

DAVID ARMANDO REVELO LUNA

SISTEMA ÓPTICO-INDUSTRIAL EMBEBIDO PARA INSPECCIÓN DE  
CARACTERÍSTICAS DE CALIDAD

Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Trabajo de grado  
Área: Instrumentación y Control  
Popayán  
2014

**SISTEMA ÓPTICO-INDUSTRIAL EMBEBIDO PARA INSPECCIÓN DE  
CARACTERÍSTICAS DE CALIDAD**



**Trabajo de Grado**

**David Armando Revelo Luna**

Director: Mg. Juan Fernando Florez Marulanda

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Electrónica, Instrumentación y Control  
Ingeniería en Automática Industrial  
Popayán, Junio de 2014**

**Hoja de Aprobación**

---

---

---

**Director** \_\_\_\_\_

**Ing. Mag. Juan Fernando Florez Marulanda**

**Jurado** \_\_\_\_\_

**Jurado** \_\_\_\_\_

**Fecha de sustentación:**

---

## **AGRADECIMIENTOS**

A todas las personas que dieron sus aportes para el desarrollo de éste proyecto: profesores, compañeros, familiares y amigos. En especial a la Universidad del Cauca y al programa de Ingeniería en Automática Industrial.

## RESUMEN

En este documento, se presenta el diseño e implementación de un sistema óptico-industrial embebido para la inspección de características de calidad, desarrollado en la Universidad del Cauca. El sistema consta de dos componentes complementarios: Hardware y software. El módulo hardware lo compone: Un sistema embebido de procesamiento *Cortex A8 1GHz (BeagleBoard-xM)*, red Modbus (*NovusDigiRail's*), cámara RGB y de profundidad (*Kinect*), *display* (Monitor HDMI), *Housing* protector del sistema. Estos componentes se integran para formar el soporte físico de una RTU (*Remote Transmission Unit*) con capacidad de procesamiento de vídeo e integración en líneas industriales de manufactura. El módulo software lo compone: un aplicativo de programación que le provee la capacidad de comunicarse a través de una red industrial Modbus, un aplicativo de programación que lo convierte en un servidor de resultados de inspección a través de sockets TCP, una interface de usuario gráfica que permite configurar las utilidades del sistema localmente, y una interface de usuario gráfica a modo de cliente TCP que permite configurar las utilidades del sistema remotamente. El sistema se ha probado en un entorno industrial dando solución al problema de producto faltante en las cajas de la Industria licorera del Cauca, en la línea envasado. El sistema identificó de un total de 30201 unidades (matrices de botellas) un 2.25% de unidades defectuosas (697 matrices de botellas incompletas, con exceso de producto o con producto en mala posición). De ésta manera se comprueba el impacto positivo del sistema desarrollado sobre la calidad del producto final entregado a los clientes y se valida el correcto funcionamiento del sistema de inspección para los efectos que fue desarrollado.

## Contenido

<b>RESUMEN</b> .....	<b>i</b>
<b>LISTA DE TABLAS</b> .....	<b>v</b>
<b>INTRODUCCION</b> .....	<b>1</b>
<b>1. MARCO CONCEPTUAL</b> .....	<b>3</b>
1.1.    Sistemas Embebidos de Procesamiento de Vídeo .....	3
1.1.1. <b>Procesamiento de Vídeo en FPGA's</b> .....	3
1.1.2. <b>Procesamiento de Vídeo en Blackfin</b> .....	4
1.1.3. <b>Procesamiento de Vídeo en ARM (Advanced RISC Machine)</b> .....	4
1.2.    Sistemas Sensores de Visión 3D .....	6
1.2.1. <b>Sensores de Vídeo 3D Industriales Convencionales</b> .....	6
1.2.2. <b>Sensor de vídeo 3D Kinect</b> .....	7
1.3.    Red Industrial Modbus .....	8
1.3.1. <b>Empaquetado de Tramas Modbus</b> .....	9
1.3.1.1 <b>Campo de Dirección [28]</b> .....	10
1.3.1.2 <b>Campo de Función [28]</b> .....	10
1.3.1.3 <b>Campo de Datos [28]</b> .....	11
1.3.1.4 <b>Campo de Control de Errores [28]</b> .....	12
1.4.    Procesamiento de Vídeo en Aplicaciones Industriales .....	13
1.4.1. <b>Generalidades de los Sistemas de Inspección de Visión Industriales</b> .....	13
1.4.2. <b>Clasificación de los Sistemas de Visión Industriales</b> .....	14
1.4.2.1 <b>Sistemas de Inspección Visual Automatizada</b> .....	15
1.4.2.2 <b>Sistemas Para Control de Procesos</b> .....	15
1.4.2.3 <b>Sistemas de Identificación de Partes</b> .....	15
1.4.2.4 <b>Sistemas Para Guiado y Control de Robots</b> .....	15
1.4.2.5 <b>Sistemas de Inspección de Calidad Dimensional</b> .....	16
1.4.2.6 <b>Sistemas de Inspección de Calidad de Superficies</b> .....	16
1.4.2.7 <b>Sistemas de Inspección de Calidad Estructural</b> .....	16
1.4.2.8 <b>Sistemas de Inspección de Calidad Operacional</b> .....	17
1.5.    Estado del Arte de sistemas ópticos industriales de inspección .....	17
1.5.1 <b>A Nivel Internacional</b> .....	17

1.5.1.1	<b>Cognex</b> .....	17
1.5.1.2	<b>Rockwell Automation</b> .....	19
1.5.1.3	<b>Vitronic</b> .....	19
1.5.2	<b>A Nivel Nacional</b> .....	20
1.5.3	<b>A Nivel Local</b> .....	20
1.6	Esquema General de un Dispositivo Industrial Para Inspección de Características de Calidad .....	21
<b>2 DISEÑO DEL SISTEMA DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD</b> .....		<b>23</b>
2.1	Generación de la idea .....	25
2.2	Especificaciones del producto.....	25
2.2.1	<b>Especificaciones Técnicas</b> .....	25
2.2.2	<b>Especificaciones Funcionales</b> .....	26
2.3	Modelo conceptual .....	27
2.4	Diseño preliminar .....	27
2.4.1	<b>Diseño Módulo de Procesamiento</b> .....	30
2.4.2	<b>Diseño Módulo de Adquisición de Vídeo</b> .....	32
2.4.3	<b>Diseño Módulo de Comunicación Modbus</b> .....	33
2.4.4	<b>Diseño Modulo de Comunicación TCP</b> .....	34
2.4.5	<b>Diseño Modulo de Visualización Local y Remoto</b> .....	36
2.4.6	<b>Diseño Eléctrico del sistema SICC</b> .....	38
2.4.7	<b>Diseño Mecánico del Sistema</b> .....	41
<b>3 IMPLEMENTACIÓN DEL SISTEMA DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD</b> .....		<b>43</b>
3.1	Implementación Módulo de Adquisición de Vídeo .....	43
3.1.1	<b>Componente Hardware del Módulo de Adquisición de Vídeo</b> .....	43
3.1.2	<b>Componente Software del Módulo de Adquisición de Vídeo</b> .....	44
3.2	Implementación Módulo de Procesamiento .....	44
3.2.1	<b>Módulo de Procesamiento Aplicación <i>Counter Objects</i></b> .....	45
3.2.2	<b>Módulo de Procesamiento Aplicación <i>Pattern Recognition</i></b> .....	50
3.3	Implementación Módulo de Comunicación Modbus .....	54
3.3.1	<b>Componente Hardware del Módulo de Comunicación Modbus</b> .....	54
3.3.2	<b>Componente Software del Módulo de Comunicación Modbus</b> .....	55
3.4	Implementación Módulo de Comunicación TCP .....	55

3.4.1	<b>Módulo de Comunicación TCP Servidor</b> .....	55
3.4.2	<b>Módulo de Comunicación TCP Cliente</b> .....	58
3.5	Implementación Módulo de Visualización Local y Remoto .....	58
3.5.1	<b>Interface Principal</b> .....	58
3.5.2	<b>Interface de Aplicación</b> .....	60
3.5.2.1	Interface de Aplicación <i>Counter Objects</i> .....	61
3.5.2.2	Interface de Aplicación <i>Pattern Recognition</i> .....	62
3.6	Implementación del Sistema Mecánico.....	64
<b>4 PRUEBAS DE VALIDACIÓN DEL SISTEMA DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD</b> .....		<b>65</b>
4.1	Problema Objetivo a Solucionar con la Prueba de Validación .....	65
4.2	Descripción de la Prueba de Validación. ....	65
4.3	Configuración y Puesta en Marcha del SICC.....	67
4.4	SICC en Operación .....	69
4.5	Resultados de la prueba de validación del SICC .....	72
4.6	Consideraciones para el manejo del SICC .....	78
<b>5.</b>	<b>CONCLUSIONES</b> .....	<b>80</b>
5.1.	Trabajos futuros .....	82
<b>ANEXOS</b> .....		<b>88</b>
<b>Anexo A - RS485 TO USB CONEVERTER NOVUS</b> .....		<b>89</b>
<b>Anexo B - CONFIGURACION SDCARD PARA BEAGLEBOARD-XM</b> .....		<b>90</b>
<b>Anexo C - CONFIGURACIÓN DIGIRAIL'S USANDO DIGICONFIG</b> .....		<b>95</b>
<b>Anexo D - MODULO SOFTWARE MODBUS (SOURCE CODE)</b> .....		<b>97</b>
<b>Anexo E – INTERACCIÓN MÓDULOS SOFTWARE</b> .....		<b>100</b>

## LISTA DE TABLAS

TABLA 1. FUNCIONES BÁSICAS Y CÓDIGOS ASOCIADOS EN EL PROTOCOLO MODBUS, TOMADO DE [28].	11
TABLA 2. COMPONENTES BÁSICOS DE ALGUNAS CÁMARAS INDUSTRIALES EXISTENTES EN EL MERCADO MUNDIAL ACTUAL. FUENTE PROPIA.	22
TABLA 3. LISTA DE PQR'S EN LA ILC EN EL AÑO 2012. FUENTE DOCUMENTACIÓN ILC	24
TABLA 4. <i>HEADERS</i> PARA EL MANEJO REMOTO DE LOS MANDOS DE LA APLICACIÓN PRINCIPAL. FUENTE PROPIA.	56
TABLA 5. <i>HEADERS</i> PARA EL MANEJO REMOTO DE LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA.	56
TABLA 6. <i>HEADERS</i> PARA EL MANEJO REMOTO DE LA CONFIGURACIÓN DE PARÁMETROS DE LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA.	57
TABLA 7. <i>HEADERS</i> PARA EL MANEJO REMOTO DE LA APLICACIÓN <i>PATTERN RECOGNITION</i> . FUENTE PROPIA.	57
TABLA 8. <i>HEADERS</i> PARA EL MANEJO REMOTO DE LA CONFIGURACIÓN DE LA APLICACIÓN <i>PATTERN RECOGNITION</i> . FUENTE PROPIA.	57
TABLA 9. REGISTRO DE EVALUACIONES BUENAS Y MALAS EN LA PRESENTACIÓN MEDIA. FUENTE PROPIA.	74
TABLA 10. REGISTRO DE EVALUACIONES BUENAS Y MALAS EN LA PRESENTACIÓN BOTELLA. FUENTE PROPIA.	75
TABLA 11. REGISTRO DE EVALUACIONES BUENAS Y MALAS EN LA PRESENTACIÓN GARRAFA. FUENTE PROPIA.	76
TABLA 12. COMPARACIÓN DE EVALUACIONES POSITIVAS Y NEGATIVAS EN LAS TRES PRESENTACIONES. FUENTE PROPIA.	76

## LISTA DE FIGURAS

FIGURA 1. APLICACIÓN DE ARM EN LA INDUSTRIA. FUENTE [15].	5
FIGURA 2. KINECT SENSOR DE VÍDEO 3D. FUENTE [22]	7
FIGURA 3. COMUNICACIÓN MAESTRO/ESCLAVO MODBUS, MODIFICADA DE [28].	9
FIGURA 4. SISTEMA TÍPICO DE VISIÓN DE MÁQUINA INDUSTRIAL, MODIFICADA DE [29].	14
FIGURA 5. ESQUEMA GENERAL DE COMPONENTES DE UN DISPOSITIVO INDUSTRIAL DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD. FUENTE PROPIA.	21
FIGURA 6. BOSQUEJO GENERAL DEL SISTEMA. FUENTE PROPIA.	28
FIGURA 7. MÓDULOS COMPONENTES DEL SICC. FUENTE PROPIA.	29
FIGURA 8. MÓDULO DE PROCESAMIENTO. FUENTE PROPIA.	31
FIGURA 9. LIBRERÍAS ASOCIADAS A LOS COMPONENTES DEL SISTEMA. FUENTE PROPIA.	31
FIGURA 10. DISEÑO CAD DE <i>ENCLOSURE</i> PARA LA CÁMARA KINECT. FUENTE PROPIA.	32
FIGURA 11. DISEÑO DEL MÓDULO MODBUS RTU DE COMUNICACIÓN. FUENTE PROPIA.	34
FIGURA 12. MÓDULO TCP DE COMUNICACIONES. FUENTE PROPIA.	35
FIGURA 13. DISEÑO GUI PARA LA INTERFACE PRINCIPAL. FUENTE PROPIA.	36
FIGURA 14. DISEÑO GUI PARA LA INTERFACE DE APLICACIÓN. FUENTE PROPIA.	37
FIGURA 15. DIAGRAMA DE POTENCIA 110V DEL SISTEMA SICC. FUENTE PROPIA.	38
FIGURA 16. DIAGRAMA DE MANDO DE LA TARJETA BB-xM. FUENTE PROPIA.	39
FIGURA 17. DIAGRAMA DE POTENCIA DE 24V DEL SISTEMA SICC. FUENTE PROPIA.	39
FIGURA 18. DIAGRAMA DE LAZO ALIMENTACIÓN DE INSTRUMENTACIÓN. FUENTE PROPIA.	39
FIGURA 19. DIAGRAMA DE LAZO ALIMENTACIÓN BEAGLEBOARD-xM. FUENTE PROPIA.	40
FIGURA 20. DIAGRAMA DE LAZO ENERGIZACIÓN DE ARMARIO. FUENTE PROPIA.	40
FIGURA 21. DIAGRAMA DE LAZO DE CONTROL. FUENTE PROPIA.	41
FIGURA 22. VISTA EXPLOSIONADA DEL DISEÑO CAD DEL PROTOTIPO. FUENTE PROPIA.	42
FIGURA 23. DISEÑO CAD DEL PROTOTIPO. FUENTE PROPIA.	42
FIGURA 24. <i>HOUSING</i> DE PROTECCIÓN PARA EL SENSOR KINECT. FUENTE PROPIA.	43
FIGURA 25. IZQUIERDA: IMAGEN RGB. DERECHA: IMAGEN DE PROFUNDIDAD EN PSEUDOCOLOR EN EL MÓDULO DE ADQUISICIÓN DE VÍDEO. FUENTE PROPIA.	44
FIGURA 26. IZQUIERDA: IMAGEN RGB, CENTRO: IMAGEN DE PROFUNDIDAD, DERECHA: IMAGEN BINARIA POR SEGMENTACIÓN DE PROFUNDIDAD. FUENTE PROPIA.	45
FIGURA 27. IZQUIERDA: IMAGEN BINARIA SIN OPERACIÓN MORFOLÓGICA. DERECHA: IMAGEN BINARIA OPERADA CON EROSIÓN Y DILATACIÓN. FUENTE PROPIA.	46
FIGURA 28. ETIQUETADO DE OBJETOS BLOBS. FUENTE PROPIA.	46
FIGURA 29. IZQUIERDA: OBJETOS DE INTERÉS SIN FILTRADO POR ÁREA. DERECHA: OBJETOS FILTRADOS POR ÁREA DE BLOBS DE INTERÉS. FUENTE PROPIA.	47
FIGURA 30. IZQUIERDA: OBJETOS SIN FILTRADO POR REGIÓN DE INTERÉS. DERECHA: OBJETOS FILTRADOS POR REGIÓN DE INTERÉS ROI EN EL MÓDULO DE PROCESAMIENTO. FUENTE PROPIA.	48
FIGURA 31. PROCESO DE EVALUACIÓN PARA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA.	49
FIGURA 32. IMAGEN PATRÓN EN EL PROCESO DE EVALUACIÓN. FUENTE PROPIA.	50
FIGURA 33. ROI DE BÚSQUEDA PARA LA EVALUACIÓN. FUENTE PROPIA.	51
FIGURA 34. RESULTADO DE UNA EVALUACIÓN EN LA APLICACIÓN <i>PATTERN RECOGNITION</i> . FUENTE PROPIA.	52
FIGURA 35. MAPA DE CORRELACIÓN ASOCIADO AL PROCESO DE EVALUACIÓN DE LA APLICACIÓN <i>PATTERN RECOGNITION</i> . FUENTE PROPIA.	52
FIGURA 36. PROCESO DE EVALUACIÓN PARA LA APLICACIÓN <i>PATTERN RECOGNITION</i> . FUENTE PROPIA.	53
FIGURA 37. DIAGRAMA DE CONEXIÓN ELÉCTRICA EN EL MÓDULO DE COMUNICACIÓN MODBUS. FUENTE PROPIA.	54
FIGURA 38. SUPERIOR: CONEXIÓN DIRECTA CLIENTE-SERVIDOR ENTRE PC Y SICC. INFERIOR: CONEXIÓN A TRAVÉS DE RED LOCAL ENTRE PC Y SICC. FUENTE PROPIA.	55
FIGURA 39. INTERFACE PRINCIPAL LOCAL. FUENTE PROPIA.	59
FIGURA 40. INTERFACE REMOTA. FUENTE PROPIA.	59
FIGURA 41. INTERFACE DE CONFIGURACIÓN DE LA COMUNICACIÓN TCP DESDE LA APLICACIÓN REMOTA. FUENTE PROPIA.	60

FIGURA 42. MINI WIRELESS KEYBOARD Y MOUSE TOUCH PAD, ACCESORIO PARA MANEJO DE INTERFACE DE USUARIO DE LA APLICACIÓN. FUENTE PROPIA. ....	60
FIGURA 43. GUI DE LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA. ....	61
FIGURA 44. GUI DE CONFIGURACIÓN DE LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA. ....	62
FIGURA 45. GUI DE LA APLICACIÓN <i>PATTERN RECOGNITION</i> . FUENTE PROPIA.....	63
FIGURA 46. GUI DE CONFIGURACIÓN DE LA APLICACIÓN <i>PATTERNRECOGNITION</i> . FUENTE PROPIA. ....	63
FIGURA 47. PROTOTIPO DEL SISTEMA SICC. FUENTE PROPIA. ....	64
FIGURA 48. ESCENARIO DE LA PRUEBA DE VALIDACIÓN. FUENTE PROPIA. ....	66
FIGURA 49. SECUENCIA DE ENCENDIDO DEL SICC. IMAGEN SUPERIOR: PANEL APAGADO. IMAGEN INFERIOR: PANEL ENCENDIDO. FUENTE PROPIA. ....	68
FIGURA 50. INTERFACE PARA ESCOGER EL NÚMERO DE OBJETOS DE REFERENCIA EN LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA. ....	69
FIGURA 51. OCURRENCIA DE EVENTOS DE SENSOR DE PRESENCIA INFRARROJO. FUENTE PROPIA. ....	69
FIGURA 52. EVALUACIÓN POSITIVA “PASS” EN LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA. ....	70
FIGURA 53. EVALUACIÓN NEGATIVA “FALL” EN LA APLICACIÓN <i>COUNTER OBJECTS</i> . FUENTE PROPIA. ....	71
FIGURA 54. MENSAJE DE AVISO DE EVALUACIÓN NEGATIVA. FUENTE PROPIA. ....	71
FIGURA 55. ALERTA VISUAL EN EL PANEL DE EVALUACIÓN NEGATIVA, PILOTO AMARILLO. FUENTE PROPIA. ....	72
FIGURA 56. PROCESO DE INSPECCIÓN DE EMPACADO EN LA PRESENTACIÓN MEDIA. SUPERIOR IZQUIERDA: BOTELLAS FALTANTES. SUPERIOR DERECHA: BOTELLA CAÍDA. INFERIOR: BOTELLA EXTRA. FUENTE PROPIA. .....	73
FIGURA 57. PROCESO DE INSPECCIÓN DE EMPACADO EN LA PRESENTACIÓN BOTELLA. SUPERIOR IZQUIERDA: BOTELLAS FALTANTES. SUPERIOR DERECHA: BOTELLA CAÍDA. INFERIOR: BOTELLA EXTRA. FUENTE PROPIA. .....	74
FIGURA 58. PROCESO DE INSPECCIÓN DE EMPACADO EN LA PRESENTACIÓN GARRAFA. IZQUIERDA: BOTELLAS FALTANTES. DERECHA: BOTELLA EXTRA. FUENTE PROPIA. ....	75
FIGURA 59. EVALUACIONES NEGATIVAS Y POSITIVAS PARA CADA PRESENTACIÓN. FUENTE PROPIA. ....	76
FIGURA 60. COMPARATIVO DE EVALUACIONES NEGATIVAS PARA LAS 3 PRESENTACIONES DEL PRODUCTO. FUENTE PROPIA. ....	77

## INTRODUCCION

Los sistemas ópticos embebidos de inspección de características de calidad en ambientes industriales integran diversas tecnologías en un sólo producto: Procesadores embebidos, hardware de adquisición, procesamiento de vídeo, periféricos especializados, buses de comunicaciones y protección industrial [1]. El desarrollo tecnológico abordado en el presente proyecto se ha encaminado al mejoramiento de la calidad de los productos en las líneas de manufactura, la integración de estos sistemas a las líneas de producción incorpora valor agregado, permite que las industrias sean más competitivas y proporciona elementos valiosos para que permanezcan en el mercado. La inspección de calidad tradicional se ha desarrollado durante largo tiempo por técnicos expertos, sin embargo este tipo de personal requiere gran inversión en tiempo de preparación. La inclusión de la automatización y monitoreo en las industrias en los últimos años ha favorecido el desarrollo de sistemas de inspección operados por máquinas con visión artificial. Procesadores cada vez más veloces y hardware especializado permiten desarrollar sistemas que logran evaluar la calidad de productos on-line en las líneas de producción. En la actualidad, en el país se han desarrollado diversos trabajos relacionados con la extracción de características de calidad para aplicaciones industriales, como los que se detallan en [2, 3, 4], sin embargo en estos trabajos se proponen sistemas que operan en computadores convencionales lo que hace de ellos dispositivos poco compactos. En general en la industria Colombiana se importan los sistemas de inspección de características de calidad de los grandes fabricantes mundiales como Rockwell [5], Vitronic [6], Cognex [7], etc.

El desarrollo de sistemas ópticos embebidos de inspección de características de calidad a nivel industrial se puede abordar desde la perspectiva del perfil de Ingeniería en Automática Industrial de la Universidad del Cauca, con el objetivo de

generar prototipos de bajo coste que puedan integrarse a líneas de manufactura locales que requieran mejorar su proceso de producción.

La monografía del presente proyecto describe los procedimientos que se llevaron a cabo para obtener los objetivos propuestos; en el capítulo uno se hace descripción de las tecnologías que integra un sistema óptico embebido de inspección de características de calidad a nivel industrial. En el capítulo dos, se describe el diseño del Sistema de Inspección de Características de Calidad (SICC), detallando cada módulo software y hardware que lo componen. En el capítulo tres, se describe el proceso de implementación de los módulos que componen el sistema: 1. Módulo de procesamiento, 2. Módulo de adquisición de vídeo, 3. Módulo de comunicación Modbus, 4. Módulo de comunicación TCP, 3. Módulo de visualización local y remota. En el capítulo cuatro se detallan las pruebas necesarias para validar el sistema en un ambiente industrial. El prototipo desarrollado se instaló en la Industria Licorera del Cauca (ILC) en la máquina encartonadora con el objetivo de solucionar el problema de producto faltante en cajas que ha generado PQR's (peticiones, quejas y reclamos) considerables de parte de los clientes. Por último, en el capítulo 5 están consignadas las conclusiones de realizar este trabajo, se mencionan además los posibles trabajos que se recomienda realizar a futuro tomando como base los resultados aquí expuestos.

## **1. MARCO CONCEPTUAL**

### **1.1. Sistemas Embebidos de Procesamiento de Vídeo**

El procesamiento de vídeo es una herramienta eficiente para diversas aplicaciones en automatización [2], procesos industriales [8] y calidad [3]. Los algoritmos asociados se ejecutan en sistemas asistidos por computador o sistemas embebidos. Los segundos se usan para aplicaciones dedicadas, en general son sistemas más compactos y de menor consumo energético. Por esta razón se ha ampliado la investigación y desarrollo de sistemas de procesamiento embebido de vídeo. Existen actualmente en el mercado sistemas embebidos con velocidades de procesamiento que permiten adquirir y procesar vídeo eficientemente, las más usadas son: FPGA's, Procesadores Digitales de Señal (DSP's), Procesadores de arquitectura ARM y Blackfin (Procesador Digital de Analog Devices). El procesador debe ser elegido teniendo en cuenta características como: alto desempeño, flexibilidad, capacidad de actualización, bajo costo, ambiente de instalación, tipo de aplicación, etc.

#### **1.1.1. Procesamiento de Vídeo en FPGA's**

Los sistemas de procesamiento de vídeo basados en FPGA tienen una ventaja particular, sobre los otros sistemas, son capaces de hacer procesamiento en paralelo. Esta característica hace que, para algunas aplicaciones, una FPGA sea la única opción viable. Como se menciona en [9] aplicaciones como codificación y decodificación H.264 HD no podrían hacerse con otros dispositivos diferentes a las FPGA's o sistemas híbridos que integren DSP y FGPA. Estos dispositivos alcanzan velocidades de procesamiento de 1.5GHz, la implementación puede resultar compleja debido a que el desarrollo es de bajo nivel, sin embargo existen kits de desarrollo de procesamiento de vídeo industrial basados en FPGA, se usan como plataformas de desarrollo si el costo no es un factor primordial, además existen entornos de desarrollo que permiten programar algoritmos usando entornos gráficos o en lenguaje C como el DK4 *Design Suite*. En [10] se presenta

un desarrollo de procesamiento de vídeo en tiempo real usando FPGA en el que se realizan operaciones básicas como filtro *Sobel* y obtención de histogramas.

