcn.

Esta rutina se inicializa justo después de que se llama al programa verificaciondeparametrosiniciales.m; aquí se crean las carpetas que contienen los datos e imágenes de las hojas de cuero en el caso de que no se hayan creado antes y cargan los valores en: Axes, Check Box y Static Text.

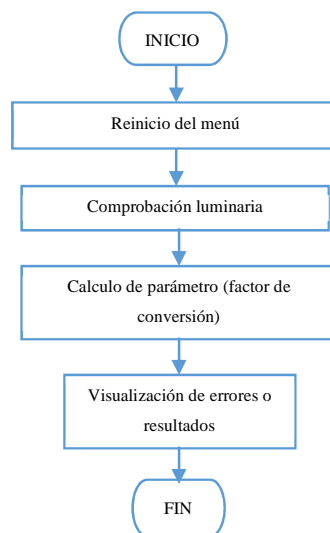
3.8.2.16.2. Rutina figure1_CloseRequestFcn.

Esta rutina se activa al intenta cerrar la Gui actual, esta acción interrumpe la adquisición de imágenes en la variable vid, inicializa una ventana de pregunta, apaga la luminaria de todo el sistema y cierra la Gui.

3.8.2.16.3. Rutina botonverificar_Callback.

La rutina es ejecutada al presionar el botón “Verificar”.

Gráfico 3.18 Diagrama de flujo del botón verificar



En ella se inicializan los axes y ckeck box y static text al igual que en la rutina verificaciondeparametrosiniciales_OpeningFcn. Inmediatamente se convoca una ventana de pregunta con el texto “Existe algún objeto sobre la mesa de trabajo?”, si la respuesta es “NO” sigue con el programa.

Seguido a esto llama a la función encenderluzinferior, encenderluzsuperior y video; a continuación, ingresa a un ciclo en el que se adquiere una imagen para luego procesarla.

De esta imagen obtenida se obtienen parámetros como la condición de luminaria inferior y superior llamando a la función rn_comprobacionluminaria. El resultado se muestra en los axes.

A continuación, se describen los errores obtenidos en el caso de que existan, por ejemplo: fallos en la luminaria inferior o superior, fallos en ambas luminarias, no se puede corregir la distorsión.

En el caso de no haber fallos se llama una ventana de pregunta con el texto: “¿Se encuentra la pieza de calibración en la mesa de trabajo?”. En el caso de que sea correcto se obtiene una nueva imagen, se llama a la función `calculodeparametro` y se muestran los resultados.

A este código adiciona un control de seguridad, este funciona cargando una variable con el valor 1 cuando se realice correctamente la verificación de parámetros iniciales y carga el valor de 0 cuando se cierre la ventana “principal”.

Todo esto ayuda a disminuir el riesgo de una toma de área equivocada y una clasificación mal ejecutada por falla en la iluminación. Con este control se obliga al usuario a realizar el control de parámetros antes de iniciar la ventana `clase_area`.

3.8.2.17. Programa `clase_area`.

Este programa tiene como finalidad entregar los resultados tanto de la clasificación, como de la medida del área.

Para llamarlo se lo hace con el nombre “`clase_area`”, de esta forma se llama a la función `.m` y su respectiva `Gui`.

La `Gui` consta de:

- Botón para proceso de medición
- Botón para proceso de clasificación
- Botón para guardar los datos e impresión
- Button group para agrupar botones
- Axes 1 contiene las imágenes de los resultados y el logo de la empresa
- Axes 2 contiene las imágenes del proceso y el logo de la universidad
- Axes 3 contiene imagen de indicaciones
- Static text para título
- Static text para texto de indicaciones
- Static text para resultado de medición de área en centímetros cuadrados
- Static text para resultado de medición de área en pies cuadrados
- Static text para resultado de clasificación
- Panel para agrupar Static text

- Dos Check List para la verificación visual de los siguientes parámetros: medida de área, clasificación por fallas.



Figura 3.38 Interfaz de usuario de programa clase_area

La programación correspondiente a esta función se encuentra en el anexo 4.22

3.8.2.17.1. Rutina clase_area OpeningFcn.

Al igual que en el programa anterior esta función se activa al iniciar la guide, se crean las carpetas que contienen los datos e imágenes de las hojas de cuero en el caso de que no se hayan creado antes y cargan los valores en: Axes, Check Box y Static Text.

Esta vez se añade la creación de un archivo en Excel el cual tiene en su primera hoja una frase de bienvenida “BIENVENIDO AL SISTEMA DE VISIÓN ARTIFICIAL PARA EL CONTROL DE CALIDAD EN HOJAS DE CUERO”.

A modo de seguridad se tiene precargado un valor en la variable controlcalibracion, esta indica si se ha realizado previamente la verificación de parámetros, en el caso de que no lo haya hecho se abre una ventana de pregunta con el texto “Desea continuar sin calibrar el sistema?” si la respuesta es afirmativa puede obtener los datos del área y la clase pero al momento de guardar sale marcado con la letra “N” indicando que no son los datos reales; caso contrario si la respuesta es negativa al momento de presionar

cualquier botón cierra la ventana actual y abre la ventana de “verificaciondeparametrosiniciales”.

3.8.2.17.2. Rutina botonguardar Callback.

Esta rutina tiene como fin guardar e imprimir una imagen de la pieza ya sea clasificada y medida el área o solo una de las dos. La imagen ira acompañada de un código el cual es el nombre de la pieza.

Gráfico 3.19 Diagrama de flujo del botón guardar

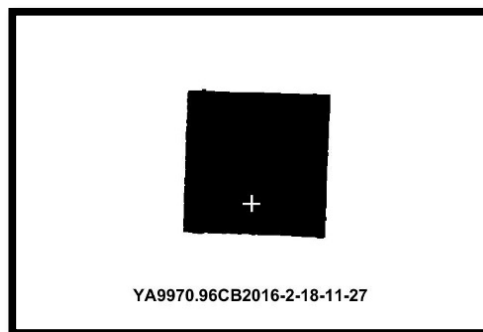
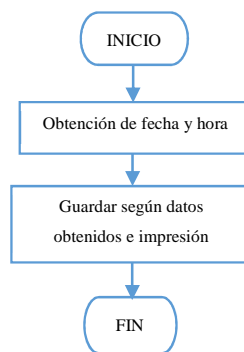


Figura 3.39 Imagen representativa al presionar boton guardar

Como se puede observar el código se encuentra en la parte inferior de la imagen. En este código podemos ver las características que tiene la pieza, para mejor comprensión se explica a continuación:

- Primera letra: “Y” nos indica que fue previamente calibrada, por lo que el resultado va a ser preciso, si se encuentra la letra “N” tendríamos un resultado no confiable.
- Segunda letra: “A” indica que el siguiente valor corresponde al área de la pieza.
- Primera cifra: “9970.96” medida del área de la pieza, se visualiza el valor 0 en el caso de no haberse analizado el área.

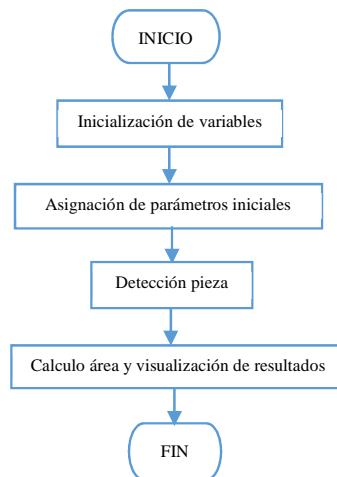
- Tercera letra: “C” indica que el siguiente valor corresponde a la clase de la pieza.
- Cuarta letra: “B” indica la clase a la que pertenece la pieza, esta letra puede ser “A”, “B”, ‘C’.
- Segunda cifra: “2016-2-18” indica la fecha en la que fue guardada la imagen
- Tercera cifra: “11-27” indica la hora a la que fue guardada la imagen

Además de guardar la imagen, se archiva el código en un registro en Excel. El cual tiene como nombre “basedatosexcel.xlsx”, a este archivo se crea una hoja nueva correspondiente al día en el que se guarde los datos.

3.8.2.17.3. Rutina botonmedirarea Callback.

Esta rutina se activa al presionar el botón “Medir Área”. Al finalizar el proceso entrega el valor del área de la pieza, este valor se muestra tanto en $[cm^2]$ como en $[ft^2]$ en la ventana actual.

Gráfico 3.20 Diagrama de flujo del botón medir área



Previamente se tiene cargado un valor en la variable “ventanaemergente”, si tiene el valor de 1 quiere decir que se desea calibrar el sistema antes de proseguir por lo que inmediatamente cierra la ventana actual y abre la ventana para verificación de parámetros.

En el caso de que ya se tenga calibrado el sistema seguiría a la obtención del área. Para ello primero manda a apagar la luminaria superior y elimina cualquier información que se encuentre en los check box y static text. Como recordatorio se abre una ventana de pregunta indicando que se retire cualquier objeto que se encuentre sobre la pieza.

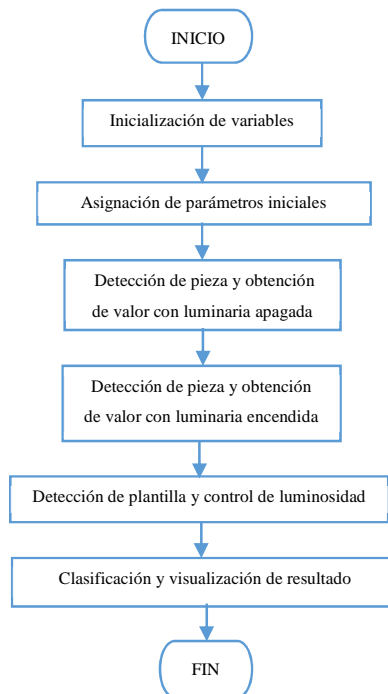
Si todo está bien manda a inicializar la toma de imágenes y adquiere una imagen representativa. A esta imagen se la procesa usando la función `deteccionpieza` y `calculoarea`, explicadas anteriormente. Se muestra la imagen resultante en el axes principal, además se muestra el área en $[cm^2]$ y $[ft^2]$ y se carga la variable dato área con el valor 1, indicando que se ha obtenido el valor del área.

3.8.2.17.4. Rutina botonclasificacionfallas Callback.

La rutina se activa al presionar el botón “CLASIFICACIÓN”, esta entrega la categoría a la que ha sido asignada la pieza de acuerdo a las fallas que este posee.

Se considera la misma restricción que el botón “MEDIR ÁREA” en la que, si previamente se seleccionó que se desea calibrar antes de continuar, al presionar este botón se abre la ventana de verificación de parámetros iniciales. Así mismo se inicializa las variables necesarias, los axes, check box, static text. Y se enciende la luminaria superior para que el usuario puede observar los lugares que existen fallas.

Gráfico 3.21 Diagrama de flujo de la función control luminosidad superior



Una vez inicializado se abre una ventana de pregunta para verificar si se han colocado las plantillas sobre las fallas del cuero. Si la respuesta es afirmativa apaga la luminaria superior y activa la adquisición de imágenes.

Posteriormente se adquieren las imágenes necesarias para obtener la clasificación, a estas imágenes se las separo en tres etapas:

- **Primera etapa:** detección de pieza y obtención de valor de un pixel con luminaria apagada.
- **Segunda etapa:** detección de pieza y obtención de valor de un pixel con luminaria encendida
- **Tercera etapa:** detección de plantilla y control de luminosidad.

3.8.2.17.5. Rutina figure1 CloseRequestFcn.

Esta función se activa al cerrar la ventana, ya sea al presionar el botón “X” o al presionar cualquier botón si se seleccionó la opción de calibrar el sistema antes de analizar.

Al igual que los otros botones, primero se verifica si se seleccionó previamente la opción de calibrar, si es así apaga las luminarias superior e inferior y cierra la ventana sin preguntar si se desea salir.

En el caso de que no se haya seleccionado esa opción el programa se ejecuta normalmente. En este caso para cerrar la ventana se debe presionar “X”, al hacerlo se abre una ventana preguntando si se desea salir del menú de clasificación, si es así apaga luces y cierra la ventana

3.8.2.18. Programa base de datos.

Este programa tiene como finalidad mostrar la imagen que le corresponde a un código pre guardado.

Para llamarlo se lo hace con el nombre “basededatos”, de esta forma se llama a la función .m y su respectiva Gui.

La Gui consta de:

- Axes 1 contiene las imágenes de los resultados y el logo de la empresa
- Axes 2 contiene el logo de la universidad

- Static text para titulo
- Static text para mostrar código erróneo
- Edit text para ingreso del código
- Botón para imprimir



Figura 3.40 Interfaz de usuario de programa basededatos

La programación correspondiente a esta función se encuentra en el anexo 4.21

3.8.2.18.1. Rutina basededatos OpeningFcn.

Esta rutina se inicializa justo después de que se llama al programa basededatos.m; aquí se crean las carpetas que contiene los datos e imágenes de las hojas de cuero en el caso de que no se hayan creado antes y cargan los valores en: Axes, edit Text y static text.

3.8.2.18.2. Rutina edit1 Callback.

Esta rutina se activa luego de ingresar un código y presionar ENTER.

Una vez ingresado el código, se comprueba que exista la imagen correspondiente; en el caso de no existir, se visualiza en un static text el código ingresado, al mismo tiempo se muestra en el edit text la frase “Ingrese correctamente el código”.

3.8.2.18.3. Rutina botón imprimir Callback.

Esta rutina se inicializa al presionar el botón Imprimir.

Con la imagen precargada correspondiente al código se tiene dos opciones, si la imagen se encuentra almacenada en la dirección asignada o no.

En el caso de no encontrarse se muestra la frase “Imagen a imprimir no existente” y se muestra la imagen de la empresa.

Caso contrario se abre una ventana grafica oculta con la imagen, se configura el tamaño del papel en el que se va a imprimir, se llama a la función imprimir y se cierra la ventana oculta.

3.8.2.19. *Programa principal.*

Este programa tiene como finalidad abrir cada programa del sistema de clasificación por separado.

Para llamarlo se lo hace con el nombre “principal”, de esta forma se llama a la función .m y su respectiva Gui.

La Gui consta de:

- Botón para programa base de datos
- Botón para programa verificación de parámetros iniciales
- Botón para programa clase_area
- Axes 1 contiene el logo de la empresa
- Axes 2 contiene el logo de la universidad
- Static text para titulo
- Static text para texto de nota
- Static text para nombre de autor



Figura 3.41 Interfaz de usuario de programa principal

Al presionar cada botón se dirige directamente al programa asignado. Al presionar el botón de salir, se apagan las luminarias tanto superior como inferior.

CAPITULO IV

4. RESULTADOS

4.1. INTRODUCCIÓN

En el capítulo, se refiere a la construcción e implementación del sistema de control de calidad para hojas de cuero, así también materiales, herramientas y recursos utilizados para su ejecución, diagramas de instalación, pruebas y ajustes a los que debe ser sometido el sistema.

4.2. IMPLEMENTACIÓN DEL SISTEMA DE VISIÓN ARTIFICIAL PARA EL CONTROL DE CALIDAD DE HOJAS DE CUERO

4.2.1. Diagrama de construcción

El sistema de clasificación consta de varios módulos para su funcionamiento los cuales se subdividen del sistema, donde cada uno de ellos tiene sus componentes y dispositivos que se agrupan para realizar el trabajo de control de calidad.

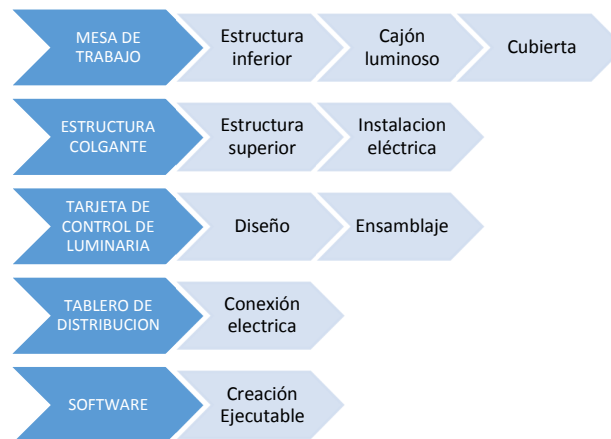


Figura 4.1 Diagrama de construcción

4.2.2. Funcionamiento

El sistema de control de calidad tiene como objetivo clasificar y medir la hoja de cuero terminada de cualquier color uniforme.

Al encender el sistema el operador debe accionar uno de los interruptores y colocarlo en automático. Para que de esta forma el sistema prenda, apague o regule la luminaria.

Una vez seleccionado el modo automático el operador inicializa la interfaz y procede a calibrar el sistema. Con el sistema calibrado a puede clasificar o medir la hoja de cuero, pulsando el botón Clasificación, Medición. El sistema pide los requisitos según la acción que se desea realizar.

La clasificación se hace por medio de plantillas colocadas por el usuario sobre las fallas, y se ejecuta al presionar el botón “Clasificación”. La medida del área igualmente se ejecutaría al presionarse el botón “Medir Área”.

Al presionar el botón “Guardar”, se almacenan los datos en una hoja de Excel y la imagen en una carpeta aparte localizada en la unidad C del computador, inmediatamente imprime la imagen.

El cuero debe ser colocado con la banda del lomo en la dirección que se indica en cubierta de la mesa, y no sobrepasar de los bordes delineados para colocarlo.

Así mismo el operador puede encender la luminaria sin necesidad de que el sistema esté en funcionamiento, colocando el selector en manual.

4.2.3. Construcción de la mesa de trabajo

Esta estructura contiene los elementos que soportan a la luminaria inferior y a la pieza de cuero.

4.2.4. Construcción de la estructura inferior

La estructura soporta las cargas producidas por los elementos que complementan a la estructura inferior como: la caja luminosa, la cubierta, el tablero de distribución, cableado, lámparas y en caso de mantenimiento cargas humanas.

4.2.4.1. Materiales

La estructura se construye como se seleccionó en el capítulo 3 de un solo tipo de material, pero con diferentes medidas, para de esta forma mantener uniformidad.

Tabla 4.1 Material estructura

MATERIAL	TIPO	FORMA	DIMENSIONES
ASTM A500	Tubería estructural	Tubo cuadrado	40x40 [mm] e= 2 [mm]
ASTM A500	Tubería estructural	Tubo cuadrado	25x25 [mm] e= 2 [mm]
ASTM A500	Tubería estructural	Tubo rectangular	40x60 [mm] e= 3 [mm]
ASTM A500	Tubería estructural	Tubo rectangular	25x40 [mm] e= 2 [mm]

4.2.4.2. Construcción

La tubería estructural se la consigue de fábrica en piezas de 6 [m] por lo cual para la construcción se las corto con las medidas establecidas en el anexo 7.12, el corte se lo hace con cierra manual de hierro y con escuadra; los cortes son a 45° de esta forma se facilita el ensamblaje y se disimulan los huecos propios de los tubos.

Para ocultar la soldadura en las uniones se utilizó masilla, una vez seca se pintó como base de fondo plomo para que posteriormente reciba el acabado con laca en color blanco en la parte superior y el resto de color negro, el proceso de construcción se puede visualizar en el anexo 11.1.

4.2.5. Construcción del cajón luminoso

El cajón debe reflejar los rayos luminosos hacia el exterior por una sola de sus caras, a su vez debe dar cabida a las lámparas para la luminaria inferior y facilitar el mantenimiento.

4.2.5.1. Materiales

El cajón se construye como se diseñó en el capítulo 3, su cuerpo es de madera MDF contrachapada de color blanco con un espesor de 15 [mm]. Se describe a continuación los materiales que conforman el cajón.

Tabla 4.2 Material cajón luminoso

MATERIAL	Dimensiones	Cantidad
Tornillo negro de Mdf	8 x 1"	6 [u]
Tornillo negro de Mdf	6 x ½"	12 [u]
Tornillo Colepato	6 x ½"	12 [u]
MDF	250 x 190 x 1.5 [cm]	1 [u]
Bisagra reforzada	1, ½"	6 [u]
Lámpara 40 W	F40WT12	4 [u]
Lámpara 40 W	F40WT10	2 [u]
Zócalo para lámpara	-	12 [u]
Broca	1/8"	1 [u]
Broca	1/2"	1 [u]
Cable solido rojo	Nº14	14 [m]
Cable solido amarillo	Nº14	14 [m]
Cable solido azul	Nº14	14 [m]
Manguera de luz	½"	8.5 [m]
Canto duro de PVC	15 [mm]	2,616 [m]
Silicona blanca	-	1

4.2.5.2. Construcción

La madera MDF se la encuentra de fábrica en planchas de 250 x 190 x 1.5 [cm], a esta se la recortó en la propia fábrica en 2 segmentos de 9.5 x 250 [cm], 2 segmentos de 9.5 x 174.5 [cm], 1 segmento de 250 x 171 [cm]; a cada uno de estos se le coloco canto duro en todo su perfil, de esta forma se tiene una terminación estéticamente mejorada.

Se atornilla los segmentos de 9.5 x 174.5 [cm] en los laterales cortos de plancha de 250 x 171 [cm] formando una caja. Los 2 segmentos restantes se atornillan directamente en los laterales de la estructura principal, de esta forma queda la caja formada.

Posterior a esto se ubican los zócalos para las lámparas tubulares, sujetándolas a presión mediante tornillos y se hace una abertura con broca delante de cada zócalo para el paso del cableado, cada terminal se conecta de acuerdo al plano de conexiones que se encuentra en el anexo 3. Para disimular el cableado se lo introduce en manguera de luz. En la parte superior se disimula el cableado que conecta a los zócalos con silicona blanca.

Todo cable tiene numeración en cada uno de sus terminales, esto con el fin de facilitar su ubicación. La unión del cableado de las lámparas con el tablero de distribución se la realiza mediante borneras colocadas debajo de la caja luminosa, el proceso de construcción se puede visualizar en el anexo 11.2.

4.2.6. Instalación de la cubierta

La cubierta debe dejar pasar la mayor cantidad de haz de luz, para que la cámara logre distinguir fácilmente la pieza del entorno.

4.2.6.1. Materiales

Como se mencionó en el capítulo 2 la cubierta es de vidrio templado de 10 [mm] de espesor, se optó por separar en 3 segmentos para facilitar el mantenimiento.

Tabla 4.3 Material cubierta

MATERIAL	Dimensiones	Cantidad
Vidrio templado	1000 x 2225 x 10 [mm]	3 [u]
Silicona transparente	-	2 [u]

Para suavizar el apoyo sobre los que va a descansar el vidrio se coloca una película de silicona en las vigas de la mesa, de forma homogénea, una vez seca se coloca el vidrio. En la unión de vidrio con vidrio se coloca silicona para evitar que pasen partículas de polvo hacia la caja luminosa.

4.2.7. Construcción de la estructura colgante

Esta estructura contiene la luminaria superior y la cámara.

4.2.7.1. Construcción de la estructura superior

La estructura contiene en su interior el cableado para la luminaria y la cámara para visión artificial.

4.2.7.1.1. Materiales

La estructura se construye como se describió en el capítulo 3 de un solo tipo de material, para de esta forma mantener uniformidad. La estructura se mantiene colgante al igual que una lámpara de techo, con una separación de 1.20 [m] de la superficie superior de la cubierta.

Tabla 4.4 Material estructura superior

MATERIAL	Dimensiones	Cantidad
Tubo estructural cuadrado	38.1 x 38.1 x 15 [mm]	12 [m]
Perfil angular	31.75 x 31.75 [mm]	30 [cm]
Acople para tubo cuadrado	35 x 35 [mm]	16 [u]
Bandola con seguro	1/8" x 1, 1/2'	4 [u]
Remache pop	5/32" x 1/2'	16 [u]
Tornillo Colepato	8 x 1/2"	76 [u]
Broca para metal	5/32"	1 [u]
Broca para metal	3/32"	1 [u]
Broca para metal	1/4"	1 [u]
Mandril con adaptador para taladro	1, 1/4"	1 [u]

4.2.7.1.2. Construcción

La tubería estructural se la consigue de fábrica en piezas de 6 [m] por lo cual para la construcción se las corto con las medidas establecidas en los anexos 7.6, 7.7 y 7.8; el corte se lo hace con cierra circular y con escuadra. Después de cada corte se lima los bordes para eliminar viruta.

Se abre orificios para el paso de remaches y tornillos de acuerdo a las especificaciones del anexo 7.5, 7.5, 7.7 en el tubo cuadrangular que forma en cuadro exterior de la estructura. A continuación, se corta el perfil angular a una distancia de 35 [mm] y se lima, obteniendo 8 piezas las cuales sirven de refuerzo y unión en los vértices del cuadro exterior.

Así mismo se perfora de acuerdo a las especificaciones y se comprueba con los huecos hechos en el cuadro exterior. Una vez comprobado se remacha quedando unidos de forma definitiva a los laterales cortos del cuadro exterior.