### **1.1.2. Procesamiento de Vídeo en Blackfin**

Blackfin es una familia de microprocesadores de 16/32 bits asistido por un pequeño microcontrolador con rendimiento energético eficiente. Usualmente el Blackfin está destinado para aplicaciones embebidas de audio, vídeo y comunicaciones. Su alto desempeño y capacidad de procesamiento en tiempo real hacen del Blackfin una interesante alternativa para procesos que requieren alto desempeño computacional como lo es el procesamiento de vídeo [11].

En [11] se hace un análisis de las principales características del procesador Blackfin y su desempeño en el campo del procesamiento de vídeo. El sistema Blackfin soporta, por ejemplo, instrucciones de aceleración como DCT (transformada discreta de coseno), y SAD (suma de diferencias absolutas) usadas para compresión de vídeo y estimación de movimiento. En [12] se realizan pruebas de *Benchmark* para procesos de dilatación y filtros *Sobel* obteniendo tiempos de 110.76us con una imagen de entrada de 32x32 pixeles para el filtro *Sobel* y 4.83us para el proceso de dilatación en una imagen de 8x16 pixeles.

### **1.1.3. Procesamiento de Vídeo en ARM (Advanced RISC Machine)**

La arquitectura ARM se ha caracterizado desde su inicio (1982) en tener en su desempeño bajo consumo energético y alta eficiencia [13]. A la fecha empresas como NXP, Atmel, Freescale, Analog Devices, TI (Texas Instruments) ofrecen un portafolio completo de soluciones para el sector industrial, de salud, automotriz, informático, computación, energético, etc. [14]. Dentro de la arquitectura ARM se encuentran varios tipos de procesadores dedicados a sectores de la industria especializados, ver Figura 1.

TI ARM Solutions: Meeting Market Needs							
	Sitara™ Processors	Stellaris® MCU	Concerto™ MCU	Hercules™ Safety MCU	DaVinci™ Video Processors	OMAP-L1x Processors	KeyStone Multicore Processors
ARM® Core	AM1x ARM9, AM3x ARM®Cortex™-A8	ARM® Cortex™-M3, M4	ARM® Cortex™-M3	ARM® Cortex™-R4, M3	ARM9™, ARM® Cortex™-A8	ARM9	ARM® Cortex™-A15
Audio	✓	✓	✓		✓	✓	✓
Automotive			✓	✓	✓	✓	✓
Communications	✓	✓	✓	✓	✓	✓	✓
Imaging					✓	✓	✓
Industrial	✓	✓	✓	✓	✓	✓	✓
Medical	✓	✓	✓	✓	✓	✓	✓
Security	✓	✓			✓	✓	✓
Wireless	✓	✓	✓		✓		
Key Features	Low power & <u>high performance</u> with extensive peripheral integration	Open architecture software, rich communication options	Dual subsystem architecture for optimized real-time control and communications	Safety	Completely tailored high performance	High performance & low power	32-bit fixed- and/or floating-point + Acceleration

**Figura 1. Aplicación de ARM en la industria. Fuente [15].**

Como se muestra en la Figura 1 para aplicaciones industriales y de procesamiento de vídeo el procesador Cortex-A8 1GHz es una posible solución. Es precisamente este procesador el que usa la tarjeta de desarrollo *BeagleBoard-xM* (BB-xM) [16]. Esta tarjeta tiene, a nivel hardware, periféricos como: USB, HDMI, Ethernet, Audio, etc. Convirtiéndose en una excelente plataforma para desarrollo de aplicaciones sobre sistemas operativos embebidos.

### 1.1.3.1 BeagleBoard-xM

BB-xM es una tarjeta de hardware libre con funcionalidades de un computador convencional de bajo consumo desarrollada por Texas Instruments [16]. El software soportado en la tarjeta es también libre. Las dimensiones de la tarjeta son 82.55x82.55mm convirtiéndola en una excelente plataforma para aplicaciones portables. Cuenta con un procesador DM3730 que lo compone una CPU Cortex-A8 de 1GHz, la cual soporta diversos sistemas operativos como: Linux (Debian, Ubuntu, Angstrom, Fedora), Android, FreeBSD, OpenBSD, RISC OS, Windows CE. Además cuenta con un TMS320C64Sx y un DSP para aceleración de vídeo y

decodificación de audio. Aceleradores 2D y 3D para renderizado soportado por OPENGL ES 2.0 [16]. Entre otras características cuenta con 512 MB de memoria RAM, conector Ethernet para comunicación de red, salida de vídeo S-Video, HDMI y un *Hub* USB con 4 puertos 2.0, lo cual permite conectar varios periféricos como teclado, mouse, cámaras USB, etc.

Todas las características anteriores convierten a la tarjeta BB-xM en una excelente plataforma para realizar procesamiento de vídeo embebido. Configurando sobre la tarjeta un sistema operativo como Ubuntu y sobre éste las librerías necesarias para adquisición y procesamiento de vídeo, es posible tener un sistema dedicado a adquirir y procesar vídeo de bajo costo, bajo consumo y alto desempeño.

## **1.2. Sistemas Sensores de Visión 3D**

En cuanto a sistemas sensores de visión 3D existen varias opciones dentro de las cuales se encuentran las cámaras típicas industriales tales como: ENSENSO N10 de IDS [17], Chameleon 360 de Gopel, VC Nano 3D de Vision Components [18], MV-D1024E-3D01 de phothon focus [19], y algunas que han resultado como desarrollos para aplicaciones particulares como *human tracking* o *motion tracking* para vídeo-juegos cómo KINECT o ASUS XTION [20].

### **1.2.1. Sensores de Vídeo 3D Industriales Convencionales**

Las cámaras industriales 3D usan diferentes tipos de técnicas para estimar la profundidad de los elementos en una escena, además pueden obtener información adicional como color, brillo, texturas, etc. Las técnicas más comunes son: Tiempo de vuelo (TOF), triangulación con luz estructurada (laser) y visión estéreo. Las cámaras basadas en TOF resuelven la medida de profundidad basadas en la velocidad constante de la luz, midiendo el tiempo de vuelo de la señal de luz que se refleja en la superficie del objeto de interés, en general éstas cámaras pueden trabajar a velocidades altas (60fps a 160fps) pero su exactitud en profundidad es limitada, valores típicos pueden ser 1cmm a 2cm. La resolución de la imagen de estas cámaras también es limitada, típicamente 200x200 pixeles. Las cámaras

basadas en triangulación laser usualmente pueden alcanzar exactitud de micrómetros, y su resolución es mayor que las TOF, valores típicos son 1024x1024 y pueden ser incluso mayores. La principal desventaja de estos dispositivos son los puntos de oclusión y proveen perfiles de profundidad sencillos. Las cámaras 3D por visión estereoscópica tienen resolución intermedia, valores usuales son 640x480. Las velocidades no suelen ser las más altas, típicamente 30fps. La exactitud de estas cámaras suele ser de 0.1mm a 1mm. La principal desventaja de estos dispositivos son los escenarios de baja textura y puntos de oclusión dependientes de la separación entre las cámaras. En general el *housing* de las tres clases de dispositivos descritos tiene algún tipo de protección industrial [21].

### 1.2.2. Sensor de vídeo 3D Kinect

Kinect es un sensor 3D de vídeo que fue desarrollado por Microsoft en principio para *human tracking* en la vídeo-consola XBOX 360 [20]. Este dispositivo cuenta con una cámara VGA de resolución 640x480 pixeles a color CMOS, con un filtro de color BAYER, además de una cámara infrarroja de resolución 640x480 pixeles CMOS, cuenta con un proyector infrarrojo además de un motor y receptores de audio, ver Figura 2.

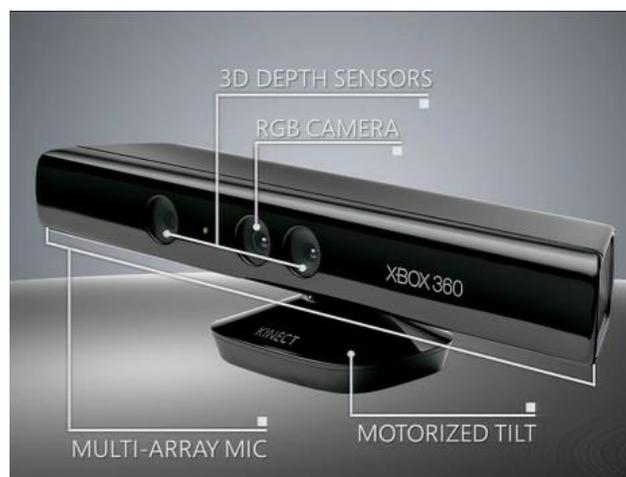


Figura 2. Kinect Sensor de Vídeo 3D. Fuente [22]

El dispositivo envía un patrón infrarrojo conocido para una profundidad determinada, éste patrón se desplaza cuando un objeto se encuentra a una profundidad diferente, ésta distorsión es medida por el sistema y representa en una imagen de profundidad (escala de grises) la distancia que existe entre la cámara y el objeto.

Las características principales del sensor son: cuenta con resolución de 640x480 pixeles, el rango de medida es de 0.5 a 4m, alcanza una precisión de 1 o 2mm, su precio es aproximadamente 100 Dólares. Proporciona 30 marcos por segundo (fps), es capaz de generar mapas de profundidad en escenarios oscuros, es relativamente robusto ante iluminación externa, genera además de la imagen de profundidad una imagen RGB. Tiene algunas limitaciones como fallos en zonas de baja reflexión infrarroja o en bordes de los objetos, el *driver* para Linux tiene poco soporte, la operación 24 horas al día y 7 días a la semana es aún cuestionable, el *housing* del dispositivo no es robusto, no cuenta con protección ante humedad, polvo o agua [23].

Por todas las características antes mencionadas el uso del Kinect se ha ampliado a aplicaciones en robótica [24], industriales [25], académicas [26], etc.

### **1.3. Red Industrial Modbus**

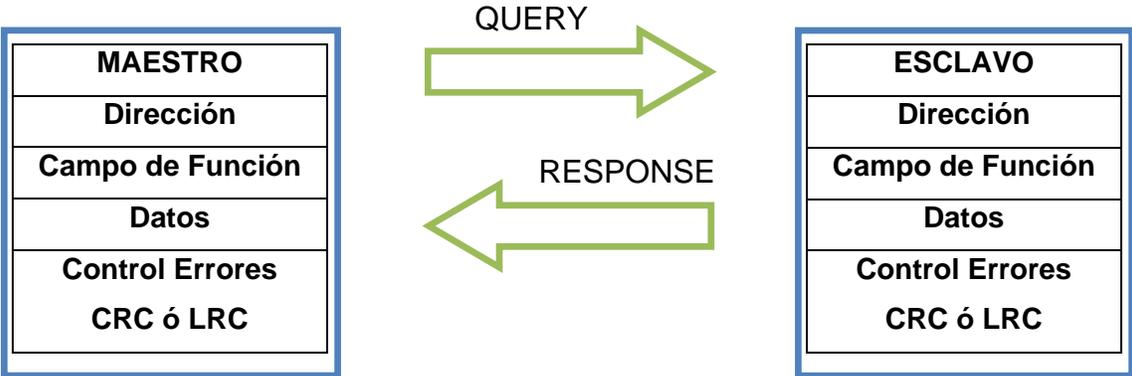
Modbus es un protocolo de comunicaciones que emplea los niveles 1, 2 y 7 del modelo OSI (*Open System Interconnection*) [27], que corresponden al nivel Físico, de Enlace y de Aplicación respectivamente. Está basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLC's). El uso de la red Modbus es superior a otras redes de comunicación debido a tres razones principales: Es de carácter público, implementación fácil y requiere poco desarrollo, maneja bloques de datos sin suponer restricciones. Modbus permite el control de una red de dispositivos, comunicación entre RTU's, sistemas de supervisión y adquisición de datos SCADA's. Existen versiones del protocolo Modbus para puerto serie y

Ethernet (Modbus TCP). La comunicación se da en forma serial asíncrona bajo los estándares RS-232 ó RS-485 para comunicación *halfduplex* o RS-422 para comunicación full dúplex, y utiliza diferentes medios físicos como son: radiofrecuencia, cables, fibra óptica o infrarrojo, y cuya velocidad de transmisión está entre 75 a 19200 baudios. Tanto las generalidades como los detalles del protocolo se encuentran en la guía de referencia de Modicon [28].

Existen dos variantes Modbus, con diferentes representaciones numéricas de los datos y detalles del protocolo ligeramente desiguales. Modbus RTU es una representación binaria compacta de los datos. Modbus ASCII es una representación legible del protocolo pero menos eficiente. El formato RTU finaliza la trama con una suma de control de redundancia cíclica (CRC), mientras que el formato ASCII utiliza una suma de control de redundancia longitudinal (LRC). La versión Modbus TCP es semejante al RTU pero estableciendo la transmisión de datos mediante paquetes TCP/IP [28].

**1.3.1. Empaquetado de Tramas Modbus**

Tanto el esclavo como el maestro en el protocolo Modbus tienen definida la estructura de la trama de datos para su comunicación. Existen 4 campos principales en la trama de comunicación: Dirección, Campo de Función, Datos, Control CRC ó LRC. La transmisión de información entre Maestro/Esclavo se lleva a cabo mediante una mecánica de pregunta/respuesta [28]. En la Figura 3 se muestra el modo de comunicación.



**Figura 3. Comunicación Maestro/Esclavo Modbus, modificada de [28].**

### **1.3.1.1 Campo de Dirección [28]**

El campo de dirección de un mensaje contiene dos caracteres (ASCII) u 8 Bits (RTU). Los valores válidos de dirección están en el rango 1-247. Si el maestro envía un mensaje al esclavo, ubica la dirección del esclavo en el campo Dirección del mensaje, cuando el esclavo responde a la petición ubica su propia dirección en el campo Dirección, así el Maestro sabe con certeza de quién proviene el mensaje.

La dirección 0 en un mensaje enviado por el Maestro indica una comunicación tipo *Broadcast* (mensaje para todos los esclavos).

### **1.3.1.2 Campo de Función [28]**

El campo de Función de un mensaje contiene dos caracteres (ASCII) u 8 bits (RTU). Valores validos están entre 1-255. Cuando un mensaje es enviado desde el Maestro al Esclavo el campo de función determina qué tipo de acción debe desarrollar el esclavo, por ejemplo: leer un estado de un elemento digital o escribir sobre él, en la Tabla 1 se muestran las funciones básicas y códigos de operación Modbus. Cuando el esclavo responde al maestro usa el campo de función para indicar una respuesta normal o si algún error ha ocurrido en el proceso. Para una respuesta normal, el esclavo simplemente responde con el mismo campo de función que escribió el maestro, para una respuesta de error o excepción el esclavo responde con un código equivalente al original pero con el bit más significativo con un valor de uno.

Por ejemplo si un código en campo de función escrito por el maestro tiene el valor: 0000 0011 (03 Hexadecimal), si el esclavo realiza la acción solicitada sin error, regresará el mismo valor en el campo de función hacia el maestro. Si por el contrario ha ocurrido una excepción, entonces el valor de retorno hacia el maestro será: 1000 0011 (83 Hexadecimal). Además para el caso de la ocurrencia de una excepción, el esclavo retorna en el campo de datos un código único que define el tipo de excepción que ha ocurrido.

Código	Hex	Descripción
0	00	Control estación esclavas
1	01	Lectura n bits de salida
2	02	Lectura n bits de entrada
3	03	Lectura n registros de entrada
4	04	Lectura n registros de salida
5	05	Escritura de 1 bit
6	06	Escritura de una palabra o registro
7	07	Lectura rápida de 8 bits
8	08	Diagnóstico
9	09	No Utilizada
10	0A	No Utilizada
11	0B	Solicitar contador de eventos de comunicaciones
12	0C	Solicitar diario de eventos de comunicaciones
13	0D	No utilizada
14	0E	No utilizada
15	0F	Escritura n bits (Coils)
16	10	Escritura n palabras ( <i>Holding Registers</i> )

**Tabla 1. Funciones básicas y códigos asociados en el protocolo Modbus, tomado de [28].**

### 1.3.1.3 Campo de Datos [28]

El campo de datos se construye usando dos grupos de dos dígitos hexadecimales, en el rango 00 a FF. Esto se realiza a partir de dos caracteres para la codificación ASCII o un carácter para la codificación RTU.

El campo de datos de los mensajes, enviados por el maestro a los esclavos, contiene información adicional que el esclavo debe interpretar para ejecutar la acción definida en el campo de función. Este campo puede incluir información adicional como direcciones iniciales y finales de entrada o salida a leer o escribir, el número de datos o registros a leer, etc. Por ejemplo, si el maestro pide al esclavo leer un grupo de registros de salida (código de función 03 hex), el campo

de datos especifica el registro de inicio y cuantos registros se deben leer. Si el maestro escribe un grupo de registros en el esclavo (código de función 10 hex), en el campo de datos se debe especificar el número del registro desde donde se debe empezar, cuantos registros se van a escribir, el número de bytes que corresponden a los datos que se transmiten en dicho mensaje, y los datos que deben ser escritos.

Si no ocurre un error en la operación, el campo de datos en el mensaje de respuesta contiene los datos solicitados. Si por el contrario ocurre un error en la operación, en el campo de datos se escribe un código de excepción que la aplicación del maestro puede usar para determinar el error y la acción a seguir.

El campo de datos puede ser inexistente (longitud cero) en algunos mensajes. Por ejemplo, en una solicitud de un dispositivo maestro a un esclavo para responder con su registro de evento de comunicaciones (código de función 0B hex), el esclavo no requiere ninguna información adicional. El código de función sólo especifica la acción.

#### **1.3.1.4 Campo de Control de Errores [28]**

Existen dos tipos de control de errores que se usan en el protocolo Modbus dependiendo de la codificación usada ASCII o RTU.

Cuando se usa la codificación ASCII el campo de control de errores contiene dos caracteres ASCII. Los caracteres de control de errores son el resultado de una operación de chequeo de redundancia longitudinal (LRC) que se calcula con el contenido del mensaje, excluyendo los caracteres dos puntos de inicio y el carácter CRLF de terminación. Los caracteres LRC se añaden al mensaje como último campo que precede los caracteres CRLF.

Cuando se usa la codificación RTU para la transmisión, el campo de control de errores contiene un valor de 16Bits como 2 bytes de 8Bits. El valor de chequeo de

error es el resultado de una operación de chequeo de redundancia cíclica (CRC) que se aplica al contenido del mensaje. El valor de CRC se añade como último campo en el mensaje. Cuando se realiza esta operación el byte menos significativo se añade primero seguido por el byte más significativo.

#### **1.4. Procesamiento de Vídeo en Aplicaciones Industriales**

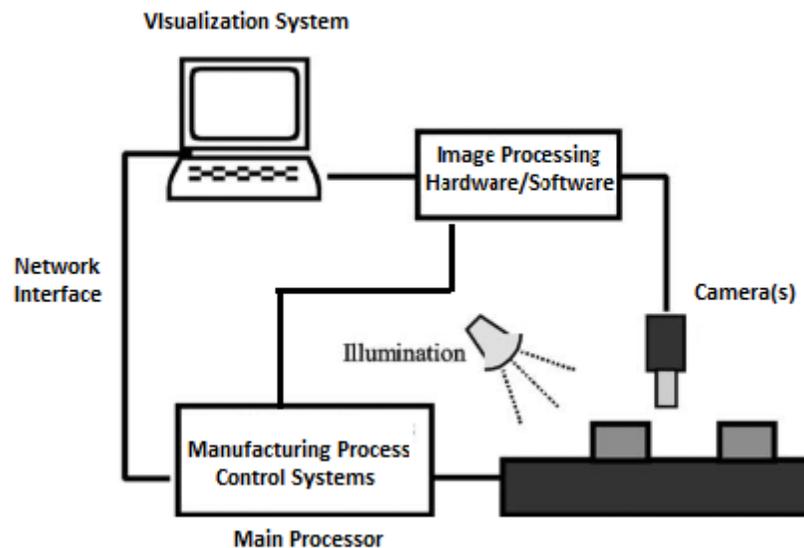
La visión de máquina en entornos industriales es ampliamente utilizada para diferentes tipos de tareas, en [29] se realiza una revisión del estado del arte de los sistemas industriales de visión artificial, se presentan trabajos en la estimación de la posición de objetos como en [30], inspección de características de calidad en una línea de manufactura [31], sistemas de control de calidad en textiles [32], análisis de corrosión en tuberías [8], etc. En [29] se listan varios tipos de aplicaciones de sistemas de visión en la industria, en [33] se usa para la inspección de calidad de madera, en [34] para la recolección de fruta automática. En general, en las publicaciones mencionadas, se coincide en que la tecnología en visión de máquina mejora la productividad y la calidad de los productos, proporcionando ventajas evidentes a las empresas que las utilizan.

##### **1.4.1. Generalidades de los Sistemas de Inspección de Visión Industriales**

Tradicionalmente, las tareas de inspección visual y control de calidad son realizadas por expertos [35]. Aunque los expertos pueden realizar mejor trabajo en muchos de los casos, las máquinas pueden ser más eficientes y más veloces. Además los expertos requieren tiempo de entrenamiento extenso, y considerando que algunas tareas deben hacerse de forma repetitiva y rápidamente, el uso de máquinas con visión artificial se convierte en una solución para una variedad de aplicaciones.

La Figura 4 muestra la estructura de un sistema típico de visión de máquina industrial. El núcleo del sistema es un módulo de adquisición y procesamiento de vídeo, el procesamiento puede ejecutarse en un sistema embebido o en una

computadora industrial donde se procesan algoritmos destinados a reconocer ciertas características particulares de los objetos en la escena. Puede o no existir un sistema de monitoreo local, además de un sistema de comunicación industrial asociado a la maquinaria de la línea de producción. Las imágenes son adquiridas por una o más cámaras en la escena, usualmente la posición de las cámaras es fija, y los sistemas son diseñados para inspeccionar objetos conocidos en posiciones fijas [29]. La escena es correctamente iluminada para facilitar la adquisición de buenas imágenes y para resaltar las características que se desean evaluar.



**Figura 4. Sistema típico de visión de máquina industrial, modificada de [29]**

#### **1.4.2. Clasificación de los Sistemas de Visión Industriales**

Los sistemas de visión industriales se han clasificado de diversas maneras, en general dependiendo del tipo de inspección visual que realicen, en [36] se estudia una clasificación dependiendo del tipo de aplicación en la que se usa el sistema de visión industrial, mientras que en [29] se realiza una clasificación asociada al tipo de característica que mide el sistema de inspección industrial. De la primera clasificación se listan 4 tipos de sistemas de visión industrial:

#### **1.4.2.1 Sistemas de Inspección Visual Automatizada**

La utilización de estos sistemas mejora considerablemente la capacidad de automatización del proceso de manufactura. Estos sistemas pueden ser usados para realizar medidas, integridad y control de calidad de objetos. Ejemplos de este tipo de sistemas son: Sistemas de inspección de productos farmacéuticos, ensamble de automóviles, evaluación de PCB's, inspección de metales usados en la industria aeroespacial, etc. [36].

#### **1.4.2.2 Sistemas Para Control de Procesos**

Estos sistemas de visión se usan para procesos de control o tracking en industrias como la farmacéutica, biomédica, automotriz, etc. Permiten controlar procesos de manufactura, donde el sistema primario de adquisición es el sensor en el lazo de control, luego existe un procesamiento de la información y un conjunto de decisiones de acuerdo a los resultados del procesamiento. En general son lazos de control tipo ON/OFF, también es común encontrar sistemas de control de posición que involucran visión de máquina industrial como en [37]. En [36] se listan aplicaciones particulares de este tipo de sistemas.

#### **1.4.2.3 Sistemas de Identificación de Partes**

Los sistemas de visión que tienen tareas de identificación y clasificación son uno de los más importantes en las aplicaciones industriales. Estos sistemas determinan un conjunto de características de los elementos en la escena como tamaño, forma, color, posición, etc. Con ésta información el sistema clasifica los objetos en grupos predefinidos [36].

#### **1.4.2.4 Sistemas Para Guiado y Control de Robots**

Los procesos de automatización en la industria están relacionados estrechamente con el uso de robots que requieren sistemas de guía. Usualmente el objetivo es el posicionamiento 2D y 3D de elementos finales en los robots. En [36] se listan posibles aplicaciones de este tipo de sistemas.

La clasificación realizada en [29] está relacionada con el tipo de inspección que realizan los sistemas de visión, y divide los sistemas de visión en 4 tipos.

#### **1.4.2.5 Sistemas de Inspección de Calidad Dimensional**

Son sistemas de inspección que verifican cuales de las dimensiones de un objeto tienen las tolerancias especificadas o la forma correcta. Tales tareas involucran inspección de características geométricas de los objetos en dos o tres dimensiones. En [29] se listan un grupo de aplicaciones de sistemas de inspección de calidad dimensional.

#### **1.4.2.6 Sistemas de Inspección de Calidad de Superficies**

Son sistemas de inspección que buscan en las superficies de los objetos características principales como ralladuras, grietas, desgaste, texturas particulares, etc. Estos sistemas se encuentran típicamente en líneas industriales de textiles [32], maderas [33], [38], metales [8]; y se usan para detectar defectos y realizan tareas de verificación de calidad. En [29] se hace un recuento de más aplicaciones particulares de este tipo de sistemas de inspección.

#### **1.4.2.7 Sistemas de Inspección de Calidad Estructural**

Son sistemas de inspección que determinan si existen componentes faltantes en el ensamble de un objeto, o determinan si existen objetos extra que no deberían estar normalmente. Este tipo de sistemas son comúnmente usados en la industria de ensamble electrónico para determinar polaridad de capacitores, identificación de IC. En [39] se usa un sistema de inspección de calidad estructural en el control de calidad de componentes en tarjetas impresas PCB's. En [40] se propone un método de reconocimiento de patrones para detección de anomalías en ensamblaje de automóviles usando este tipo de sistemas. En [29] se hace una revisión más amplia de este tipo de sistemas de inspección.