A los laterales largos se avellana los huecos con broca de ¼” para estilizar el acabado. Estos se unen al cuadro exterior con tornillo de 8 x ½”.

El resto de la estructura ira anclada a al cuadro exterior por medio de anclajes para tubo cuadrado, para ello se toma las medidas de su ubicación del anexo 7.4, 7.5, 7.6 y 7.7, con este situado se toma de guía para perforar los huecos para los tornillos y para el paso de cables. Del mismo modo se sitúa y se toma de guía los zócalos para las lámparas halógenas. El hueco correspondiente para el lente de la cámara se lo realiza con el mandril con adaptación para el taladro según las especificaciones del anexo 7.4. La unión de la estructura se la realiza una vez conectado todo el cableado, el proceso de construcción se puede visualizar en el anexo 11.3.

4.2.7.2. Instalación eléctrica

La conexión eléctrica se realiza en el interior de la estructura, tanto para la luminaria como para la cámara.

4.2.7.2.1. Materiales

Con todas las aberturas realizadas, se puede poner cada elemento en su lugar.

Tabla 4.5 Material instalación eléctrica

MATERIAL	Dimensiones	Cantidad
Cámara	150 x 33 x 30 [mm]	1
Zócalos mr16	-	14
Transformador	92 x 34 x 20 [mm]	5
Cable gemelo	n° 18	8 [m]
Bornera	5/32” x ½”	10
Tonillo de madera	4 x 3/4”	28

Para ello primero se coloca la cámara en su posición, con mucho cuidado de no rayar el lente, seguido se coloca el cableado por dentro del tubo central de la estructura. Se colocan los zócalos pasando suavemente los cables por los huecos asignados, una vez que estén bien alineados se ajusta con los tornillos. Se pasa el cable del zócalo por el agujero situado en el medio del acople cuadrado; a este se lo une con el cable gemelo, de acuerdo al diagrama de conexiones situado en el anexo 3, mediante borneras. A continuación, se une toda la estructura, metiendo los tubos pequeños en sus respectivos acoples y atornillando el marco exterior de la estructura.

Por último, se coloca las lámparas halógenas y se asegura las bandolas, el proceso de instalación se puede visualizar en el anexo 11.4.

4.2.8. Construcción de la tarjeta de control de luminaria

Esta tarjeta contiene los elementos que permiten controlar el encendido, apagado y regular la luminaria.

4.2.8.1. Diseño de la tarjeta

La tarjeta unifica los módulos explicados en el capítulo 3. Para reducir el tamaño de la misma se realiza el grabado de las pistas en ambas caras.

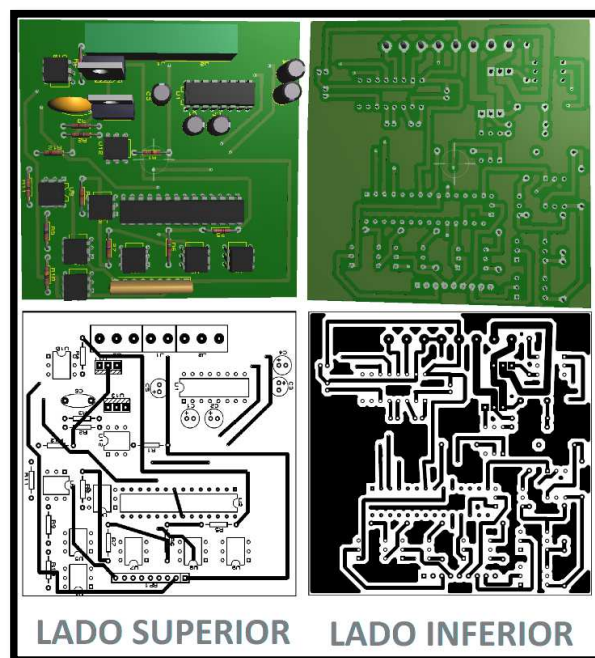


Figura 4.2 Diseño de la tarjeta.

El montar los elementos en un solo lado facilita el proceso de soldadura y mejora la presentación. Los circuitos impresos se encuentran en el anexo 2.

4.2.8.2. *Ensamblaje de la tarjeta*

Una vez diseñada la posición de cada elemento en la tarjeta se imprime el diseño en papel fotográfico, para luego traspasarla a la baquelita por medio de calor.

4.2.8.2.1. Material

Tabla 4.6 Material tarjeta de control de luminaria.

MATERIAL	Cantidad
Zócalo para CI de 8 pines	9 [u]
Zócalo para CI de 28 pines	1 [u]
Zócalo para CI de 16 pines	1 [u]
MOC 3010	9 [u]
Atmega88PA	1 [u]
Max 232	1 [u]
Resistencia 21K Ω	1 [u]
Resistencia 10K Ω	1 [u]
Resistencia 5.1K Ω	2 [u]
Resistencia 2.7K Ω	1 [u]
Resistencia 1.2K Ω	1 [u]
Resistencia 680 Ω	1 [u]
Resistencia 330 Ω	2 [u]
Resistencia 180 Ω	2 [u]
Resistencia 2K Ω	1 [u]
Paquete de 8 resistencias 330 Ω	1 [u]
BTA-12	1 [u]
BTA-10	1 [u]
Capacitor electrolítico 10uF	5 [u]
Capacitor cerámico 100nF	1 [u]
Terminal para PCB C/tornillo 2 vías	1 [u]
Terminal para PCB C/tornillo 3 vías	2 [u]
Broca 1/32	1 [u]
Baquelita de fibra de vidrio doble cara	1 [u]
Acido férrico	3 [u]
Alambre de estaño	2 [m]

4.2.8.2.2. Ensamble

Una vez pasadas las pistas de la tarjeta se la introduce al ácido férrico, para que se elimine el exceso de cobre sobre la baquelita. Como resultado se obtiene una placa con las pistas de cobre, comprobamos que estas conduzcan correctamente y se realizan los respectivos huecos para que entren los pines de los elementos. Se sitúa todos los elementos en sus lugares verificando que todos embonen.

Antes de soldarlos a la placa se da una capa de pintura para protegerla de la corrosión. Por último, se suelda cada elemento, el proceso de ensamble se puede visualizar en el anexo 11.5.

4.2.9. Construcción del tablero de distribución

El tablero permite visualizar y seleccionar al operador el estado de cada uno de los circuitos.

4.2.9.1. Ensamble del tablero

Con los circuitos ensamblados para la luminaria superior e inferior, se puede identificar que elementos van dentro del tablero, los elementos que van en el exterior se dio a conocer en el capítulo 3.

4.2.9.1.1. Materiales

Tabla 4.7 Material tablero de control.

MATERIAL	Cantidad
Caja térmica 30 x 40 x16 [cm]	1 [u]
Luz piloto verde	2 [u]
Luz piloto blanca	4 [u]
Conector DB9 hembra	1 [u]
Selector 3 posiciones	1 [u]
Selector 2 posiciones	1 [u]
Dimmer	1 [u]
Toma corriente de 2 entradas	1 [u]
Balastro 2 x 40W	3 [u]
Interruptor diferencial	1 [u]
Interruptor termoelectrico	1 [u]
Fuente 5Vdc	1 [u]
Tarjeta de control de luminaria	2 [u]
Canaleta para cableado	1 [m]
Carril DIN	0.5 [m]
Cable de hilo verde N° 14	1 [m]
Cable de hilo plomo N°14	3 [m]
Cable de hilo azul N°14	3 [m]
Cable de audio	1 [m]
Cable 3 vías N°16	5 [m]
Bornera	2[u]
Terminal plano para cable	50 [u]

Como referencia se establece que el cable de color verde es para tierra, azul es para neutro y plomo para la fase.

4.2.9.1.2. Ensamble

Una vez situado todos los elementos que van dentro del tablero, se atornillan en la placa base de color naranja. Así mismo se realizan los respectivos huecos en la tapa del tablero de acuerdo a los planos en el anexo 7.13.

Posterior se cablea de acuerdo al anexo 3, cada cable tiene una numeración guía en cada extremo, así como sus terminales.

Una vez cableado se comprueba continuidad y se realiza las respectivas pruebas de funcionalidad.

4.2.10. Software

Una vez diseñado y programado la interfaz del usuario que tiene el software se obtiene un ejecutable, el cual puede ser instalado en cualquier ordenador. Este ejecutable disminuye el riesgo de modificar o alterar las líneas de código del programa

4.2.10.1. Obtención del ejecutable

Una vez verificado el código del software, se puede crear el ejecutable a partir de la extensión de matlab “Application Compiler”, localizada en “Application Deployment”.

Seleccionamos la toolbox y se abre un menú, el cual se llena con los datos del autor, descripción del programa, se cargan todos los archivos que tienen relación con el programa. Se incluye las bibliotecas compartida para la ejecución de aplicaciones de Matlab, de este modo no es necesario tener instalado previamente matlab en el ordenador en el cual se instale este ejecutable.

Figura 4.3 Carga de datos y archivos para ejecutable

Con todos los campos llenos creamos el ejecutable, presionando en el botón package. Como resultado se tiene una carpeta con tres carpetas en su interior. Una contiene el instalador de la biblioteca de Matlab, otra los archivos del ejecutable y la ultima un ejecutable para simulaciones y prueba.

4.3. PRUEBA DE FUNCIONAMIENTO DE LA TARJETA DE CONTROL DE LA LUMINARIA

Para la realización de esta prueba se procede a activar desde la tarjeta de control cada una de las salidas. Las cuales comandan el funcionamiento de la luminaria inferior y superior.

Los resultados obtenidos se detallan en la siguiente tabla.

Tabla 4.8 Resultados de prueba de funcionamiento en la tarjeta de control de luminosidad.

COMANDO ENVIADO POR MATLAB	NOMBRE DE BORNERA	RESULTADOS	
		VOLTAJE DE SALIDA	ESTADO
1	Dim Out	110 [V~]	Normal
2	Dim Out	109 [V~]	Normal
3	Dim Out	118.4[V~]	Normal
4	Dim Out	0.8 [V~]	Normal
5	ON OFF Out	118.4 [V~]	Normal
6	ON OFF Out	24.5[V~]	Normal
	110 V~ In	118.4[V~]	Normal
	5V	5.46 [V]	Normal
	GND	0 [V]	Normal
	TX	0 [V]	Normal
	RX	5.28 [V]	Normal
	GND	0 [V]	Normal

4.4. PRUEBA DE NIVEL DE ILUMINACIÓN DE LA LUMINARIA SUPERIOR

Esta prueba permite constatar que la cantidad de luxes reflejado sobre las hojas de cuero sea el adecuado tanto para el operario como para la cámara.

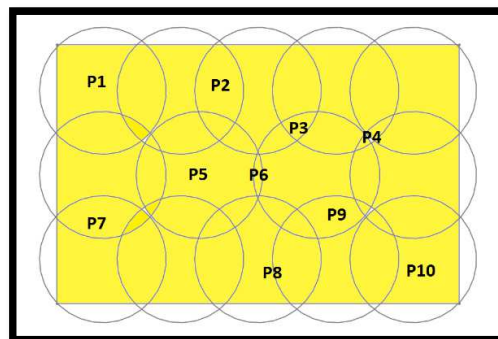


Figura 4.4 Ubicación para la prueba del nivel de luminosidad

Los ensayos se realizaron con el luxómetro de marca Testo 545, en los puntos designados en la figura 4.4, con la luminaria inferior apagada, la luminaria superior encendida, y en condiciones de nivel de luminosidad natural 0, obteniendo la tabla 4.9.

Tabla 4.9 Nivel de iluminación en la luminaria superior

# Prueba	Nivel de iluminación de luz natural +artificial (Luxes)
1	592
2	596
3	591
4	592
5	590
6	588
7	590
8	591
9	589
10	592
Promedio=	591.1

La prueba revela que la luminaria superior emite en promedio 591.1 luxes a una altura de 120 cm, estando dentro del rango mínimo que propone el Ministerio de Trabajo que es de 500 luxes.

4.5. PRUEBA DE FUNCIONAMIENTO DE LA CÁMARA

La verificación del funcionamiento de la cámara se realiza mediante el uso toolbox Imagen Adquisición de Matlab. Dentro de esta se puede encontrar los nombres y el tipo de calidad de la imagen de las cámaras conectadas al ordenador.

Verificamos que se encuentre el nombre y la calidad de la cámara pre programada en el software, para evitar errores posteriores.

A continuación, se confirma que este enfocada, esto mediante el movimiento del ajuste manual que se encuentra en el lente de la misma.

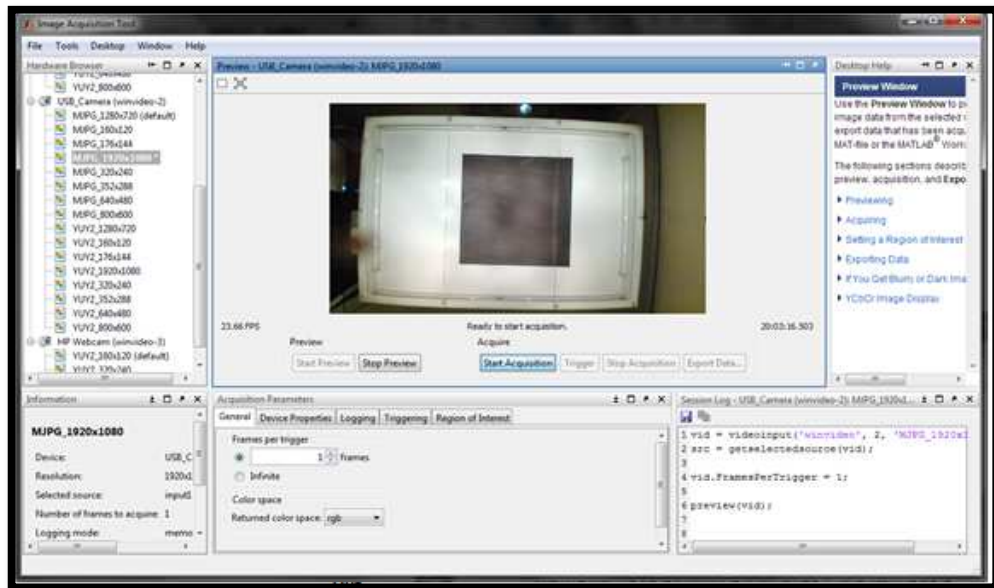


Figura 4.5 Prueba funcionamiento de la cámara

4.6. PRUEBA DEL SOFTWARE

Para realizar las pruebas del software se unió cada uno de los módulos tanto estructurales como eléctricos, además del ordenador, la cámara y la impresora.

Las pruebas del sistema se muestran en forma secuencial.

4.6.1. Programa principal

Al programa principal se realiza prueba de comportamiento de acuerdo a cada botón.

Tabla 4.10 Prueba de comportamiento del programa principal

Acción realizada	Consecuencia	Estado
Presionar botón base de datos	Abre ventana basededatos	Correcto
Presionar botón calibración del sistema	Abre ventana verificaciondeparametrosiniciales	Correcto
Presionar botón clasificación medición	Abre ventana clase_area	Correcto

4.6.2. Programa basededatos

En el programa que contiene la base de datos, se realizan simulaciones de acciones equivocadas que puede realizar el usuario, para comprobar su comportamiento.

Tabla 4.11 Prueba de comportamiento del programa basededatos

Proceso	Acción realizada	Consecuencia	Estado
Búsqueda de imagen por código	Ingreso de código incorrecto	Muestra texto: “Ingrese correctamente el código”, “Usted ingreso: *****”	Correcto
	Ingreso de código correcto	Muestra texto: “Código correcto” y la imagen correspondiente al código.	Correcto
Impresión de imagen	Presionar botón sin ingresado código.	Muestra texto: “Imagen a imprimir no existente” “Usted ingreso:”	Correcto
	Presionar botón después de haber ingresado código incorrecto.	Muestra texto: “Imagen a imprimir no existente” “Usted ingreso: *****”	Correcto
	Presionar botón después de haber ingresado código correcto.	Envía a imprimir imagen	Correcto

4.6.3. Programa verificaciondeparametrosiniciales

Las pruebas que se realizan al programa de verificación se las divide en dos segmentos, el primero comprobando el accionamiento de la luminaria de acuerdo al proceso que se encuentra realizando, el segundo segmento corresponde a errores simulados que puedan suscitarse luego de haber presionado el botón llamado “Verificar”.

Tabla 4.12 Prueba de comportamiento de luminaria en el programa verificaciondeparametrosiniciales

Proceso	Localización luminaria	Activación	Estado
Verificación luminaria	Luminaria inferior	Si	Correcto
	Luminaria superior	Si	Correcto
Calculo de parametros	Luminaria inferior	Si	Correcto
	Luminaria superior	No	Correcto

Tabla 4.13 Prueba de comportamiento del programa verificaciondeparametrosiniciales

Proceso	Error premeditado	Consecuencia	Estado
Verificación luminaria	Extracción de una lámpara en luminaria superior	Abre ventana con texto: “Fallo de iluminación superior, verificar circuito”	Correcto
	Extracción de una lámpara en la luminaria inferior	Abre ventana con texto: “Fallo de iluminación inferior, verificar circuito”	Correcto

Calculo de parámetros	Interrupción de brazo sobre la cubierta de la mesa.	Abre ventana con texto: “Fallo de iluminación inferior, verificar circuito”	Correcto
	Pieza de calibración sobre la cubierta	Abre ventana con texto: “Fallo de iluminación superior e inferior, verificar circuitos”	Correcto
	Sin error premeditado	Prosigue con el proceso	Correcto
	Interrupción de brazo sobre la pieza de calibración.	Recalcula parámetros	Correcto
	No presencia de la pieza de calibración	Muestra texto: “Proceso concluido sin éxito”	Correcto
	Luminaria inferior apagada	Muestra texto: “Proceso concluido sin éxito”	Correcto
	Sin error premeditado	Entrega de resultados	Correcto

4.6.4. Programa clase_area

EL programa clase_area posee 5 procesos, estos son: ingreso al programa, medición de área, clasificación por fallas, guardar resultados, salir del programa; cada uno se activa al presionar su respectivo botón.

Las pruebas que se realizan al programa clase_area se las divide en dos segmentos, el primero comprobando el accionamiento de la luminaria de acuerdo al proceso que se encuentra realizando, el segundo segmento corresponde a errores simulados que puedan suscitarse luego de haber presionado cada botón”.

Tabla 4.14 Prueba de comportamiento de luminaria en el programa clase_area

Proceso	Localización luminaria	Activación	Estado
Ingreso al programa	Luminaria inferior	Si	Correcto
	Luminaria superior	No	Correcto
Medición de Área	Luminaria inferior	Si	Correcto
	Luminaria superior	No	Correcto
Clasificación por fallas	Luminaria inferior	Si	Correcto
	Luminaria superior	Si	Correcto
	Control de cantidad lumínica en luminaria superior	Si	Correcto
Guardar resultados	Luminaria inferior	Si	Correcto
	Luminaria superior	No	Correcto
Salir del programa	Luminaria inferior	No	Correcto
	Luminaria superior	No	Correcto

Tabla 4.15 Prueba de comportamiento del programa clase_area

Proceso	Error premeditado	Consecuencia	Estado
Ingreso al programa	Proseguir sin previamente calibrar el sistema	Guarda con identificador N en código	Correcto
	Seleccionar calibrar el sistema previamente en ventana emergente	Todo botón llama a programa verificaciondeparametrosiniciales	Correcto
	Abrir el programa después de haber calibrado el sistema	Guarda con identificador Y en código	Correcto
Medición de área	Extracción de una lámpara en la luminaria inferior, colocada la pieza de cuero.	Entrega resultado del área medida	Correcto
	Interrupción de brazo sobre la pieza de cuero.	No entrega resultado	Correcto
	Sin pieza de cuero sobre la cubierta	No entrega resultado	Correcto
	Luz inferior apagada	No entrega resultado	Correcto
	Sin error premeditado	Entrega resultado del área medida	Correcto
Clasificación por fallas	Extracción de una lámpara en la luminaria inferior, colocada la pieza de cuero.	Entrega resultado de la clase de la pieza	Correcto
	Extracción de una lámpara en la luminaria superior, colocada la pieza de cuero.	Entrega resultado da la clase de la pieza	Correcto
	Interrupción de brazo sobre la pieza de cuero.	No entrega resultado	Correcto
	Sin pieza de cuero sobre la cubierta	No entrega resultado	Correcto
	Sin plantillas para clasificación	Cuero de clase A	Correcto
	Con plantillas de clasificación	Entrega resultado da la clase de la pieza	Correcto
	Luz inferior desconectada	No entrega resultado	Correcto
	Luz superior desconectada	No entrega resultado	Correcto
	Sin error premeditado	Entrega resultado de la clase de la pieza	Correcto
	Sin resultados del área	Guarda e imprime.	Correcto

Guardar resultados	Sin resultado de la clase	Guarda e imprime.	Correcto
	Sin resultados	No guarda ni imprime	Correcto
	Con resultado de clase y área	Guarda e imprime	Correcto
Salir del programa	Mientras está ejecutándose algún proceso	Interrumpe proceso	Correcto
	Sin proceso en ejecución	Pregunta si se desea cerrar la ventana	Correcto.

La imagen de la etiqueta, se puede visualizar en el anexo 11.6.

Se realiza prueba para constatar precisión de medida y repetitividad de datos. Para ello se realizan dos tipos de pruebas.

4.6.5. Exactitud de medida

Se coloca la pieza de calibración para comprobar la medida, previamente se sabe que esta posee un área de $14400 [cm^2] = 15.5 [ft^2]$, se realizan un total de 10 mediciones en diferentes posiciones y con diferente color de la pieza de calibración.

Se estable la tolerancia calculando un error de $\pm 1\%$ de la medida real. De acuerdo a la siguiente ecuación:

Ecuación 4.1 Calculo tolerancia para medida de una pieza

$$T = ES - EI = 15.655 - 15.345 = 0.31$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

T: Tolerancia; ES: Error superior; Ei: Error inferior

Ecuación 4.2 Calculo promedio de medida

$$\overline{medida} = \frac{\sum_{i=1}^n medida}{n}$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

\overline{medida} : Promedio de medida; $\sum_{i=1}^n medida$: Total de medida; n: cantidad de pruebas

Ecuación 4.3 Calculo % de exactitud

$$\%Exactitud = \frac{|\overline{medida} - Medida\ real|}{Tolerancia} \times 100$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Tabla 4.16 Estudio de exactitud con una pieza patrón

Color pieza de calibración	Prueba N°	Posición	Medida [ft ²]
Negro	1	Esquina superior izquierda	15.35
	2	Esquina superior derecha	15.37
	3	Esquina inferior izquierda	15.36
	4	Esquina inferior derecha	15.35
	5	Centro	15.65
Blanco	6	Esquina superior izquierda	15.36
	7	Esquina superior derecha	15.35
	8	Esquina inferior izquierda	15.36
	9	Esquina inferior derecha	15.35
	10	Centro	15.64
<i>medida</i>			15.414
Valor real			15.5
T			0.31
%Exactitud			27.74

Tabla 4.17 Criterio de aceptación

CRITERIO DE ACEPTACIÓN	
Abajo de 10%	→ Excelente proceso
De 10 a 20%	→ Bueno, aceptable
De 20% a 30%	→ Marginalmente aceptable
Arriba de 30%	→ Inaceptable y debe ser corregido

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

De esta tabla se saca como resultado un porcentaje de exactitud de 27.14%, a este valor se lo compara con la tabla 4.17, demostrando que se encuentra dentro de un rango Marginalmente Aceptable.

4.6.6. Repetitividad de resultados

Del mismo modo se comprueba que exista repetitividad de resultados, para ello se realiza la medición y clasificación de 10 diferentes cueros, en 2 ocasiones cada uno. Previamente se separan 5 cueros de color claro y 5 de color oscuro tomados al azar.

Ecuación 4.4 Calculo de rangos

$$R = |P1 - P2|$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

R: Rango; P1: Prueba 1; P2: Prueba 2

Ecuación 4.5 Calculo promedio de rangos

$$\bar{R} = \frac{\sum_{i=1}^n R}{n}$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

\bar{R} : Promedio de rangos; **$\sum_{i=1}^n R$:** Total de rangos; **n:** cantidad de pruebas

Ecuación 4.6 Calculo error máximo (repetitividad)

$$EM = k_2(\bar{R})$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

EM: Error de medición; **k_2 :** constante correspondiente al número de operadores; **\bar{R} :** Promedio de rangos

Ecuación 4.7 Desviación estándar del error medido

$$\sigma_{R\&R} = \frac{EM}{5.15}$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

$\sigma_{R\&R}$: Desviación estándar del error medido; **EM:** Error máximo

Se establece la tolerancia calculando un error de $\pm 1\%$ del promedio de la medida de cada prueba.