#### **1.4.2.8 Sistemas de Inspección de Calidad Operacional**

Los sistemas de inspección de calidad operacional están relacionados con la verificación de la correcta operación y precisión de los objetos inspeccionados de acuerdo a los estándares de manufactura. Por ejemplo la inspección de la soldadura realizada por un sistema laser [41]. En [42] se usa este tipo de sistemas para verificar la operación correcta de relojes de manecillas usando un reloj de referencia para comparar la operación de los objetos evaluados.

### **1.5 Estado del Arte de sistemas ópticos industriales de inspección**

Durante varios años se han desarrollado e integrado en la industria los sistemas de visión artificial, actualmente ésta es una tecnología a tener en cuenta en todas las industrias, principalmente para extraer características de calidad de los productos en la línea de producción. Sistemas de adquisición (cámaras 2D o 3D), sistemas de procesamiento (FPGA, DSP, ARM), buses de campo (Modbus RTU, Modbus TCP, Profibus), módulos software de procesamiento, sistemas software y hardware complementarios componen los sistemas de inspección de características de calidad (SICC) industriales. A continuación se hace referencia a los principales desarrolladores de ésta tecnología a nivel nacional e internacional.

#### **1.5.1 A Nivel Internacional**

En el mundo existen pocas empresas que desarrollan tecnología de visión artificial para aplicaciones industriales, sus desarrollos y dispositivos se usan en líneas de manufactura de muchas partes del mundo. A continuación se presentan tres de las más importantes empresas que desarrollan tecnología en este campo.

##### **1.5.1.1 Cognex**

Empresa creada por un ex-profesor del MIT y dos de sus más destacados estudiantes. En 1982 fabrican su primer dispositivo de visión industrial y para 1995 ya era líder del mercado mundial en visión artificial industrial [7]. Actualmente

Cognex cuenta con una línea de visión muy estructurada, sus más destacados productos son:

In-Sight System Vision: Son sistemas de visión autónomos que combinan una herramienta de visión artificial con captura y procesamiento de alta velocidad. El sistema cuenta con un software de configuración llamado *EasyBuilder* que permite a los usuarios manejar los sistemas aún si no tienen conocimiento de programación. El sistema también cuenta con una interface software de inspección y control de la cámara llamado *VisionView*, además de periféricos VGA para supervisar en tiempo de ejecución la captura del sensor. La línea de visión ofrece diversos tipos de sistemas dependiendo de la aplicación particular [43].

In-Sight 500: Sistema de visión con protección industrial IP65, intensifica la velocidad de procesamiento de imágenes hasta 500 cuadros por segundo (fps). Permite la inspección por blobs (grandes objetos binarios), histogramas y bordes. Además de compatibilidad con EthernetIP y Profinet como protocolos de comunicación [43].

In-Sight 5000: Es una serie especial de hardware (Cámara, lentes y procesador) resistente. Tiene solidas carcasas de aluminio fundido y de acero inoxidable, conectores impermeables y protección IP67 e IP68. El sistema cuenta con una herramienta de visión artificial que incluye: lecturas de códigos de aprendizaje sencillos, OCR, inspección, medición, color y localización de piezas [43].

In-Sight Micro: Es un sistema de visión completo y compacto, ya que sólo mide 30mmx30mmx60mm. Contiene una herramienta de calibrado no lineal para montar con ángulos de hasta 45°, fácil de instalar en espacios muy reducidos. Usa el software de configuración *EasyBuilder*. No necesita un PC, el sistema *VisionView* permite monitorear el proceso, con hasta 9 cámaras, usando redes industriales [43].

In-Sight 7000: Cuenta con iluminación configurable y lentes integrados, autoenfoco y captura de imágenes veloz (100fps). Todo esto en una carcasa industrial compacta IP67. Los protocolos de comunicación con los que cuenta son: Ethernet, RS232 serial y múltiples entradas y salidas discretas. El sistema puede comunicarse con PLC's, permite controlarse y configurarse en red [43].

#### **1.5.1.2 Rockwell Automation**

Rockwell Automation hace presencia en el área de control de calidad en visión de máquina con un producto con procesamiento embebido [5], que se describe a continuación:

MultiSight 48MS: Es un sensor óptico con salida PNP de paso/rechazo. El sistema usa cuatro diferentes métodos de evaluación (detección de patrones, contraste, brillo y detección de contornos). La principal aplicación es en el campo del control de calidad en líneas industriales. El sistema puede detectar presencia o ausencia de objetos, objetos incompletos, posición de marquillas, etiquetas, empaques y detección de componentes. Tiene la capacidad de trabajar en modo *Stand-Alone*. Su carcasa es industrial con protección IP67, tiene iluminación integrada. Los tiempos de evaluación y de ejecución de los algoritmos van desde 50ms para detección de brillo y contraste hasta 500ms para detección de contornos. Tiene la capacidad de configuración de 32 detectores virtuales, los que pueden ser operados lógicamente para crear detectores compuestos. El sistema posee comunicación Ethernet/IP y capacidad de integrarse con PLC's RSLogix 5000 [44].

#### **1.5.1.3 Vitronic**

Vitronic tiene operaciones en cuatro continentes y es una de las empresas pioneras en visión de máquina a nivel mundial, es una empresa fundada en Alemania en el año 1984 [6]. Sus productos más destacados en cuanto a visión artificial industrial embebida para la inspección de características de calidad se detallan a continuación:

VICAMSnap: Es un sistema compacto, embebido, que tiene la capacidad de reconocer códigos de barras y códigos 2D. El proceso de identificación se hace mientras se manipulan los objetos haciendo eficiente la identificación de los productos sin importar si están siendo ordenados, recogidos o almacenados. El sistema tiene iluminación incorporada y un software de monitoreo de inspección [45].

VICAMssi2: Es un sistema ligero de reconocimiento de barras, códigos 2D, OCR. La unidad consta de una cámara de alta velocidad, compacta, iluminación LED incorporada, y un mini PC integrado con un paquete de procesamiento básico para tareas específicas y funcionamiento en modo *Stand-Alone*. Puede usarse en bandas transportadoras con velocidades de hasta 885 ft/min. Mecanismo de enfoque adaptable según las velocidades de adquisición e imágenes con resolución de hasta 170dpi [46].

### **1.5.2 A Nivel Nacional**

Si bien en el país se han desarrollado diversos trabajos relacionados con la extracción de características de calidad para aplicaciones industriales, tales como: Control de calidad de empaques usados en compresores centrífugos en [47], sistema de control de calidad de piezas mecanizadas en una línea de manufactura, sistema 3D de reconstrucción de piezas para comparación de similitudes con diseños CAD para detectar defectos mecánicos en las piezas. Estos trabajos se desarrollan como sistemas poco compactos, con capacidad de comunicaciones limitadas y se ejecutan en computadores convencionales. En general, a nivel industrial, se integran sistemas de visión importados de los grandes fabricantes mundiales.

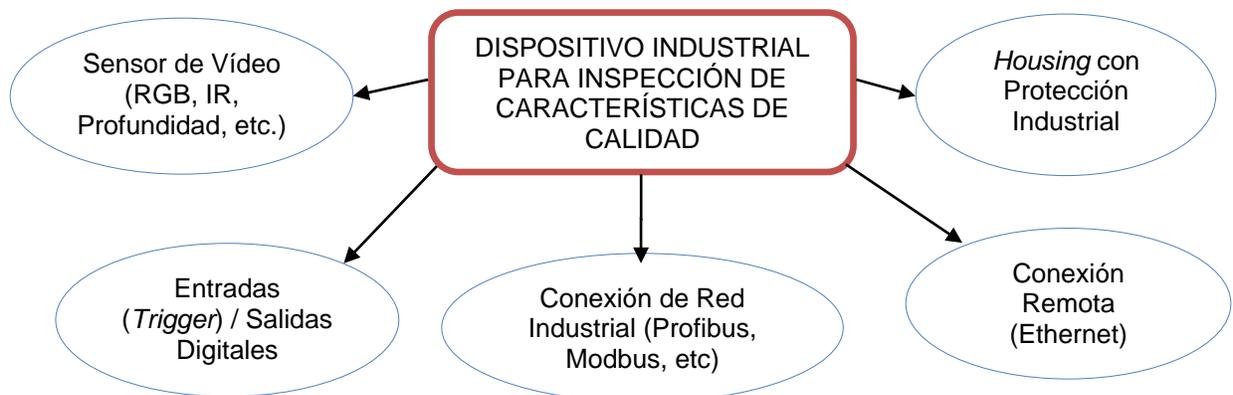
### **1.5.3 A Nivel Local**

Localmente en el suroccidente Colombiano se han diseñado prototipos académicos que potencialmente pueden integrarse en una línea de producción

industrial, como el presentado en [48], en donde se aplica la visión de máquina para la detección de defectos de frutas que permiten su clasificación. Finalmente se encuentra un desarrollo realizado para la ILC donde se integró a la línea de envasado un sistema de control de calidad y gestión de información basado en una RTU de vídeo con capacidad de procesamiento ejecutada en un computador industrial. Este sistema fue realizado por investigadores locales de la empresa *BeAutomation S.A.S.* El sistema actualmente ejecuta tareas de control de calidad en la unidad formadora de cajas con resultados satisfactorios. El sistema incorporado a la línea de producción de la ILC está orientado al conteo de botellas antes de que sean empacadas en las cajas de las diferentes presentaciones con el fin de garantizar que no exista producto faltante.

### 1.6 Esquema General de un Dispositivo Industrial Para Inspección de Características de Calidad

En general, todos los productos industriales descritos anteriormente pueden representarse esquemáticamente como una variación particular del esquema general mostrado en la Figura 5. Algunos dispositivos pueden contar con los 5 componentes mostrados en el esquema, y otros pueden tener algunos de ellos.



**Figura 5. Esquema general de componentes de un dispositivo industrial de inspección de características de calidad. Fuente propia.**

Para cada producto industrial descrito en la sección anterior se presenta en la Tabla 2 los componentes específicos asociados al esquema de la Figura 5.

<b>Dispositivo</b>	<b>Sensor de Video</b>	<b>Entradas/Salidas Digitales</b>	<b>Conexión de Red Industrial</b>	<b>Conexión Remota</b>	<b>Housing Industrial</b>
In-Sight 500	CMOS RGB	4 Entradas – 4 Salidas de Relé	Profinet, Modbus TCP	Ethernet TCP	IP65
In-Sight 5000	CMOS RGB	Conector M12	Profinet, Modbus TCP	Ethernet TCP	IP67-IP68
In-Sight Micro	CMOS RGB	Conector M8	<b>X</b>	Ethernet TCP	Zinc y Acero
In-Sight 7000	CMOS RGB	Conector M12	Profinet, Modbus TCP	Ethernet TCP	IP67
MultiSight 48MS	CCD	2 Entradas – 4 Salidas de Relé	<b>X</b>	Ethernet TCP	IP67
VICAMSnap	CMOS RGB	Hasta 24 Entradas. 6 Salidas Led de Estatus.	RS232	Ethernet TCP	IP54
VICAMssi2	CMOS RGB	<b>X</b>	<b>X</b>	Protocolo propietario	IP54

**Tabla 2. Componentes básicos de algunas cámaras industriales existentes en el mercado mundial actual. Fuente propia.**

## **2 DISEÑO DEL SISTEMA DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD**

El presente proyecto está enfocado hacia el diseño y desarrollo de un producto de carácter industrial que sea capaz de evaluar características de calidad en una línea de producción usando visión artificial en un sistema embebido. El sistema se piensa como un sistema que identifica características de calidad en una línea de manufactura. Como industria caso de estudio se trabaja con la Industria Licorera del Cauca (ILC).

La ILC presenta considerable cantidad de PQR's (Peticiónes Quejas y Reclamos) por factores como problemas con la etiqueta hasta producto faltante, ver Tabla 3. Como se aprecia, las PQR's más altas se deben a aspectos relacionados con la etiqueta (46.12%), líquido bajo (15.72%), tapa dañada (12.20%), etc. Las PQR's por producto faltante en caja presenta el porcentaje más bajo (2,20 %), sin embargo el problema de botellas faltantes en cajas se extiende a otros aspectos del negocio y no sólo molesta a los clientes por la falta del producto, también afecta la imagen de la licorera e internamente al área de producción.

La máquina encartonadora es la que se encarga de empacar en cajas las botellas de las presentaciones media, botella y garrafa. La banda transportadora lleva las botellas hasta el lugar de empacado organizadas en una matriz de 24, 12 o 6 botellas dependiendo de la presentación del producto, hasta donde se forma una caja a su alrededor. Este proceso consume en promedio 5 segundos para las 3 presentaciones. En ocasiones, las matrices llegan a la máquina encartonadora con botellas faltantes, extras o con botellas caídas. Si la matriz tiene botellas faltantes, las cajas se empacan con producto faltante, llegan al cliente y se generan las PQR's asociadas. Si la matriz tiene botellas extras o botellas caídas, los mecanismos que empacan la matriz en las cajas o las varillas que transportan las botellas provocan que las botellas se quiebren y se ocasiona pérdida de líquido y botellas, además de tiempos muertos en la línea de producción. Por todo lo

anterior se decide de común acuerdo con Be Automation S.A.S que el sistema a diseñar se concentre en atacar el problema de botellas faltantes.

<b>PQR's 2012 (Enero - Diciembre)</b>	<b>Cantidad (Unidades)</b>	<b>Porcentaje (%)</b>
Tapa dañada, cintillo de seguridad roto o mal tapado	222	12,20%
Etiqueta rota o ausente	839	46,12%
Defecto en el vidrio, roto, fisurado o rayado.	149	8,19%
Líquido bajo o sin líquido	286	15,72%
Sedimentación o partículas en suspensión	175	9,62%
Faltantes en caja	40	2,20%
Análisis	108	5,94%
<b>TOTAL (154 Reclamos):</b>	<b>1819</b>	<b>100%</b>

**Tabla 3. Lista de PQR's en la ILC en el año 2012. Fuente documentación ILC**

Para dar solución al problema mencionado, el sistema a diseñar debe tener la capacidad de: contar objetos (paso 1), verificar que la matriz este completa (paso 2), de lo contrario parar la encartonadora (paso 3) y posteriormente reiniciarla cuando el operario lo indique (paso 4)<sup>1</sup>. Se propone el diseño y construcción de un producto industrial siguiendo las etapas de diseño descritas en [49].

1. Generación de la idea.
2. Investigación de Mercado Exploratoria
3. Especificaciones funcionales del producto.
4. Modelo conceptual.
5. Diseño preliminar.
6. Diseño detallado.
7. Diseño del proceso de producción.

---

<sup>1</sup> Los tres primeros pasos se deben realizar en menos del tiempo promedio de tránsito de la matriz en la banda transportadora.

Con el objetivo de llegar a un prototipo funcional que cumpla con las especificaciones requeridas se siguen los puntos 1, 3, 4 y 5 de la lista anteriormente mencionada para la generación de un producto industrial.

## **2.1 Generación de la idea**

El producto diseñado en este trabajo parte de una necesidad real en la industria local. Un problema que desde el enfoque de este trabajo puede solucionarse construyendo un producto que se instale en la línea de producción. En síntesis, la idea se orientó al desarrollo de un sistema prototipo industrial escalable para realizar la inspección de características particulares de calidad en una industria local.

## **2.2 Especificaciones del producto**

Debido a que el proyecto está orientado al desarrollo de un sistema óptico embebido de inspección de características de calidad de una línea de producción industrial, se deben considerar especificaciones técnicas y funcionales en el diseño final del producto que se mencionan a continuación:

### **2.2.1 Especificaciones Técnicas**

Dentro de las especificaciones técnicas del producto se consideran: composición física, velocidad de procesamiento, componentes del sistema, características eléctricas.

1. El sistema debe tener un *housing* protector de tipo industrial, que le brinde protección física.
2. El sistema debe estar soportado en una plataforma embebida de procesamiento.
3. El sistema debe tener alimentación eléctrica de 110V.
4. El sistema debe contar con protección contra sobre flujo de corriente de 5A.

5. El sistema debe contar con pasa-cables para ingresar el cableado desde el exterior hacia la circuitería interna.
6. El sistema debe trabajar a velocidad de procesamiento menor a 2 marcos por segundo.
7. El sistema debe contar con un sensor de vídeo RGB, un sensor de vídeo de profundidad y un sensor de presencia.
8. El sistema debe contar con un extractor para la protección contra el polvo.
9. El sistema debe contar con un módulo de comunicación Industrial Modbus RTU para el manejo de Entradas/Salidas digitales.
10. El sistema debe contar con un módulo de comunicación de red Ethernet para la configuración y mando remoto.

## **2.2.2 Especificaciones Funcionales**

Dentro de las especificaciones funcionales se requiere que el prototipo cumpla con las siguientes características:

1. El sistema debe tener visualización local de vídeo, eventos de procesamiento y resultados de inspección.
2. El sistema debe tener visualización remota de eventos de procesamiento y resultados de inspección.
3. El sistema debe contar con una interface de usuario de monitoreo y control local.
4. El sistema debe contar con una interface de usuario de control remota.
5. El sistema debe contar con un piloto de estatus del encendido eléctrico.
6. El sistema debe contar con un piloto de alarma de evaluación negativa en la inspección de las características de calidad.
7. El sistema debe contar con una llave de encendido eléctrico.
8. El sistema debe contar con un interruptor de encendido del módulo de procesamiento y adquisición de vídeo.
9. El sistema debe contar con una interface Ethernet para conexión de red local.

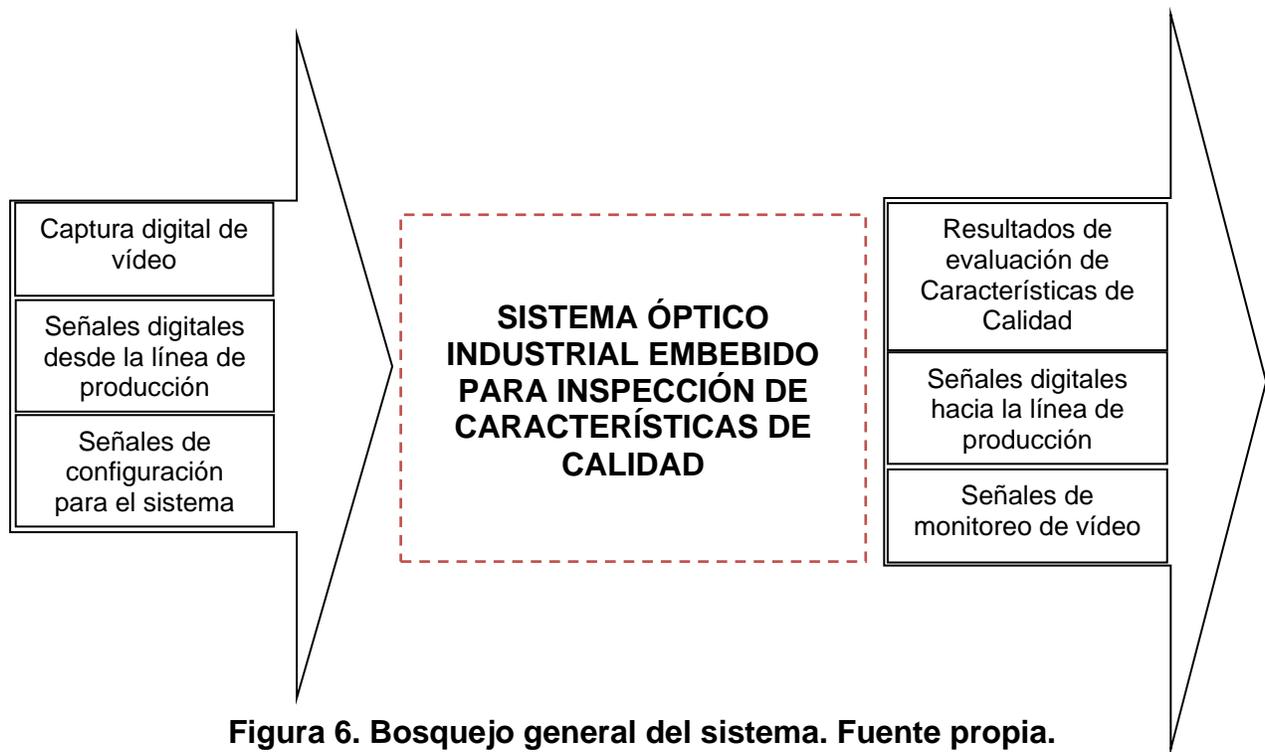
10. El sistema debe tener una aplicación de conteo de objetos.
11. El sistema debe tener una aplicación de detección de patrones.
12. El sistema debe tener la capacidad de parar o arrancar una máquina a través de una salida digital de relé.
13. El sistema debe permitir la visualización de vídeo continua en el tiempo.
14. El sistema debe permitir fácil acceso a su interior para efectos de mantenimiento.

### **2.3 Modelo conceptual**

Partiendo de las especificaciones funcionales y técnicas, el sistema prototipo se resume de la siguiente manera: El sistema debe contar con un elemento hardware de carácter industrial y software que permita la adquisición, procesamiento de vídeo embebido en un ambiente industrial y realizar inspección de características de calidad particulares. El procesamiento se iniciará una vez señales digitales externas (por ejemplo una señal de *trigger*) marquen la presencia de los objetos de interés. Su configuración se puede realizar en forma local o remota usando sockets TCP. El vídeo se monitoreará de manera local usando una pantalla e interface de usuario. Los resultados de inspección se monitorearán de forma local y remota usando sockets TCP. Contará con una aplicación cliente TCP para el control y configuración remoto. Según el resultado de la inspección el sistema debe tomar acciones sobre la línea industrial a través de dispositivos de entradas/salidas digitales de una red industrial Modbus RTU.

### **2.4 Diseño preliminar**

Partiendo del modelo conceptual, los requerimientos y especificaciones de diseño, se establece un diagrama de caja negra genérico del sistema, Figura 6, en donde las entradas principales son una captura digital de vídeo, señales de configuración y señales digitales emitidas por sensores en la línea de producción y las salidas son resultados de evaluación, señales de monitoreo y órdenes digitales dirigidas a la línea de producción.



**Figura 6. Bosquejo general del sistema. Fuente propia.**

El sistema está compuesto por 6 módulos que se describen a continuación y se ilustran en la Figura 7.

\_Módulo de adquisición de vídeo: Lo compone una cámara especializada de profundidad (Kinect) y un módulo software de adquisición embebido. Este módulo satisface la especificación funcional 2 y 6.

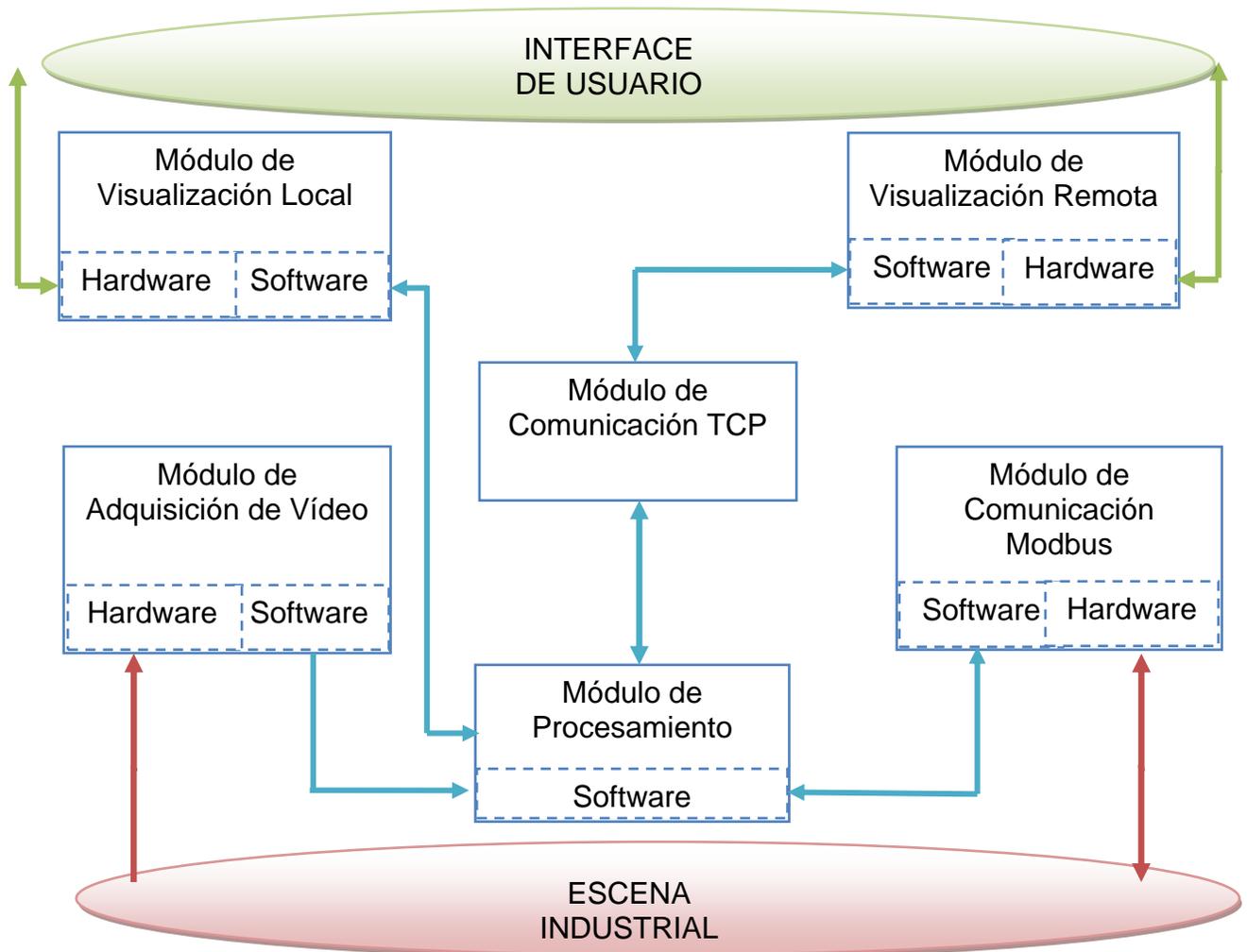
\_Módulo de procesamiento de vídeo: Lo compone un aplicativo software soportado sobre un sistema embebido de procesamiento de vídeo que realiza dos tareas particulares: contar objetos segmentados por profundidad y detectar patrones particulares. Además de recibir mensajes de otros módulos y ejecutar acciones sobre el sistema. Este módulo satisface la especificación funcional 3, 6, 13 y 14.

\_Módulo de comunicación TCP: Lo compone un software de comunicación TCP representado como un servidor de datos de procesamiento. Este módulo satisface la especificación funcional 4 y 8.

\_Módulo de comunicación Modbus RTU: Lo compone un software de comunicación Modbus RTU y un hardware I/O Modbus asociado (*DigiRail's*). Este módulo satisface la especificación funcional 5, 10 y 11.

\_Módulo de Visualización Local: Lo compone un software GUI que presenta el vídeo adquirido y procesado, además de los resultados de evaluación y estado de las comunicaciones. Este módulo satisface la especificación funcional 7 y 12.

\_Módulo de Visualización Remota: Lo compone una aplicación remota cliente TCP que tiene acceso a la información de procesamiento e inspección de las características de calidad que puede ejecutarse on-line. Este módulo satisface la especificación funcional 9.



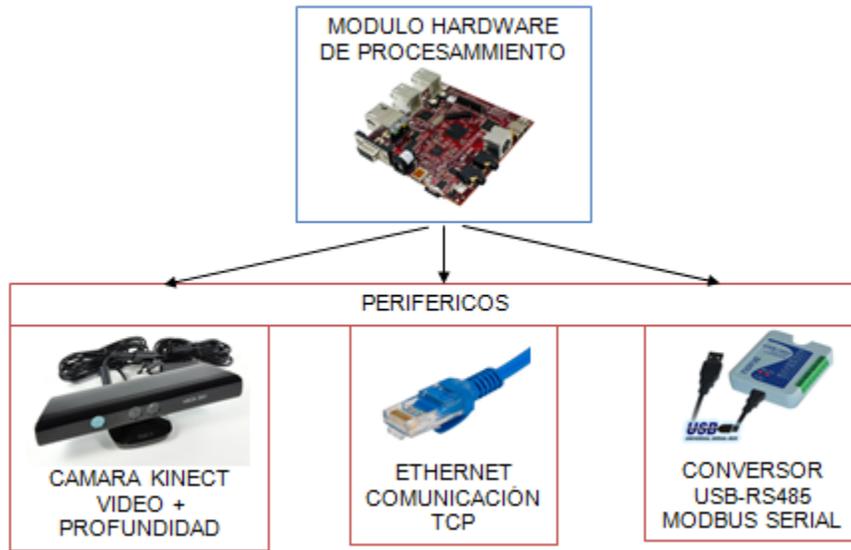
**Figura 7. Módulos componentes del SICC. Fuente propia.**

Los módulos que interactúan directamente con la escena industrial son los módulos de adquisición de vídeo a través de un sensor de vídeo y el módulo de comunicación Modbus a través de dispositivos I/O. El módulo de adquisición de vídeo digitaliza la imagen y la envía codificada al módulo de procesamiento. A su vez el módulo de procesamiento realiza cálculos y estimaciones a partir de la imagen digital y envía los resultados al módulo de visualización local, además envía la imagen para la visualización hacia la interface de usuario. Por otro lado los resultados de procesamiento se envían al módulo de visualización remota a través del módulo de comunicación TCP. Los eventos digitales en la escena industrial son capturados por el módulo de comunicación Modbus y se envían al módulo de procesamiento, éste módulo envía de vuelta al módulo de comunicación Modbus salidas de alarma y salidas digitales hacia la escena industrial.

De acuerdo al diseño modular descrito anteriormente se procede a diseñar cada uno de los módulos descritos de acuerdo a las especificaciones funcionales mencionadas.

#### **2.4.1 Diseño Módulo de Procesamiento**

El módulo de procesamiento es el núcleo software y hardware del sistema. En concordancia con las especificaciones requeridas, el sistema está soportado sobre un sistema de procesamiento embebido basado en arquitectura ARM Cortex8, particularmente en la tarjeta *BeagleBoard-xM* (BB-xM). El módulo de procesamiento ejecutado sobre la tarjeta con las especificaciones mencionadas en la sección **1.1.3.1**, es soportado sobre un Sistema Operativo (SO) Linux, en particular la distribución Ubuntu embedded 12.10. La Figura 8 muestra el módulo hardware de procesamiento (BB-xM) y los periféricos de interface que suministran la información necesaria para el procesamiento.



**Figura 8. Módulo de procesamiento. Fuente propia.**

Una vez el módulo de procesamiento hardware está definido, es necesario definir el módulo software asociado, ver Figura 9. El desarrollo software principal está soportado sobre las siguientes librerías:

- \_QtProject para el desarrollo de la interface gráfica de usuario GUI.
- \_Libfreenect para la adquisición y el procesamiento básico de la cámara Kinect.
- \_Librerías Nativas de Linux para el manejo de comunicaciones TCP y Modbus.

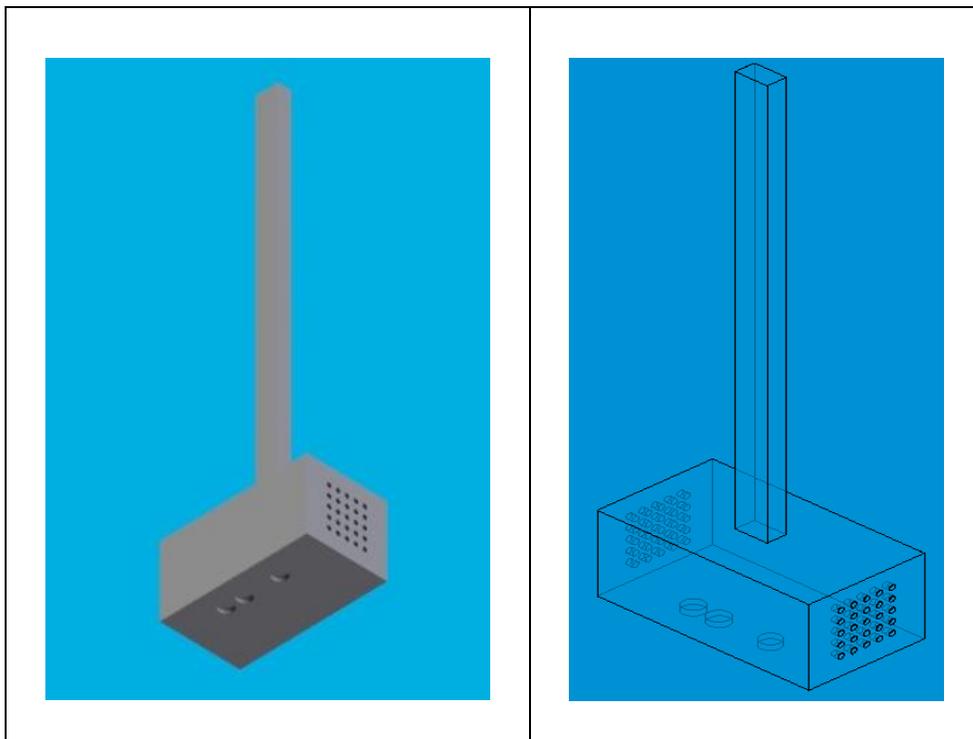


**Figura 9. Librerías asociadas a los componentes del sistema. Fuente propia.**

### 2.4.2 Diseño Módulo de Adquisición de Vídeo

El sistema de adquisición de vídeo está constituido por una cámara de vídeo y un módulo software de adquisición. La cámara de vídeo debe ser dispuesta de tal forma que pueda incluirse en un ambiente industrial típico. Además, la cámara elegida (Kinect) debe ser capaz de integrarse a un sistema de procesamiento embebido.

El sistema de adquisición de vídeo está compuesto por una cámara de profundidad y RGB conocida como Kinect la que se describe en la sección 1.2.2. La interface hardware de la cámara es USB. Se diseñó el modelo CAD del *enclosure* para que la cámara Kinect pueda integrarse en el ambiente industrial, ver Figura 10.



**Figura 10. Diseño CAD de *Enclosure* para la cámara Kinect. Fuente propia.**

El diseño del *enclosure* mostrado en la Figura 10 cuenta con tres orificios circulares que corresponden con los lentes de la cámara RGB y el sensor de

profundidad. Además de una rejilla de ventilación y una barra para dar soporte mecánico.

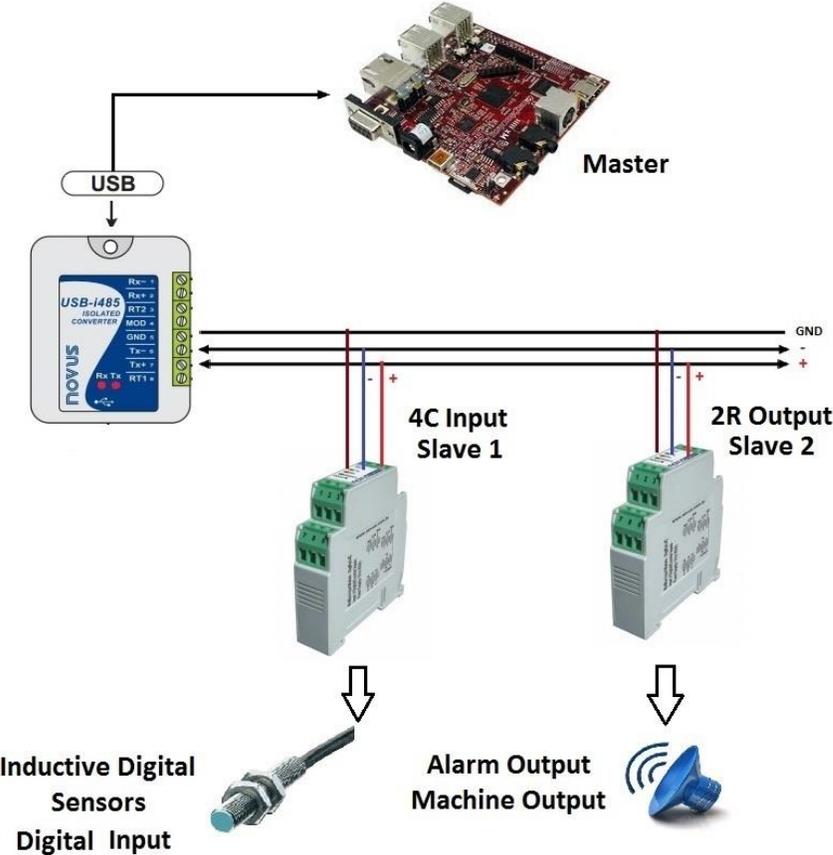
Desde la perspectiva de software, la adquisición de vídeo, se realiza usando las librerías libfreenect, con estas se obtiene, del sensor de vídeo, una imagen RGB y una imagen en mapa de profundidad listas para procesar. En caso de no usar el sensor Kinect si no una cámara convencional, se usan las librerías OpenCV para la adquisición de vídeo. Si eventualmente se prefiere usar una cámara industrial se debe integrar el API que corresponda en la aplicación desarrollada.

### **2.4.3 Diseño Módulo de Comunicación Modbus**

La comunicación entre SICC y el entorno industrial es crítica para la aplicación. Existen diversos protocolos de comunicación industrial, sin embargo se decide realizar el diseño de este prototipo usando comunicación industrial Modbus RTU por dos motivos principalmente: 1. El protocolo Modbus RTU es un estándar industrial ampliamente usado por su fácil desarrollo e implementación y su confiabilidad. 2. En concordancia con lo descrito en la sección **2.1** el sistema está orientado a la solución de un problema particular de la industria local ILC, en este caso particular la red industrial de comunicación en esta línea de manufactura es Modbus RTU.

El sistema cuenta con una *Gateway* que comunica el nivel físico del Modelo OSI [27] con la aplicación convirtiendo la señal de RS485 (*Half-Duplex*, 2 hilos) a USB, usando el conversor RS485-USB de *Novus*, las especificaciones del dispositivo se presentan en el Anexo A. El dispositivo maestro lo compone la tarjeta *BeagleBoard-xM* que ejecuta una aplicación desarrollada para manejar dos esclavos básicos de I/O. Los esclavos son: Un DigiRail de *Novus* 2R, dos salidas digitales, y un DigiRail de *Novus* 4C, cuatro entradas digitales. Los dispositivos mencionados (*GateWay*, DigiRail 2R, DigiRail 4C, Maestro Modbus) más un módulo software desarrollado para la comunicación Maestro-Esclavo constituyen el módulo Modbus RTU del sistema.

Las entradas digitales corresponden a un sensor de presencia que informará la presencia de un objeto de interés y un pulsador que activa una salida digital de relé en la línea industrial. Mientras que las salidas digitales serán una alarma y una acción de control sobre la máquina que transporta o clasifica los objetos, en la Figura 11 se presenta el diseño propuesto.



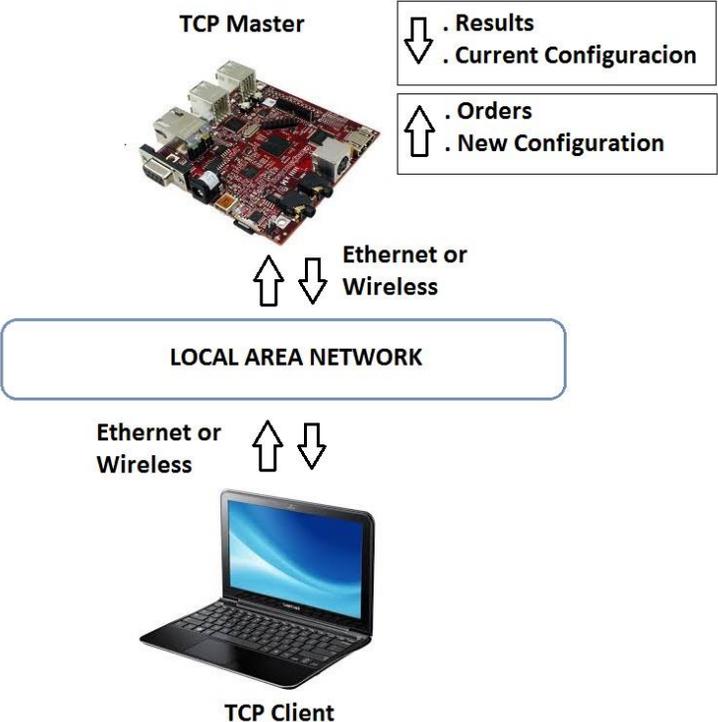
**Figura 11. Diseño del Módulo Modbus RTU de Comunicación. Fuente propia.**

**2.4.4 Diseño Modulo de Comunicación TCP**

Según los requerimientos, el sistema debe estar dotado de un módulo de comunicación TCP para la transferencia de información desde la aplicación a sistemas remotos que requieran la información, y para ejecutar órdenes desde los sistemas remotos. El módulo de comunicación TCP está constituido por un elemento hardware que hace referencia al nivel físico de conexión Ethernet o WiFi,

la tarjeta de procesamiento *BeagleBoard-xM* cuenta con el nivel físico cableado Ethernet y cuenta con la posibilidad de configurar receptores WiFi locales. Por otra parte, el módulo de comunicación TCP cuenta con una componente software desarrollada para manejar las comunicaciones desde la aplicación hacia los clientes remotos que necesiten información específica o que requieran dar órdenes al sistema.

La aplicación software es desarrollada usando las librerías nativas del sistema operativo (*Linux Embedded*) para el servidor de información, y se usan las librerías de *QtProject* para la aplicación de los clientes remotos. La información que se envía desde el servidor es referente a los resultados de evaluación y al estado de las comunicaciones TCP y Modbus. Por otra parte, la información que se envía desde los clientes remotos es referente a órdenes de arranque y parada del sistema, configuración de las aplicaciones, pedido de información de resultados de evaluación, etc. En la Figura 12 se presenta el diseño del módulo TCP de comunicaciones.



**Figura 12. Módulo TCP de comunicaciones. Fuente propia.**

### 2.4.5 Diseño Modulo de Visualización Local y Remoto

El sistema cuenta con un módulo de visualización local y remoto de resultados y eventos. Los dos sistemas de visualización se componen de 3 interfaces gráficas de usuario GUI's: 1. Interface principal: Contiene un menú de acceso a dos aplicaciones básicas (Conteo de objetos y detección de patrones) según se especificó en la sección 2 de este documento. 2. Interface de la aplicación Conteo de Objetos: Contiene visualización de resultados, visualización de vídeo (local) y menú de configuración para la aplicación. 3. Interface de la aplicación Detección de Patrones: Contiene visualización de resultados, visualización de vídeo (local) y menú de configuración para la aplicación.

En la Figura 13 se presenta el diseño de la plantilla GUI propuesta para la interface principal.

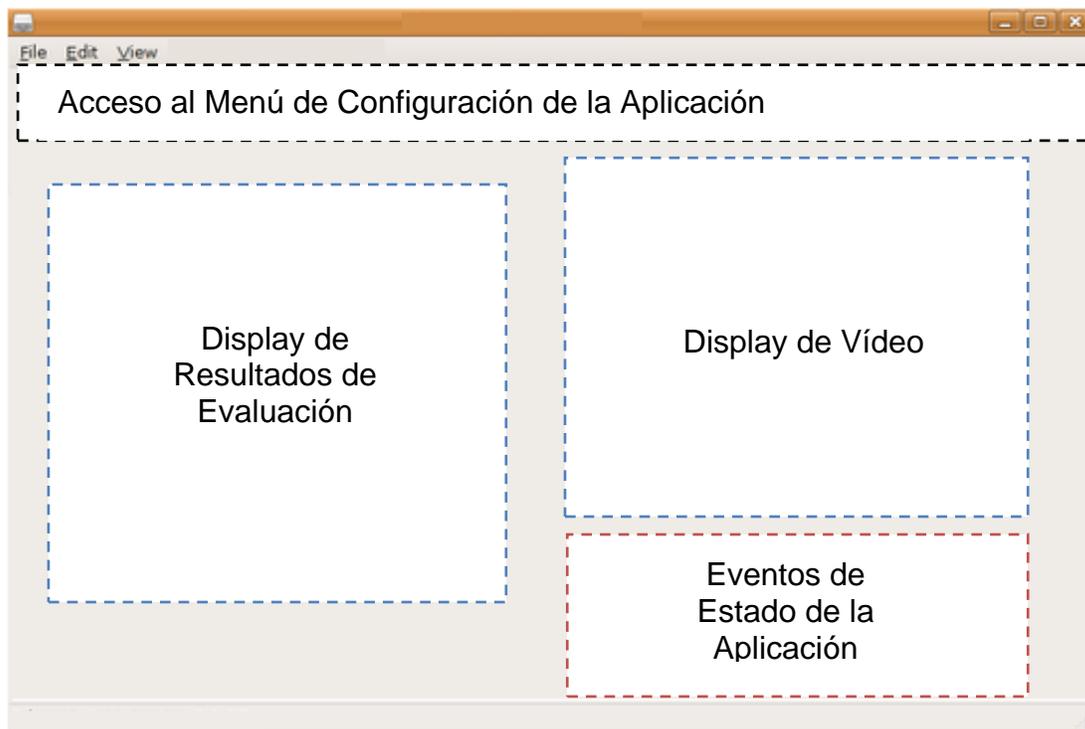


**Figura 13. Diseño GUI Para la Interface Principal. Fuente propia.**

El acceso a las aplicaciones de Conteo de Objetos y Reconocimiento de Patrones se realiza desde la barra de herramientas o desde los iconos en la interface. Existe también un *TextBox* que presenta todos los eventos que ocurran referentes a las comunicaciones Modbus y TCP. En la barra de estatus se lleva la información básica del servidor: IP Local y Puerto de comunicación.

Se plantea el mismo diseño gráfico para la interface de visualización remota y local. La aplicación remota está constituida por un cliente TCP desarrollado con librerías *QtProject*, la información desplegada en la interface es suministrada por el servidor TCP implementado en la tarjeta *BeagleBoard-xM*.

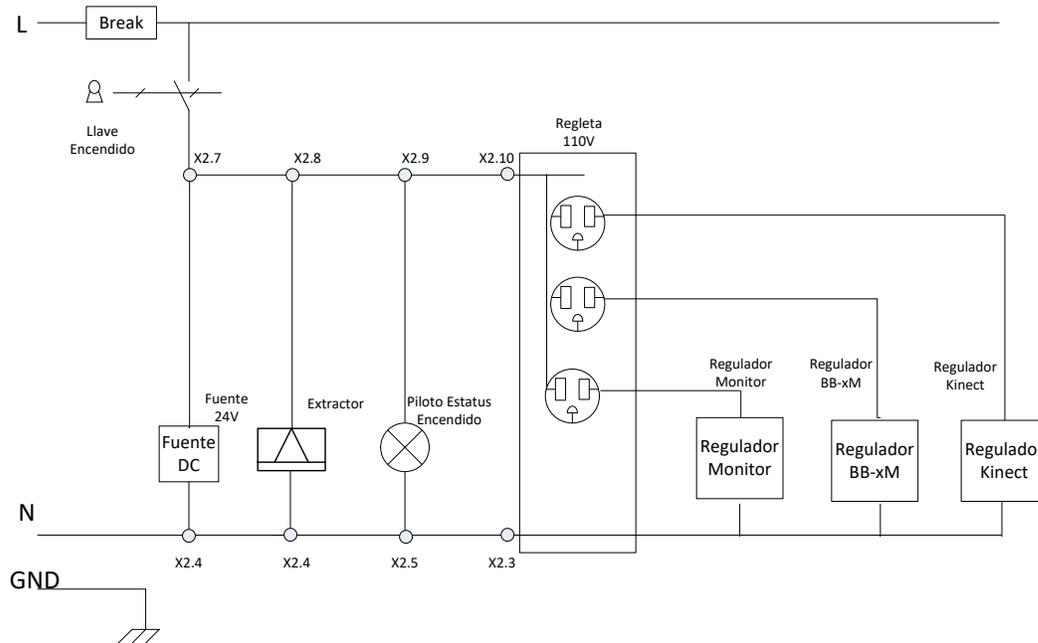
En la Figura 14 se presenta el diseño de la plantilla GUI propuesta para la interface de las aplicaciones Conteo de Objetos y Reconocimiento de Patrones.



**Figura 14. Diseño GUI Para la Interface de Aplicación. Fuente propia.**

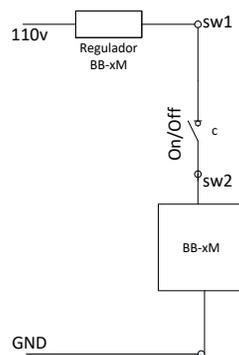
## 2.4.6 Diseño Eléctrico del sistema SICC

El diseño eléctrico de SICC lo compone el circuito de cableado de la red Modbus, DigiRail's de entrada y salida digitales, *break* de protección eléctrica, luz piloto de estatus de la potencia, interruptor de encendido, regulador de 24V, pantalla HDMI, cable de conexión *Ethernet*, cable USB para conexión del sensor Kinect, y conexión de potencia para el extractor. El diagrama de potencia y mando de 110v se presenta en la Figura 15.

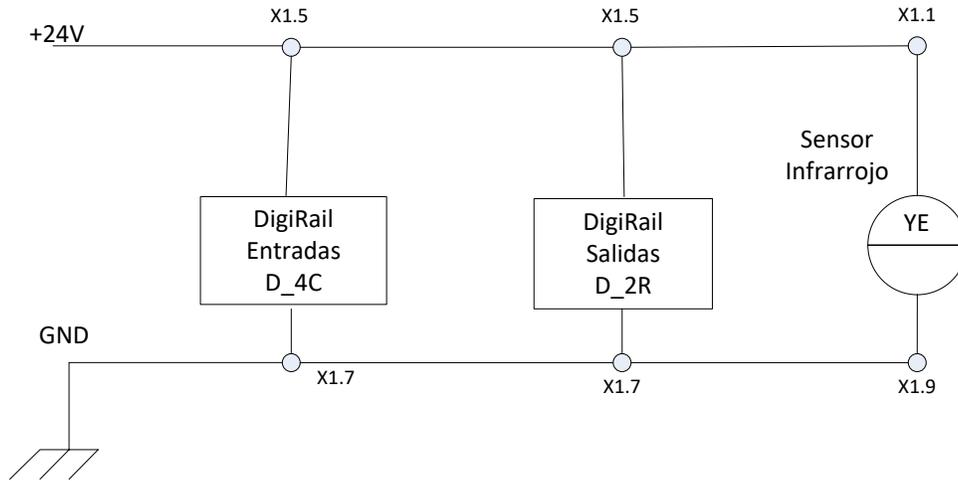


**Figura 15. Diagrama de potencia 110V del sistema SICC. Fuente propia.**

El diagrama de mando asociado al encendido de la tarjeta BB-xM se presenta en la Figura 16.

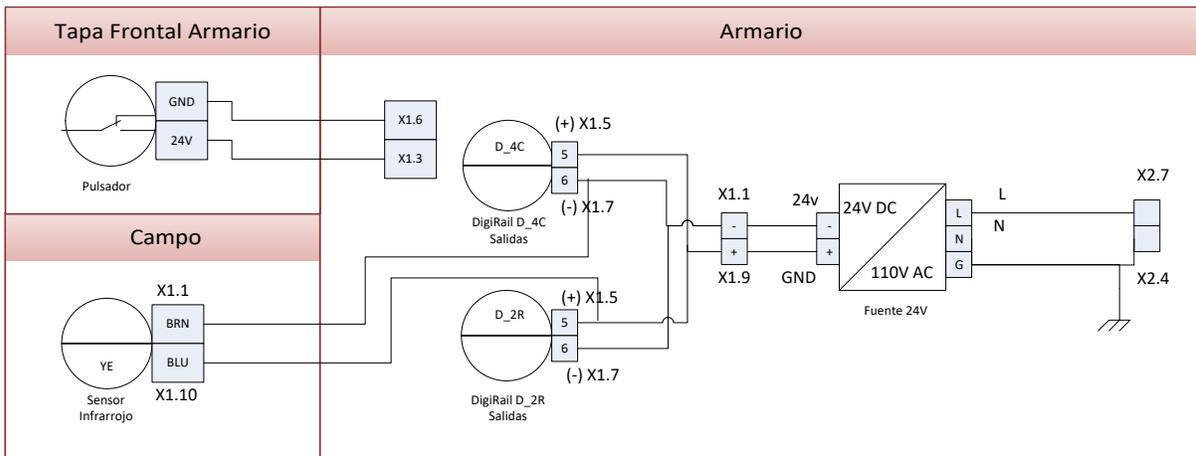


**Figura 16. Diagrama de mando de la tarjeta BB-xM. Fuente propia.**  
 En la Figura 17 se presenta el diagrama de potencia de 24V de SICC.