Ecuación 4.8 Tolerancia

$$T = 0.01 x (P1 + P2)$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

%EM: Porcentaje de error máximo; EM: Desviación estándar del error medido;
Tolerancia: Tolerancia

Ecuación 4.9 Porcentaje del error medido (% de repetitividad)

$$\%EM = \frac{EM(100)}{Tolerancia}$$

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Donde:

%EM: Porcentaje de error máximo; EM: Desviación estándar del error medido;
Tolerancia: Tolerancia

Tabla 4.18 Constante para expandir el rango promedio

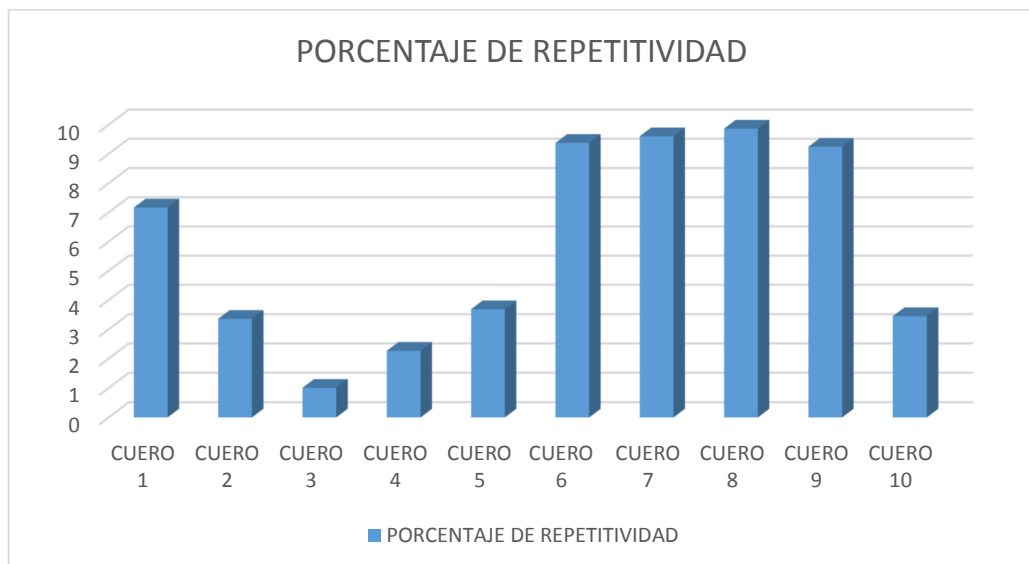
k_2 Constante para expandir el rango promedio a 5.15 desviaciones estándar				
Número de pares	Número de operadores			
	2	3	4	5
1	3.65	2.7	2.3	2.08
2	4.02	2.85	2.4	2.15
3	4.19	2.91	2.43	2.16
4	4.26	2.94	2.44	2.17
5	4.33	2.96	2.45	2.18
6	4.36	2.98	2.46	2.19
7	4.30	2.98	2.46	2.19
8	4.40	2.99	2.48	2.19
9	4.44	2.99	2.48	2.20
10	4.44	2.99	2.48	2.20

Fuente: Gutiérrez, Humberto; Control estadístico de calidad y seis sigma, 2009.

Tabla 4.19 Prueba repetitividad y reproductibilidad de resultados

Color de la hoja de cuero	Cuero N°	Prueba N°	Medida [ft^2]	Clase	Rango	EM	$\sigma_{R\&R}$	T	%EM
Obscuro	1	1	25.456	B	0.01	0.037	0.0071	0.509	7.17
		2	25.446	B					
	2	1	16.267	C	0.003	0.011	0.002	0.325	3.36
		2	16.270	C					
	3	1	35.826	B	0.002	0.007	0.0014	0.716	1.01
2		35.824	B						
4	1	32.147	A	0.004	0.015	0.0028	0.641	2.27	
	2	32.151	A						
5	1	19.731	B	0.004	0.015	0.0028	0.395	3.69	
	2	19.735	B						
Claro	6	1	23.336	A	0.012	0.043	0.0085	0.467	9.38
		2	23.348	A					
	7	1	11.401	B	0.006	0.022	0.0042	0.228	9.61
		2	11.395	B					
	8	1	36.984	A	0.02	0.073	0.0141	0.740	9.87
		2	37.004	A					
	9	1	17.758	C	0.009	0.032	0.0063	0.355	9.25
		2	17.767	C					
	10	1	26.423	B	0.005	0.018	0.0035	0.529	3.45
		2	26.428	B					

Gráfico 4.1 Grafica de porcentaje error medido (% de repetitividad)



De esta tabla se obtiene como resultado un porcentaje de repetitividad de 9.87%, a este valor se lo compara con la tabla 4.17, demostrando que este sistema tiene una capacidad de repetitividad y reproductividad de resultados excelente.

Además, se puede apreciar que la columna de Clase se repite el resultado en cada cuero, mostrando repetitividad de datos.

4.6.7. Esquema general del sistema de visión artificial

A continuación, se muestra el hardware ensamblado para el sistema de visión artificial, seguido de su descripción.

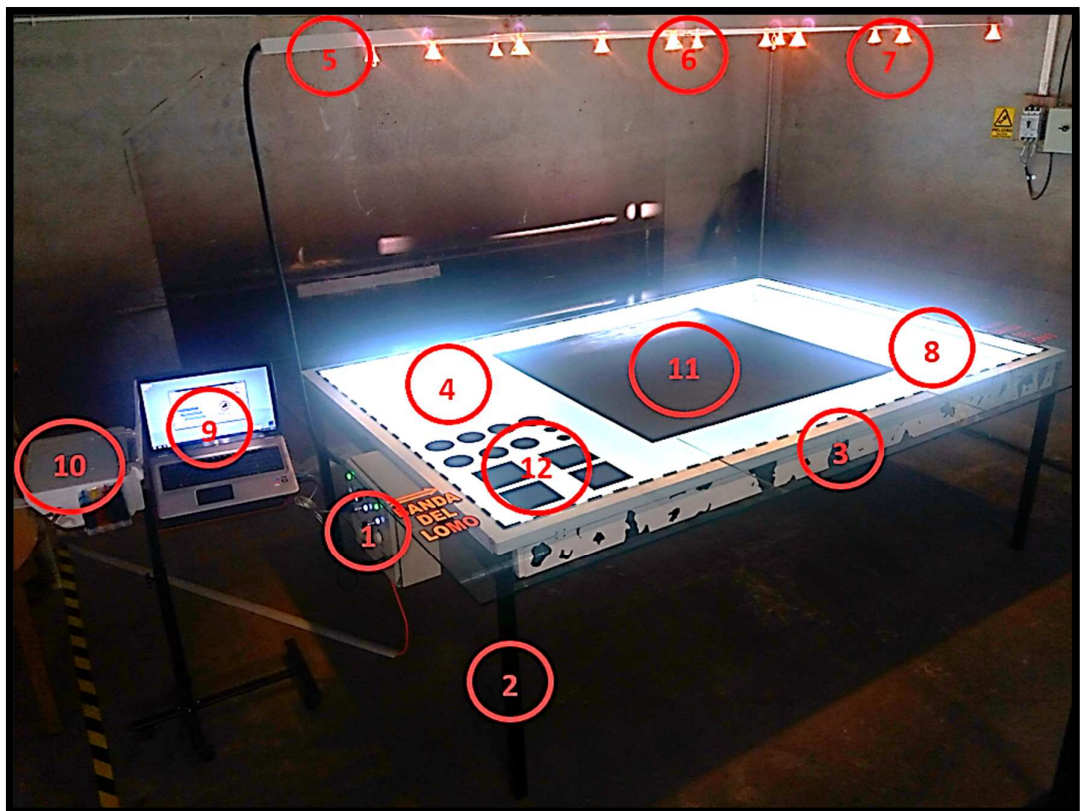


Figura 4.6 Esquema del sistema de vision artificial para el control de calidad de hojas de cuero

Tabla 4.20 Detalle de los componentes

NUMERO	NOMBRE	DESCRIPCIÓN
1	Tablero de distribución	Muestra y activa la luminaria, permite la conexión de periféricos a la red eléctrica.
2	Estructura inferior	Da soporte a: cajón luminoso, tablero de distribución, cubierta, hoja de cuero
3	Cajón luminoso	Concentra en una sola dirección los rayos luminosos de la luminaria inferior
4	Cubierta	Da soporte a la hoja de cuero y permite el paso de los rayos luminosos
5	Estructura superior	Da soporte a: cámara, luminaria superior
6	Cámara	Captura imágenes
7	Luminaria superior	Permite la visualización del borde de la hoja de cuero
8	Luminaria inferior	Permite la visualización de las plantillas sobre la hoja de cuero
9	Ordenador	Periférico usado de hardware para el software del sistema
10	Impresora	Periférico que suministra la etiqueta referencial del cuero
11	Pieza de calibración	Ayuda a calibrar los parámetros del sistema
12	Plantillas de control	Permite distinguir las fallas propias del cuero

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- El algoritmo planteado permite eliminar las distorsiones del lente de la cámara correctamente por medio de la toolbox Camera Calibrator de Matlab, obteniendo una imagen bien delineada, nítida y sin distorsiones.
- La normalización a dos colores básicos (negro y blanco) permite obtener el área de trabajo y la clasificación de las hojas mediante plantillas de cualquier color de hoja de cuero.
- La técnica de uso de dióicos seleccionada para la luminaria superior facilita una correcta distribución de la luz en el área de trabajo permitiendo un buen control de calidad ya que estas proyectan un haz de luz circular
- La programación del sistema está diseñada para eliminar cualquier elemento que no pertenezca a la hoja de cuero o interferencia lumínica exteriores al sistema, eliminando la mayor cantidad de respuestas erróneas.
- Un sistema basado en visión artificial puede identificar patrones, en este caso se usó plantillas como patrón, esto debido a que no se tiene una correcta clasificación cuando se identificaba las fallas directamente, ya que el sistema presentaba confusión en la discriminación y discernición de las mismas.
- La categorización del cuero bajo normas exige la identificación de todas las fallas existentes sobre la hoja, que bajo el sistema de visión propuesto son complejas en su identificación por lo que se implementó plantillas como patrón de falla eliminando el error en la discriminación y discernición de las mismas.
- El error presentado en la medida (27%) y repetitividad (9.87%) del área es debido a que crear un programa que sirva para hojas de cuero de colores claros como oscuros demanda el uso de multiplicadores de píxeles, los cuales ayudan a unir píxeles sueltos, esto provoca aleatoriamente líneas más pronunciadas o lo contrapuesto.
- El diseño propuesto de la estructura del sistema de visión artificial permite soportar hasta 180 kg de masa sobre la pesa sin que sufra ninguna deformación, además se la aisló de la estructura superior para evitar vibraciones que provoquen error en el proceso.
- Luego de realizar las pruebas se observa que no es necesario un control de graduación de la cantidad lumínica cuando se tiene un ambiente controlado, y

que la medida y la clasificación se encuentran en un rango aceptable de error bajo las normas que lo rigen.

- Al ser un sistema basado en redes neuronales es necesario la calibración previa del mismo para certificar las mediciones, por lo que se realizó un manual de usuario.

5.2. RECOMENDACIONES

- No rotar el lente de la cámara debido a que podría cambiar las distorsiones y arrojaría datos erróneos.
- No usar luminaria con un alto grado lumínico en la luminaria superior ya que podría saturar el brillo de la imagen.
- Se recomienda mantener las manos fuera del área marcada como límite, ya que puede arrojar resultados erróneos
- Tener a consideración que la norma para la clasificación es de acuerdo a un informe emitido por la corporación ganadera CORFOGA, ya que no existe una normativa ecuatoriana.
- Para próximos diseños se sugiere elegir materiales y un mecanismo que permita fácil transportación y peso liviano, para que se puede brindar servicio del sistema en otras localidades.
- Se recomienda la ubicación de la estructura superior a una distancia superior a 2.23 [m] de la cubierta en la mesa.
- Matlab tiene un retraso en la inicialización de sus toolbox por primera vez, por lo que se recomienda esperar un tiempo prudencial después de inicializar un ciclo, en cualquiera de sus ventanas.
- Para próximos diseños eliminar el control de luminosidad, pero mantener el valor lumínico moderado de la luminaria, ya que no es muy necesario; usar luminaria led.
- Se recomienda que el suelo sobre el que se va a colocar la estructura inferior este correctamente nivelado.

LISTA DE SIGLAS

CCD: Dispositivos de acoplamiento de carga

CID: Dispositivo de inyección de carga

CMOS: Semiconductor complementario de óxido metálico

Mp: Megapíxeles

Fps: Frames per second (Cuadros por segundo)

AI: Alternativa

GUIDE: Graphical User Interface Desing Enviroment

CI: Circuito Integrado

ASCII: American Standard Code for Information Interchange

μC: Micro-controlador

RNA: Redes Neuronales Artificiales

IA: Inteligencia Artificial

BIBLIOGRAFÍA

- Boylestad, R. L. (2011). *Introducción al análisis de circuitos*. México: Pearson Educación.
- Bueno, J. (1999). *Introducción a la óptica instrumental*. Murcia: Universidad de murcia.
- CEDAL. (12 de Marzo de 2016). *Perfiles de aluminio arquitectónico y estructural*. Obtenido de Cedal: http://www.cedal.com.ec/pdf/cedal_ref.jpg
- CORFOGA. (2 de Abril de 2014). *Clasificación de Cueros*. Obtenido de CORFOGA: www.corfoga.org
- Cromer, A. (2006). *Física en la ciencia y en la industria*. Barcelona: Reverté. S.A.
- CUERONET. (31 de Marzo de 2014). *Defectos más habituales que afectan la calidad de los cueros y pieles*. Obtenido de CUERONET: www.cueronet.com
- CUERONET. (23 de Marzo de 2014). *División Superficie de la piel*. Obtenido de CUERONET: <http://www.cueronet.com/>
- FLACSO - MIPRO. (Diciembre de 2011). Boletín mensual de análisis sectorial de MIPYMES-Ropa de vestir de cuero para exportación. *Boletín N°22*. (H. Jácome, & O. Julio, Edits.) Quito, Pichincha, Ecuador.
- Flórez López, R., & Fernández, J. M. (2008). *Las Redes Neuronales Artificiales (Fundamentos teóricos y aplicaciones prácticas)*. España: Netbiblo, S.L.
- Floyd, T. L. (2008). *Dispositivos Electrónicos*. México: Pearson Educación.
- García, I. (2008). *Visión artificial y procesamiento digital de imágenes usando Matlab*. Ibarra: PUCE.
- Genius, KYE Systems Corp. (15 de Abril de 2014). *Cámara web Full HD gran angular*. Obtenido de Genius: <http://www.geniusnet.com/>
- Gormaz, I. (2010). *Técnicas y procesos en las instalaciones singulares en los edificios*. Madrid: Parainfo.

- Gutierrez, H., & De la Vara, R. (2009). *Control estadístico de calidad y seis sigma*. Mexico: McGraw-Hill .
- Havells Sylvania. (10 de Agosto de 2016). *MR16 CLFT 12V 20W 38° GU5.3 SV*. Obtenido de Havells Sylvania: www.havells-sylvania.com/product/es-es/pdf/DataSheet/0022516
- Industrias JQ. (7 de Marzo de 2016). *Datos tecnicos policarbonato: Industrias JQ*. Obtenido de Industrias JQ Plasticos de Ingenieria: <http://www.jq.com.ar/Imagenes/Productos/Policarbonato/dtecnicos/dtecnicos.htm>
- IPAC. (3 de Marzo de 2016). *Tuberia Estructural Rectangular: IPAC*. Obtenido de IPAC S.A.: http://www.ipac-acero.com/assets/img/upload/big/CatalogoIPAC_12.pdf
- IPAC. (10 de Marzo de 2016). *Tuberia estructural cuadrada: IPAC*. Obtenido de IPAC S.A.: http://www.ipac-acero.com/assets/img/upload/big/CatalogoIPAC_11.pdf
- Li, F., & Nathan, A. (2005). *CCD Image Sensors in Deep-Ultraviolet: Degradation Behavior and Damage Mechanisms*. Canada: Springer.
- Mann, I. (1978). *Técnicas de curticion rural* .
- Mathworks. (22 de Abril de 2014). *Productos & Servicios: Computer Vision System Toolbox*. Obtenido de Mathworks: <http://www.mathworks.es>
- Mathworks. (22 de Abril de 2014). *Productos y servicios: Image Acquisition Toolbox*. Obtenido de Matworks: <http://www.mathworks.com/>
- Mathworks. (20 de Abril de 2014). *Productos y servicios: MATLAB*. Obtenido de Mathworks: <http://www.mathworks.es>
- Ministerio del Trabajo. (10 de 08 de 2016). *Reglamento de Seguridad y Salud de los Trabajadores y Mejoramiento del Medio Ambiente de Trabajo – Decreto Ejecutivo 2393*. Obtenido de Ministerio del Trabajo : <http://www.trabajo.gob.ec/seguridad-y-salud-en-el-trabajo/>

- Monsó, J. (1994). *Sistema de identificación y control automáticos, Volumen 2*. Barcelona: Marcombo S.A.
- Mott, R. (2006). *Diseño de elementos de máquinas*. Pearson Educación.
- National Instrument. (15 de Abril de 2014). *NI 1778 Smart Camera*. Obtenido de National Instrument: <http://latam.ni.com/>
- NBN Automação Industrial. (2 de Abril de 2014). *Sistemas de medição para cueros semi-acabado/acabado*. Obtenido de NBN: www.nbn.com.br
- Nilsson, N. J. (2001). *Inteligencia Artificial: una nueva síntesis*. Mc Graw-Hill.
- Pearson, C., & CAVIPLAN. (s.f.). *Manual del vidrio plano*. Argentina: TOD Producciones S.A.
- Riba, C. (2002). *Diseño concurrente*. Barcelona: Edicions UPC.
- Sterlacci, F. (2010). *Diseño de moda en piel*. BLUME (Naturart).
- Torres, F., Pomaras, J., & Gil, P. (2002). *Robots y sistemas sensoriales*. Madrid: Pearson Educación.
- VASA. (8 de Marzo de 2016). *Vidrios para tapas de mesa y estantes*. Obtenido de Vasa vidrieria argentina S.A.: <http://www.vasa.com.ar/DocVasa/pdf/BOLETIN-VASA-25-TAPAS-DE-MESAS.pdf>
- Young, H., & Freedman, R. (2009). *Física universitaria, con física moderna volumen 2*. México: Pearson Educación.

ANEXO 1 MANUAL DE USUARIO Y MANTENIMIENTO

ANEXO 1

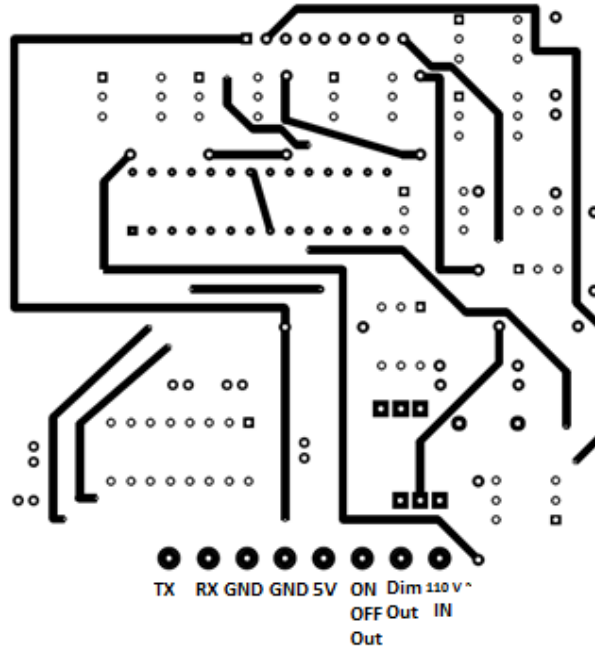
**Manual de usuario y
mantenimiento**

**ANEXO 2 CIRCUITO IMPRESO DE LA TARJETA DE
LUMINOSIDAD**

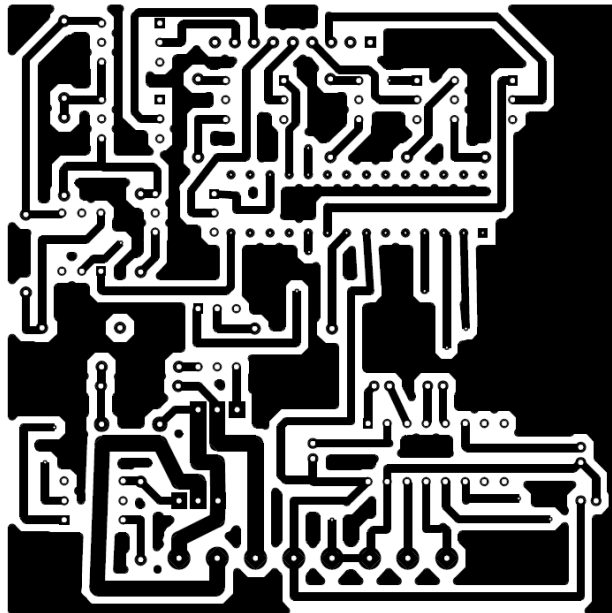
ANEXO 2

**Circuito impreso de la tarjeta de
luminosidad**

CIRCUITO IMPRESO, LADO SUPERIOR



CIRCUITO IMPRESO, LADO INFERIOR



ANEXO 3 DIAGRAMA ELÉCTRICO

ANEXO 3

Diagrama electrico

ANEXO 4 PROGRAMACIÓN SOFTWARE

ANEXO 4

Programación software

Anexo 4.1 Función apagarluzinferior

```
function apagarluzinferior
%% carga de dato a enviar en variable
dato=num2str(6);%carga en dato el valor 6 en formato string
%% busqueda de puertos serial conectados en el ordenador
serialInfo = instrhwinfo('serial');%busqueda de puertos seriales conectados
serialdisponible=serialInfo.AvailableSerialPorts;%devuelve el nombre del puerto activo
verificacionserial=isempty(serialdisponible);%comprueba estado de variable serialdisponible
if(verificacionserial==1)
else
%% configuracion del puerto serial
serialconfig=serial(char(serialdisponible));%crea puerto serial
set(serialconfig,'BaudRate',9600);%carga frecuencia de comunicacion
set(serialconfig,'DataBits',8);%cantidad de bits por caracter
set(serialconfig,'Parity','none');%sin bit de paridad
set(serialconfig,'StopBits',1);%bit de parada
set(serialconfig,'FlowControl','none');%sin control de flujo
%% transmision de dato por puerto serial
fopen(serialconfig);%abre puerto serial configurado
fprintf(serialconfig,'%s',dato);%transmite variable dato
fprintf(serialconfig,13);%transmite caracter ASCII de control ENTER
fclose(serialconfig);%cierra puerto serial configurado
delete(serialconfig);%borra la variable serialconfig
clear serialconfig;%limpia la variable serialconfig
end
end
```

Anexo 4.2 Función apagarluzsuperior

```
function apagarluzsuperior
%% carga de dato a enviar en variable
dato=num2str(4);%carga en dato el valor 4 en formato string
%% busqueda de puertos serial conectados en el ordenador
serialInfo = instrhwinfo('serial');%busqueda de puertos seriales conectados
serialdisponible=serialInfo.AvailableSerialPorts;%devuelve el nombre del puerto activo
verificacionserial=isempty(serialdisponible);%comprueba estado de variable serialdisponible
if(verificacionserial==1)
else
%% configuracion del puerto serial
serialconfig=serial(char(serialdisponible));%crea puerto serial
set(serialconfig,'BaudRate',9600);%carga frecuencia de comunicacion
set(serialconfig,'DataBits',8);%cantidad de bits por caracter
set(serialconfig,'Parity','none');%sin bit de paridad
set(serialconfig,'StopBits',1);%bit de parada
set(serialconfig,'FlowControl','none');%sin control de flujo
%% transmision de dato por puerto serial
fopen(serialconfig);%abre puerto serial configurado
fprintf(serialconfig,'%s',dato);%transmite variable dato
fprintf(serialconfig,13);%transmite caracter ASCII de control ENTER
fclose(serialconfig);%cierra puerto serial configurado
delete(serialconfig);%borra la variable serialconfig
clear serialconfig;%limpia la variable serialconfig
end
end
```