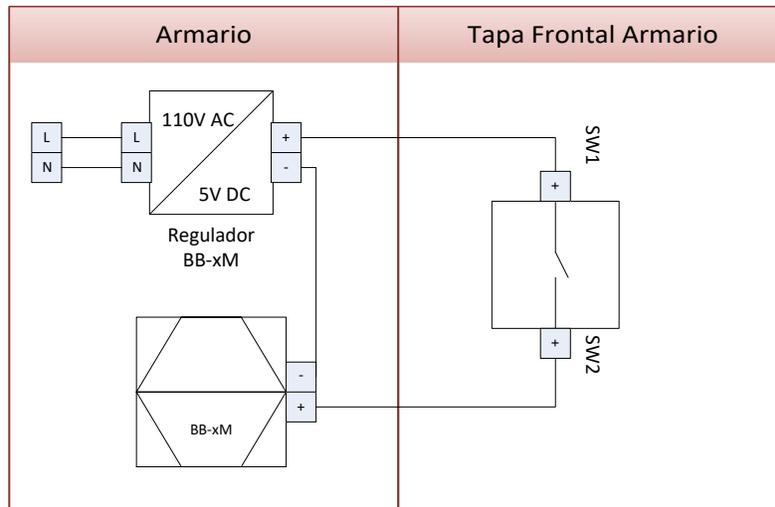


**Figura 17. Diagrama de potencia de 24V del sistema SICC. Fuente propia.**

Los diagramas de lazo asociados a la alimentación de instrumentación y de la tarjeta BB-xM se presentan en Figura 18 y Figura 19.

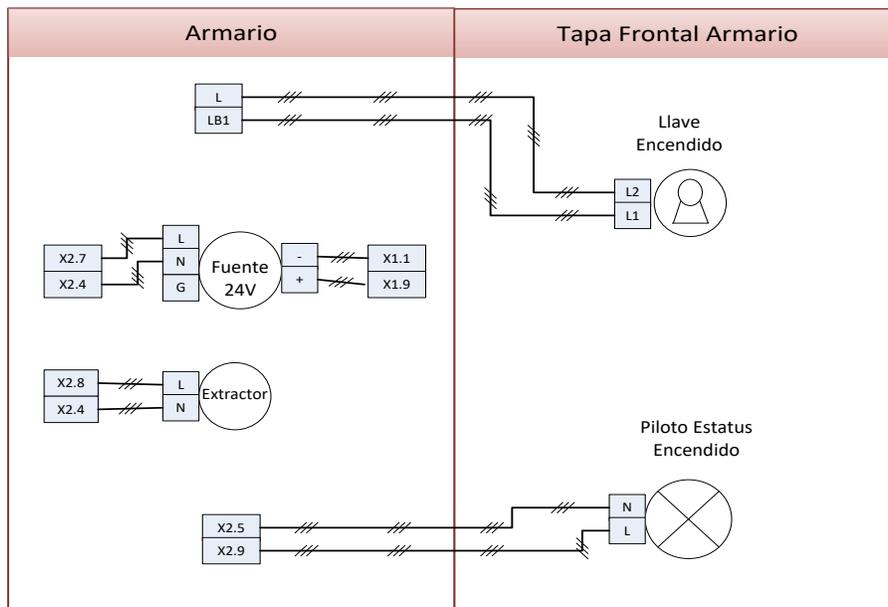


**Figura 18. Diagrama de lazo alimentación de instrumentación. Fuente propia.**



**Figura 19. Diagrama de lazo alimentación BeagleBoard-xM. Fuente propia.**

El diagrama de lazo asociado a la energización del armario se presenta en la Figura 20.



**Figura 20. Diagrama de lazo energización de armario. Fuente propia.**

Finalmente el diagrama de Lazo de control se presenta en la Figura 21.

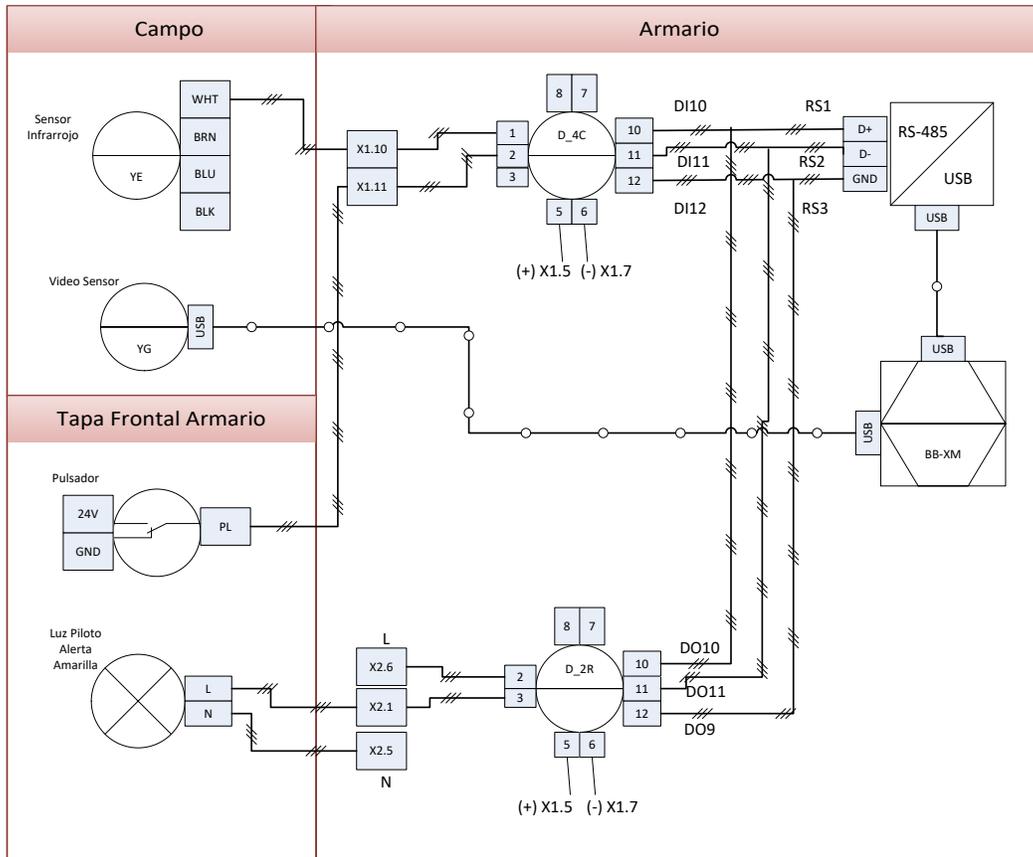


Figura 21. Diagrama de lazo de control. Fuente propia.

### 2.4.7 Diseño Mecánico del Sistema

El diseño mecánico del sistema corresponde con los requerimientos presentados anteriormente. El sistema mecánico lo constituye un *enclosure* que le proporciona protección industrial, cuenta con ranuras de ventilación, internamente se encuentra un monitor para el despliegue de la interface de visualización local, cuenta con acceso a conexión Ethernet. Internamente el panel está conformado por los *DigiRail's* I/O, break de protección, un regulador de 24V, un interruptor de encendido y una luz piloto de estatus de suministro de potencia para todo el sistema.

Una vez definidas las características generales del sistema mecánico y las partes constituyentes, se realiza el diseño CAD del prototipo, las Figura 22 y Figura 23 muestran dos vistas del diseño mecánico propuesto.



### 3 IMPLEMENTACIÓN DEL SISTEMA DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD

Siguiendo los requerimientos de diseño especificados en la sección 2 se realiza la implementación modular hardware y software del sistema óptico embebido industrial de inspección de características de calidad.

#### 3.1 Implementación Módulo de Adquisición de Vídeo

En concordancia con lo mencionado en el diseño de éste módulo, la implementación cuenta con una componente software y una hardware.

##### 3.1.1 Componente Hardware del Módulo de Adquisición de Vídeo

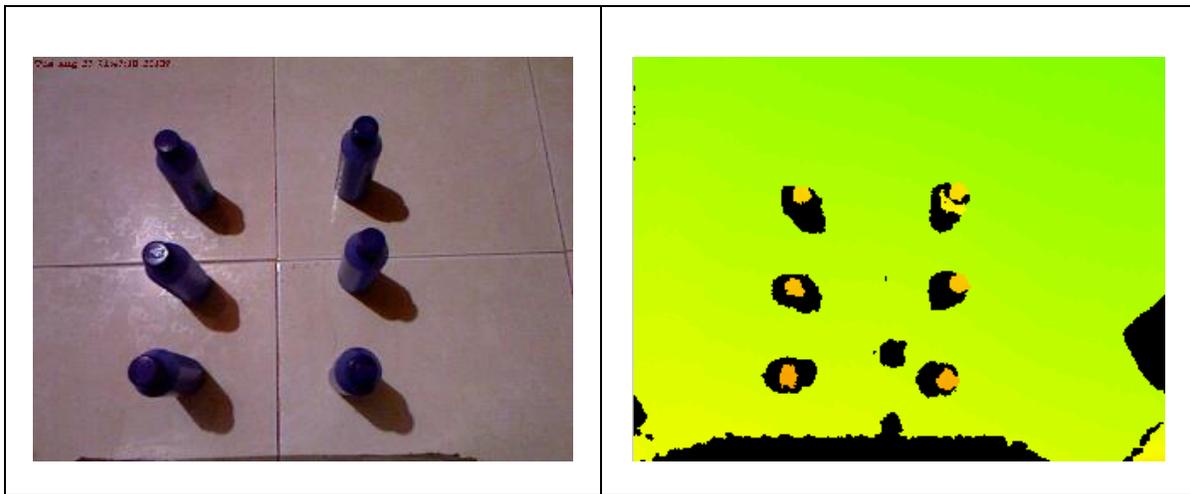
El *housing* como parte de la componente hardware del sistema brinda protección al sensor Kinect ante ambientes industriales típicos. En la Figura 24 se muestra el *housing* fabricado para tal fin. El *housing* del prototipo no cuenta con certificación de protección IP.



Figura 24. *Housing* de protección para el sensor Kinect. Fuente propia.

### 3.1.2 Componente Software del Módulo de Adquisición de Vídeo

La componente software del módulo de adquisición de vídeo parte de la compilación y configuración de las librerías para el acceso a la información del sensor Kinect, en el [Anexo B](#) se muestran los pasos de configuración de las librerías libfreenect. El desarrollo del sistema de adquisición de vídeo cuenta con funciones propias de la librería libfreenect que proveen a la aplicación de una imagen de profundidad de 11 bits y una imagen RGB de la escena, como se muestra en la Figura 25.



**Figura 25. Izquierda: Imagen RGB. Derecha: Imagen de profundidad en pseudocolor en el módulo de Adquisición de Vídeo. Fuente propia.**

### 3.2 Implementación Módulo de Procesamiento

El sistema de procesamiento está soportado sobre un procesador ARM Cortex8 1GHz sobre la tarjeta de desarrollo *BeagleBoard-xM* (BB-xM), la tarjeta permite la configuración de un sistema operativo mediante tarjeta microSD. La configuración se realiza descargando sobre la microSD una imagen del sistema operativo, en particular en este proyecto se usa *Ubuntu 12.10 embedded*. El proceso de configuración se realiza mediante comandos de consola Linux. En el [Anexo B](#) se presentan los comandos de configuración detallados para la tarjeta microSD.

Una vez se cuenta con el sistema operativo sobre la tarjeta BB-xM, se inicia el desarrollo del componente software del módulo de procesamiento. Usando librerías *QtProject*, el IDE *QtCreator* y *OpenCV* se plantea la solución de desarrollo para el módulo de procesamiento que se divide en dos sub-módulos: Módulo de procesamiento para la aplicación *Counter Objects* y Módulo de procesamiento para la aplicación *Pattern Recognition*.

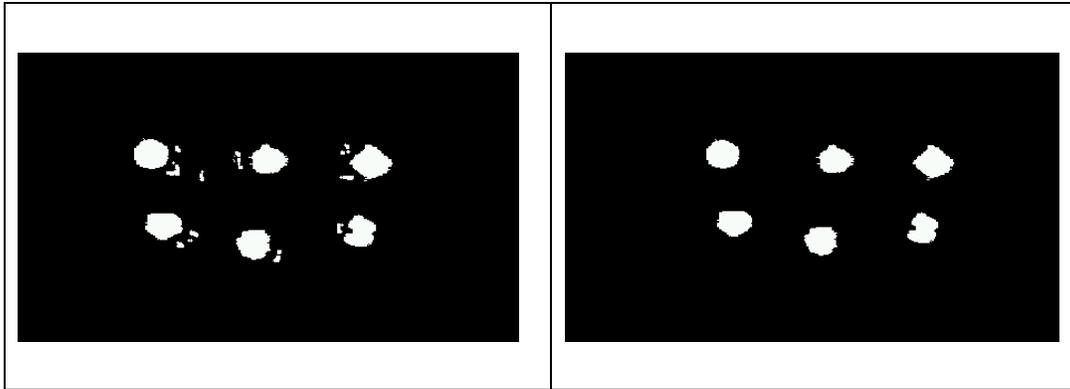
### 3.2.1 Módulo de Procesamiento Aplicación *Counter Objects*

Este sub-módulo se provee de información del módulo software de adquisición de vídeo, usa la imagen de profundidad para segmentar objetos que se encuentren en un rango determinado de profundidad. El nivel umbral de profundidad se configura usando controles *Slide* que pueden ser modificadas por el usuario. De este proceso se obtiene una imagen binaria con los posibles objetos de interés. En la Figura 26 se muestra el resultado de la operación en donde se ha eliminado el fondo de la escena y el cuerpo de las botellas, los objetos de interés en éste caso son las tapas que se encuentran a una profundidad menor (respecto al sensor Kinect) que los demás objetos de la escena.



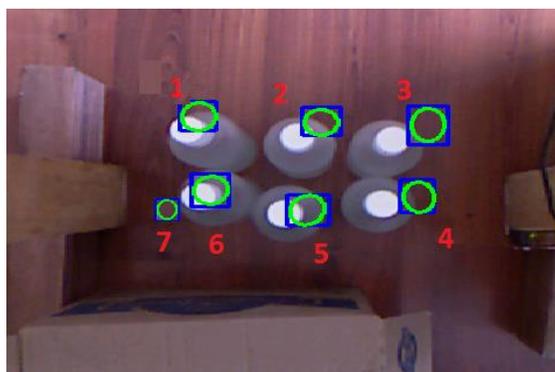
**Figura 26. Izquierda: Imagen RGB, Centro: Imagen de Profundidad, Derecha: Imagen binaria por segmentación de profundidad. Fuente propia.**

La imagen binaria obtenida es tratada usando operaciones morfológicas de Erosión y Dilatación con la finalidad de obtener objetos más definidos y eliminar falsos positivos de la escena. En la Figura 27 se muestra el resultado de las operaciones morfológicas aplicadas. Los niveles de Erosión y Dilatación también pueden ser configurados gráficamente por el usuario.



**Figura 27.** Izquierda: Imagen binaria sin operación morfológica. Derecha: Imagen binaria operada con Erosión y Dilatación. Fuente propia.

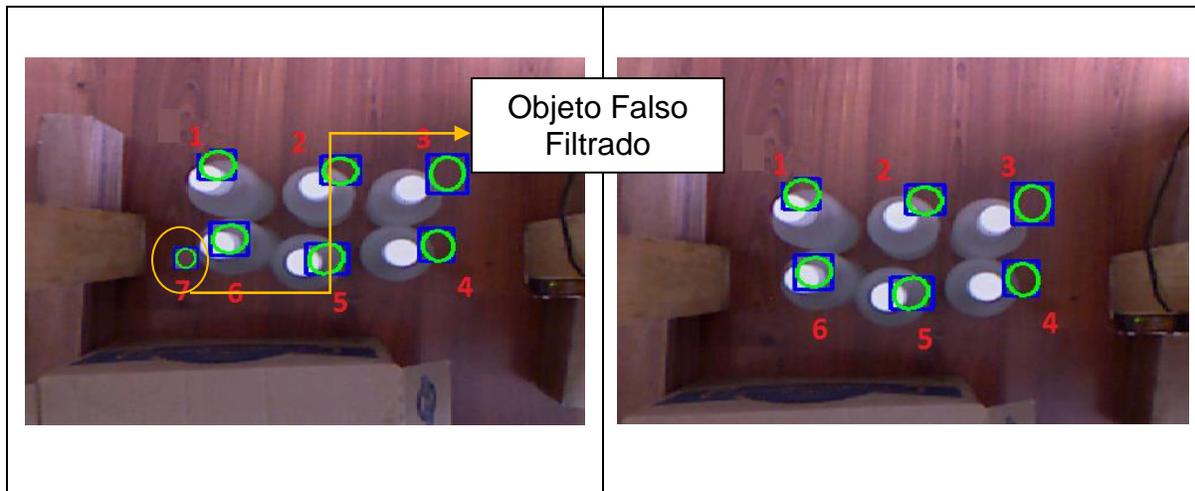
Las operaciones anteriores dan como resultado una imagen binaria segmentando objetos por profundidad, sin embargo es importante individualizar los posibles objetos de interés que contenga dicha imagen binaria. Con el objetivo de separar unos objetos de otros sobre la imagen se usan técnicas de etiquetado de objetos, de esta manera se logran individualizar Blobs (Objetos etiquetados), además se obtienen características importantes de cada objeto de interés como: Área, perímetro, circunferencia, rectángulo o elipse que circunscribe el objeto, centro geométrico de la figura asociada, entre otras. En la Figura 28 se muestra el etiquetado de posibles objetos de interés en la escena.



**Figura 28.** Etiquetado de Objetos Blobs. Fuente propia.

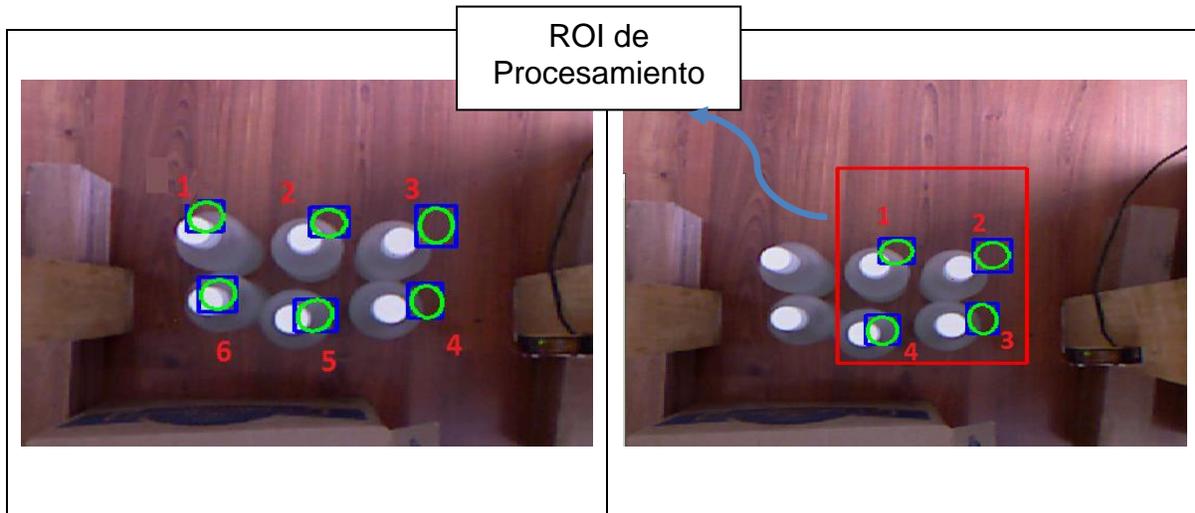
Al individualizar los objetos y obtener propiedades características es posible filtrar usando dichas propiedades, en este caso se filtran los objetos por área. Es decir los objetos de interés deben cumplir con tener un área mayor a un valor de Área

Mínima y un área menor a un valor de Área Máxima. El rango de área también puede elegirse por el usuario gráficamente. En la Figura 29 se presenta los resultados del filtrado por área de los objetos de interés.



**Figura 29. Izquierda: Objetos de interés sin filtrado por área. Derecha: Objetos filtrados por área de Blobs de interés. Fuente propia.**

Además del filtrado por área de blobs, es importante definir regiones de interés ROI's dentro de la imagen. Las regiones de interés definen el espacio de trabajo de los algoritmos de procesamiento, permiten liberar carga computacional a los algoritmos y permiten reducir la probabilidad de fallas y errores. La ROI se define como un rectángulo en la imagen sobre el cual se aplica los algoritmos de procesamiento. En este caso la ROI de procesamiento se escoge por el usuario gráficamente. En la Figura 30 se muestra la funcionalidad de región de interés en la aplicación.



**Figura 30. Izquierda: Objetos sin filtrado por región de interés. Derecha: Objetos filtrados por región de interés ROI en el módulo de procesamiento. Fuente propia.**

Una vez filtrada y procesada la imagen se procede a contar los objetos individualizados que han pasado por los filtros. Claramente pueden realizarse procedimientos adicionales según el tipo de aplicación específica o de los objetos de interés. De esta forma queda el sistema básico de conteo de objetos desarrollado.

Las señales digitales desde la escena industrial, como por ejemplo un sensor de presencia, activan en el módulo software el aviso para realizar un proceso de evaluación, en ese instante se iniciará el proceso de conteo de objetos y finalizará comparando el número de objetos encontrado con el número de objetos patrón. Según el valor positivo o negativo de la evaluación se realizarán acciones sobre la escena industrial, como por ejemplo la parada de una banda transportadora o el aviso usando alarmas visuales o auditivas. Estas señales de salida digitales se ejecutan a través del módulo hardware Modbus del SICC. Además se enviará el número de objetos encontrados a la interface de usuario y la información de la evaluación al módulo software de la comunicación TCP para su visualización remota. En la Figura 31. Proceso de Evaluación para aplicación *Counter Objects*. Fuente propia. Figura 31 se describe el proceso de Evaluación para la aplicación *Counter Objects*.

Una señal digital de entrada representada con un pulsador se lee a través del DigiRail de Entradas Digitales



Una señal de sensor discreto (*trigger*) se lee a través de un DigiRail de Entradas Digitales



La señal de *trigger* se comunica a través del protocolo Modbus con el núcleo de procesamiento BB-xM e informa la presencia de objetos de interés



La tarjeta BB-xM estima el número de objetos de interés y envía: señales discretas hacia la escena Industrial, y resultados de inspección para el monitoreo



El monitoreo de vídeo se presenta en el monitor local del sistema, mientras que los resultados de inspección se presentan local y de forma remota, al igual que la configuración del SICC



La imagen de profundidad de la escena industrial se envía hacia el núcleo de procesamiento BB-xM

El SICC se comunica con la escena industrial a través de salidas digitales enviando señales de alarma o parada de máquina

**Figura 31. Proceso de Evaluación para aplicación *Counter Objects*. Fuente propia.**

### 3.2.2 Módulo de Procesamiento Aplicación *Pattern Recognition*

Este sub-módulo se provee de información del módulo software de adquisición de vídeo, usa la imagen RGB para procesar y buscar el patrón objetivo. Desde la interface de usuario GUI es posible elegir el patrón de referencia (imagen de dimensión wxh), para este propósito se ha definido una región de interés ROI. La imagen patrón se guarda en la memoria de la tarjeta BB-xM y se usa para el proceso de *Matching* en las evaluaciones correspondientes. Es posible también escoger una imagen patrón que ya exista en la memoria del sistema. En la Figura 32 se muestra el proceso de elegir la imagen patrón. Sobre la Interface de usuario, existen dos *checkbox* asociados a la selección del patrón objetivo: *View Pattern ROI* y *Config Pattern ROI*, el primero permite ver el rectángulo que define el patrón sobre la imagen de vídeo, el segundo *checkbox* habilita la selección de una imagen patrón. Una vez estén activos los *checkbox* se debe seleccionar con el *click* izquierdo del mouse el origen del rectángulo que representa la ROI y la esquina opuesta. Cada evento que suceda en la selección del patrón se verá reflejado en el *TextBox* de estatus ubicado debajo de la imagen de vídeo.



Figura 32. Imagen patrón en el proceso de evaluación. Fuente propia

Luego de definir la imagen patrón, se define la región de búsqueda de ese patrón específico. Existen dos *checkbox* asociados a la selección de la región de búsqueda: View ROI, Config Interest ROI. La ROI debe ser de dimensiones mayores a las dimensiones de la imagen patrón. En la Figura 33 se muestra la región de interés de búsqueda en un caso particular.