Anexo 4.3 Función areapieza

```
function areapieza
global imundgraycon areal imselecta imrellena opcion impieza imcontmorf e
%% resalte de contornos
opcion=1;
eliminamesaaumentadocontraste
if imselecta==~ones(1080,1920)
    imselecta=ones(1080,1920);
    imrellena=ones(1080,1920);
else
imcondet=edge(imundgraycon,'sobel');%deteccion de contornos a imagen contrastada
%% realce de características
imcontmorf=bwmorph(imcondet,'dilate',2);%operacion morfologica dilate, 2 veces repetida
    %para aclarar bordes
imcontmorf=bwmorph(imcontmorf,'bridge');%operacion morfologica bridge, usada para conectar
    %pixeles cercanos sueltos
imcontmorf=bwmorph(imcontmorf,'diag');%operacion morfologica diag, usada para conectar en
    %diagonal pixeles cercanos sueltos

%% discercion de pixeles
e=bwmorph(imselecta,'dilate',4);
impieza=and(imcontmorf,bwmorph(imselecta,'dilate',4));%se elimina pixeles de bordes que
    %se encuentran dentro de pieza de cuero 15 25

%% seleccion borde pieza
piezacon=bwconncomp(impieza);%conecta pixeles formando regiones
proppiezacon=regionprops(piezacon,'Perimeter');%obtiene propiedad de cada region
busqueda=find([proppiezacon.Perimeter]>2000);%localiza regiones con perimetros mayor
    %al 33% de la imagen total
imselecta=ismember(labelmatrix(piezacon),busqueda);%toma las regiones de la imagen conectada
    %que cumplen con la busqueda anterior

%% relleno de imagen pieza
imrellena=imfill(imselecta,'holes');%rellena agujeros
%% calculo area en pixeles
if(imrellena==~ones(1080,1920))%si imagen esta totalmente en negro
areal.Area=0;%area es 0
imshow(ones(1080,1920));%muestra imagen en blanco
else
pieza=bwconncomp(imrellena);%conecta pixeles formando regiones
areal=regionprops(pieza,'BoundingBox','Area','Centroid');%obtiene propiedad (Area)
    %de cada region

end
end
end
```

Anexo 4.4 Función aumentarluminosidadsuperior

```
function aumentarluminosidadsuperior
%% carga de dato a enviar en variable
dato=num2str(1);%carga en dato el valor 1 en formato string
%% busqueda de puertos serial conectados en el ordenador
serialInfo = instrhwinfo('serial');%busqueda de puertos seriales conectados
serialdisponible=serialInfo.AvailableSerialPorts;%devuelve el nombre del puerto activo
verificacionserial=isempty(serialdisponible);%comprueba estado de variable serialdisponible
if(verificacionserial==1)
else
%% configuracion del puerto serial
serialconfig=serial(char(serialdisponible));%crea puerto serial
set(serialconfig,'BaudRate',9600);%carga frecuencia de comunicacion
set(serialconfig,'DataBits',8);%cantidad de bits por caracter
set(serialconfig,'Parity','none');%sin bit de paridad
set(serialconfig,'StopBits',1);%bit de parada
set(serialconfig,'FlowControl','none');%sin control de flujo
%% transmision de dato por puerto serial
fopen(serialconfig);%abre puerto serial configurado
fprintf(serialconfig,'%s',dato);%transmite variable dato
fprintf(serialconfig,13);%transmite caracter ASCII de control ENTER
fclose(serialconfig);%cierra puerto serial configurado
delete(serialconfig);%borra la variable serialconfig
clear serialconfig;%limpia la variable serialconfig
end
end
```

Anexo 4.5 Función calculoarea

```
function calculoarea
global fc areal area2calstr area2calstrpies
%%verificacion de existencia de dato
verificacionarea=isempty(areal);%existe dato en variable areal
if(verificacionarea==1)%no existe dato
else
%%conversion de pixeles a ft2
amgi=areal.Area+0.01*(areal.Area);%extrae valor de area de variable areal
load('fc.mat');%carga archivo fc.mat (factor de conversion)
arpn=amgi*fc;%convierte de pixeles a cm2
area2calstr=num2str(arpn);%transforma formato de numeros a string
arpnpies2=arpn*0.00107639;%convierte a pies cuadrado
area2calstrpies=num2str(arpnpies2);%transforma formato de numeros a string
end
end
```

Anexo 4.6 Función calculodeparametros

```
function calculodeparametros
global vid imcor fc areal imselec val
%% variables de inicio
contador1=1;
var1=0;
val=0;

%% adquisicion de imagen y procesamiento (eliminar distorsión y mesa)
while contador1==1
    flushdata(vid)%borra memoria variable vid
    eliminadistorsion%llama a funcion eliminadistorsion
    eliminamesa%llama a funcion eliminamesa
    if (imcor==ones(1080,1920))
        imshow(ones(1080,1920));%muestra imagen en blanco
        pause(0.01)%pausa
        flushdata(vid)%borra memoria variable vid
    else
%% discernimiento de piezas por areas
        imcorcomp=bwconncomp(~imcor);%conecta pixeles con valor 1 formando regiones
        propimcor1=regionprops(imcorcomp,'Area');%recolecta informacion de
            %propiedades(Area) de cada una de las regiones
        var3=find([propimcor1.Area]>207360);%seleccion de datos de regiones superiores
            %al 10% del area total
        imselec=ismember(labelmatrix(imcorcomp),var3);%imagen solo con bordes de las
            %plantilla
%% adquisicion de parametro ubicacion de la pieza de calibracion en la imagen
        if(imselec==~ones(1080,1920))
            imshow(ones(1080,1920));%muestra imagen en blanco
        else
            imseleccomp=bwconncomp(imselec);%conecta pixeles con valor 1 formando
                %regiones(plantilla)
            propimcor2=regionprops(imseleccomp);%recolecta informacion de propiedades de cada
                %una de las regiones
            cordcentroimselec=round(propimcor2.Centroid);%punto central de la plantilla
%% union de imagen de pieza de calibracion con imagen de flechas para centrar pieza
            hold on
            if(cordcentroimselec(1)>960 && cordcentroimselec(2)==540)
                imselecind=and(imselec,imread('flechaizquierda.bmp'));%une imagen pieza y flecha
                imshow(imselecind)%muestra imagen pieza y flecha
            end
            if(cordcentroimselec(1)<960 && cordcentroimselec(2)==540)
                imselecind=and(imselec,imread('flechaderecha.bmp'));%une imagen pieza y flecha
                imshow(imselecind)%muestra imagen pieza y flecha
            end
            if(cordcentroimselec(2)>540 && cordcentroimselec(1)==960)
                imselecind=and(imselec,imread('flechaarriba.bmp'));%une imagen pieza y flecha
                imshow(imselecind)%muestra imagen pieza y flecha
            end
            if(cordcentroimselec(2)<540 && cordcentroimselec(1)==960)
                imselecind=and(imselec,imread('flechaabajo.bmp'));%une imagen pieza y flecha
                imshow(imselecind)%muestra imagen pieza y flecha
            end
        end
    end
    contador1=contador1+1;
end
```

```

end
if(cordcentroimselec(1)>960 && cordcentroimselec(2)>540)
    imselecind=and(and(imselec,imread('flechaizquierda.bmp')),...
        imread('flechaarriba.bmp')); %une imagen pieza y flechas
    imshow(imselecind)%muestra imagen pieza y flecha
end
if(cordcentroimselec(1)<960 && cordcentroimselec(2)<540)
    imselecind=and(and(imselec,imread('flechaderecha.bmp')),...
        imread('flechaabajo.bmp')); %une imagen pieza y flechas
    imshow(imselecind)%muestra imagen pieza y flechas
end
if(cordcentroimselec(1)<960 && cordcentroimselec(2)>540)
    imselecind=and(and(imselec,imread('flechaderecha.bmp')),...
        imread('flechaarriba.bmp')); %une imagen pieza y flechas
    imshow(imselecind)%muestra imagen pieza y flechas
end
if(cordcentroimselec(1)>960 && cordcentroimselec(2)<540)
    imselecind=and(and(imselec,imread('flechaizquierda.bmp')),...
        imread('flechaabajo.bmp')); %une imagen pieza y flechas
    imshow(imselecind)%muestra imagen pieza y flechas
end
plot(960,540,'co','markersize',4,'markerfacecolor','c')%muestra punto del centro
    %de la imagen en color azul
%% aceptacion o rechazo de pieza de calibracion en centro de la imagen
if(cordcentroimselec(1)>=940 && cordcentroimselec(1)<=980 &&...
    cordcentroimselec(2)>=520 && cordcentroimselec(2)<=560 )
    plot(cordcentroimselec(1), cordcentroimselec(2),'go','markersize',4,...
        'markerfacecolor','g')%muestra punto del centro de la pieza en color verde
    var1=1;
else
    plot(cordcentroimselec(1), cordcentroimselec(2),'ro','markersize',4,...
        'markerfacecolor','r')%muestra punto del centro de la pieza en color rojo
end
%% calculo de area de pieza de referencia en pixeles
hold off
pause(0.01)%pausa
if (var1==1)
    stop(vid)%detiene grabacion de video
    flushdata(vid)%borra memoria variable vid
    areapieza%llama a funcion areapieza
    if (areal.Area==0)
        contador1=1;%vuelve a iniciar labusueda de la pieza
        val=0;%carga valor 0 en variable val indicando que no a acabado el proceso
    else
%% obtencion de factor de conversion
    aipg=areal.Area;%carga con el valor del area de la plantilla en pixeles
    save('aipg.mat','aipg');%guarda variable aipg en aipg.mat
    fc=(14400/aipg);%area real de la plantilla guia(120cm*120cm)
    save('fc.mat','fc');%guarda variable fc en fc.mat
    val=1;%carga valor 1 en variable val indicando que acabado el proceso
    contador1=0;%carga valor 0 en variable para poder salir del ciclo

    end
end
end
end
end
end
end
end

```

Anexo 4.7 Función clasificación

```
function clasificacion
global datosregiones2 imfiltcom proppieza clase
%% inicializacion de variables
load('aipg.mat');
claseA(1)=0;
claseA(2)=0;
claseA(3)=0;
claseB(1)=0;
claseB(2)=0;
claseB(3)=0;
claseC=0;
primertercio=0;
segundotercio=0;
tercertercio=0;
contadorplant=0;
contadorprimertercio=0;
contadoresegundotercio=0;
contadortercertercio=0;
proplanci5cm=[];
proplanci10cm=[];
proplancu15cm=[];
proplancircular=[];
areacuadrado15cm=0;
areacirculo5cm=0;
areacirculo=0;
%% clasificaion de plantillas por area
%plantilla circular 5 cm de diametro
datosbusq=find([datosregiones2.Area]<1500 & [datosregiones2.Area]>=300);%seleccion de datos
%necesarios de acuerdo a la medida de las plantilla:
planci5cm=ismember(labelmatrix(imfiltcom),datosbusq);%imagen con plantillas circular
%5 cm de diametro

%plantilla circular 10 cm de diametro
datosbusq2=find([datosregiones2.Area]<5000 & [datosregiones2.Area]>=1501);%seleccion de datos
%necesarios de acuerdo a la medida de las plantilla:
planci10cm=ismember(labelmatrix(imfiltcom),datosbusq2);%imagen con plantillas circular
%10 cm de diametro

%plantilla cuadrada 15 cm de lado
datosbusq3=find([datosregiones2.Area]<8000 & [datosregiones2.Area]>=5001);%seleccion de datos
%necesarios de acuerdo a la medida de las plantilla:
plancu15cm=ismember(labelmatrix(imfiltcom),datosbusq3);%imagen con plantillas cuadrada 15 cm
%de lado

%% obtencion de nuevas propiedades de las plantillas separadas por area
%propiedades plantilla circular 5 cm de diametro
if(planci5cm==~ones(1080,1920))
else
    planci5cmcon=bwconncomp(planci5cm);%conecta pixeles con valor 1 formando regiones
    proplanci5cm=regionprops(planci5cmcon,'BoundingBox','Centroid');%recolecta informacion
%de propiedades decada una de las regiones
end
%propiedades plantilla circular 10 cm de diametro
```

```

if(planci10cm==~ones(1080,1920))
else
    planci10cmcon=bwconncomp(planci10cm);%conecta pixeles con valor 1 formando regiones
    proplanci10cm=regionprops(planci10cmcon,'BoundingBox','Centroid');%recolecta informacion
                                %de propiedades decada una de las regiones
end
%%propiedades plantilla cuadrada 15 cm de lado
if(plancu15cm==~ones(1080,1920))
else
    plancu15cmcon=bwconncomp(plancu15cm);%conecta pixeles con valor 1 formando regiones
    proplancu15cm=regionprops(plancu15cmcon,'BoundingBox','Centroid');%recolecta informacion
                                %de propiedades decada una de las regiones
end
%%propiedades plantilla circular
plancircular=or(planci5cm,planci10cm);%union de plantillas circulares
if(plancircular==~ones(1080,1920))
else
    plancircularcon=bwconncomp(plancircular);%conecta pixeles con valor 1 formando regiones
    proplancircular=regionprops(plancircularcon,'BoundingBox','Centroid'); %recolecta
                                %informacion de propiedades decada una de las regiones
end
%% segmentacion de la pieza por bandas y por tercios
if (length(propieza)>1)
else
    limiteprimertercio=propieza.BoundingBox(1)+(propieza.BoundingBox(3)/3);%limite primer ter
    limitesegundotercio=propieza.BoundingBox(1)+(propieza.BoundingBox(3)/3)*2;%limite segun
                                %tercio
    limitebandapanza=propieza.BoundingBox(2)+(propieza.BoundingBox(4)/3);%limite banda
                                %de la panza
    limitebandacentral=propieza.BoundingBox(2)+((propieza.BoundingBox(4)/3)*2);%limite banda
                                %del lomo
%% preclasificacion pieza de CLASE A
if(length(proplanci5cm)<=1 && length(proplanci10cm)<1 && length(proplancu15cm)<=2)
    %max:1 circulo 5cm,0 cirulo 10cm,2 cuadrados 15cm
    if(plancu15cm==~ones(1080,1920))%si plantilla cuadrada 15cm no existe
        claseA(2)=1;%condicion marcas de fierro cumple
        areacuadrado15cm=0;%area afectada por marcas de fierro es 0
    else
        for k=1:length(proplancu15cm)
            if(proplancu15cm(k).Centroid(2)>limitebandacentral)%si plantilla cuadrada
                %esta en la banda del lomo
            if(proplancu15cm(k).Centroid(1)<limiteprimertercio)%si centroide se
                %encuentra en el primer tercio
                primertercio=1;%existe plantilla en el primer tercio
                areacuadrado15cm=areacuadrado15cm+8000;%area afectada por marcas de
                    %fierro se anade 8000 pixeles
            elseif(proplancu15cm(k).Centroid(1)>=limiteprimertercio &&...
                proplancu15cm(k).Centroid(1)<=limitesegundotercio)
                %si centroide se encuentra en el segundo
                segundotercio=1;%existe plantilla en el segundo tercio
                areacuadrado15cm=areacuadrado15cm+8000;%area afectada por marcas de

```



```

                                                                    %fierro se anade 8000 pixeles
elseif (proplancu15cm(k).Centroid(1)>limitesegundotercio)%si centroide se
                                                                    %encuentra en el tercer tercio
    tercertercio=1;%existe plantilla en el tercer tercio
    areacuadrado15cm=areacuadrado15cm+8000;%area afectada por marcas de
                                                                    %fierro se anade 8000 pixeles
end
if (primertercio+segundotercio+tercertercio<=1)%si se encuentran las
                                                                    %plantillas en un solo tercio
    claseA(2)=1;%condicion marcas de fierro cumple
else%si no se encuentran las plantillas en un solo tercio
    claseA(2)=0;%condicion marcas de fierro cumple
end
end
end
end
if (planci5cm==~ones(1080,1920))%si plantilla circular 5cm no existe
    claseA(1)=1;%condicion defectos cumple
    areacirculo5cm=0;%area afectada por defectos es 0
else
    for k=1:length(proplanci5cm)
        if (proplanci5cm.Centroid(2)<=limitebandacentral &&...
            proplanci5cm.Centroid(2)>=limitebandapanza )
            %si plantilla circular 5cm esta dentro de la banda central
            claseA(1)=0; %condicion defectos no cumple
            areacirculo5cm=0;%area afectada por defectos es 0
        else
            claseA(1)=1;%condicion defectos cumple
            areacirculo5cm=1500;%area afectada por defectos es 0
        end
    end
end
end
end
if (propieza.Area-areacuadrado15cm-areacirculo5cm>=0.9*propieza.Area)%si area util es
                                                                    %mayor al 90%
    claseA(3)=1;%condicion area cumple
else
    claseA(3)=0;%condicion area no cumple
end
claseA=and (and (claseA(1),claseA(2)),claseA(3));%si cumple las 3 condiciones claseA=1
%% preclasificacion pieza de CLASE B
if (length (proplancircular)<=6 && length (proplancu15cm)<=3)%maximo: 6 circulos -
                                                                    %3 cuadrados 15 cm
    if (plancu15cm==~ones(1080,1920))%si plantilla cuadrada 15cm no existe
        claseB(2)=1;%condicion marcas de fierro cumple
        areacuadrado15cm=0;%area afectada por marcas de fierro es 0
    else
        for k=1:length (proplancu15cm)
            if (proplancu15cm(k).Centroid(2)>limitebandacentral)%si plantilla cuadrada
                                                                    %esta en la banda del lomo
                contadorplant=contadorplant+1;%cuenta la cantidad de plantillas cuadradas
            end
        end
    end
end
end

```

```

        areacuadrado15cm=areacuadrado15cm+8000;%area afectada por marcas de fierro
                                %se anade 8000 pixeles
    else%si plantilla cuadrada esta fuera de la banda del lomo
        claseB(2)=0;%condicion marcas de fierro no cumple
    end
end
if(contadorplant<=3)%si plantilla cuadrada es menor en cantidad o igual a 3
    claseB(2)=1;%condicion marcas de fierro cumple
else%si plantilla cuadrada es mayor en cantidad a 3
    claseB(2)=0;%condicion marcas de fierro no cumple
end
end
if(plancircular==~ones(1080,1920))%si plantilla circular no existe
    claseB(1)=1;%condicion defectos cumple
    areacirculo=0;%area afectada por defectos es 0
else
    for k=1:length(proplancircular)
        if(proplancircular(k).Centroid(1)<limiteprimertercio)%si centroide se
                                %encuentra en el primer tercio
            contadorprimertercio=contadorprimertercio+1;%cuenta cantidad de
                                %plantillas en el primer tercio
            areacirculo=areacirculo+3000;%area afectada por defectos se anade
                                %3000 pixeles
        elseif(proplancircular(k).Centroid(1)>=limiteprimertercio &&...
            proplancircular(k).Centroid(1)<=limitesegundotercio)
            %si centroide se encuentra en el segundo tercio
            contadorsegundotercio=contadorsegundotercio+1;%cuenta cantidad de
                                %plantillas en el segundo tercio
            areacirculo=areacirculo+3000;%area afectada por defectos se anade
                                %3000 pixeles
        elseif(proplancircular(k).Centroid(1)>limitesegundotercio)%si centroide
                                %se encuentra en el tercer tercio
            contadortercertercio=contadortercertercio+1;%cuenta cantidad de
                                %plantillas en el segundo tercio
            areacirculo=areacirculo+3000;%area afectada por defectos se anade 3000
                                %pixeles
        end
    end
    if (contadorprimertercio<=2 && contadorsegundotercio<=2 && contadortercertercio<=2)
        %si existen menos de 2 plantillas circulares en un tercio
        claseB(1)=1;%condicion defectos cumple
    else
        claseB(1)=0;%condicion defectos no cumple
    end
end
Areautil=proppieza.Area-areacuadrado15cm-areacirculo;
if((Areautil)>=(0.6*proppieza.Area))%si area util es mayor al 60%
    claseB(3)=1;%condicion area cumple
else
    claseB(3)=0;%condicion area no cumple
end
end

    claseB=and(and(claseB(1),claseB(2)),claseB(3));%si cumple las 3 condiciones claseB=1
%% preclasificcion pieza de CLASE C
if (length(proplancircular)>6 || length(proplancu15cm)>3)%mas de 6 circulos y 3 cuadrados 1!
    claseC=1;%pieza de clase C
end
if(claseA==0 && claseB==0)%si no es clase A o clase B
    claseC=1;%pieza de clase C
end
%% clasificacion
if(claseB==1 && claseA==1)%Si es pieza de clase A y clase B
    claseA=1;%pieza de clase A
    claseB=0;%NO pieza de clase C
end

if(claseA==1 && claseB==0 && claseC==0)%Si es pieza de clase A y No clase B y No clase C
    clase='A';%pieza de clase A
elseif(claseA==0 && claseB==1 && claseC==0)%Si NO es pieza de clase A y SI clase B y No cla:
    clase='B';%pieza de clase B
elseif(claseA==0 && claseB==0 && claseC==1)%Si NO es pieza de clase A y NO clase B y SI cla:
    clase='C'; %pieza de clase C
end
end
end
end

```

Anexo 4.8 Función control luminosidad superior

```
function controlluminosidadsuperior
global valorluzencendida valorluzapagada imundgray vid
%% inicializacion de variables
cont=0;
ciclo=0;
ciclo1=5;
%% delimitacion valores maximos y minimos aceptables de cantidad luminica
limiteinfvalorluzencendida=0.98*valorluzencendida;%limite inferior aceptable para luz encen
limitesupvalorluzencendida=1.20*valorluzencendida;%limite superior aceptable para luz encen
%% control luminico
if valorluzapagada>=valorluzencendida%si valor de luminosidad apagada la luminaria
    %es mayor a valor de luminosidad encendida la luminaria
    apagarluzsuperior%se apaga la luminaria superior
else%si valor de luminosidad apagada la luminaria es menor a valor de luminosidad encendida

    while ciclo==0 && ciclo1==5%ciclo video encendido
        aumentarluminosidadsuperior%aumenta en 1 la luminosidad de la luminaria superior
    %% adquisicion de valores luminico en tiempo real
        if (vid.FramesAcquired>=60)%cantidad de capturas adquiridas mayor a 60
            im=getsnapshot(vid);%adquiere imagen
            flushdata(vid)%elimina datos de imagenes almacenadas en variable
            load('cameraParameters.mat','cameraParams');%carga de parametros para eliminar
                %distorsiones
            imundrgb=undistortImage(im,cameraParams);%elimina distorsion
            imundgray=rgb2gray(imundrgb);%convierte imagen de rgb a escala de grises
            imundgray=imrotate(imundgray,180,'nearest');%gira la imagen 180 grados
            valoractual=imundgray(540,960);%extrae valor de pixel ubicado en posicion(540,9)
            cont=cont+1;%aumenta contador guia para evitar desbordamiento de repeticion
    %% evaluacion de control luminico
        if(valoractual>=limiteinfvalorluzencendida &&...
            valoractual<=limitesupvalorluzencendida) %intervalo aceptable de luminosidad
            ciclo1=6;%sale de ciclo de video encendido
        end
        if(valoractual<=limiteinfvalorluzencendida && cont>150)%si valor actual es
            %inferior al limite minimo aceptable y a aumentado 150 veces
            ciclo1=6;%sale del ciclo de video encendido
        end
        if(valoractual>limitesupvalorluzencendida)%si valor actual es superior
            %al limite maximo aceptable
            while valoractual>limitesupvalorluzencendida
                im=getsnapshot(vid);%adquiere imagen
                flushdata(vid)%elimina datos de imagenes almacenadas en variable
                imundrgb=undistortImage(im,cameraParameters);%elimina distorsion
                imundgray=rgb2gray(imundrgb);%convierte imagen de rgb a escala de grises
                imundgray=imrotate(imundgray,180,'nearest');%gira la imagen 180 grados
                valoractual=imundgray(540,960);%extrae valor de pixel ubicado en posicion(540,9)
                disminuirluminosidadsuperior%disminuye luminosidad hasta apagarla
                flushdata(vid)%borra datos de variable vid
            end
            ciclo1=6;
        end

        flushdata(vid)%borra datos de variable vid
    end
end
%% finalizacion de adquisicion de valores luminicos
stop(vid)%detiene adquisicion de video
flushdata(vid)%borra datos de variable vid
end
end
```

Anexo 4.9 Función deteccionpieza

```
function deteccionpieza
global vid imcor imundgraycon imbin imselec imselecta propimcor2 propimcor1
%% adquisicion de imagen y procesamiento(eliminar distorsion y mesa)
flushdata(vid)%elimina datos d imagenes almacenadas en variable
eliminadistorsion %llama a funcion eliminadistorsion
stop(vid)%detiene adquisicionde imagenes
flushdata(vid)%elimina datos de imagenes almacenadas en variable
eliminamesa%llama a funcion eliminamesa
%% en caso de no encontrar pieza mostrar imagen en blanco
if (imcor==ones(1080,1920))%siimagen esta en blanco
imshow(ones(1080,1920));%muestra imagen en blanco
pause(0.01)%pausa
flushdata(vid)%elimina datos de imagenes almacenadas en variables
else
%% llamar a funcion areapieza
areapieza%llama a funcion area pieza
end
end
```