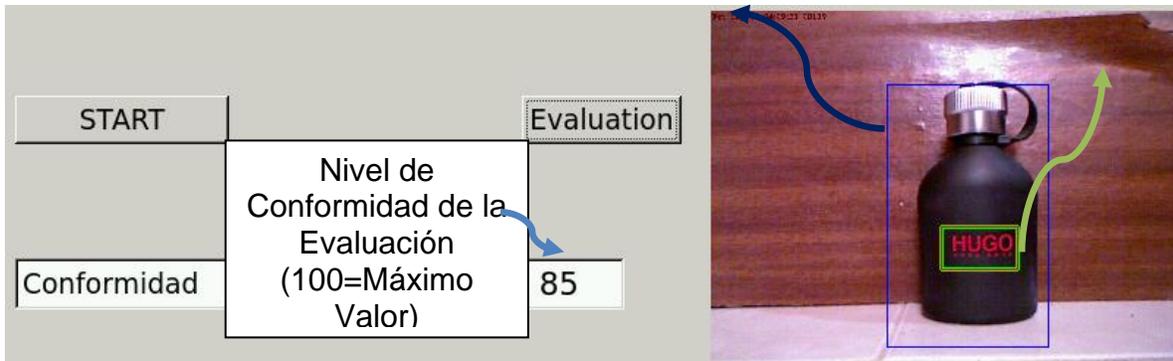


**Figura 33. ROI de búsqueda para la evaluación. Fuente propia.**

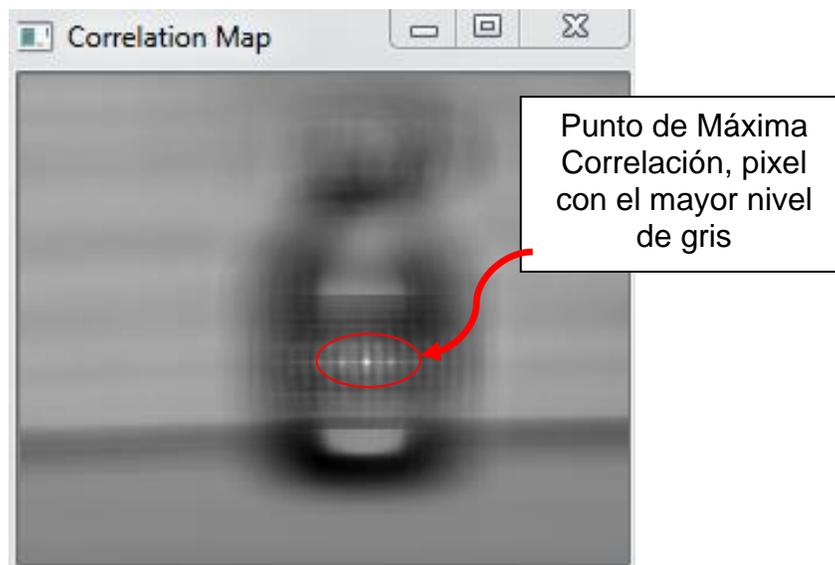
Una vez definidas las regiones de interés y el patrón objetivo, el sistema está listo para realizar evaluaciones, el resultado de evaluación es un valor de 0-100 que mide el nivel de conformidad de la evaluación. Es tarea del usuario elegir el valor umbral para aceptar o rechazar una evaluación. En caso de encontrar el patrón en la región de búsqueda, el sistema mostrará un rectángulo verde ubicando la región de mayor correlación. En la Figura 34 y Figura 35 se muestra el resultado de la evaluación y el mapa de correlación asociado a la operación.

ROI de  
Procesamiento.  
Recuadro Azul

Patrón  
Encontrado.  
Recuadro Verde



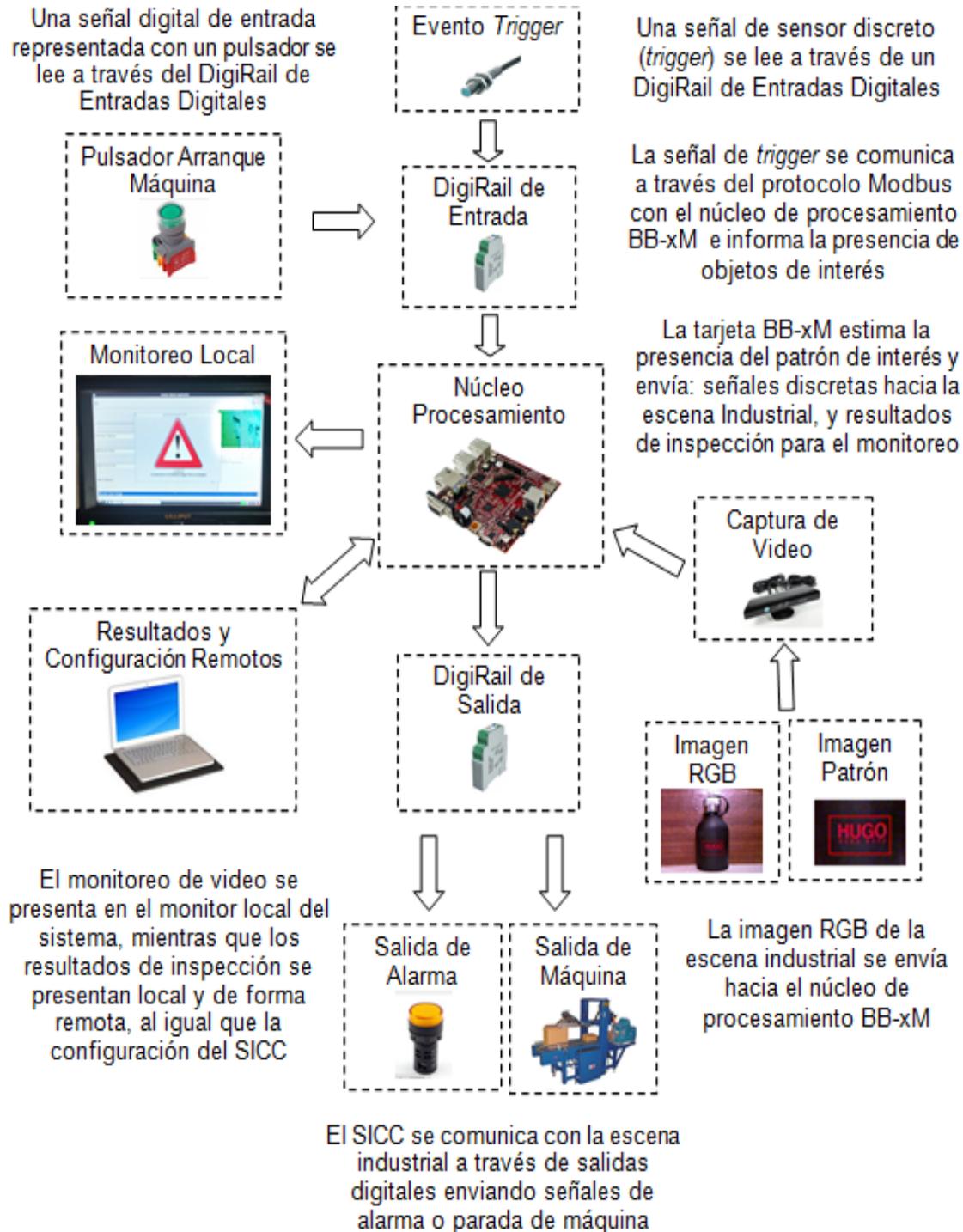
**Figura 34. Resultado de una evaluación en la aplicación *Pattern Recognition*. Fuente propia.**



**Figura 35. Mapa de Correlación asociado al proceso de evaluación de la aplicación *Pattern Recognition*. Fuente propia.**

Al igual que para la aplicación *Counter Objects*, las señales digitales desde la escena industrial, como por ejemplo un sensor de presencia, activan en el módulo software el aviso para realizar un proceso de evaluación, en ese instante se iniciará el proceso de *Matching* y finalizará con un valor de conformidad normalizado de 0 a 100, donde 100 es la mayor conformidad. Según el valor positivo o negativo de la evaluación se realizarán acciones sobre la escena industrial, como por ejemplo la parada de una banda transportadora o el aviso usando alarmas visuales o auditivas. Estas señales de salida digitales se ejecutan a través del módulo hardware Modbus del SICC. Además se enviará el valor de

conformidad a la interface de usuario y la información de la evaluación al módulo software de la comunicación TCP para su visualización remota. En la Figura 36 se muestra el proceso de evaluación en la aplicación *Pattern Recognition*.



**Figura 36. Proceso de evaluación para la aplicación *Pattern Recognition*. Fuente propia.**

### 3.3 Implementación Módulo de Comunicación Modbus

#### 3.3.1 Componente Hardware del Módulo de Comunicación Modbus

La parte hardware del módulo de comunicación Modbus la componen dos *DigiRail's* de *Novus*: *DigiRail 2R* de dos salidas digitales y *DigiRail 4C* de 4 entradas digitales y una *GateWay* RS485 - USB para la comunicación con la tarjeta BB-xM. La configuración de los *DigiRail's* se muestra en el Anexo C usando la interface *DigiConfig* de *Novus*, como parte de la configuración se asignan *Tags*, se define el *Baud Rate* de comunicación y se configura el USB Serial para la interface con la tarjeta. En la Figura 37 se muestra el diagrama de conexión eléctrica de los dispositivos hardware que componen este módulo.

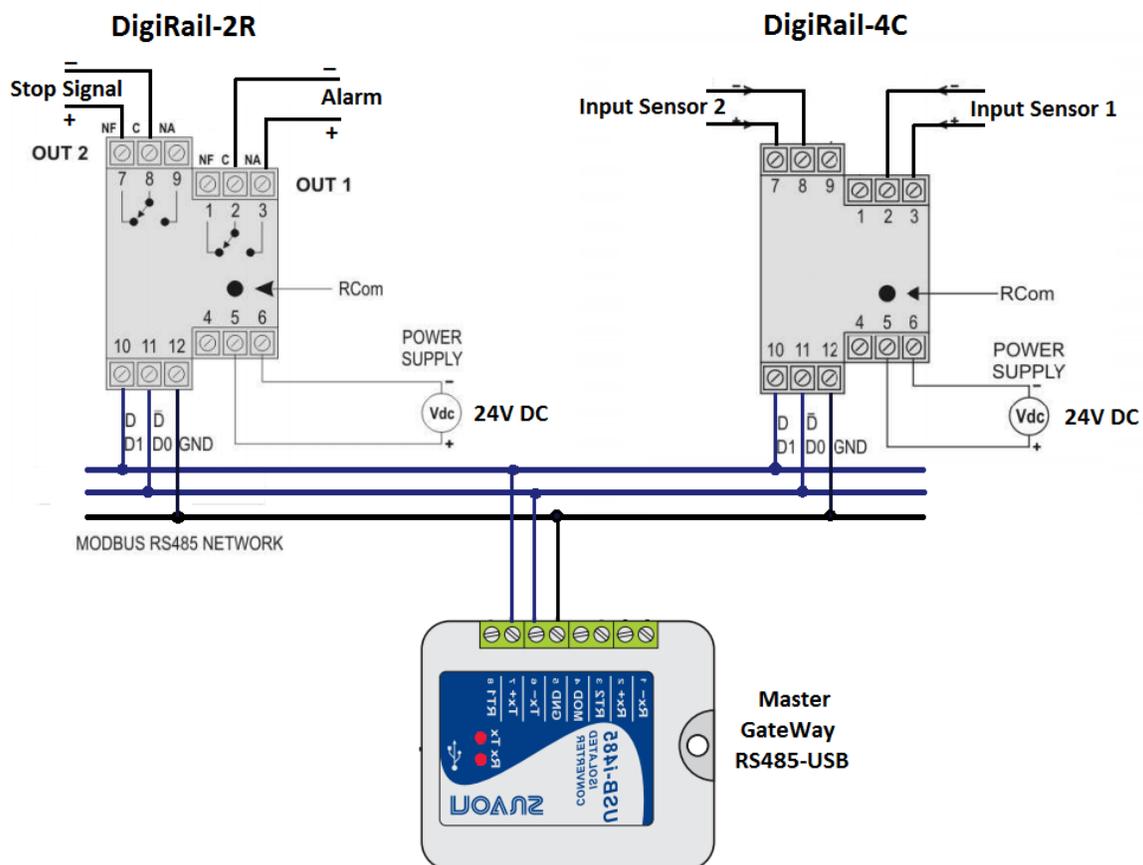


Figura 37. Diagrama de conexión eléctrica en el módulo de comunicación Modbus. Fuente propia.

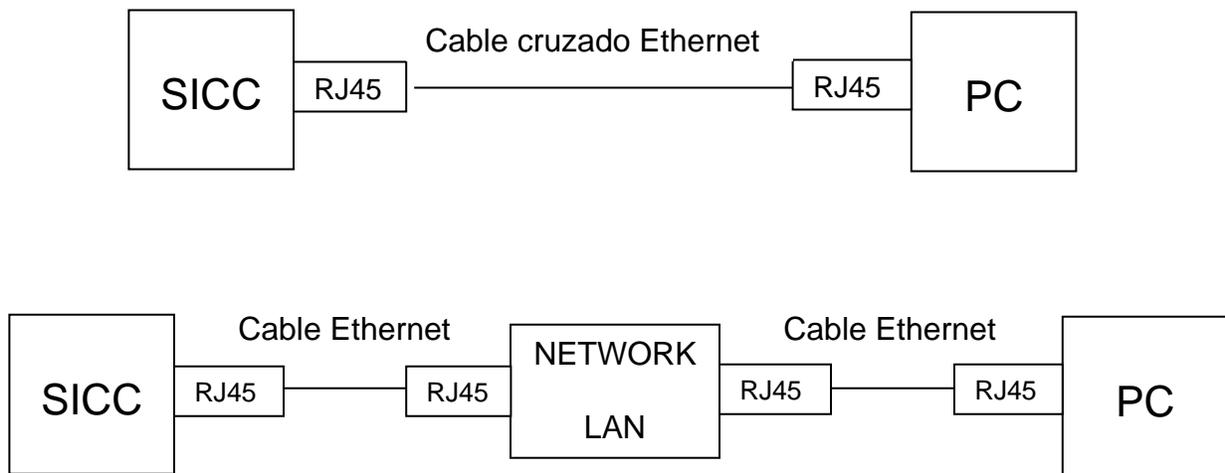
### **3.3.2 Componente Software del Módulo de Comunicación Modbus**

La parte software del módulo de comunicación Modbus la componen un conjunto de funciones que se ejecutan en el ciclo principal de un hilo de procesamiento llamado ModbusThread. Se desarrollaron tres funciones básicas principales: 1. Lectura del dispositivo DigiRail 4C. 2. Escritura del dispositivo 2R. 3. Calculo del parámetro Cyclical Redundancy Checking CRC. En el

**Anexo D** se presenta el código fuente del desarrollo de las funciones mencionadas.

### 3.4 Implementación Módulo de Comunicación TCP

La aplicación principal se diseñó como un servidor TCP capaz de suministrar información de evaluación y estatus del sistema a clientes conectados a la red local que necesiten información particular. La conexión servidor y cliente puede establecerse usando un cable Ethernet cruzado, conexión inalámbrica o a través de un *switch* o concentrador de red. En la Figura 38 se muestra la conexión Cliente-Servidor Típica.



**Figura 38. Superior: Conexión directa Cliente-Servidor entre PC y SICC. Inferior: Conexión a través de red local entre PC y SICC. Fuente propia.**

#### 3.4.1 Módulo de Comunicación TCP Servidor

El desarrollo software de la aplicación se realizó usando librerías nativas del sistema operativo Ubuntu 12.10 *embedded*. Se estableció un protocolo de comunicación entre Cliente y Servidor para lograr compatibilidad en los mensajes escritos y leídos. Los mensajes cuentan básicamente con uno o dos campos de cabecera (*Header*) y uno o varios campos de datos. Una vez se inicie la conexión TCP a través de sockets, el Servidor escribe en el socket una trama de inicio particular que únicamente los clientes TCP de la aplicación reconocen y a la que

responden con un mensaje de confirmación. Únicamente si el mensaje de confirmación es correcto se inicia la interacción TCP a través de los dispositivos.

En la Tabla 4 a Tabla 8 se consignan los *Headers* de los mensajes entre Cliente y Servidor.

<b>Mensaje</b>	<b>Descripción</b>
Ap/CO	Inicio aplicación <i>Counter Objects</i>
Ap/PD	Inicio aplicación <i>Pattern Recognition</i>
Ap/DP	Inicio configuración de parámetros para aplicación <i>Pattern Recognition</i>
Ap/DK	Inicio configuración de parámetros para aplicación <i>Counter Objects</i>
Ap/MB	Inicio Bus Modbus

**Tabla 4. Headers para el manejo remoto de los mandos de la aplicación principal. Fuente propia**

<b>Mensaje</b>	<b>Descripción</b>
CO/SA	Inicio vídeo en aplicación <i>Counter Objects</i>
CO/RG	Selección imagen RGB para visualización
CO/BN	Selección imagen binaria para visualización
CO/PS	Selección imagen pseudocolor para visualización
CO/OU	Salir de aplicación <i>Counter Objects</i>
CO/EV	Evaluación en aplicación <i>Counter Objects</i>
CO/NO/xx	Número de Objetos (xx) en aplicación <i>Counter Objects</i>
CO/LP/xx/yy	Punto de origen (xx,yy) para ROI
CO/WH/xx/yy	Ancho xx y alto yy de la ROI

**Tabla 5. Headers para el manejo remoto de la aplicación *Counter Objects*. Fuente propia.**

<b>Mensaje</b>	<b>Descripción</b>
CO/DK/SA	Inicio dialogo de configuración de parámetros <i>Counter Objects</i>
CO/DK/OK	Salir del dialogo aceptando los parámetros de configuración

CO/DK/LA/xx	Área mínima (xx) de filtrado de blobs
CO/DK/HA/xx	Área máxima (xx) de filtrado de blobs
CO/DK/S1/xx	Valor del umbral menor (xx) para la segmentación de profundidad
CO/DK/S2/xx	Valor del umbral mayor (xx) para la segmentación de profundidad
CO/DK/RV	Activa la visualización de la ROI
CO/DK/CR	Habilita la configuración de la ROI
CO/DK/ER/xx	Número de veces que se aplica erosión a la imagen segmentada
CO/DK/DI/xx	Número de veces que se aplica dilatación a la imagen segmentada

**Tabla 6. Headers para el manejo remoto de la configuración de parámetros de la aplicación *Counter Objects*. Fuente propia.**

Mensaje	Descripción
PD/SA	Inicio vídeo en aplicación <i>Pattern Recognition</i>
PD/OU	Salir de la aplicación <i>Pattern Recognition</i>
PD/LL/xx/yy	Punto de origen (xx,yy) ROI de patrón
PD/LW/xx/yy	Ancho xx y alto yy de ROI de patrón
PD/RL/xx/yy	Punto de origen (xx,yy) ROI de procesamiento
PD/RW/xx/yy	Ancho xx y alto yy de ROI de procesamiento

**Tabla 7. Headers para el manejo remoto de la aplicación *PatternRecognition*. Fuente propia.**

Mensaje	Descripción
DP/SA	Inicio dialogo de configuración de parámetros <i>Pattern Recognition</i>
DP/SP	Grabar la imagen como patrón de referencia
DP/RV	Activar visualización de ROI de patrón
DP/CV	Habilitar configuración de ROI de patrón
DP/IV	Activar visualización de ROI de procesamiento
DP/IC	Habilitar configuración de ROI de procesamiento

**Tabla 8. Headers para el manejo remoto de la configuración de la aplicación *Pattern Recognition*. Fuente propia.**

### 3.4.2 Módulo de Comunicación TCP Cliente

El desarrollo software de la aplicación se realizó usando librerías *QtProject* para el manejo de *sockets*. El sistema Cliente TCP está desarrollado con librerías

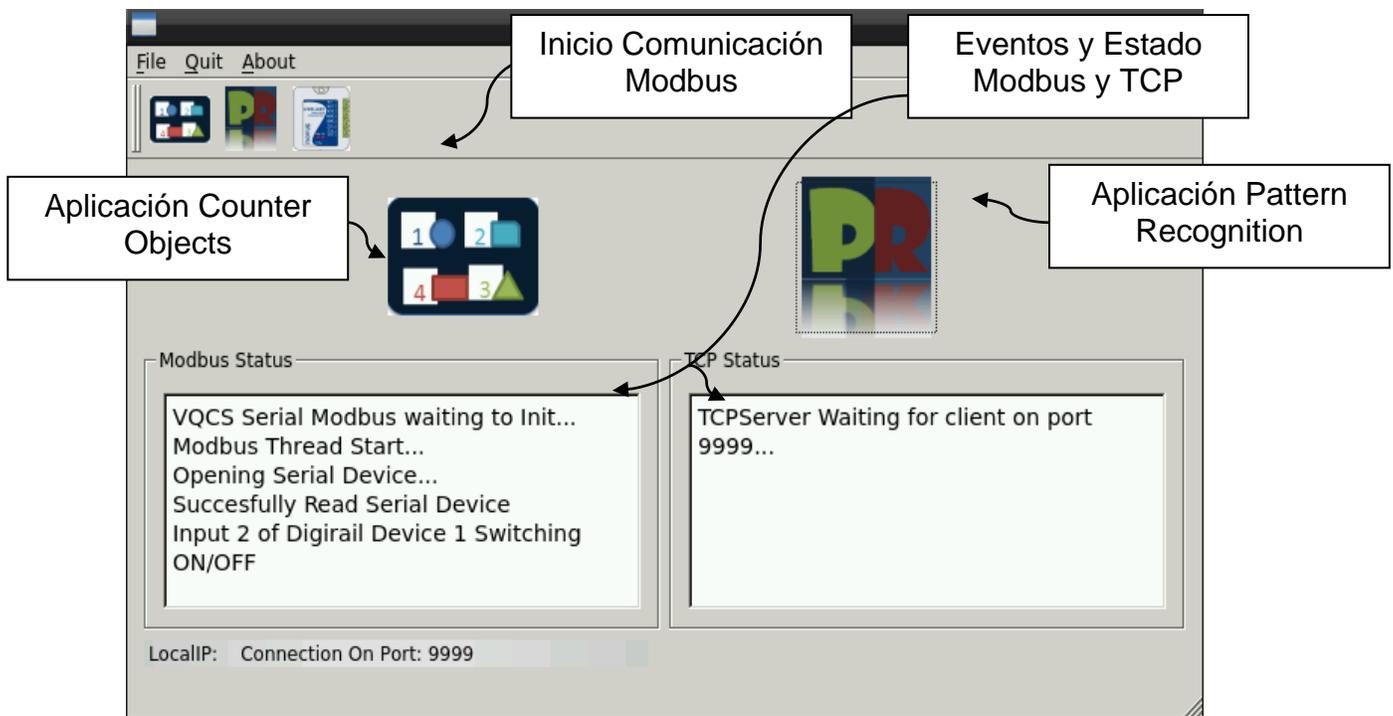
multiplataforma, sin embargo, actualmente se compila en Windows. Al igual que la comunicación desde el Servidor al Cliente, se establecieron tramas de comunicación particulares para la comunicación del Cliente al Servidor, que cuentan con campos de cabecera (*Header*) y campo de Datos. En general la comunicación del Cliente al Servidor consta de órdenes hacia el sistema o solicitud de información. La interacción entre los módulos software se presenta en el Anexo E.

### **3.5 Implementación Módulo de Visualización Local y Remoto**

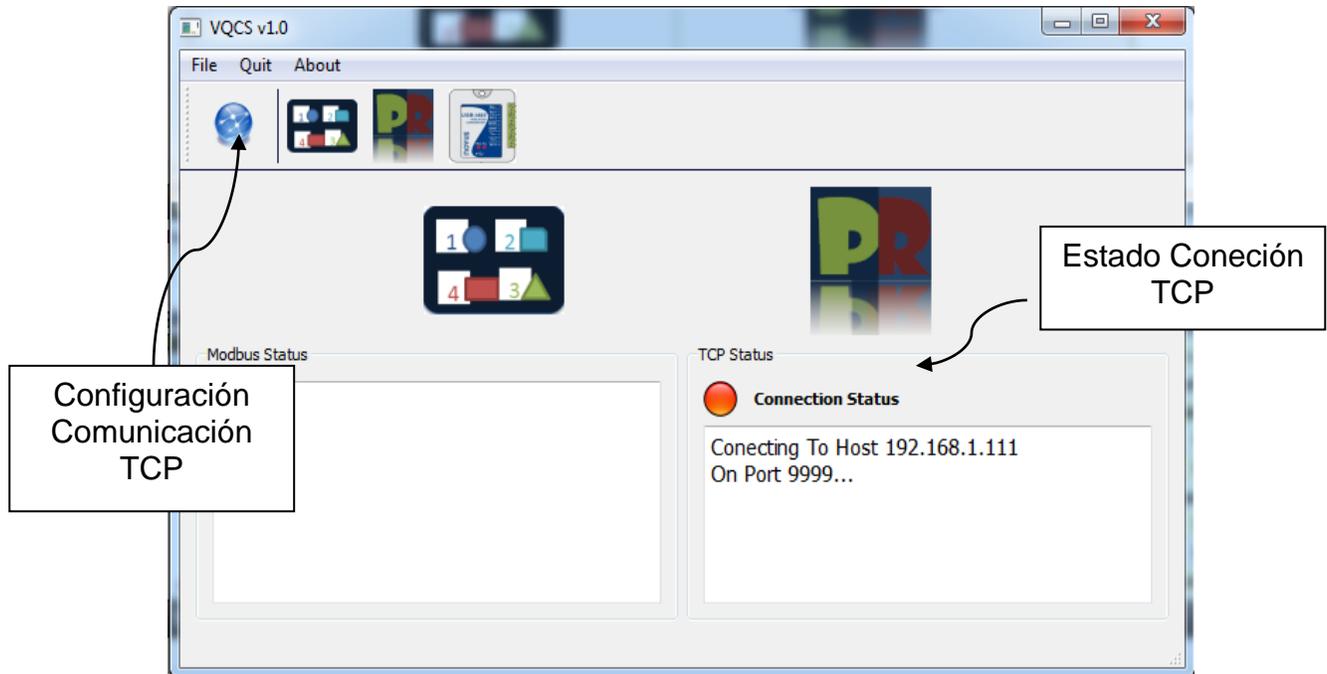
En concordancia con lo mencionado en la sección de diseño **2.4.5** se establecieron dos interfaces de usuario: 1. Interface Principal y 2. Interface de aplicación. La interface de aplicación contiene a su vez una Interface de Configuración. El aspecto de las interfaces Local y Remota se diseñaron iguales para que la interface Remota sea una réplica de la Local y la única característica que las diferencie sea su modo de ejecución.

#### **3.5.1 Interface Principal**

La interface principal se desarrolló siguiendo la plantilla mostrada en la Figura 13. Este GUI cuenta con acceso a las dos aplicaciones del sistema y con información de estatus del sistema de comunicaciones TCP y Modbus RTU. En la Figura 39 se muestra la interface local y en la Figura 40 se muestra la interface remota. Tanto la interface remota como local cuentan con dos iconos principales que llevan al usuario a las dos aplicaciones del SICC: *Counter Objects*, *Pattern Recognition*. Además cuentan con un *TextBox* de estatus del módulo Modbus y un *TextBox* de estatus del módulo TCP.