Anexo 4.10 Función deteccionplantilla

```
function deteccionplantilla
global vid imcor imundgraycon imfiltcom datosregiones2 imrellena opcion ...
    implantillas implantilla imcontmorf a impiezarellena
%% adquisicion de imagen y procesamiento (eliminar distorsion y mesa)
    flushdata(vid)%borra memoria variable vid
    eliminadistorsion%llama a funcion eliminadistorsion
    stop(vid)%detiene grabacion de video
    flushdata(vid)%borra memoria variable vid
    eliminamesa%llama a funcion eliminamesa
    if (imcor==ones(1080,1920))
        imshow(ones(1080,1920));%muestra imagen en blanco
        pause(0.01)%pausa
        flushdata(vid)%borra memoria variable vid
    else
%% resalte de contornos
        opcion=0;
        eliminamesa%aumentadocontraste%llama a funcion eliminamesa%aumentadocontraste
        imcondet=edge(imundgraycon,'sobel');%deteccion de contornos a imagen contrastada
%% realce de características
        imcontmorf=bwmorph(imcondet,'dilate');%operacion morfologica dilate 1 vez para
            %aclarar bordes
        imcontmorf=bwmorph(imcontmorf,'bridge');%operacion morfologica bridge, usada para
            %conectar pixeles cercanos sueltos
        imcontmorf=bwmorph(imcontmorf,'diag');%operacion morfologica diag, usada para
            %conectar en diagonal pixeles cercanos sueltos
%% discercion de pixeles
        implantilla=and(imcontmorf,bwmorph(impiezarellena,'dilate',3));%se elimina pixeles
            %de bordes que esten fuera de la pieza de cuero
%% seleccion borde plantilla
        plantillabordecon=bwconncomp(implantilla);%conecta pixeles formando regiones
        propreregion=regionprops(plantillabordecon,'Perimeter');%obtiene propiedad de cada region
        busqueda=find([propreregion.Perimeter]<(600));%localiza regiones con perimetros menor
            %al 10% de la imagen total
        imbordesplantillas=ismember(labelmatrix(plantillabordecon),busqueda);%toma las regiones
            %de la imagen conectada que cumplen con la busqueda anterior
%% relleno de imagen plantilla
        imrellena1=imfill(imbordesplantillas,'holes');%rellena el interior de las plantilla
        sinperfil=and(imrellena1,~implantilla);%elimina bordes de plantillas
%% seleccion plantillas
        plantillacon=bwconncomp(sinperfil);%conecta pixeles formando regiones
        propreregion=regionprops(plantillacon,'Area');%obtiene propiedad de cada region
        busqueda=find([propreregion.Area]>300);%localiza regiones con perimetros menor al 10% de :
            %imagen total
        plantillas=ismember(labelmatrix(plantillacon),busqueda);%toma las regiones de la imagen
            %conectada que cumplen con la busqueda anterior
        imfiltcom=bwconncomp(plantillas);%conecta pixeles con valor 1 formando regiones
        datosregiones2=regionprops(imfiltcom,'Area','Centroid');%recolecta informacion de
            %propiedades decada una de las regiones
        implantillas=or(~impiezarellena,plantillas);%une imagen de pieza con plantillas
        imshow(implantillas)
    end
end
```

Anexo 4.11 Función disminuir luminosidad superior

```
function disminuir_luminosidad_superior
%% carga de dato a enviar en variable
dato=num2str(2);%carga en dato el valor 2 en formato string
%% busqueda de puertos serial conectados en el ordenador
serialInfo = instrhwinfo('serial');%busqueda de puertos seriales conectados
serialdisponible=serialInfo.AvailableSerialPorts;%devuelve el nombre del puerto activo
verificacionserial=isempty(serialdisponible);%comprueba estado de variable serialdisponible
if(verificacionserial==1)
else
%% configuracion del puerto serial
    serialconfig=serial(char(serialdisponible));%crea puerto serial
    set(serialconfig,'BaudRate',9600);%carga frecuencia de comunicacion
    set(serialconfig,'DataBits',8);%cantidad de bits por caracter
    set(serialconfig,'Parity','none');%sin bit de paridad
    set(serialconfig,'StopBits',1);%bit de parada
    set(serialconfig,'FlowControl','none');%sin control de flujo
%% transmision de dato por puerto serial
    fopen(serialconfig);%abre puerto serial configurado
    fprintf(serialconfig,'%s',dato);%transmite variable dato
    fprintf(serialconfig,13);%transmite caracter ASCII de control ENTER
    fclose(serialconfig);%cierra puerto serial configurado
    delete(serialconfig);%borra la variable serialconfig
    clear serialconfig;%limpia la variable serialconfig
end
end
```

Anexo 4.12 Función eliminar distorsión

```
function eliminar_distorsion
global vid imundrgb imundgray imundbin elimdist im
%% adquisicion de imagen
    im=getsnapshot(vid);%adquiere imagen
    flushdata(vid);%elimina datos de imagenes almacenadas en variable
%% elimina distorsion
    load('cameraParameters.mat', 'cameraParams')%carga de parametros para eliminar
                                                %distorsiones
    imundrgb=undistortImage(im,cameraParams); %elimina distorsion
%% procesamiento de imagen
    imundgray=rgb2gray(imundrgb);%convierte imagen de rgb a escala de grises
    imundbin=im2bw(imundgray,0.9);%convierte imagen de escala de gris a blanco y negro
    elimdist=1;%carga valor 1 (indicando que se concluyo el proceso)
end
```

Anexo 4.13 Función eliminamesa

```
function eliminamesa
global imundgray imundgraycon imbin imcor
%% distinción entre pieza y el entorno
imundgraycon=imadjust(imundgray);%incrementa contraste de la imagen
imundgraycon=medfilt2(imundgraycon);%filtro mediana (elimina ruido salt&peper)
umbral=graythresh(imundgraycon);%calculo de umbral
imbin=im2bw(imundgraycon,umbral-0.2);%transforma a binario
%% resalte de características
imbin=bwmorph(imbin,'dilate',2);%dilata pixeles
imb=bwmorph(imbin,'skel');%elimina bordes
imbn=~imb;%invierte valores de la imagen
imbn=bwmorph(imbn,'dilate',2);%expande los pixeles
%% disernimiento de pixeles
etiquetaobj=logical(imbn);%etiqueta objetos conectados
seccion=regionprops(etiquetaobj,'Image');%secciona la imagen
marco=seccion(1).Image;%separa objeto con etiqueta 1 de la imagen
imcor=or(imb,marco);%suma la imagen con borde negro y la imagen del marco
imcor=bwmorph(imcor,'dilate');%dilata pixeles en los bordes
end
```

Anexo 4.14 Función eliminamesaadocontraste

```
function eliminamesaadocontraste
global imundgraycon imbin opcion imcor imselecta
%% distinción entre pieza y el entorno
filtro=fspecial('unsharp');%filtro de realce de contraste por enfoque
imundgraycon=roifilt2(filtro,imundgraycon,imbin);%filtra la imagen por enfoque
umbral=graythresh(imundgraycon);%calculo de umbral
imbin=im2bw(imundgraycon,umbral+0.15);%transforma a binario
%% resalte de características
imb=bwmorph(imbin,'skel');%elimina bordes
imbn=~imb;%invierte valores de la imagen
imbn=bwmorph(imbn,'dilate',4);% expande los pixeles
%% disernimiento de pixeles
etiquetaobj=logical(imbn); %etiqueta objetos conectados
seccion=regionprops(etiquetaobj,'Image');%secciona la imagen
marco=seccion(1).Image;%separa objeto con etiqueta 1 de la imagen
imcor=or(imb,marco);%suma la imagen con borde negro y la imagen del marco
imcor=bwmorph(imcor,'dilate',3);% dilata pixeles en los bordes para eliminar
    %partes negras no queridas

%% disernimiento de piezas por areas
if opcion==1
imcorcomp=bwconncomp(~imcor);%conecta pixeles con valor 1 formando regiones
propimcor=regionprops(imcorcomp,'Area');%recolecta informacion de propiedades(Area)
    %de cada una de las regiones
var3=find([propimcor.Area]>207036);%seleccion de datos necesarios de acuerdo
    %a la medida de las plantillas
imselecta=ismember(labelmatrix(imcorcomp),var3);%si es miembro extrae seccion
    %de imagen anterior a nueva imagen
else
end
end
```

Anexo 4.15 Función encenderluzinferior

```
function encenderluzinferior
%% carga de dato a enviar en variable
dato=num2str(5);%carga en dato el valor 5 en formato string
%% busqueda de puertos serial conectados en el ordenador
serialInfo = instrhwinfo('serial');%busqueda de puertos seriales conectados
serialdisponible=serialInfo.AvailableSerialPorts;%devuelve el nombre del puerto activo
verificacionserial=isempty(serialdisponible);%comprueba estado de variable serialdisponible
if(verificacionserial==1)
else
%% configuracion del puerto serial
    serialconfig=serial(char(serialdisponible));%crea puerto serial
    set(serialconfig,'BaudRate',9600);%carga frecuencia de comunicacion
    set(serialconfig,'DataBits',8);%cantidad de bits por caracter
    set(serialconfig,'Parity','none');%sin bit de paridad
    set(serialconfig,'StopBits',1);%bit de parada
    set(serialconfig,'FlowControl','none');%sin control de flujo
%% transmision de dato por puerto serial
    fopen(serialconfig);%abre puerto serial configurado
    fprintf(serialconfig,'%s',dato);%transmite variable dato
    fprintf(serialconfig,13);%transmite caracter ASCII de control ENTER
    fclose(serialconfig);%cierra puerto serial configurado
    delete(serialconfig);%borra la variable serialconfig
    clear serialconfig;%limpia la variable serialconfig
end
end
```

Anexo 4.16 Función encenderluzsuperior

```
function encenderluzsuperior
%% carga de dato a enviar en variable
dato=num2str(3);%carga en dato el valor 3 en formato string
%% busqueda de puertos serial conectados en el ordenador
serialInfo = instrhwinfo('serial');%busqueda de puertos seriales conectados
serialdisponible=serialInfo.AvailableSerialPorts;%devuelve el nombre del puerto activo
verificacionserial=isempty(serialdisponible);%comprueba estado de variable serialdisponible
if(verificacionserial==1)
else
%% configuracion del puerto serial
    serialconfig=serial(char(serialdisponible));%crea puerto serial
    set(serialconfig,'BaudRate',9600);%carga frecuencia de comunicacion
    set(serialconfig,'DataBits',8);%cantidad de bits por caracter
    set(serialconfig,'Parity','none');%sin bit de paridad
    set(serialconfig,'StopBits',1);%bit de parada
    set(serialconfig,'FlowControl','none');%sin control de flujo
%% transmision de dato por puerto serial
    fopen(serialconfig);%abre puerto serial configurado
    fprintf(serialconfig,'%s',dato);%transmite variable dato
    fprintf(serialconfig,13);%transmite caracter ASCII de control ENTER
    fclose(serialconfig);%cierra puerto serial configurado
    delete(serialconfig);%borra la variable serialconfig
    clear serialconfig;%limpia la variable serialconfig
end
end
```


Anexo 4.17 Función imagenclase

```
function imagenclase
global datosregiones2 imrellena
%% inicializacion de variables
cont=0;
%% visualizacion imagen pieza
imshow(~imrellena);% muestra imagen de la pieza
%% adision de cruz representando a plantilla en imagen pieza
hold on
for k=1:length(datosregiones2)
    cont=cont+1;%aumenta contador en 1
    plot(datosregiones2(k).Centroid(cont),datosregiones2(k).Centroid(cont+1),...
        'go','marker','+','markersize',20,'linewidth',3,'color','w')
        %pone cruz blanca en el centro donde estan las plantillas
    cont=0;
end
hold off
end
```

Anexo 4.18 Función video

```
function video
global vid%crea la variable global vid
%% seleccion de dispositivo
vid = videoinput('winvideo', 2, 'MJPG_1920x1080');%selecciona el dispositivo de entrada
%% configuracion de parametros
vid.FramesPerTrigger = Inf;%numero de fotos por disparo = infinito
vid.ReturnedColorspace = 'rgb';%formato de color
%% inicializa adquisicion
start(vid);%inicia adquisicion
end
```

Anexo 4.19 Función imprimir

```
function imprimir
global hfig
%% configuracion orientacion hoja
orient 'landscape';%orientacion de la hoja a imprimir (HORIZONTAL)
%% determinacion de impresora
impresora='Canon MG2400 series Printer';%nombre impresora preseleccionada
%% busqueda de impresora conectada
[~,printers] = findprinters;%busqueda de impresoras en el sistema
impresoraconectada=find(ismember(printers,{impresora}), 1);
                                % comprobacion de existencia la
                                %impresora pre seleccionada en el sistema
verificacionimpresora=isempty(impresoraconectada); %si existe carga 0 si no existe carga 1
if(verificacionimpresora==1)%si no existe en el sistema no realiza impresion
else
                                %si existe en el sistema
%% impresion
print(hfig,['-P',impresora])%imprime la ventana grafica oculta
end
end
```

Anexo 4.20 Función rn_comprobacionluminaria

```
function rn_comprobacionluminaria
global imundbin imundbinresize validacion1 validacion iluminf ilumsup
%% carga de datos de entrada y salida
%% ingreso de imagenes de referencia
imagen1=im2double(reshape(imread('todoapagadobin.bmp'), [],1));%imagen transformada a vector
%de 1 columna y 220 filas
imagen2=im2double(reshape(imread('lamparaencendidabin.bmp'), [],1));%imagen transformada a
%vector de 1 columna y 220 filas
imagen3=im2double(reshape(imread('mesaencendidabin.bmp'), [],1));%imagen transformada a
%vector de 1 columna y 220 filas
imagen4=im2double(reshape(imread('todoencendidobin.bmp'), [],1));%imagen transformada a
%vector de 1 columna y 220 filas

%vector de entrada de referencia
Entradas=[imagen1 imagen2 imagen3 imagen4];%carga de imagenes en vector de entrada
%vector de salida de referencia
Salidas=[0 0 1 1;0 1 0 1];%carga de imagenes en vector de salida
%fila 1 luminaria inferior, fila 2 luminaria superior
%(0 apagado 1 encendido)

%% configuracion de la red neuronal
Red=newff(Entradas,Salidas,50,{'logsig','logsig'},'trainscg');%50 neuronas en capa oculta,
%y funcion de transferencia sigmoidal

Red.divideFcn='';%sin divicion de datos
Red.inputs{1}.processFcns={};%sin funcion de pre procesamiento de datos para la entrada
Red.outputs{2}.processFcns={};%sin funcion de pre procesamiento de datos para la salida
Red.trainParam.goal=0;%objetivo de rendimiento 0
Red.trainParam.epochs=200;%maximo numero de iteraciones permitidas 200

%% entrenamiento de la red neuronal y primera validacion
Red=train(Red,Entradas,Salidas);
%validacion de red entrenada
validacion=sim(Red,Entradas);

%% Validacion de la red con nueva imagen
%carga de imagen a validar
imundbinresize=imresize(bwmorph(imundbin,'dilate',50),0.01,'nearest');%carga de nueva
%imagen y procesamiento
imagenavalidar=im2double(reshape(imundbinresize, [],1));%imagen transformada a vector
%de 1 columna y 220 filas

%validacion con la nueva imagen
validacion1=sim(Red,imagenavalidar);
ntraintool('close')% cierra ventana ntraintool

%% reconocimiento de resultados
%iluminacion de mesa y lampara encendida
if (validacion1(1)>=validacion(7) && validacion1(2)>=validacion(8))
    iluminf=1;
    ilumsup=1;

%iluminacion de mesa apagada o con falla y lampara encendida
elseif (validacion1(1)<validacion(7) && validacion1(2)>=validacion(8))
    iluminf=0;

    ilumsup=1;

%iluminacion de mesa encendida y lampara apagada o con falla
elseif (validacion1(1)>=validacion(7) && validacion1(2)<validacion(8))
    iluminf=1;
    ilumsup=0;

%iluminacion de mesa y lampara apagada o con falla
elseif (validacion1(1)<=validacion(7) && validacion1(2)<validacion(8))
    iluminf=0;
    ilumsup=0;
end
end
```

Anexo 4.21 Programa basededatos

```
function varargout = basededatos(varargin)
% BASEDEDATOS MATLAB code for basededatos.fig
%   BASEDEDATOS, by itself, creates a new BASEDEDATOS or raises the existing
%   singleton*.
%
%   H = BASEDEDATOS returns the handle to a new BASEDEDATOS or the handle to
%   the existing singleton*.
%
%   BASEDEDATOS('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in BASEDEDATOS.M with the given input arguments.
%
%   BASEDEDATOS('Property','Value',...) creates a new BASEDEDATOS or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before basededatos_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to basededatos_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help basededatos

% Last Modified by GUIDE v2.5 14-Nov-2015 22:57:21

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @basededatos_OpeningFcn, ...
                  'gui_OutputFcn',  @basededatos_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before basededatos is made visible.
function basededatos_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to basededatos (see VARARGIN)

% Choose default command line output for basededatos
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
%asignacion de parametros iniciales
axes(handles.axes1);%llama a axes1
imshow(imread('logogamacueros.jpg'));%lee y muestra imagen logo empresa
axes(handles.axes2); %llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
set(handles.edit1,'string','Ingrese el codigo aqui');%carga texto el edit text

%creacion de direccion para almacenaje
if ~exist('C:\SISTEMA_CLASE_AREA\imagenes','dir')%si no existe la direccion
    mkdir('C:\SISTEMA_CLASE_AREA\imagenes');%crea direccion
end
if ~exist('C:\SISTEMA_CLASE_AREA\basededatosexcel','dir')%si no existe la direccion
    mkdir('C:\SISTEMA_CLASE_AREA\basededatosexcel');%crea direccion
end

% UIWAIT makes basededatos wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = basededatos_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
global codigocuero
codigocuero=get(handles.edit1,'string');

if ~exist(strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigocuero,'.jpg'),'file')
    %busca la imagen por su codigo si no existe

```

```

set(handles.edit1,'string','Ingrese correctamente el codigo')%carga en edit text frase
set(handles.text2,'string',strcat('Usted ingreso: ',codigocuero))
    %carga en static text el texto ingresado
axes(handles.axes1);%llama a axes1
imshow(imread('logogamacueros.jpg')); %lee y muestra imagen logo empresa
else%si existe
    set(handles.edit1,'string','CODIGO CORRECTO')%carga en edit text frase
    axes(handles.axes1);%llama a axes1
    imshow(imread(strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigocuero,'.jpg')));
        %lee y muestra imagen correspondiente a codigo
    set(handles.text2,'string','') %pone en blanco static text
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: delete(hObject) closes the figure
delete(hObject);

% --- Executes on button press in imprimir.
function imprimir_Callback(hObject, eventdata, handles)
% hObject    handle to imprimir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global hfig codigocuero
if ~exist(strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigocuero,'.jpg'),'file')
    %busca la imagen por su codigo, si no existe
    set(handles.edit1,'string','Imagen a imprimir no existente') %carga en edit text frase
    set(handles.text2,'string',strcat('Usted ingreso: ',codigocuero))
        %carga en static text el texto ingresado
    axes(handles.axes1);%llama a axes1
    imshow(imread('logogamacueros.jpg'));%lee y muestra imagen logo empresa
else %si existe

hfig=figure('visible','off');%abre ventana grafica pero la mantiene oculta
set(gcf,'PaperType','A4') %configura tamaño de papel
imshow(imread(strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigocuero,'.jpg')));
    %lee y muestra imagen correspondiente a codigo
imprimir%llama a funcion imprimir
close(hfig)%cierra ventana grafica
end

```

Anexo 4.22 Programa clase_area

```
function varargout = class_area(varargin)
% CLASE_AREA MATLAB code for classearea.fig
%   CLASE_AREA, by itself, creates a new CLASE_AREA or raises the existing
%   singleton*.
%
%   H = CLASE_AREA returns the handle to a new CLASE_AREA or the handle to
%   the existing singleton*.
%
%   CLASE_AREA('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in CLASE_AREA.M with the given input arguments.
%
%   CLASE_AREA('Property','Value',...) creates a new CLASE_AREA or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before class_area_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to class_area_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help class_area

% Last Modified by GUIDE v2.5 26-Mar-2016 10:48:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @class_area_OpeningFcn, ...
                  'gui_OutputFcn',  @class_area_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before class_area is made visible.
function class_area_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

```

% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to clase_area (see VARARGIN)

% Choose default command line output for clase_area
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% Get default command line output from handles structure
varargout{1} = handles.output;

% UIWAIT makes clase_area wait for user response (see UIRESUME)
% uiwait(handles.figure1);
%inicializacion de variables
global val ciclo ciclo1 datoarea datoclase controlcalibracion ventanaemergente seguridad
val=0;%val 0 indica que no existe area
ciclo=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
ciclo1=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
datoarea=0;%carga variable co 0(indicando que no se obtenido el area)
datoclase=0;%carga variable co 0(indicando que no se obtenido la clase)
ventanaemergente=0;
%carga variable con 0(indicando que se cierra ventana sin preguntar si se desea salir)
%asignacion de parametros iniciales
load('controlcalibracion.mat');
%carga archivo con dato indicador de si se realizo verificacion de parametros o no)
encenderluzinferior %llama a funcion (enciende la luminaria inferior)
axes(handles.axes1); %llama a axes1
imshow(imread('logogamacueros.jpg'));%lee y muestra imagen logo empresa
axes(handles.axes2);%llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
axes(handles.axes3);%llama a axes2
imshow(imread('piezasdereferencia.PNG'));%lee y muestra imagen logo UTN
set(handles.checkmediarea, 'value', 0);%carga a checkbox de medicion de area con valor 0
set(handles.checkclasifallas, 'value', 0);
        v%carga a checkbox de clasificacion por fallas con valor 0
set(handles.clasecuero, 'String', ' ');
        %pone en blanco el static text correspondiente a la clase de pieza
set(handles.areafinalcm, 'String', ' ');
        %pone en blanco el static text correspondiente al area de pieza en [cm]
set(handles.areafinalpies, 'String', ' ');
        %pone en blanco el static text correspondiente al area de pieza en [in]
%creacion de direccion para almacenaje
if ~exist('C:\SISTEMA_CLASE_AREA\imagenes', 'dir')%si no existe la direccion
    mkdir('C:\SISTEMA_CLASE_AREA\imagenes'); %crea direccion
end
if ~exist('C:\SISTEMA_CLASE_AREA\basededatosexcel', 'dir')%si no existe la direccion
    mkdir('C:\SISTEMA_CLASE_AREA\basededatosexcel'); %crea direccion
end

```



```

if ~exist('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx','file')
    %si no existe archivo
    texto={'BIENVENIDO',' AL SISTEMA','DE VISION','ARTIFICIAL';'PARA EL',...
        'CONTROL DE','CALIDAD','EN HOJAS';'',',','DE CUERO',''};
    %texto de bienvenida
    xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
        texto,'INICIO')%crea archivo
end
%contol de calibracion
if controlcalibracion==0%si variable indicando que no a obtenido parametros
    resp=questdlg('Desea continuar sin calibrar el sistema?','ADVERTENCIA',...
        'Si','No','No'); %pregunta para proseguir
    switch resp
        case 'No'%si respuesta es no
            ventanaemergente=1;
            %carga variable con 1(indicando que se cierra ventana sin preguntar si se desea salir)
            case 'Si'%si respuesta es Si
                seguridad=sprintf('N');
                %carga letra N(indicando que no se realizo la calibracion del sistema)
            end
        else
            ventanaemergente=0;%carga variable con 1(indicando que hay una ventana emergente)
            seguridad=sprintf('Y');
            %carga letra Y(indicando que se realizo la calibracion del sistema)
        end
end

% --- Outputs from this function are returned to the command line.
function varargout = clase_area_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in botonguardar.
function botonguardar_Callback(hObject, eventdata, handles)
% hObject handle to botonguardar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global area2calstropies clase datoarea datoclase imrellena seguridad ...
    posicionanterior ventanaemergente
if ventanaemergente==1
    %si se habrio ventana indicadora de no haber parametros de inicio
    close clase_area; %cierra ventana clase_area
    verificaciondeparametrosiniciales
    %abre ventana de verificacion de parametros iniciales
else
%obtencion de hora y fecha

```

```

reloj=clock;%obtiene datos de hora y fecha del sistema
fecha=sprintf('%d-%d-%d',reloj(1),reloj(2),reloj(3));
    %selecciona datos correspondientes a la fecha
hora=sprintf('%d-%d',reloj(4),reloj(5));
    %selecciona datos correspondientes a la hora
areaaprox=num2str(roundn(str2double(area2calstrpies),-2));
    %redondea el valor de area con dos decimales
%guardar segun datos obtenidos
if (datoarea==1 && datoclase==1)%si se tiene dato de area y clase
    axes(handles.axes1);%llama a axes1
    imagenclase%muestra imagen de clase modificada
    pause(0.01)%pausa
    codigoalcl1= strcat(seguridad,'A',areaaprox,'C',clase,fecha,hora);
        %concatena datos en un string para mostrar
    set(title(gca,codigoalcl1,'fontsize',20),'Position',[960,1080,100]);
        %muestra el codigo de el cuero analizado en ubicacion
    impieza = getframe(gca);%obtiene imagen del axes
    iml=and(im2bw(impieza.cdata),imread('logo.bmp'));
        %imagen de pieza con codigo y logo gamacueros
    imareareducido=imresize(imarea,0.10);
        %reduce imagen correspondiente a area al 10%
    imagenarea=padarray(imareareducido,[394 562],'replicate','post' );
        %imagen del area minimizada
    imfinal=and(iml,imagenarea);%une imagen del area,de la clase y el logo
    imwrite(imfinal, strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigoalcl1,'.jpg'));
        %guarda imagen obtenida
    hfig=figure('visible','off');%abre ventana grafica pero la mantiene oculta
    set(gcf,'PaperType','A4') %configura tamaño de papel
    imshow(imfinal)%muestra imagen a imprimir
    imprimir%llama a funcion imprimir
    close(hfig)%cierra ventana grafica
    [A,B] = xlsfinfo('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx');
        %obtiene informacion del archivo .xlsx
    if any(strcmp(B,fecha))
        %comprueba si existe hoja de calculo con el nombre de la fecha actual
        load('posicionanterior.mat')
        %carga archivo con dato de la ultima fila en la que se escribio
        posicionanterior=posicionanterior+1;%aumenta en uno la posicion anterior
        save('posicionanterior.mat','posicionanterior');%guarda la nueva posicion
        posicion= strcat('A',num2str(posicionanterior));
            %selecciona columna A y la posicion que le toca
        xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
            {codigoalcl1},fecha,posicion);
            %escribe en el archivo de excel el codigo en la hoja con la fecha indicada
            %y en la casilla que le tocaba si no existe hoja de calculo con la fecha actual
        posicionanterior=1;%carga variable con valor1 (indicando la primera fila)
        save('posicionanterior.mat','posicionanterior');%guarda la nueva posicion
        posicion= strcat('A',num2str(posicionanterior));
            %selecciona columna A y la posicion que le toca
        xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
            {codigoalcl1},fecha,posicion);
    end
end

```

```

        %escribe en el archivo de excel el codigo, en la hoja con la fecha indicada
        %y en la casilla que le tocaba
end

elseif (datoarea==1 && datoclase==0)%si se tiene dato de area y no de la clase
axes(handles.axes1);%llama a axes1
imshow(imarea);%muestra imagen de la pieza
pause(0.01)%pausa
codigoalcl0=strcat(seguridad,'A',areaaprox,'C','0',fecha,hora);
    %concatena datos en un string para mostrar
set(title(gca,codigoalcl0,'fontsize',20),'Position',[960,1080,100]);
    % muestra el codigo de el cuero analizado en ubicacion
impieza = getframe(gca);%obtiene imagen del axes
imfinal=and(im2bw(impieza.cdata),imread('logo.bmp'));
    %imagen de pieza con codigo y logo gamacueros
imwrite(imfinal, strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigoalcl0, '.jpg'));
    %guarda imagen obtenida
hfig=figure('visible','off');%abre ventana grafica pero la mantiene oculta
set(gcf,'PaperType','A4')%configura tamaño de papel
imshow(imfinal)%muestra imagen a imprimir
imprimir%llama a funcion imprimir
close(hfig)%cierra ventana grafica
[A,B] = xlsfinfo('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx');
    %obtiene informacion del archivo .xlsx
if (strcmp(B,fecha)%comprueba si existe hoja de calculo con
    %el nombre de la fecha actual
load('posicionanterior.mat')
    %carga archivo con dato indicador de la ultima posicion en la que se escribio
posicionanterior=posicionanterior+1;
    %aumenta en uno la posicion anterior
save('posicionanterior.mat','posicionanterior');%guarda la nueva posicion
posicion=strcat('A',num2str(posicionanterior));
    %selecciona columna A y la posicion que le toca
xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
    {codigoalcl0,fecha,posicion});
    %escribe en el archivo de excel el codigo,en la hoja con la fecha indicada
    %y en la casilla que le tocaba si no existe hoja de calculo con la fecha actual
posicionanterior=1;%carga variable con valor1 (indicando la primera fila)
save('posicionanterior.mat','posicionanterior');%guarda la nueva posicion
posicion=strcat('A',num2str(posicionanterior));
    %selecciona columna A y la posicion que le toca
xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
    {codigoalcl0,fecha,posicion});
    %escribe en el archivo de excel el codigo,en la hoja con la fecha indicada
    %y en la casilla que le tocaba
end

elseif (datoarea==0 && datoclase==1)%no se tiene dato de area y si de la clase
axes(handles.axes1);%llama a axes1
imagenclase%muestra imagen de clase modificada
pause(0.01)%pausa

```

```

codigoa0c11=strcat(seguridad,'A','0','C',clase,fecha,hora);
    %concatena datos en un string para mostrar
set(title(gca,codigoa0c11,'fontsize',20),'Position',[960,1080,100]);
    % muestra el codigo de el cuero analizado en ubicacion
impieza = getframe(gca);%obtiene imagen del axes
imfinal=and(im2bw(impieza.cdata),imread('logo.bmp'));
    %imagen de pieza con codigo y logo gamacueros
imwrite(imfinal, strcat('C:\SISTEMA_CLASE_AREA\imagenes\',codigoa0c11, '.jpg'));
    %guarda imagen obtenida
hfig=figure('visible','off'); %abre ventana grafica pero la mantiene oculta
set(gcf,'PaperType','A4');%configura tamaño de papel
imshow(imfinal)%muestra imagen a imprimir
imprimir%llama a funcion imprimir
close(hfig)%cierra ventana grafica
[A,B] = xlsfinfo('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx');
    %obtiene informacion del archivo .xlsx
if any(strcmp(B,fecha))
    %comprueba si existe hoja de calculo con el nombre de la fecha actual
load('posicionanterior.mat')
    %carga archivo con dato indicador de la ultima posicion en la que se escribio
posicionanterior=posicionanterior+1;%aumenta en uno la posicion anterior
save('posicionanterior.mat','posicionanterior');%guarda la nueva posicion
posicion=strcat('A',num2str(posicionanterior));
    %selecciona columna A y la posicion que le toca
xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
    {codigoa0c11,fecha,posicion});
    %escribe en el archivo de excel el codigo,%en la hoja con la fecha indicada
    %y en la casilla que le tocaba si no existe hoja de calculo con la fecha actual
posicionanterior=1;%carga variable con valor1 (indicando la primera fila)
save('posicionanterior.mat','posicionanterior');%guarda la nueva posicion
posicion=strcat('A',num2str(posicionanterior));
    %selecciona columna A y la posicion que le toca
xlswrite('C:\SISTEMA_CLASE_AREA\basededatosexcel\basededatosexcel.xlsx',...
    {codigoa0c11,fecha,posicion});%escribe en el archivo de excel el codigo,
    %en la hoja con la fecha indicada y en la casilla que le tocaba
end
else%no se tiene dato de area y tampoco de la clase
axes(handles.axes1);%llama a axes1
imshow(ones(1080,1920));%muestra imagen en blanco
pause(0.01)%pausa
end
set(handles.clasecuero,'String',' ');
    %pone en blanco el static text correspondiente a la clase de pieza
set(handles.areafinalcm,'String',' ');
    %pone en blanco el static text correspondiente al area de pieza en [cm]
set(handles.areafinalpies,'String',' ');
    %pone en blanco el static text correspondiente al area de pieza en [in]
set(handles.checkmediarea,'value',0);
    %carga a checkbox de medicion de area con valor 0
set(handles.checkclasifallas,'value',0)
    %carga a checkbox de clasificacion por fallas con valor 0

```

```

datoarea=0;%resetea variable
datoclase=0;%resetea variable
end

% --- Executes on button press in botonmedirarea.
function botonmedirarea_Callback(hObject, eventdata, handles)
% hObject    handle to botonmedirarea (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global area2calstr area2calstrpies imrellena vid ciclo ciclo1 datoarea ventanaemergente ima:
if ventanaemergente==1%si se habrio ventana indicadora de no haber parametros de inicio
    close clase_area;%cierra ventana clase_area
    verificaciondeparametrosiniciales%abre ventana de verificacion de parametros iniciales
else
%inicializacion de variables
ciclo=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
ciclo1=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
%asignacion de parametros iniciales
axes(handles.axes1); %llama a axes1
imshow(imread('logogamacueros.jpg'));%lee y muestra imagen logo empresa
axes(handles.axes2);%llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
apagarluzsuperior%llama a funcion (apaga la luminaria superior)
set(handles.checkmediarea,'value',0);%carga a checkbox de medicion de area con valor 0
set(handles.areafinalcm,'String',' ');
    %pone en blanco el static text correspondiente al area de pieza en [cm]
set(handles.areafinalpies,'String',' ');
    %pone en blanco el static text correspondiente al area de pieza en [in]
resp=questdlg('Revise que no se encuentren objetos sobre la pieza de cuero, en caso de que '
    %pregunta para proseguir
    switch resp
        case 'SI hay Objetos'%si respuesta es si hay objetos
            return;%retorna a ventana clase_area
        case 'NO hay Objetos'%si respuesta es no hay objetos
            video%llama a funcion video
            ciclo=1;%carga variable con 1(indica video inicializado)
        end
    end
    while ciclo==1 && ciclo1==0
        if (vid.FramesAcquired>=90)%cantidad de capturas adquiridas mayor a 90
            deteccionpieza%llama a funcion deteccion pieza
            calculoarea %llama a funcion calculo area
            set(handles.checkmediarea,'value',1) %carga a checkbox de medicion de area
            axes(handles.axes1);%llama a axes1
            imshow(~imrellena);%visualiza pieza en binario
            pause(0.01)%pausa
            imarea=~imrellena;%imagen del area
            set(handles.areafinalcm,'String',strcat('Área: ',area2calstr,'cm2')); %carga
            set(handles.areafinalpies,'String',strcat('Área: ',area2calstrpies,'ft2'));
            datoarea=1;%carga variable con 1(indica que se a completado el proceso)
            ciclo1=1;%carga variable con 1(indica ciclo concluido)
        end
    end

    flushdata(vid) %borra datos cargados en la memoria de variable vid
end

% --- Executes on button press in botonclasificacionfallas.
function botonclasificacionfallas_Callback(hObject, eventdata, handles)
% hObject    handle to botonclasificacionfallas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ciclo vid ciclo1 clase imrellena datoclase imundgray valorluzapagada valorluzencend:
if ventanaemergente==1
    close clase_area;
    verificaciondeparametrosiniciales %abre ventana de verificacion de parametros ini:
else

```

```

%inicializacion de variables
ciclo=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
ciclo1=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
%asignacion de parametros iniciales
axes(handles.axes1);%llama a axes1
imshow(imread('logogamacueros.jpg'));%lee y muestra imagen logo empresa
axes(handles.axes2);%llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
encenderluzsuperior%llama a funcion (enciende la luminaria superior)
set(handles.checkclasifallas,'value',0)
    %carga a checkbox de clasificacion por fallas con valor 0
set(handles.clasecuero,'String',strcat(''));
    %pone en blanco el static text correspondiente al area de pieza en [cm]
resp=questdlg('A colocado las plantillas de referencia en fallas sobre el cuero? ',...
    ' PLANTILLAS PARA FALLAS','SI','NO','NO'); %pregunta para proseguir
switch resp
    case 'NO' %si respuesta es NO
        return;%retorna a ventana clase_area
    case 'SI' %si respuesta es SI
        apagarluzsuperior%llama a funcion (apaga la luminaria superior)
        video%llama a funcion video
        ciclo=1;%carga variable con 1(indica video inicializado)
    end
%deteccion de pieza y obtencion de valor luminaria apagada
while ciclo==1 && ciclo1==0
    if (vid.FramesAcquired>=60)%cantidad de capturas adquiridas mayor a 60
        deteccionpieza%llama a funcion deteccion pieza
        valorluzapagada=imundgray(540,960);
            %extraccion valor de pixel en escala de grises(0(obscuros) a 255(claros))
        if imrellena==~ones(1080,1920)%si imagen esta totalmente en negro
            imshow(ones(1080,1920));%muestra imagen en blanco
            pase(0.01)%pausa
            ciclo=0;%termina proceso de clasificacion
        else%si se detecto pieza
            proppieza=areal;%obtiene propiedad de la pieza
            axes(handles.axes2);%llama a axes2
            imshow(~imrellena);%visualiza pieza en binario
            pause(0.01)%pausa
            ciclo1=1;%carga variable con 1 (activa siguiente ciclo)
        end
    end
    flushdata(vid)
end
%deteccion de pieza y obtencion de valor luminaria encendida
start(vid);%activa obtencion de imagenes
while ciclo==1 && ciclo1==1
    encenderluzsuperior%llama a funcion (enciende la luminaria superior)
    if (vid.FramesAcquired>=60)%cantidad de capturas adquiridas mayor a 60
        deteccionpieza%llama a funcion deteccion pieza
        apagarluzsuperior%llama a funcion (apaga la luminaria superior)
        valorluzencendida=imundgray(540,960);
    end
end

```

```

        %extraccion valor de pixel en escala de grises(0(obscuros) a 255(claros))
        ciclol=2;%carga vaiable con 2 (activa siguiente ciclo)
    end
    flushdata(vid)%borra datos cargados en la memoria de variable vid
end
%deteccion de plantilla y control de luminosidad
start(vid);%activa obtencion de imagenes
while ciclo==1 && ciclol==2
    controlluminosidadsuperior%llama a funcion que controla la cantidad de luminosidad
    if (vid.FramesAcquired>=60)%cantidad de capturas adquiridas mayor a 60
        axes(handles.axes2);%llama a axes2
        deteccionplantilla %llama a funcion deteccion de plantilla
        apagarluzsuperior%llama a funcion (apaga la luminaria superior)
        clasificacion%llama a funcion claificacion
        axes(handles.axes1);%llama a axes1
        imshow(implantillas)%muestra imagen de plantillas
        set(handles.clasecuero,'String',strcat('Pieza de clase:',clase));
        % carga string clase
        set(handles.checkclasifallas,'value',1)
        %carga a checkbox de clasificacion por fallas con valor 0
        axes(handles.axes2);%llama a axes2
        imshow(imread('utnlogo.jpg'));%Lee y muestra imagen logo UTN
        datoclase=1;%carga variable con 1(indica que se a completado el proceso)
        ciclol=3;%carga variable con 3(indica ciclo concluido)
    end
    flushdata(vid)%borra datos cargados en la memoria de variable vid
end
flushdata(vid)%borra datos cargados en la memoria de variable vid
end

% --- Executes on button press in checkclasifallas.
function checkclasifallas_Callback(hObject, eventdata, handles)
% hObject    handle to checkclasifallas (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkclasifallas

% --- Executes on button press in checkmediarea.
function checkmediarea_Callback(hObject, eventdata, handles)
% hObject    handle to checkmediarea (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkmediarea

% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global ciclo vid ciclol ventanaemergente
if ventanaemergente==1
    apagarluzsuperior           %llama a funcion (apaga la luminaria superior)
    apagarluzinferior          %llama a funcion (apaga la luminaria inferior)
    delete(hObject);          %cierra la ventana
else
    if(ciclo==1)                %si variable tiene valor 1
        stop(vid)              %detiene video
        flushdata(vid)         %borra datos cargdos en la memoria de variable vid
    end
    resp=questdlg('¿Desea salir del menu de clasificación?', 'SALIR', 'Si', 'No', 'No');
                                %llama a ventana de pregunta y carga con texto
    switch resp
        case 'No'               %si respuesta es NO
            return;             %regresa a la vetana actual
        case 'Si'               %si respuesta es SI
            apagarluzsuperior    %llama a funcion (apaga la luminaria superior)
            apagarluzinferior    %llama a funcion (apaga la luminaria inferior)
            ciclo=0;             %carga valor 0 a variable (indica video detenido)
            ciclol=3;            %carga valor 0 a variable (indica ciclo concluido)
            delete(hObject);     %cierra la ventana
        end
    end
end

```


Anexo 4.23 Programa principal

```
function varargout = principal(varargin)
% PRINCIPAL MATLAB code for principal.fig
% PRINCIPAL, by itself, creates a new PRINCIPAL or raises the existing
% singleton*.
%
% H = PRINCIPAL returns the handle to a new PRINCIPAL or the handle to
% the existing singleton*.
%
% PRINCIPAL('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in PRINCIPAL.M with the given input arguments.
%
% PRINCIPAL('Property','Value',...) creates a new PRINCIPAL or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before principal_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to principal_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help principal

% Last Modified by GUIDE v2.5 13-Nov-2015 13:05:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @principal_OpeningFcn, ...
                  'gui_OutputFcn', @principal_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before principal is made visible.
function principal_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
```

```

% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to principal (see VARARGIN)

% Choose default command line output for principal
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes principal wait for user response (see UIRESUME)
% uiwait(handles.figure1);
axes(handles.axes1);           %Carga la imagen
imshow(imread('logogamacueros.jpg')); %Lee y muestra imagen logo empresa
axes(handles.axes2);           %Carga la imagen
imshow(imread('utnlogo.jpg'));  %Lee y muestra imagen logo UTN

% --- Outputs from this function are returned to the command line.
function varargout = principal_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in botonbasedatos.
function botonbasedatos_Callback(hObject, eventdata, handles)
% hObject    handle to botonbasedatos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
basededatos%llama a programa basededatos

% --- Executes on button press in botoncalibracionsistema.
function botoncalibracionsistema_Callback(hObject, eventdata, handles)
% hObject    handle to botoncalibracionsistema (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
verificaciondeparametrosiniciales%llama a programa verificaciondeparametrosiniciales

% --- Executes on button press in botonsistemaclasificacion.
function botonsistemaclasificacion_Callback(hObject, eventdata, handles)
% hObject    handle to botonsistemaclasificacion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clase_area%llama a programa clase_area

% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global controlcalibracion
controlcalibracion=0;%carga varianlecon 0 (indicando que se debe calibrar)
save('controlcalibracion.mat','controlcalibracion');%almacena variable
apagarluzsuperior%llama a funcion (apaga la luminaria superior)
apagarluzinferior%llama a funcion (apaga la luminaria inferior)
delete(hObject);%cierra ventana

```

Anexo 4.24 Programa verificaciondeparametrosiniciales

```
function varargout = verificaciondeparametrosiniciales(varargin)
% VERIFICACIONDEPARAMETROSINICIALES MATLAB code for verificaciondeparametrosiniciales.fig
%   VERIFICACIONDEPARAMETROSINICIALES, by itself, creates a new VERIFICACIONDEPARAMETROSINICIALES
%   singleton*.
%
%   H = VERIFICACIONDEPARAMETROSINICIALES returns the handle to a new VERIFICACIONDEPARAMETROSINICIALES
%   the existing singleton*.
%
%   VERIFICACIONDEPARAMETROSINICIALES('CALLBACK',hObject,eventData,handles,...) calls the
%   function named CALLBACK in VERIFICACIONDEPARAMETROSINICIALES.M with the given input arguments.
%
%   VERIFICACIONDEPARAMETROSINICIALES('Property','Value',...) creates a new VERIFICACIONDEPARAMETROSINICIALES
%   existing singleton*. Starting from the left, property value pairs are applied to the GUI before
%   verificaciondeparametrosiniciales_OpeningFcn gets called. An unrecognized property name or invalid
%   value makes property application stop. All inputs are passed to verificaciondeparametrosiniciales_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help verificaciondeparametrosiniciales

% Last Modified by GUIDE v2.5 13-Feb-2016 15:54:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @verificaciondeparametrosiniciales_OpeningFcn, ...
                  'gui_OutputFcn',  @verificaciondeparametrosiniciales_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before verificaciondeparametrosiniciales is made visible.
function verificaciondeparametrosiniciales_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to verificaciondeparametrosiniciales (see VARARGIN)

% Choose default command line output for verificaciondeparametrosiniciales
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes verificaciondeparametrosiniciales wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% asignacion de parametros iniciales
axes(handles.axes1); %llama a axes1
imshow(imread('logogamacueros.jpg')); %lee y muestra imagen logo empresa
axes(handles.axes2); %llama a axes2
imshow(imread('utnlogo.jpg')); %Lee y muestra imagen logo UTN
set(handles.checkelimdist, 'value', 0) %carga a checkbox de eliminacion de distorsiones con va
set(handles.checkiluminf, 'value', 0) %carga a checkbox de iluminación inferior con valor 0
set(handles.checkilumsup, 'value', 0) %carga a checkbox de iluminación superior con valor 0
set(handles.checkareaobj, 'value', 0) %carga a checkbox de la area del objeto con valor 0
set(handles.texto, 'String', ' '); %pone en blanco el static text

% creacion de direccion para almacenaje
if ~exist('C:\SISTEMA_CLASE_AREA\imagenes', 'dir') % si no existe la direccion
    mkdir('C:\SISTEMA_CLASE_AREA\imagenes'); % crea direccion
end
if ~exist('C:\SISTEMA_CLASE_AREA\basededatosexcel', 'dir') % si no existe la direccion
    mkdir('C:\SISTEMA_CLASE_AREA\basededatosexcel'); % crea direccion
end

%%

% --- Outputs from this function are returned to the command line.
function varargout = verificaciondeparametrosiniciales_OutputFcn(handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in botonverificar.
function botonverificar_Callback(hObject, eventdata, handles)
% hObject     handle to botonverificar (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%%
global vid elimdist iluminf ilumsup val imundbin imundbinresize imrellena ...
    ciclo controlcalibracion

```

```

elimdist=0;%iniciamos variable eliminar distorsion en 0
iluminf=0;%iniciamos variable iluminacion inferior en 0
ilumsup=0;%iniciamos variable iluminacion superior en 0
ciclo=0;%iniciamos variable ciclo en 0 (indica video no inicializado)
%reinicio menu
axes(handles.axes1);%llama a axes1
imshow(imread('logogamacueros.jpg'));%lee y muestra imagen logo empresa
axes(handles.axes2);%llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
set(handles.checkelimdist,'value',0)%carga a checkbox de eliminacion de distorsiones con
set(handles.checkiluminf,'value',0)%carga a checkbox de iluminación inferior con valor 0
set(handles.checkilumsup,'value',0)%carga a checkbox de iluminación superior con valor 0
set(handles.checkareaobj,'value',0)%carga a checkbox de la area del objeto con valor 0
set(handles.texto,'String',' ');%pone en blanco el static text
resp=questdlg('Existe algun objeto sobre la mesa de trabajo?','PREGUNTA','Si','No','No');
%llama a ventana de pregunta y carga con texto

switch resp
    case 'No'%respuesta no
        encenderluzinferior%llama a funcion (enciende la luminaria inferior)
        encenderluzsuperior%llama a funcion (enciende la luminaria superior)
        video%llama a funcion video
        ciclo=1;%iniciamos variable ciclo en 1 (indica video inicializado)
    case 'Si'%respuesta si
        %no realiza accion
end
while ciclo==1%mientras video inicializado
%comprobacion de luminaria
if (vid.FramesAcquired>=90)%cantidad de capturas adquiridas mayor a 90
    eliminadistorsion%adquisicion de imagen y eliminacion de distorsiones
    stop(vid)%detiene video
    flushdata(vid)%borra datos cargados en la memoria de variable vid
    rn_comprobacionluminaria%llama funcion comprobacion de luminaria
    axes(handles.axes1);%llama a axes1
    imshow(imundbin);%lee y muestra imagen en binario
    axes(handles.axes2);%llama a axes2
    imshow(imundbinresize);%lee y muestra imagen en binario reducida de tamaño
    %% calculo de parametros
    if (elimdist==1&&iluminf==1&&ilumsup==1)
        set(handles.checkelimdist,'value',elimdist)
        %carga a checkbox de eliminacion de distorsiones con valor 1
        set(handles.checkiluminf,'value',iluminf)
        %carga a checkbox de iluminación inferior con valor 1
        set(handles.checkilumsup,'value',ilumsup)
        %carga a checkbox de iluminación superior con valor 1
        resp=questdlg('¿Se encuentra la pieza de calibración en la mesa de trabajo?','...
        'PREGUNTA','SI','NO','NO'); %llama a ventana de pregunta y carga con texto
        if strcmp(resp,'SI') %si respuesta es afirmativa
            apagarluzsuperior%llama a funcion (apaga la luminaria superior)
            video %llama a funcion video
            axes(handles.axes1);%llama a axes1
            calculodeparametros%llama a funcion de calculo de parametros
        end
    end
end

```