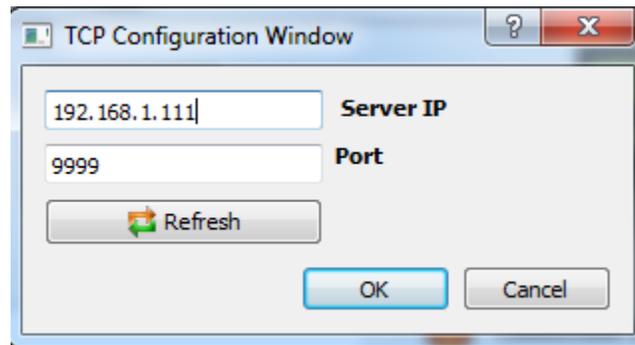


**Figura 39. Interface Principal Local. Fuente propia**



**Figura 40. Interface Remota. Fuente propia**

La interface para la aplicación remota cuenta con un dialogo adicional para la configuración de los parámetros básicos de comunicación TCP. En éste dialogo se establecen la dirección IP del servidor y el puerto de comunicación. En la Figura 41 se muestra el dialogo de configuración de la comunicación TCP.



**Figura 41. Interface de Configuración de la Comunicación TCP desde la aplicación Remota. Fuente propia**

Para el manejo de la interface de usuario de forma local se debe usar un periférico tipo mouse. En éste caso particular se optó por usar un *keyboard-mouse* inalámbrico Rii6718 mostrado en la Figura 42, para facilitar tareas en el manejo de los controles de las interfaces de usuario (aplicación y configuración).



**Figura 42. Mini Wireless Keyboard y Mouse Touch Pad, accesorio para manejo de interface de usuario de la aplicación. Fuente propia.**

### 3.5.2 Interface de Aplicación

La interface de aplicación también se desarrolló siguiendo la plantilla propuesta en la sección de diseño. En ésta interface se encuentra el despliegue de vídeo de la

cámara, un campo de eventos de la aplicación, un campo de resultados y un acceso a la configuración de la aplicación.

### 3.5.2.1 Interface de Aplicación *Counter Objects*

La interface de aplicación *Counter Objects* en particular presenta el número de objetos encontrados en la evaluación, el área en pixeles del objeto más pequeño encontrado y el área en pixeles del objeto de mayor tamaño. En la Figura 43 se muestra el GUI de ésta aplicación. Se muestra la imagen de vídeo con las etiquetas de los objetos de interés encontrados, debajo de la imagen se encuentra el *TextBox* de estatus en donde se presentan todos los eventos que ocurren en el transcurso del proceso de evaluación. Existe un control en la parte inferior donde es posible elegir el tipo de imagen: RGB, Binaria o profundidad en pseudocolor.

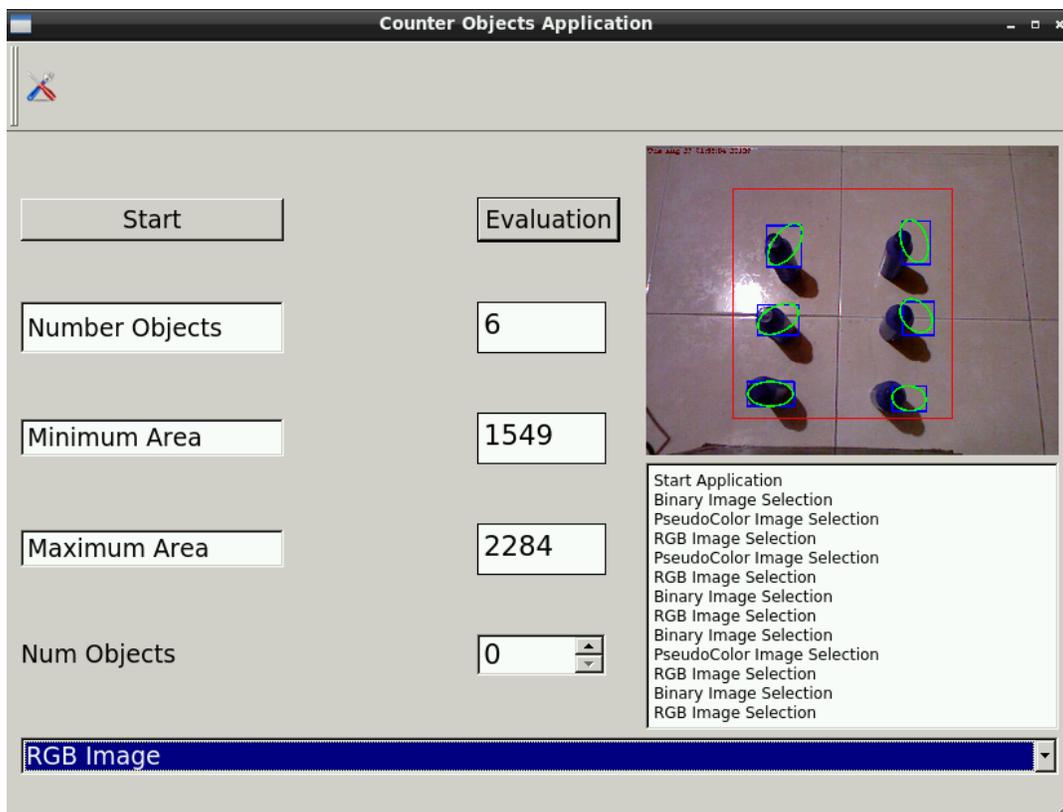
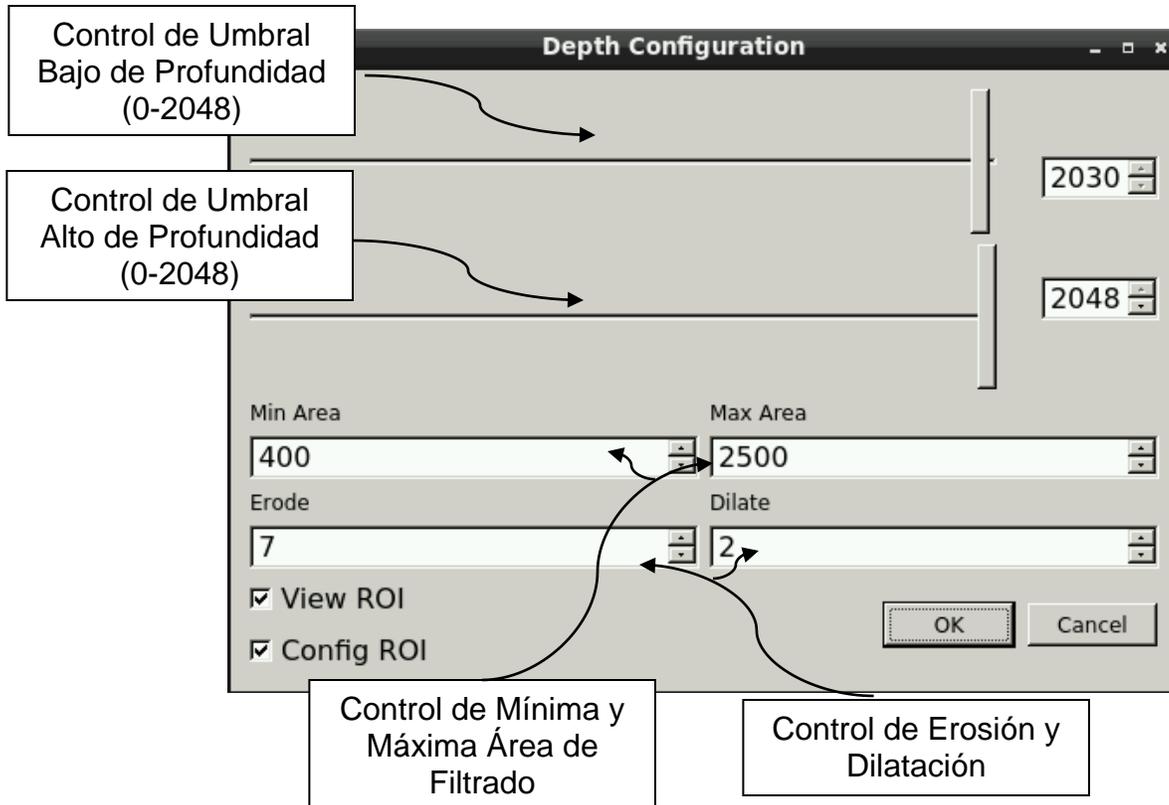


Figura 43. GUI de la aplicación *Counter Objects*. Fuente propia.

La interface de configuración de la aplicación permite modificar: el número de objetos de referencia, el área mínima y máxima de filtrado de Blobs, el número de veces de erosión y dilatación de la imagen binaria, el rango de distancia para la segmentación en profundidad y habilita o no la opción de escoger la región de interés. En la Figura 44 se muestra la interface de configuración de la aplicación *Counter Objects*.

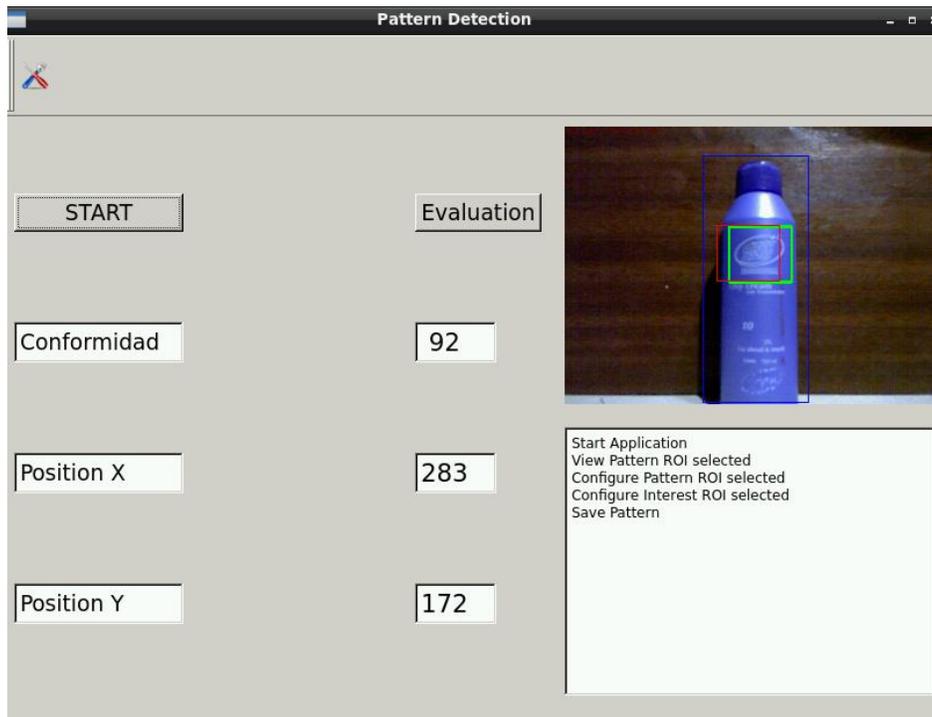


**Figura 44. GUI de configuración de la aplicación *Counter Objects*. Fuente propia.**

### 3.5.2.2 Interface de Aplicación *Pattern Recognition*

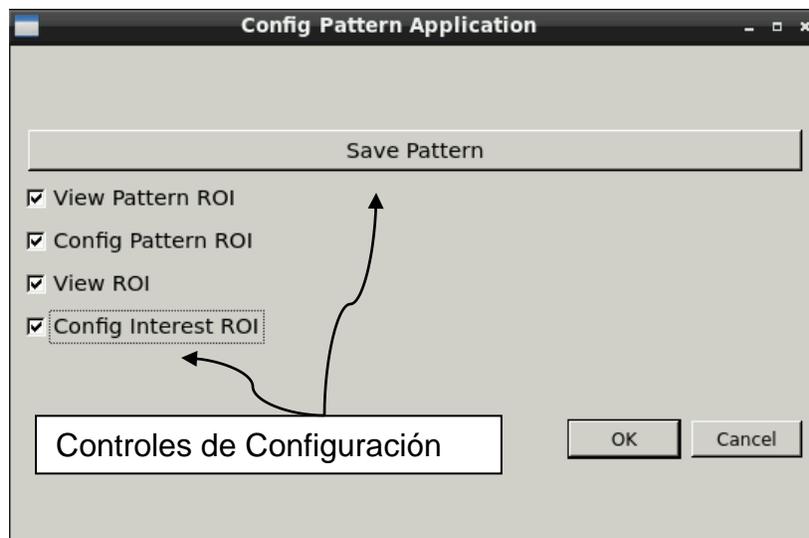
La interface de aplicación *Pattern Recognition* en particular presenta el resultado de conformidad de la evaluación en una escala normalizada de 0 a 100, la posición en pixeles en la coordenada x e y de la imagen. En la Figura 45 se muestra el GUI de ésta aplicación. Se muestra la imagen de vídeo con las etiquetas del patrón de interés encontrado. Debajo de la imagen se encuentra el

TextBox de estatus en donde se presentan todos los eventos que ocurren en el transcurso del proceso de evaluación.



**Figura 45. GUI de la aplicación *Pattern Recognition*. Fuente propia.**

La interface de configuración permite elegir la región de interés de evaluación, además permite elegir y guardar la imagen patrón. En la Figura 46 se muestra el GUI de la interface de configuración de la aplicación *Pattern Recognition*.



**Figura 46. GUI de configuración de la aplicación *Pattern Recognition*. Fuente propia.**

### 3.6 Implementación del Sistema Mecánico

El *enclosure* mecánico para resguardar los componentes del sistema está basado en una *Industrial Box* con protección IP67 (protección contra polvo e inmersión en agua) de dimensiones 14x10x6 pulgadas. Se adecuaron orificios y perforaciones para adaptar una pantalla HDMI de 7 pulgadas, una luz piloto que revela el estatus del circuito de potencia, un interruptor de encendido y los periféricos de la tarjeta, un pasa cables industrial para llevar los cables al exterior de la caja sin poner en riesgo la seguridad del sistema. En la Figura 47 se muestra el prototipo del *enclosure* terminado. El *enclosure* del prototipo después del proceso de maquinado no cuenta con certificación de protección IP.



Figura 47. Prototipo del sistema SICC. Fuente propia.

## **4 PRUEBAS DE VALIDACIÓN DEL SISTEMA DE INSPECCIÓN DE CARACTERÍSTICAS DE CALIDAD**

El sistema de inspección de características de calidad se somete a pruebas de validación en un ambiente industrial con el fin de evaluar su desempeño. A continuación se describe el escenario y condiciones de la prueba de validación.

### **4.1 Problema Objetivo a Solucionar con la Prueba de Validación**

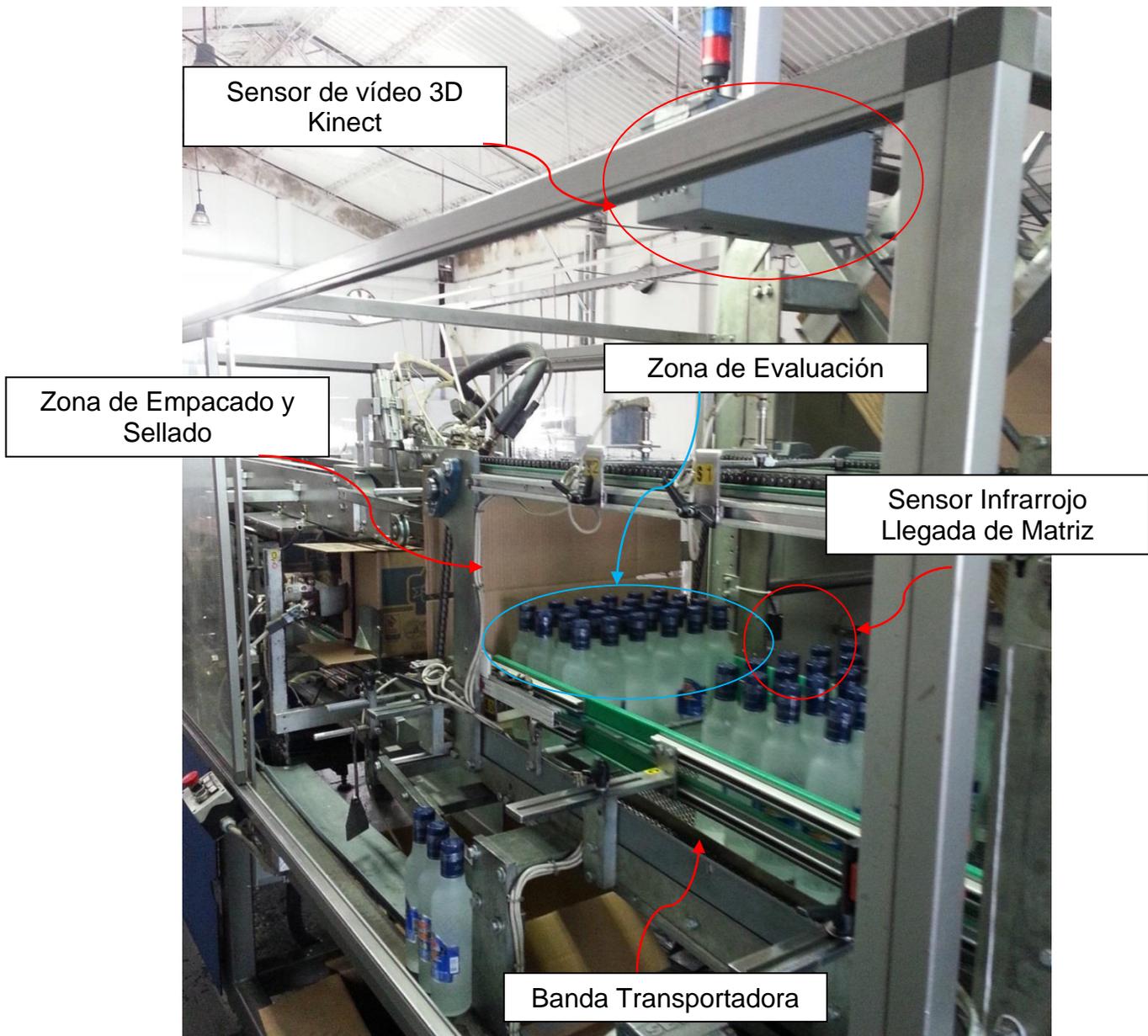
En la Industria Licorera del Cauca se establecieron siete (7) principales situaciones causantes de Quejas y Reclamos en el producto en sus diferentes presentaciones. Una de estas corresponde a botellas faltantes en las diferentes presentaciones del producto. Este problema corresponde al 2.2% de las quejas y reclamos totales de la empresa en el año 2012, ésta cifra equivale a 40 quejas por los motivos mencionados. Es conveniente retomar, de acuerdo a la introducción del capítulo 2, que además existe pérdida de unidades cuando la máquina encartonadora intenta empaquetar más botellas de las que debería contener una caja y se quiebran los envases de vidrio, provocando además de las pérdidas de líquido y envases, demoras en los tiempos de producción y trabajo extra no productivo de los operarios, sin mencionar los peligros de seguridad industrial generados cuando los operarios intentan corregir la matriz con la encartonadora en operación.

El objetivo particular del sistema SICC en la prueba de validación es detectar los eventos en los cuales se encuentra producto faltante, producto en exceso o mal posicionado antes de que se empaquen y distribuyan las cajas de las diferentes presentaciones.

### **4.2 Descripción de la Prueba de Validación.**

El escenario particular de validación es la banda transportadora de la unidad formadora de cajas en la línea de envasado de aguardiente en las presentaciones Media, Botella y Garrafa. El sistema se instalará con el objetivo de contar el

número de botellas antes de que la máquina encartonadora empaque y selle las cajas con las botellas de aguardiente. La prueba tomará 30 días durante los cuales se guardará la información de matrices con producto faltante, producto extra o mal posicionado, se registrará estadísticamente la cantidad de sucesos ocurridos de cada defecto y se tomarán fotos cada vez que ocurra una inspección negativa con el objetivo de tener evidencia visual de la inspección. En la Figura 48 se presenta el escenario de la prueba de validación.



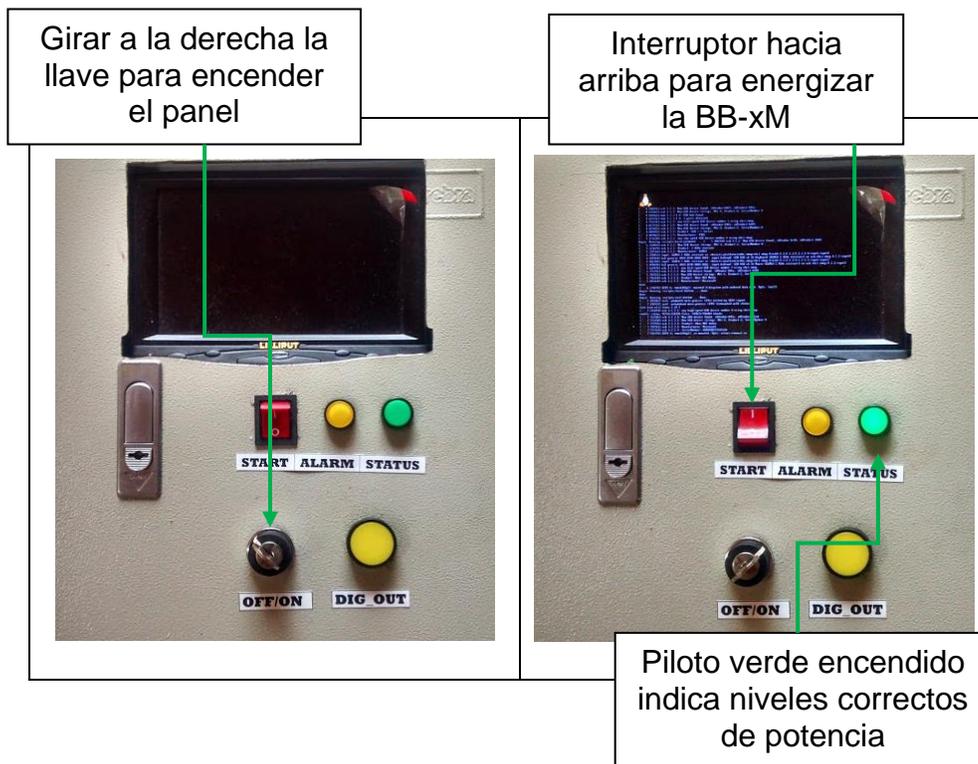
**Figura 48. Escenario de la prueba de validación. Fuente propia.**

El sensor de vídeo 3D Kinect se ubica aproximadamente a 1.5m sobre la banda transportadora, ésta lleva la matriz de botellas hasta la zona de empaqueo y sellado junto a la encartonadora donde se detienen (aquí se empaquen y se sellan en la caja respectiva), un sensor infrarrojo de presencia informa al SICC que existe una matriz lista para ser inspeccionada en la zona de evaluación. El sistema de inspección estima el número de objetos en la matriz, si el número de objetos encontrados no corresponde con el número patrón el SICC envía una señal digital de relé para detener la banda transportadora, además informa al operario mediante una señal de alarma visual en el panel (luz piloto amarilla), y en la interface de monitoreo (un mensaje de alarma en el GUI). Una vez el operario soluciona el defecto en la matriz, pulsa el *push-button* de re-arranque de la banda transportadora *Dig\_Out*, ubicado en la tapa frontal del SICC, y el proceso de sellado y embalado continúa, finalmente la caja con el número correcto de botellas en su interior sale de la línea.

### **4.3 Configuración y Puesta en Marcha del SICC**

El SICC deberá iniciarse junto con los demás módulos de la línea de envasado garantizando que la formadora de cajas esté deshabilitada, es decir que esté encendida pero no detenida. Desde el gabinete se debe dar inicio al sistema, para esto se deben seguir los pasos a continuación:

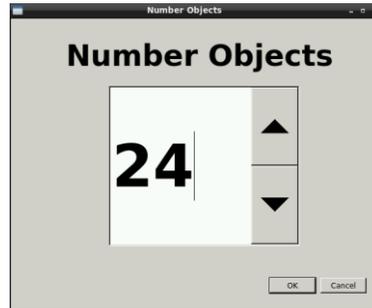
1. Girar el interruptor de energía del sistema “Off/On” al estado On. El led piloto verde deberá encenderse indicando que todo está correctamente energizado.
2. Accionar el interruptor rojo “Start”. Éste encenderá la tarjeta BB-xM, ver Figura 49.



**Figura 49. Secuencia de Encendido del SICC. Imagen superior: Panel apagado. Imagen Inferior: Panel encendido. Fuente propia.**

3. Esperar algunos segundos a que el sistema operativo inicie su funcionamiento y se ejecute la aplicación software. Esto se hace de manera automática. Si por algún motivo se ha puesto en marcha la banda transportadora con anterioridad, ésta se detendrá.
  
4. Al iniciar la aplicación software se desplegará la interface mostrada en la Figura 39, si se está operando de forma local, o la interface de la Figura 40 si se está operando remotamente. Si se realiza de forma local se debe usar un periférico tipo *mouse* para navegar en la interface de usuario (No está incluido en el SICC, sin embargo se usa como un periférico o accesorio extra).
  
5. Escoger la aplicación que desee ejecutar (para este caso particular *Counter Objects*).
  
6. Si se ha escogido la aplicación *Counter Objects* se debe seleccionar el número de objetos de referencia (24 objetos para la presentación de media,

12 para la presentación de botella de litro y 6 para la presentación de garrafa), ver Figura 50.