```

if (val==0)%val 0 indica que no existe area por error en el proceso
set(handles.texto,'String',strcat('PROCESO CONCLUIDO SIN EXITO'));
                                     %carga texto en el static text
set(handles.checkareaobj,'value',val);
                                     %carga a checkbox area del objeto con valor 1
axes(handles.axes1);%llama a axes1
imshow(imrellena);%lee y muestra imagen de solo la pieza
axes(handles.axes2);%llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
controlcalibracion=0;
                                     %carga variable con 0(indicando que no a obtenido parametros)
save('controlcalibracion.mat','controlcalibracion');
                                     %guarda variable en formato .mat
else%val 1 indicando que se tiene un area
set(handles.texto,'String',strcat('PROCESO CONCLUIDO CON EXITO'));
                                     %carga el valor del area en el static text
set(handles.checkareaobj,'value',val);
                                     %carga a checkbox de la distancia del objeto con valor 1
axes(handles.axes1);%llama a axes1
imshow(imrellena);%lee y muestra imagen de solo la pieza
axes(handles.axes2);%llama a axes2
imshow(imread('utnlogo.jpg'));%lee y muestra imagen logo UTN
controlcalibracion=1;
                                     %carga variable con 1(indicando que si a obtenido parametros)
save('controlcalibracion.mat','controlcalibracion');
                                     %guarda variable en formato .mat
end
end
%fallos en el sistema de verificacion de parametros
elseif (elimdist==1&&iluminf==0&&ilumsup==1)
set(handles.checkelimdist,'value',elimdist)
                                     %carga a checkbox de eliminacion de distorsiones con valor 1
set(handles.checkiluminf,'value',iluminf)
                                     %carga a checkbox de iluminaci3n inferior con valor 0
set(handles.checkilumsup,'value',ilumsup)
                                     %carga a checkbox de iluminaci3n superior con valor 1
errordlg('FALLO DE ILUMINACI3N INFERIOR, VERIFICAR CIRCUITO','ALERTA')
                                     %llama a ventana de error y carga con texto
set(handles.checkareaobj,'value',0)
                                     %carga a checkbox de area del objeto con valor 0
elseif (elimdist==1&&iluminf==1&&ilumsup==0)
set(handles.checkelimdist,'value',elimdist)
                                     %carga a checkbox de eliminacion de distorsiones con valor 1
set(handles.checkiluminf,'value',iluminf)
                                     %carga a checkbox de iluminaci3n inferior con valor 1
set(handles.checkilumsup,'value',ilumsup)
                                     %carga a checkbox de iluminaci3n superior con valor 0
errordlg('FALLO DE ILUMINACI3N SUPERIOR, VERIFICAR CIRCUITO','ALERTA')
                                     %llama a ventana de error y carga con texto
set(handles.checkareaobj,'value',0)
                                     %carga a checkbox de area del objeto con valor 0

```

```

elseif (elimdist==1&&iluminf==0&&ilumsup==0)
    set(handles.checkelimdist,'value',elimdist)
        %carga a checkbox de eliminacion de distorsiones con valor 1
    set(handles.checkiluminf,'value',iluminf)
        %carga a checkbox de iluminación inferior con valor 0
    set(handles.checkilumsup,'value',ilumsup)
        %carga a checkbox de iluminación superior con valor 0
    errordlg('FALLO DE ILUMINACIÓN INFERIOR Y SUPERIOR, VERIFICAR CIRCUITOS',...
        'ALERTA') %llama a ventana de error y carga con texto
    set(handles.checkareaobj,'value',0)
        %carga a checkbox de area del objeto con valor 0
else
    set(handles.checkelimdist,'value',elimdist)
        %carga a checkbox de eliminacion de distorsiones con valor 0
    set(handles.checkiluminf,'value',iluminf)
        %carga a checkbox de iluminación inferior con valor 0
    set(handles.checkilumsup,'value',ilumsup)
        %carga a checkbox de iluminación superior con valor 0
    set(handles.checkareaobj,'value',0)
        %carga a checkbox de area del objeto con valor 0
    set(handles.texto,'String',' ');
        %pone en blanco el static text
    errordlg('NO SE PUEDE CORREGIR LA DISTORSION, LOS DATOS OBTENIDOS PODRIAN NO SER OPTIMO!
        'ALERTA')%llama a ventana de error y carga con texto
end
ciclo=0;%carga valor 0 a variable (indica video detenido)
end
flushdata (vid)%borra datos cargdos en la memoria de variable vid
end

%%

% --- Executes on button press in checkilumsup.
function checkilumsup_Callback(hObject, eventdata, handles)
% hObject    handle to checkilumsup (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkilumsup

% --- Executes on button press in checkelimdist.
function checkelimdist_Callback(hObject, eventdata, handles)
% hObject    handle to checkelimdist (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkelimdist

% --- Executes on button press in checkareaobj.

```

```

function checkkareaobj_Callback(hObject, eventdata, handles)
% hObject    handle to checkkareaobj (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkkareaobj

% --- Executes on button press in checkkiluminf.
function checkkiluminf_Callback(hObject, eventdata, handles)
% hObject    handle to checkkiluminf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkkiluminf

% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%%
global ciclo vid

    if(ciclo==1)                %si variable tiene valor 1
    stop(vid)                   %detiene video
    flushdata(vid)              %borra datos cargdos en la memoria de variable vid
    end
    resp=questdlg('¿Desea salir del menu de calibración?', 'SALIR', 'Si', 'No', 'No');
                                %llama a ventana de pregunta y carga con texto
    switch resp
        case 'No'
            return;             %regresa a ventana actual
        case 'Si'
            apagarluzsuperior    %llama a funcion (apaga la luminaria superior)
            apagarluzinferior    %llama a funcion (apaga la luminaria inferior)
            ciclo=0;              %carga valor 0 a variable (indica video detenido)
            delete(hObject);      %Hint: delete(hObject) closes the figure
    end

```


ANEXO 5 PROGRAMACIÓN MICROCONTROLADOR

ANEXO 5

Programación microcontrolador

```
/*
This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
*/
```

```
Project : atmega88pa_luminaria
Version : 1.0
Date : 05/03/2016
Author : Manuel Fuentes
Company : Universidad Tecnica del Norte
Comments: Trabajo de Grado
```

```
Chip type : ATmega88PA
Program type : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256
*/
```

```
#include <mega88a.h>

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

void main(void)
{
// Declare your local variables here
int ilum, subrilusu=1, babrilusu=2, enlusu=3, apalusu=4, enluinf=5, apaluinf=6, val=200, limen=255, limapa=110;

// Crystal Oscillator division factor: 1
#pragma optsize-
CLKPR=0x80;
CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif

// Input/Output Ports initialization
// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=0 State1=T State0=T
PORTD=0x00;
DDRD=0x04;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OCA output: Disconnected
// OCB output: Disconnected
TCCR0A=0x00;
TCCR0B=0x00;
TCNT0=0x00;
OCR0A=0x00;
OCR0B=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OCA output: Discon.
// OCB output: Discon.
```

```

// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2A output: Disconnected
// OC2B output: Disconnected
ASSR=0x00;
TCCR2A=0x00;
TCCR2B=0x00;
TCNT2=0x00;
OCR2A=0x00;
OCR2B=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// Interrupt on any change on pins PCINT0-7: Off
// Interrupt on any change on pins PCINT8-14: Off
// Interrupt on any change on pins PCINT16-23: Off
EICRA=0x00;
EIMSK=0x00;
PCICR=0x00;

// Timer/Counter 0 Interrupt(s) initialization
TIMSK0=0x00;

// Timer/Counter 1 Interrupt(s) initialization
TIMSK1=0x00;

// Timer/Counter 2 Interrupt(s) initialization
TIMSK2=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART0 Mode: Asynchronous
// USART Baud Rate: 9600
UCSR0A=0x00;
UCSR0B=0x10;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1:
ACSR=0x80;
ADCSRB=0x00;
DIDR1=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

while (1)

```

```

{
scanf("%d", &ilum);
if (ilum==enlusu)
{
PORTB=255;
}
if (ilum==apalusu)
{
PORTB=0;
}
if (ilum==enluinf)
{
PORTD=0b00000100;
}
if (ilum==apaluinf)
{
PORTD=0b00000000;
}

if (ilum==subrilusu)
{
val=val+1;
if (val>limen)
{
val=limen;
PORTB=val;
}
else
{
PORTB=val;
}
}
if (ilum==babrilusu)
{
val=val-1;
if (val<limapa)
{
val=160;
PORTB=0;
}
else
{
PORTB=val;
}
}
}
}

```

ANEXO 6 ESPECIFICACIONES COMPONENTES

ANEXO 6

Especificaciones componentes

Anexo 6.1 Dicroicos halógenos Sylvania

SYLVANIA

MR16 Coolfit Home

MR16 CLFT 12V 20W 38° GU5.3 SV

0022516



Range features

- Virtually all heat and light dissipated away from back of the lamp keeping fixture cool
- Wear resistant aluminium reflector for constant output and colour throughout life
- Easily recognisable by blue cement in lamp cap
- Can replace any 12V 50mm dichroic low voltage lamp
- Suitable for use in open luminaires following Luminaire Standard EN 60598-1
- Excellent colour rendering (CRI=100)
- Average rated life 3000hrs (66% longer life than standard MR16 lamps)



DESCRIPCIÓN DEL PRODUCTO

Número de orden	0022516
Color de la luz	829 Warm white
Tecnología	Halogen
Clase de energía	C
Promedio de vida útil (clasificada) (h)	3000
Acabado de lámpara	Aluminium
Forma de la lámpara	Reflector
Regulable	Yes
Casquillo/Base	GU5.3
Tipo	MR16
Código EAN	5410288225166
CRI (Ra)	100
Ángulo de haz (°)	38
Temperatura de color (K)	3000
Useful luminous flux (Rated) (lm)	180
Potencia (nominal) (W)	20
Tensión (V)	12

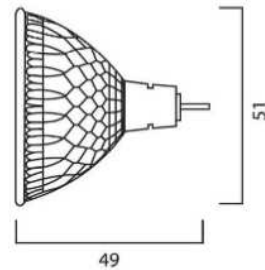
FOTOMETRÍA

Coolfit 50

		38°		
		20W	35W	50W
689mm Ø	1m	660lx	1200lx	1950lx
1377mm Ø	2m	165lx	300lx	488lx
2066mm Ø	3m	73lx	133lx	217lx
2755mm Ø	4m	41lx	75lx	122lx

DIBUJOS TÉCNICOS

GX/GU5.3

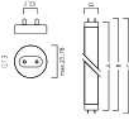


Anexo 6.2 Lámpara tubular Sylvania

T12 Estándar



- Gama amplia de lámparas fluorescentes T12 estándar en varias potencias y colores.
- Para empleo en luminarias e envolucros duros. El Estándar no es estanco.
- Ef. lumínico de máxima, en conformidad con EN 60321.
- 100W y 125W con inversión en el cátodo ofrecen una elevada reproducción cromática (Ra85Clases II).



	A	B	C	D
3000	3000	410	206	92mm
3500	3500	410	206	92mm
4000	3500	410	206	92mm
4500	3500	410	206	92mm
5000	3500	410	206	92mm
6000	3500	410	206	92mm
8000	3500	410	206	92mm
10000	3500	410	206	92mm
12500	3500	410	206	92mm

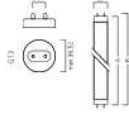
T12 ESTÁNDAR

Código	Descripción	Potencia Watt	Longitud mm	Diámetro mm	Temp. ambiente °C	Flujo luminoso lm	Clase de eficiencia energética
000150	T12 Estándar Blanca Universal (D1-L-18)	20W	570	0.230	25	1700	A
000151	T12 Estándar Blanca Universal (D1-L-18)	30W	670	0.230	25	2500	A
000152	T12 Estándar Blanca Universal (D1-L-18)	40W	770	0.230	25	3300	A
000153	T12 Estándar Blanca Universal (D1-L-18)	50W	870	0.230	25	4100	A
000154	T12 Estándar Blanca Universal (D1-L-18)	60W	970	0.230	25	4900	A
000155	T12 Estándar Blanca Universal (D1-L-18)	80W	1170	0.230	25	6500	A
000156	T12 Estándar Blanca Universal (D1-L-18)	100W	1370	0.230	25	8100	A
000157	T12 Estándar Blanca Universal (D1-L-18)	125W	1570	0.230	25	9700	A
000158	T12 Estándar Blanca Universal (D1-L-18)	20W	570	0.230	25	1700	A
000159	T12 Estándar Blanca Universal (D1-L-18)	30W	670	0.230	25	2500	A
000160	T12 Estándar Blanca Universal (D1-L-18)	40W	770	0.230	25	3300	A
000161	T12 Estándar Blanca Universal (D1-L-18)	50W	870	0.230	25	4100	A
000162	T12 Estándar Blanca Universal (D1-L-18)	60W	970	0.230	25	4900	A
000163	T12 Estándar Blanca Universal (D1-L-18)	80W	1170	0.230	25	6500	A
000164	T12 Estándar Blanca Universal (D1-L-18)	100W	1370	0.230	25	8100	A
000165	T12 Estándar Blanca Universal (D1-L-18)	125W	1570	0.230	25	9700	A
000166	T12 Estándar Blanca Universal (D1-L-18)	20W	570	0.230	25	1700	A
000167	T12 Estándar Blanca Universal (D1-L-18)	30W	670	0.230	25	2500	A
000168	T12 Estándar Blanca Universal (D1-L-18)	40W	770	0.230	25	3300	A
000169	T12 Estándar Blanca Universal (D1-L-18)	50W	870	0.230	25	4100	A
000170	T12 Estándar Blanca Universal (D1-L-18)	60W	970	0.230	25	4900	A
000171	T12 Estándar Blanca Universal (D1-L-18)	80W	1170	0.230	25	6500	A
000172	T12 Estándar Blanca Universal (D1-L-18)	100W	1370	0.230	25	8100	A
000173	T12 Estándar Blanca Universal (D1-L-18)	125W	1570	0.230	25	9700	A
000174	T12 Estándar Blanca Universal (D1-L-18)	20W	570	0.230	25	1700	A
000175	T12 Estándar Blanca Universal (D1-L-18)	30W	670	0.230	25	2500	A
000176	T12 Estándar Blanca Universal (D1-L-18)	40W	770	0.230	25	3300	A
000177	T12 Estándar Blanca Universal (D1-L-18)	50W	870	0.230	25	4100	A
000178	T12 Estándar Blanca Universal (D1-L-18)	60W	970	0.230	25	4900	A
000179	T12 Estándar Blanca Universal (D1-L-18)	80W	1170	0.230	25	6500	A
000180	T12 Estándar Blanca Universal (D1-L-18)	100W	1370	0.230	25	8100	A
000181	T12 Estándar Blanca Universal (D1-L-18)	125W	1570	0.230	25	9700	A

T12 Estándar / Luxline® Plus



- Gama amplia de lámparas fluorescentes T12 estándar en varias potencias y colores.
- Para empleo en luminarias e envolucros duros. El Estándar no es estanco.
- Ef. lumínico de máxima, en conformidad con EN 60321.
- 100W y 125W con inversión en el cátodo ofrecen una elevada reproducción cromática (Ra85Clases II).



	A	B	C	D
3000	3000	410	206	92mm
3500	3500	410	206	92mm
4000	3500	410	206	92mm
4500	3500	410	206	92mm
5000	3500	410	206	92mm
6000	3500	410	206	92mm
8000	3500	410	206	92mm
10000	3500	410	206	92mm
12500	3500	410	206	92mm

T12 ESTÁNDAR/LUXLINE® PLUS

Código	Descripción	Potencia Watt	Longitud mm	Diámetro mm	Temp. ambiente °C	Flujo luminoso lm	Clase de eficiencia energética
000190	T12 Estándar Blanca Plus (D1-L-18)	20W	570	0.230	25	1800	A
000191	T12 Estándar Blanca Plus (D1-L-18)	30W	670	0.230	25	2600	A
000192	T12 Estándar Blanca Plus (D1-L-18)	40W	770	0.230	25	3400	A
000193	T12 Estándar Blanca Plus (D1-L-18)	50W	870	0.230	25	4200	A
000194	T12 Estándar Blanca Plus (D1-L-18)	60W	970	0.230	25	5000	A
000195	T12 Estándar Blanca Plus (D1-L-18)	80W	1170	0.230	25	6600	A
000196	T12 Estándar Blanca Plus (D1-L-18)	100W	1370	0.230	25	8200	A
000197	T12 Estándar Blanca Plus (D1-L-18)	125W	1570	0.230	25	9800	A
000198	T12 Estándar Blanca Plus (D1-L-18)	20W	570	0.230	25	1800	A
000199	T12 Estándar Blanca Plus (D1-L-18)	30W	670	0.230	25	2600	A
000200	T12 Estándar Blanca Plus (D1-L-18)	40W	770	0.230	25	3400	A
000201	T12 Estándar Blanca Plus (D1-L-18)	50W	870	0.230	25	4200	A
000202	T12 Estándar Blanca Plus (D1-L-18)	60W	970	0.230	25	5000	A
000203	T12 Estándar Blanca Plus (D1-L-18)	80W	1170	0.230	25	6600	A
000204	T12 Estándar Blanca Plus (D1-L-18)	100W	1370	0.230	25	8200	A
000205	T12 Estándar Blanca Plus (D1-L-18)	125W	1570	0.230	25	9800	A
000206	T12 Estándar Blanca Plus (D1-L-18)	20W	570	0.230	25	1800	A
000207	T12 Estándar Blanca Plus (D1-L-18)	30W	670	0.230	25	2600	A
000208	T12 Estándar Blanca Plus (D1-L-18)	40W	770	0.230	25	3400	A
000209	T12 Estándar Blanca Plus (D1-L-18)	50W	870	0.230	25	4200	A
000210	T12 Estándar Blanca Plus (D1-L-18)	60W	970	0.230	25	5000	A
000211	T12 Estándar Blanca Plus (D1-L-18)	80W	1170	0.230	25	6600	A
000212	T12 Estándar Blanca Plus (D1-L-18)	100W	1370	0.230	25	8200	A
000213	T12 Estándar Blanca Plus (D1-L-18)	125W	1570	0.230	25	9800	A
000214	T12 Estándar Blanca Plus (D1-L-18)	20W	570	0.230	25	1800	A
000215	T12 Estándar Blanca Plus (D1-L-18)	30W	670	0.230	25	2600	A
000216	T12 Estándar Blanca Plus (D1-L-18)	40W	770	0.230	25	3400	A
000217	T12 Estándar Blanca Plus (D1-L-18)	50W	870	0.230	25	4200	A
000218	T12 Estándar Blanca Plus (D1-L-18)	60W	970	0.230	25	5000	A
000219	T12 Estándar Blanca Plus (D1-L-18)	80W	1170	0.230	25	6600	A
000220	T12 Estándar Blanca Plus (D1-L-18)	100W	1370	0.230	25	8200	A
000221	T12 Estándar Blanca Plus (D1-L-18)	125W	1570	0.230	25	9800	A

ANEXO 7 PLANOS

ANEXO 7

Planos

Anexo 7.1 Estructura superior

Anexo 7.2 Ángulo exterior

Anexo 7.3 Ángulo interior

Anexo 7.4 Viga central

Anexo 7.5 Viga exterior corta para anclaje

Anexo 7.6 Viga exterior corta

Anexo 7.7 Viga exterior larga

Anexo 7.8 Vigueta

Anexo 7.9 Mesa

Anexo 7.10 Cubierta

Anexo 7.11 Cajón luminoso

Anexo 7.12 Estructura inferior

Anexo 7.13 Tablero de distribución

Anexo 7.14 Placa de montaje

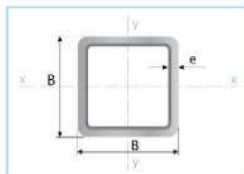
ANEXO 8 CATÁLOGO TUBERÍA ESTRUCTURAL

ANEXO 8

Catálogo tubería estructural

Anexo 8.1 Tubería estructural cuadrada

Tubería Estructural Cuadrada



Largo Normal:
6 metros
Recubrimiento:
Negro o Galvanizado
Norma de Calidad:
ASTM A 500 Gr. A, B ó C
Norma de Fabricación:
NTE INEN 2415
Espesores:
Desde 1,50 a 6,00 mm
Observaciones:
Otras dimensiones, espesores y
largos previa consulta

Aplicaciones

- Estructuras: columnas para galpones y naves industriales.
- Señalización y vialidad: alumbrado, soportes.
- Equipos y máquinas de todo tipo.
- Construcción en general.

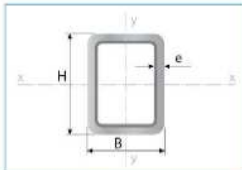
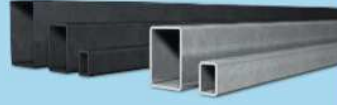
INEN



Designaciones	Área	Peso	Propiedades Estáticas			
			Eje x/x e y/y			
B	e	A	P	Momento de inercia	Módulo de resistencia	Radio de giro
mm	mm	cm ²	kg/m	cm ⁴	cm ³	cm
20	1,50	1,05	0,83	0,58	0,58	0,74
	2,00	1,34	1,05	0,69	0,69	0,72
25	1,50	1,35	1,06	1,41	1,11	0,97
	2,00	1,74	1,36	1,48	1,18	0,92
30	1,50	1,65	1,30	2,19	1,46	1,15
	2,00	2,14	1,68	2,71	1,81	1,12
	3,00	3,01	2,36	3,50	2,34	1,08
40	1,50	2,25	1,77	5,48	2,74	1,56
	2,00	2,94	2,31	6,92	3,46	1,53
	3,00	4,21	3,30	9,28	4,64	1,48
50	1,50	2,85	2,24	12,56	4,92	2,01
	2,00	3,74	2,93	14,13	5,65	1,94
	3,00	5,41	4,25	19,40	7,76	1,89
60	1,50	3,45	2,71	18,94	6,31	2,43
	2,00	4,54	3,56	25,12	8,37	2,35
	3,00	6,61	5,19	35,06	11,69	2,30
75	2,00	5,74	4,50	51,90	13,84	3,00
	3,00	8,41	6,60	76,13	19,99	2,97
	4,00	10,95	8,59	94,02	24,74	2,92
100	2,00	7,74	6,07	122,99	24,60	3,99
	3,00	11,41	8,96	190,94	37,53	4,04
	4,00	14,95	11,73	235,87	46,54	3,96
125	2,00	9,74	7,64	350,00	56,00	6,00
	3,00	14,41	11,31	355,00	56,80	4,96
	4,00	18,95	14,87	457,00	73,12	4,91
	6,00	27,63	21,69	1193,92	105,20	31,46
135	4,00	20,55	16,13	581,38	86,13	5,32
	5,00	25,36	19,90	704,23	104,33	5,27
	6,00	30,03	23,58	818,50	121,26	5,22

Anexo 8.2 Tubería estructural rectangular

Tubería Estructural Rectangular



Largo Normal:
6 metros
Recubrimiento:
Negro o Galvanizado
Norma de Calidad:
ASTM A 500 Gr. A, B & C
Norma de Fabricación:
NTE INEN 2415
Espesores:
Desde 1,50 a 6,00 mm
Observaciones:
Otras dimensiones, espesores y
largos previa consulta

➤ **Aplicaciones**

- Estructuras: galpones y naves industriales, edificios, soporte de techos.
- Automotriz y de autopartes: carrocería y remolques.
- Señalización y vialidad: soportes.
- Construcción en general.



Designaciones	Área	Peso	Propiedades Estáticas							
			Eje x-x				Eje y-y			
			Momento de inercia	Módulo de resistencia	Radio de giro	Momento de inercia	Módulo de resistencia	Radio de giro		
B	H	e	A	P	I	W	i	I	W	i
mm	mm	mm	cm ²	kg/m	cm ⁴	cm ³	cm	cm ⁴	cm ³	cm
20	40	1,50	1,65	1,30	3,26	1,63	1,40	1,09	1,09	0,81
		2,00	2,14	1,68	4,04	2,02	1,37	1,33	1,33	0,79
25	50	1,50	2,10	1,65	7,65	3,02	0,82	2,50	2,02	1,05
		2,00	2,74	2,15	8,37	3,35	1,75	2,80	2,24	1,01
		3,00	3,91	3,07	12,90	5,08	1,75	4,12	3,26	0,99
30	50	1,50	2,25	1,77	7,53	3,01	1,83	3,41	2,27	1,23
		2,00	2,94	2,31	9,52	3,81	1,80	4,28	2,85	1,21
		3,00	4,21	3,30	12,78	5,11	1,74	5,66	3,77	1,16
40	60	1,50	2,85	2,24	14,40	4,79	2,26	7,71	3,85	1,65
		2,00	3,74	2,93	18,39	6,13	2,22	9,81	4,90	1,62
		3,00	5,41	4,25	25,31	8,44	2,16	13,37	6,89	1,57
30	70	1,50	2,85	2,24	18,08	5,17	2,49	4,76	3,17	1,28
		2,00	3,74	2,93	22,20	6,34	2,44	5,85	3,90	1,25
		3,00	5,41	4,25	30,50	8,71	2,37	7,84	5,23	1,20
40	80	2,00	4,54	3,56	37,32	9,33	2,87	12,70	6,35	1,67
		3,00	6,61	5,19	52,16	13,04	2,81	17,49	8,75	1,63
		4,00	8,55	6,71	53,20	17,78	2,62	10,80	11,50	1,18
50	100	2,00	5,74	4,50	73,63	14,50	3,66	26,46	10,03	2,15
		3,00	8,41	6,60	113,57	22,29	3,61	38,40	15,08	2,10
		4,00	10,95	8,59	140,19	27,53	3,56	47,01	18,52	2,06
50	150	2,00	7,74	6,07	207,45	27,66	5,18	37,17	14,87	2,19
		3,00	11,41	8,96	317,82	41,79	5,21	56,16	22,12	2,18
		4,00	14,95	11,73	397,70	52,27	5,13	69,06	27,20	2,14
		5,00	18,36	14,41	455,54	60,74	4,98	77,40	30,96	2,05
75	175	3,00	14,41	11,31	556,65	63,62	6,22	149,40	39,84	3,22
		4,00	18,95	14,87	718,11	82,07	6,16	191,03	50,94	3,18
		5,00	23,36	18,33	867,95	99,19	6,10	228,83	61,02	3,13
		6,00	27,63	21,69	1.006,41	115,02	6,03	262,93	70,11	3,08
100	150	3,00	14,41	11,31	461,00	61,47	5,65	248,00	49,60	4,15
		4,00	18,95	14,87	619,84	81,28	5,69	331,14	65,22	4,17
		5,00	23,36	18,33	719,00	95,90	5,55	384,00	76,80	4,04
		6,00	27,63	21,69	873,60	114,71	5,59	461,76	90,95	4,06
200	70	3,00	15,61	12,25	749,65	214,19	6,93	145,09	14,51	0,96
		4,00	20,55	16,13	968,54	96,85	6,87	185,17	52,91	3,00
		5,00	25,36	19,90	1.172,89	117,29	6,80	221,55	63,30	2,96
		6,00	30,03	23,58	1.362,69	136,27	6,74	254,26	72,65	2,91

ANEXO 9 SELECCIÓN DE VIDRIO PARA CUBIERTA

ANEXO 9

Selección de vidrio para cubierta

TAPAS DE MESA

Para determinar la carga que puede soportar una tapa de mesa se tomaron ejemplos de espesores y dimensiones máximas establecidos en base a aplicaciones que resultaron satisfactorias a lo largo de muchos años de uso sin presentar una deflexión exagerada.

El Cuadro 2 permite determinar el espesor de **Float** adecuado para una tapa de mesa en función de su tamaño y de la relación entre las dimensiones de sus lados.

Los diferentes diseños y modos de soporte de una tapa **Float** deben ser evaluados antes de especificar el cristal.

El soporte adecuado deberá tener en cuenta la masa del **Float** -cuyo peso es igual a 2,5 kg/m² por milímetro de espesor- y los medios necesarios para impedir que el paño se desplace, deslice o incline.

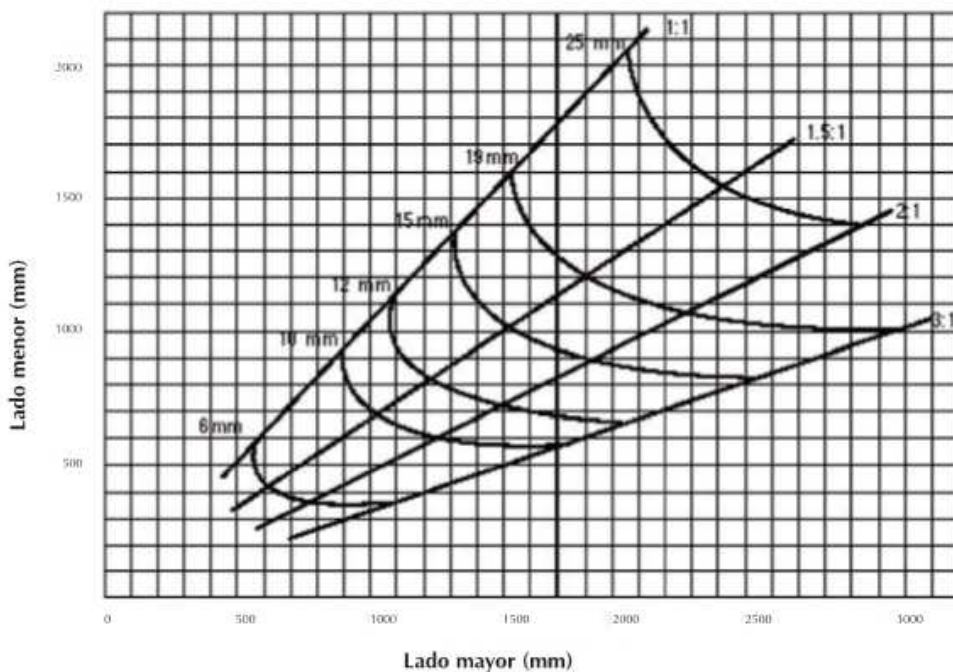
Los puntos de apoyo entre el **Float** y el soporte deben estar aislados para impedir cualquier contacto del cristal con metales y/o materiales de dureza equivalente.

CUADRO 2

Ejemplos de diferentes tapas de Float según sus dimensiones (mm)

Espesor (mm)	Relación entre lados		
	1:1	2:1	3:1
6	560x560	760x380	1080x360
10	915x915	1220x610	1740x580
12	1090x1090	1520x760	2040x680
15	1350x1350	1830x915	2580x860
19	1600x1600	2180x1090	3045x1015
25	2100x2100	2900x1450	- -

Figura 2 TAPAS DE MESA (Apoyadas en forma continua en sus cuatro lados)



ANEXO 10 COMPONENTES ELECTRÓNICOS

ANEXO 10

Componentes electrónicos



BTA/BTB12 and T12 Series

12A TRIACS

SNUBBERLESS™, LOGIC LEVEL & STANDARD

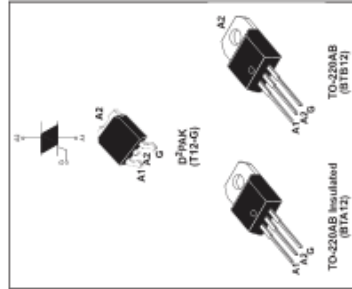
MAIN FEATURES:

Symbol	Value	Unit
$I_{T(RMS)}$	12	A
V_{CEM}/V_{RSM}	600 and 600	V
I_{GT} (a.)	5 to 50	mA

Available in through-hole or surface-mount packages, the BTA/BTB12 and T12 series can be used as an ON/OFF function in applications such as static relays, heating regulation, induction motor starting circuits... or for speed controllers. The standard 12A triac operation in light dimmers, motor speed controllers (BTA/BTB12...AV and T12 series) are specially recommended for use on inductive loads, thanks to their high commutation peak current capability. The standard 12A triac series are designed for surface-mount applications in microcontrollers. By using an internal ceramic pad, the BTA series provides voltage insulated lab standards (file ref.: E81734).

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
I_T (RMS)	RMS on-state current (0.1 ms pulse)	12	A
I_{TSM}	TO-220AB Ins.	$T_c = 100^\circ\text{C}$	A
	TO-220AB Ins.	$T_c = 90^\circ\text{C}$	
I_T	Non repetitive surge peak on-state current (0.1 ms pulse, $T_j = 25^\circ\text{C}$)	120	A
	$F = 60\text{ Hz}$	$t = 16.7\text{ ms}$	
I_T	t_1 value for latching	78	A
dI/dt	Critical rate of rise of on-state current	$F = 120\text{ Hz}$	A/ μs
	$I_g = 2\text{ A } I_{GT}, I_T \leq 100\text{ mA}$	$T_j = 125^\circ\text{C}$	
V_{CEM}/V_{RSM}	Non repetitive surge peak off-state voltage	$t_p = 10\text{ ms}$	V
I_{GM}	Peak gate current	$t_p = 20\text{ }\mu\text{s}$	
$P_{GM(AV)}$	Average gate power dissipation	4	W
T_{STG}	Storage junction temperature range	1	$^\circ\text{C}$
T_j	Operating junction temperature range	$-40\text{ to }+150$ $-40\text{ to }+125$	



BTA/BTB12 and T12 Series

ELECTRICAL CHARACTERISTICS ($T_j = 25^\circ\text{C}$, unless otherwise specified)

■ SNUBBERLESS™ and LOGIC LEVEL (3 Quadrants)

Symbol	Test Conditions	T12			BTA/BTB12			Unit
		T1235	SW	CW	SW	CW	BW	
I_{GT} (1)	$V_G = 12\text{ V}$ $R_L = 30\ \Omega$	MAX.	35	5	10	35	50	mA
V_{GT}		MAX.			1.3			V
V_{GO}	$V_G = V_{ORM}$ $I_T = 125^\circ\text{C}$	MAX.						V
I_H (2)	$I_T = 100\text{ mA}$	MAX.	35	10	15	35	50	mA
I_L	$I_T = 1.2\text{ } I_{GT}$	I-III	MAX.	50	10	25	50	70
		II	MAX.	69	15	30	60	60
dI/dt (2)	$V_G = 67\text{ } \mu\text{s}$ slow gate open $I_T = 125^\circ\text{C}$	MIN.	500	20	40	500	1000	V/ μs
dI/dt (2)	(drive) $V_G = 10\text{ V}$ fast $I_T = 125^\circ\text{C}$ Without snubber	MIN.	-	3.5	6.5	-	-	A/ μs
			6.5	-	-	6.5	12	

■ STANDARD (4 Quadrants)

Symbol	Test Conditions	Quadrant	BTA/BTB12			Unit	
			C	B	BW		
I_{GT} (1)	$V_G = 12\text{ V}$ $R_L = 30\ \Omega$	I-III-III IV	MAX.	25	50	100	mA
V_{GT}		ALL	MAX.			1.3	V
V_{GD}	$V_G = V_{GSM}$ $I_T = 500\text{ mA}$	ALL	MIN.			0.2	V
I_H (2)	$I_T = 1.2\text{ } I_{GT}$	I-III-IV	MAX.	25	50	50	mA
		II	MAX.	40	50	100	mA
dI/dt (2)	$V_G = 67\text{ } \mu\text{s}$ slow gate open $I_T = 125^\circ\text{C}$ (drive) $V_G = 5.3\text{ A}$ fast $I_T = 125^\circ\text{C}$	MIN.	MIN.	200	400	400	V/ μs

STATIC CHARACTERISTICS

Symbol	Test Conditions	Value		Unit
		MIN.	MAX.	
V_T (2)	$I_H = 17\text{ A}$ $t_p = 350\text{ }\mu\text{s}$ $T_j = 25^\circ\text{C}$	MAX.	1.55	V
V_G (2)	Threshold voltage $T_j = 125^\circ\text{C}$	MAX.	0.85	V
$R_{\theta jC}$ (2)	Dynamic resistance $T_j = 125^\circ\text{C}$	MAX.	35	m Ω
I_{GSM}	$V_{GSM} = V_{GSM}$ $T_j = 25^\circ\text{C}$	MAX.	5	μA
		$T_j = 125^\circ\text{C}$	MAX.	1

Note 1: I_{GSM} and I_{TSM} are measured at $t_p = 100\text{ ns}$.
Note 2: For both packages of BTA/BTB12 and T12.

TRIACS

FEATURES

- LOW I_T = 13mA max
- HIGH SURGE CURRENT : $I_{TSM} = 120A$
- I_{GT} SPECIFIED IN FOUR QUADRANTS
- INSULATING VOLTAGE = 2500V(RMS) (UL RECOGNIZED : E81734)



DESCRIPTION
The BTA10 GP's use high performance, glass passivated chips.
The insulated TO220AB package, the high surge current and low holding current make this family well adapted to LIGHT DIMMER applications.

ABSOLUTE RATINGS (limiting values)

Symbol	Parameter	Value	Unit
I_T (RMS)	RMS on-state current (50% conduction angle)	10	A
I_{TSM}	Non repetitive surge peak on-state current (T_J initial = 25°C)	125	A
	$t_p = 8.3$ ms	120	
I_{T1}	$t_p = 10$ ms	72	A 2s
	$t_p = 10$ ms	10	A μ s
dI/dt	Critical rate of rise of on-state current Gate supply: $I_G = 500mA$, $dI_G/dt = 1A/\mu s$	10	A μ s
T_{stg} T_J	Storage and operating junction temperature range	- 40 to + 150	°C
		- 40 to + 125	°C
T_l	Maximum lead temperature for soldering during 10 s at 4.5 mm from case	200	°C

Symbol	Parameter	BTA10-		Unit
		495 GP	600 GP	
V_{DRM} V_{PRM}	Repetitive peak off-state voltage $T_l = 125^\circ C$	400	600	V

BTA10 GP

THERMAL RESISTANCES

Symbol	Parameter	Value	Unit
$R_{th(j-a)}$	Junction to ambient	80	°C/W
$R_{th(j-c)}$	Junction to case for DC	4	°C/W
$R_{th(j-c)}$	Junction to case for 360° conduction angle ($F = 50$ Hz)	3	°C/W

GATE CHARACTERISTICS (maximum values)

P_G (AV) = 1W $P_{GM} = 10W$ ($t_p = 20 \mu s$) $I_{GM} = 4A$ ($t_p = 20 \mu s$) $V_{GDM} = 16V$ ($t_p = 20 \mu s$)

ELECTRICAL CHARACTERISTICS

Symbol	Test Conditions	Quadrant	Unit	
			GP	GP
I_{GT}	$V_D = 12V$ (DC) $R_L = 50\Omega$	II/III IV	MAX	mA
V_{GT}	$V_D = 12V$ (DC) $R_L = 50\Omega$	II/III	MAX	V
		IV	MIN	V
V_{GD}	$V_D = V_{DRM}$ $I_G = 3A/5\mu s$	II/III/IV	TYP	μs
I_G	$V_D = V_{DRM}$ $I_G = 500mA$ $dI_G/dt = 3A/\mu s$	II/III/IV	TYP	mA
I_{H1}	$I_G = 1.2$ IGT	II	MAX	mA
I_{VTM1}	$I_T = 100mA$ gate open	II	MAX	V
I_{DRM} I_{PRM}	$I_T = 14A$ $t_p = 35\mu s$	II/III	MAX	mA
		IV	MAX	mA
dV/dt	Linear slope up to $V_D = 67V/V_{DRM}$ gate open	II/III	MIN	V/ μs
		IV	MIN	V/ μs
$(dV/dt)_{crit}$	$(dI/dt)_{crit} = 2.5A/\mu s$	II/III/IV	TYP	V/ μs

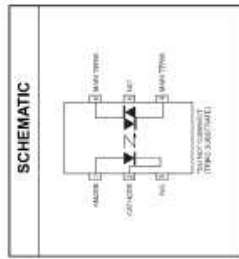
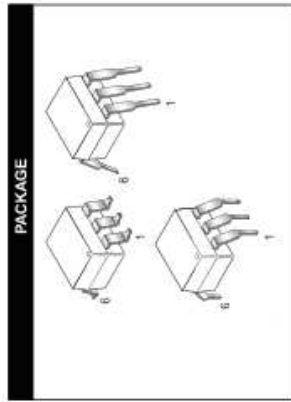
* For other polarity of electrode A, voltage with reference to electrode A.

Anexo 10.3 Datasheet Moc-3021

FAIRCHILD
SEMICONDUCTOR®

**6-PIN DIP RANDOM-PHASE
OPTOISOLATORS TRIAC DRIVER OUTPUT
(250/400 VOLT PEAK)**

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M



DESCRIPTION

The MOC301XM and MOC302XM series are optically isolated triac driver devices. These devices contain a GaAs infrared emitting diode and a light activated silicon lateral switch, which functions like a triac. They are designed for interfacing between electronic controls and power loads to control resistive and inductive loads for 115 VAC operations.

FEATURES

- Excellent f_T stability—IR emitting diode has low degradation
- High isolation voltage—minimum 5300 VAC RMS
- Underwriters Laboratory (UL) recognized—File #E60700
- Peak 866mW output
- VDE recognized (File #A4786)
- ISO9001 certified
- Ordering option V (e.g. MOC3023VM)

APPLICATIONS

- Industrial controls
- Solid state relays
- Washing machines
- Solid state relay
- Lamp ballasts
- Servo drive controls
- Static AC power switches
- Incandescent lamp dimmers
- Motor control

FAIRCHILD
SEMICONDUCTOR®

**6-PIN DIP RANDOM-PHASE
OPTOISOLATORS TRIAC DRIVER OUTPUT
(250/400 VOLT PEAK)**

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M

ABSOLUTE MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$, unless otherwise noted)

Parameters	Symbol	Device	Value	Units
TOTAL DEVICE				
Storage Temperature	T_{STG}	All	-40 to +150	$^\circ\text{C}$
Operating Temperature	T_{OP}	All	-40 to +85	$^\circ\text{C}$
Lead Solder Temperature	T_{SOL}	All	260 for 10 sec	$^\circ\text{C}$
Junction Temperature Range	T_J	All	-40 to +100	$^\circ\text{C}$
Isolation Surge Voltage ⁽¹⁾ (peak AC voltage, 60Hz, 1 sec duration)	V_{ISO}	All	7500	(Vcrpks)
Total Device Power Dissipation @ 25 $^\circ\text{C}$	P_D	All	330	mW
Derate above 25 $^\circ\text{C}$		All	4.4	mW/ $^\circ\text{C}$
EMITTER				
Continuous Forward Current	I_F	All	50	mA
Reverse Voltage	V_R	All	3	V
Total Power Dissipation 25 $^\circ\text{C}$ Ambient	P_D	All	100	mW
Derate above 25 $^\circ\text{C}$		All	1.33	mW/ $^\circ\text{C}$
DETECTOR				
Off-State Output Terminal Voltage	V_{IORM}	MOC3010M/11M/2M MOC3020M/12M/3M	250	V
Peak Repetitive Surge Current (Pulse Width = 1 ms, 120 pulses)	I_{FSM}	All	1	A
Total Power Dissipation @ 25 $^\circ\text{C}$ Ambient	P_D	All	300	mW
Derate above 25 $^\circ\text{C}$		All	4	mW/ $^\circ\text{C}$

Note

¹ Isolation surge voltage, V_{ISO} , is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

**6-PIN DIP RANDOM-PHASE
OPTOISOLATORS TRIAC DRIVER OUTPUT
(250/400 VOLT PEAK)**

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M

ELECTRICAL CHARACTERISTICS (T_A = 25°C Unless otherwise specified)

INDIVIDUAL COMPONENT CHARACTERISTICS						
Parameters	Test Conditions	Symbol	Device	Min	Typ	Max Units
EMITTER						
Input Forward Voltage	I _F = 10 mA	V _F	All		1.15	1.5 V
Reverse Leakage Current	V _R = 3 V, T _A = 25°C	I _R	All		0.01	100 µA
DETECTOR						
Peak Bleeding Current, Either Direction	Rated V _{CE} , I _E = 0 (note 1)	I _{BR}	All		10	100 nA
Peak On-State Voltage, Either Direction	I _{FM} = 100 mA peak, I _E = 0	V _{FM}	All		1.8	3 V

TRANSFER CHARACTERISTICS (T_A = 25°C Unless otherwise specified.)

DC Characteristics	Test Conditions	Symbol	Device	Min	Typ	Max Units
LED Trigger Current	Voltage = 3V (note 3)	I _T	MOC3020M			30
			MOC3010M			15
			MOC3021M			
			MOC3011M			10
			MOC3022M			
Holding Current, Either Direction		I _H	All		100	5 µA

Note

- Test voltage must be applied within dv/dt rating.
- This is static dv/dt. See Figure 5 for test circuit. Commutating dv/dt is a function of the load-driving thyristor(s) only.
- All devices are guaranteed to trigger at an I_T value less than or equal to max I_T. Therefore, recommended operating I_T lies between max I_T (30 mA for MOC3020M, 15 mA for MOC3010M and MOC3021M, 10 mA for MOC3011M and MOC3022M, 5 mA for MOC3012M and MOC3023M) and absolute max I_T (80 mA).

**6-PIN DIP RANDOM-PHASE
OPTOISOLATORS TRIAC DRIVER OUTPUT
(250/400 VOLT PEAK)**

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M

- The mercury switch relay provides a high speed retriggered pulse to the D.U.T.
- 100x scope probes are used, to allow high speeds and voltages.
- The worst-case condition for static dv/dt is established by triggering the D.U.T. with a normal LED input current, then removing the current. The variable R_{EXT} allows the dv/dt to be gradually increased until the D.U.T. continues to trigger in response to the applied voltage pulse, even after the LED current has been removed. The dv/dt is then decreased until the D.U.T. stops triggering. dv/dt is measured at this point and recorded.

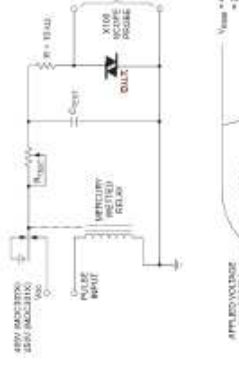


Figure 5. Static dv/dt Test Circuit

Note: This optoisolator should not be used to drive a load directly. It is intended to be a trigger device only.

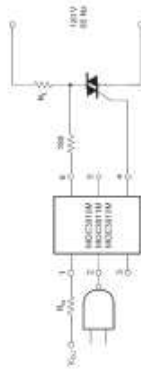


Figure 6. Inductive Load

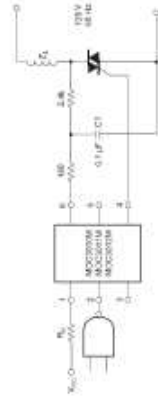


Figure 7. Inductive Load with Sensitive Gas Tube (I_{FM} = 15 mA)

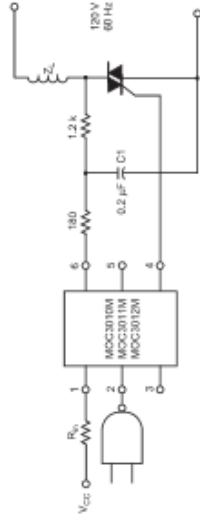
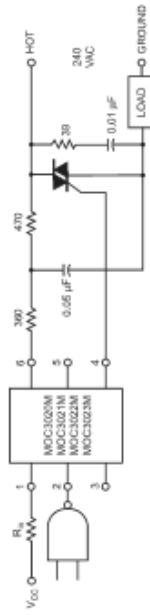


Figure 8. Inductive Load with Sensitive Gate Triac ($I_{GT} \leq 15$ mA)



In this circuit the "hot" side of the line is switched and the load connected to the cold or ground side. The 39 ohm resistor and 0.01µF capacitor are for snubbing of the triac, and the 470 ohm resistor and 0.05 µF capacitor are for snubbing the coupler. These components may or may not be necessary depending upon the particular and load used.

Figure 9. Typical Application Circuit

**ANEXO 11 FOTOGRAFÍAS IMPLEMENTACIÓN DEL
SISTEMA**

ANEXO 11

**FOTOGRAFÍAS
IMPLEMENTACIÓN
DEL SISTEMA**

Anexo 11.1 Construcción estructura inferior



Anexo 11.2 Construcción cajón luminoso



Anexo 11.3 Construcción estructura superior

Anexo 11.3 Instalación eléctrica



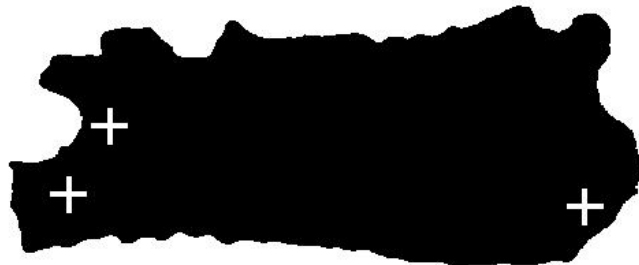
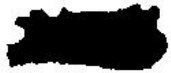
Anexo 11.4 Ensamble de la tarjeta.



Anexo 11.5 Ensamble del tablero de distribución.



Anexo 11.6 Imagen representativa de la etiqueta.



YA15.22CB2016-7-28-12-1