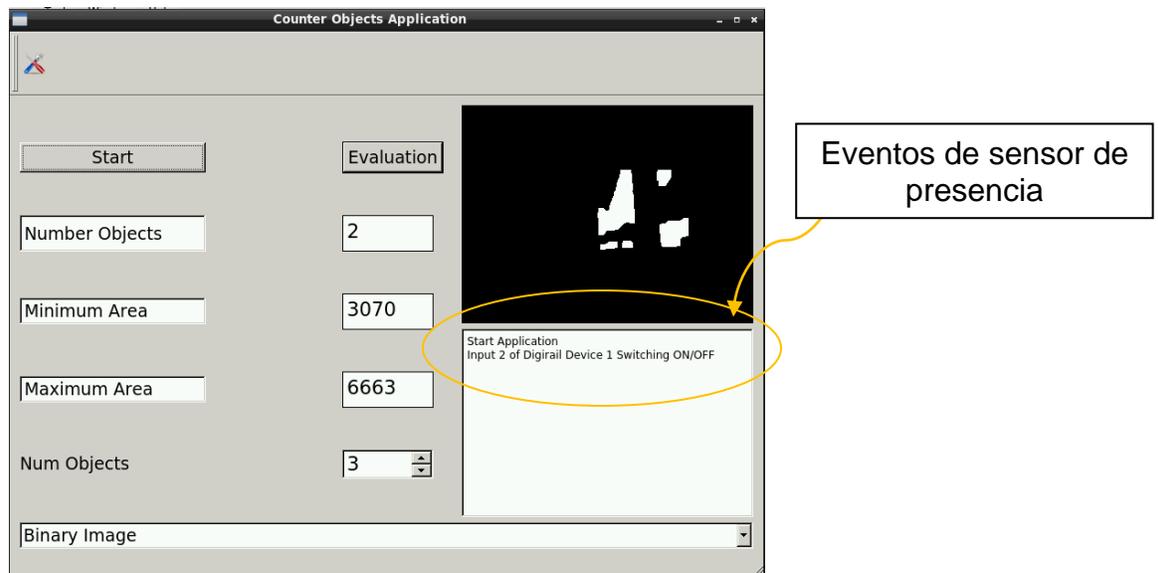


**Figura 50. Interface para escoger el número de objetos de referencia en la aplicación *Counter Objects*. Fuente propia.**

7. Luego de escoger el número de objetos referencia (equivalente al formato de envasado) hacer *click* en *start* en la aplicación.

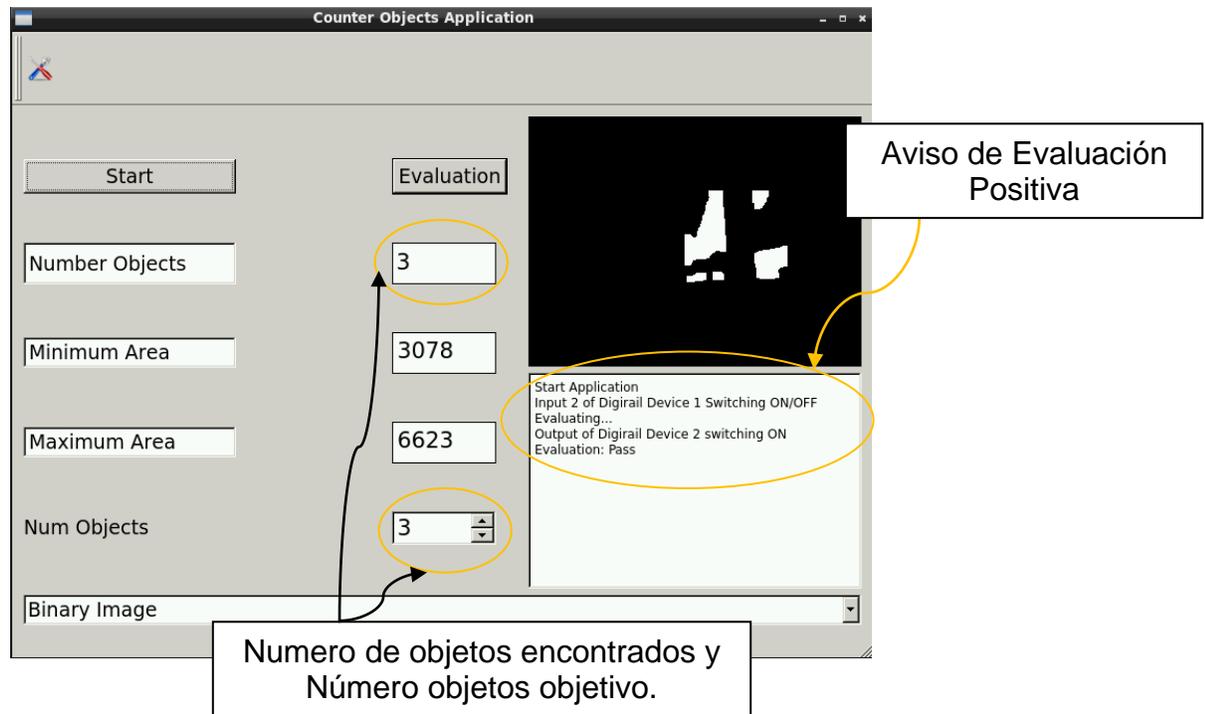
#### 4.4 SICC en Operación

El sistema SICC una vez en operación se encuentra a la espera de una matriz de botellas para evaluar. Una vez se activa el sensor infrarrojo se inicia la evaluación. En el *TextBox* de estatus del sistema se informa de la ocurrencia de eventos de los sensores de presencia como se muestra en la Figura 51.



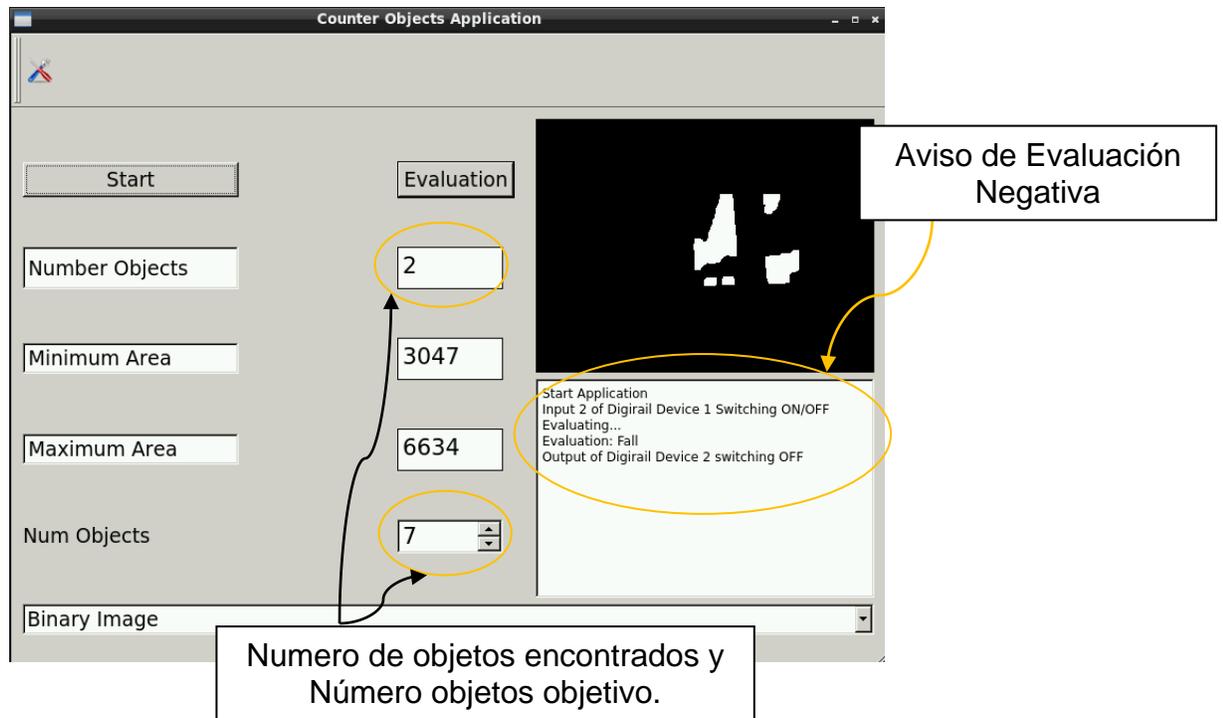
**Figura 51. Ocurrencia de eventos de sensor de presencia infrarrojo. Fuente propia.**

El sistema toma varias muestras de la escena y calcula el número de objetos presentes, si el número de objetos encontrados en la escena es igual al número de objetos de referencia, entonces la evaluación será positiva y se contará como una matriz aceptable incrementando un contador de eventos positivos “*pass*”. En la Figura 52 se ilustra el procedimiento.



**Figura 52. Evaluación positiva “*pass*” en la aplicación *Counter Objects*.  
Fuente propia.**

Si por el contrario el número de objetos encontrado en la escena es mayor o menor al número de objetos de referencia, entonces la evaluación será negativa y se contará como una matriz no aceptable incrementando un contador de eventos negativos “*fall*”. Además la ocurrencia de una evaluación “*fall*” dará lugar a una señal de *stop* emitida hacia el *DigiRail 2R* que detendrá la banda transportadora antes de que se empaque y selle la caja que contendrá las botellas evaluadas. En la Figura 53 se ilustra el procedimiento.



**Figura 53. Evaluación negativa “fall” en la aplicación *Counter Objects*. Fuente propia.**

Una vez el operario ha corregido el problema que puede ser agregar las botellas faltantes o quitar las botellas extras debe poner en marcha nuevamente la máquina haciendo *click* en Aceptar en el dialogo “*Warning Evaluation Fall*”, ver Figura 54 o usando el pulsador *Dig\_Out*, ver Figura 55. Con esto se pone en funcionamiento nuevamente la banda transportadora y se continúa el proceso de inspección con las matrices de botellas siguientes.



**Figura 54. Mensaje de aviso de evaluación negativa. Fuente propia.**

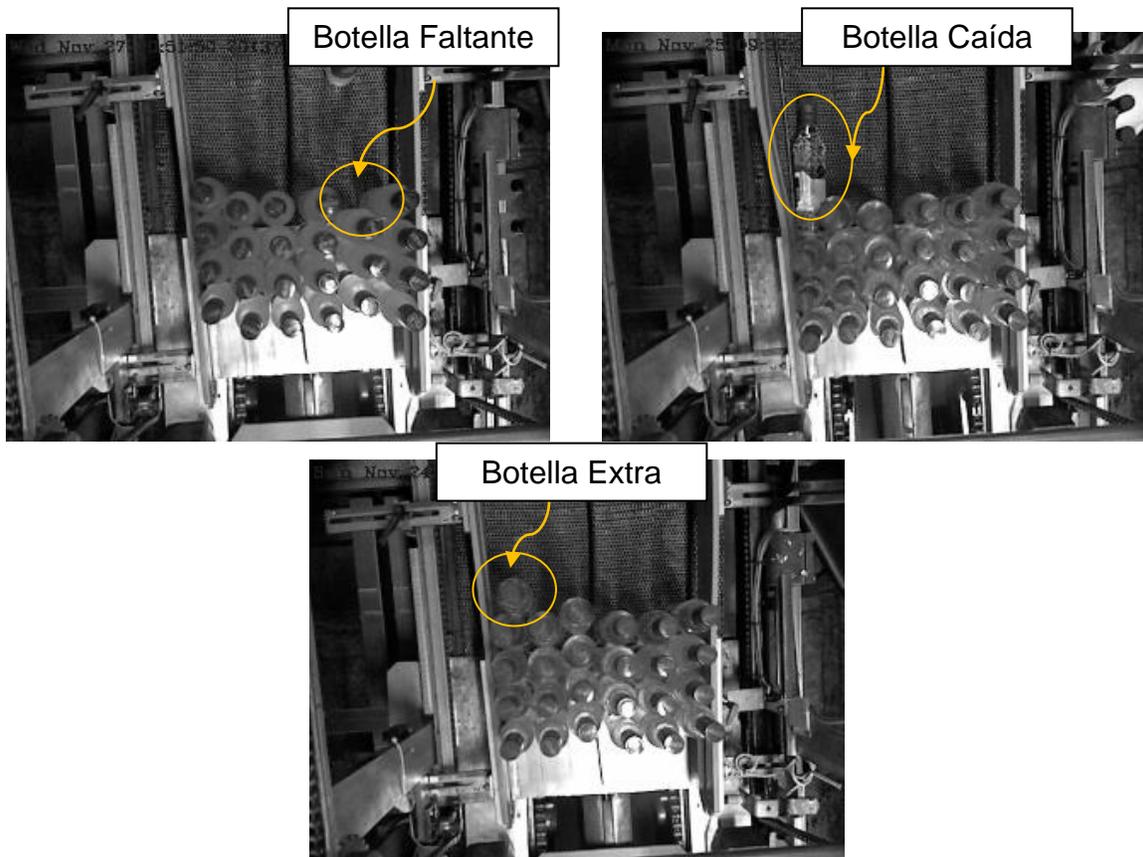


Piloto amarillo indica un proceso de evaluación negativa

**Figura 55. Alerta visual en el panel de evaluación negativa, piloto amarillo. Fuente propia.**

#### **4.5 Resultados de la prueba de validación del SICC**

Luego de 30 días de pruebas de validación bajo las condiciones y procedimientos mencionados anteriormente, se obtuvo un registro de 30201 matrices evaluadas en total, de las cuales para la presentación media fueron 13294 en 19 días de operación, 9611 matrices evaluadas para la presentación botella en 6 días de operación y 7296 matrices evaluadas para la presentación garrafa en 5 días de operación. Del total de matrices evaluadas se obtuvo el número de matrices con evaluación positiva “*pass*” 29559 y el número de matrices con evaluación negativa “*fail*” 679. En la Figura 56, Figura 57 y Figura 58 se presentan las imágenes registradas por el sistema debido a botellas faltantes y extras en el proceso de inspección.



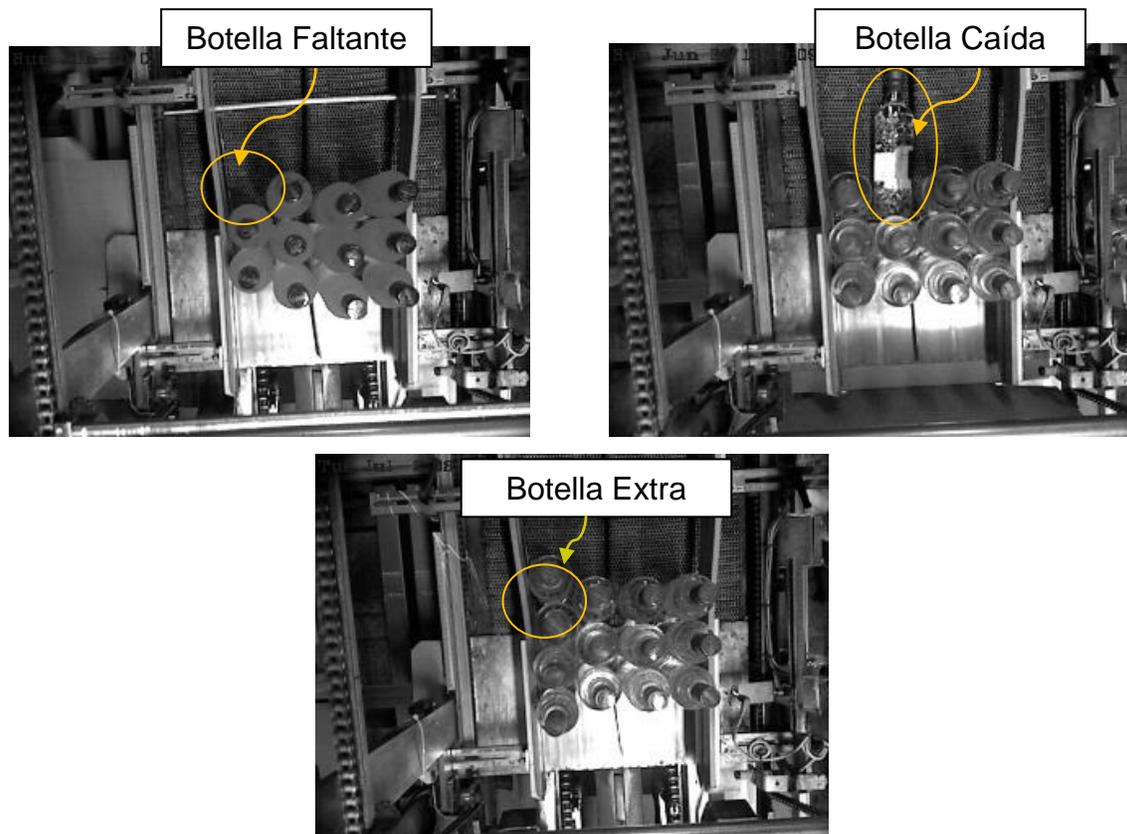
**Figura 56. Proceso de inspección de empaqueo en la presentación media.  
 Superior Izquierda: Botellas faltantes. Superior Derecha: Botella caída.  
 Inferior: Botella extra. Fuente propia.**

El SICC se operó para la presentación “Media” durante 19 días para los que se obtuvo un registro de matrices de botellas aceptables y no aceptables como se muestra en la Tabla 9.

Día	Buenas	Malas	Tiempo de Trabajo (minutos)
1	682	14	774
2	515	9	664
3	646	14	419
4	680	77	791
5	528	30	327
6	479	21	304
7	704	38	776
8	754	25	411
9	326	17	405
10	1033	42	533

11	31	5	289
12	1037	29	433
13	1149	65	505
14	864	34	349
15	734	47	293
16	388	25	336
17	274	2	177
18	984	21	440
19	936	35	397
<b>Total</b>	12744	550	8623

**Tabla 9. Registro de evaluaciones buenas y malas en la presentación Media.**  
Fuente propia.

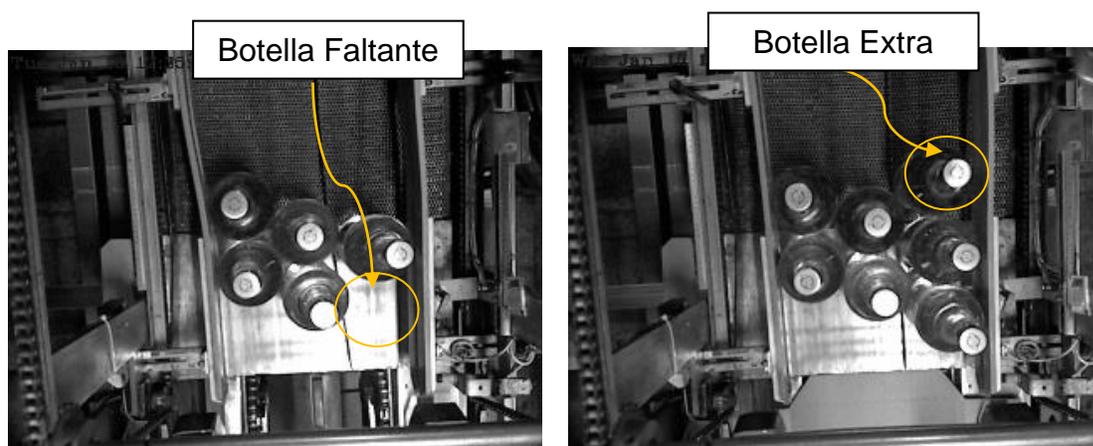


**Figura 57. Proceso de inspección de empaclado en la presentación botella.**  
Superior Izquierda: Botellas faltantes. Superior Derecha: Botella caída.  
Inferior: Botella extra. Fuente propia.

El SICC se operó para la presentación “Botella” durante 6 días para los que se obtuvo un registro de matrices de botellas aceptables y no aceptables como se muestra en la Tabla 10.

Día	Buenas	Malas	Tiempo de Trabajo (minutos)
1	1491	4	589
2	1389	14	665
3	1293	13	549
4	2191	24	892
5	2151	25	905
6	1004	12	728
<b>Total</b>	<b>9519</b>	<b>92</b>	<b>4328</b>

**Tabla 10. Registro de evaluaciones buenas y malas en la presentación Botella. Fuente propia.**



**Figura 58. Proceso de inspección de empaquetado en la presentación garrafa. Izquierda: Botellas faltantes. Derecha: Botella extra. Fuente propia.**

El SICC se operó para la presentación “Garrafa” durante 5 días para los que se obtuvo un registro de matrices de botellas aceptables y no aceptables como se muestra en la Tabla 11.

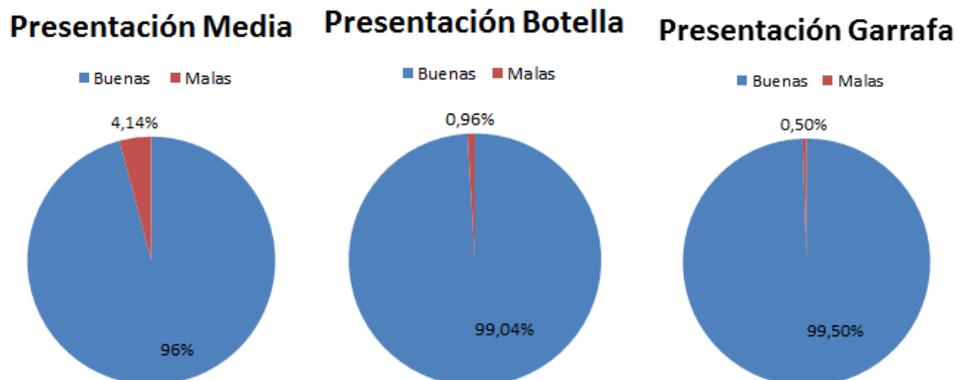
Día	Buenas	Malas	Tiempo de Trabajo (minutos)
1	574	3	302
2	731	10	589
3	1910	11	582
4	1193	3	553
5	2851	10	831
<b>Total</b>	<b>7259</b>	<b>37</b>	<b>2857</b>

**Tabla 11. Registro de evaluaciones buenas y malas en la presentación Garrafa. Fuente propia.**

En la Tabla 12 se consignan los resultados consolidados del registro de evaluaciones que realizó el SICC en el proceso de validación, gráficamente se presenta la comparación entre evaluaciones positivas y negativas de las tres presentaciones en la Figura 59.

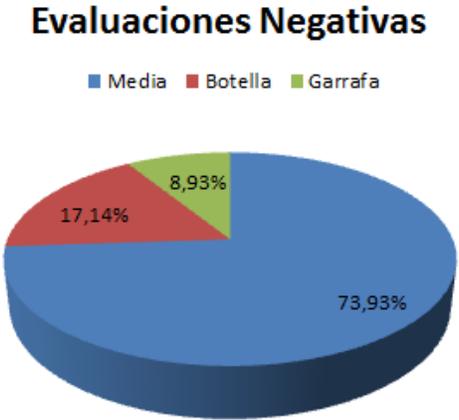
Presentación	#total de Matrices	Evaluaciones Positivas	Evaluaciones Negativas	%Evaluaciones Negativas
Media	13294	12744	550	4.14%
Botella	9611	9519	92	0.96%
Garrafa	7296	7259	37	0.51%

**Tabla 12. Comparación de evaluaciones positivas y negativas en las tres presentaciones. Fuente propia.**



**Figura 59. Evaluaciones Negativas y Positivas para cada presentación. Fuente propia.**

Del total de evaluaciones negativas registradas en los días de prueba (679 matrices de botellas) se obtuvo el mayor porcentaje para la presentación de media, situación que concuerda con la experiencia de los operarios de turno, la presentación del producto que produjo la menor cantidad de problemas en el empackado fue la de garrafa, en la Figura 60 se muestra la diferencia porcentual de evaluaciones negativas obtenidas en las pruebas de validación.



**Figura 60. Comparativo de evaluaciones negativas para las 3 presentaciones del producto. Fuente propia.**

La presentación de Media resultó más propensa a fallas en cuanto a la cantidad correcta de botellas que llegan a la zona de empackado. En los días de validación más del 70% de matrices defectuosas fueron causadas en esa presentación. En ocasiones la cantidad de elementos en la matriz era correcta pero debido al movimiento de la línea una o varias botellas caían en la banda lo que ocasionaba que las barras de arrastre golpearan las botellas hasta quebrarlas, el sistema permitió detener la máquina antes de que esto sucediera.

Con lo anterior se refleja evidentemente que el sistema SICC afectó positivamente la calidad del producto entregado a los clientes, se registraron 679 matrices con defectos que eventualmente habrían podido llegar a los clientes causando malestar, esto equivale al 2.25% del total de matrices empackadas<sup>2</sup>. Además es

---

<sup>2</sup> La estadística asociada a falsos positivos en las evaluaciones no se realizaron por pérdida de información de la base de datos.

apropiado resaltar que el operario antes encargado de vigilar la cantidad de unidades que se debían empacar usó el tiempo para estar pendiente de otras zonas cercanas a la zona de empacado. El sistema SICC facilitó el proceso de vigilancia del sistema de empacado y permitió usar tiempo valioso del operario en tareas más críticas. Por otra parte se evidenció la reducción de pérdida de unidades por ruptura de envases cuando exceso de botellas pasaban a la zona de empacado.

Por todo lo anterior mencionado se concluye que el proceso de validación del sistema SICC desarrollado en este proyecto fue exitoso y cumplió con su objetivo principal. Como retroalimentación del proceso de prueba y validación se consolidaron un conjunto de consideraciones presentadas en la siguiente sección.

#### **4.6 Consideraciones para el manejo del SICC**

Para el uso del SICC no se requiere personal con conocimientos informáticos, sin embargo, se recomienda al usuario prestar atención a los mensajes de estatus del sistema y a la documentación de operación registrada en este u otro documento.

Verificar que el sensor infrarrojo reflectivo emita una interrupción (indicada por un led rojo sobre el sensor) en el momento que las botellas pasan justo enfrente del emisor. Es importante garantizar en la instalación que no ocurran interrupciones causadas por eventos diferentes a la llegada de botellas al campo de visión de la cámara, como por ejemplo movimientos indebidos de los operarios u objetos extraños al sistema.

Verificar que se encuentre encendido el sistema de ventilación, si no lo está es posible que el SICC tenga daños por sobre calentamiento.

Una vez se inicia el Bus Modbus del sistema, haciendo *click* sobre el icono correspondiente en la barra de herramientas en las interfaces de aplicación, verificar que los leds Rx y Tx de los elementos I/O del bus (*DigiRail 2R* y *4C*) estén

en operación intermitente, esto indica el proceso de *polling* del maestro hacia los esclavos, y evidencia la actividad del bus.

## 5. CONCLUSIONES

Se diseñó y construyó un Sistema de Inspección de Características de Calidad (SICC) como un sistema óptico-industrial embebido que permite realizar inspección de características de calidad en entornos industriales. El sistema consta de dos componentes complementarios: Hardware y software, soportado sobre un sistema embebido *BeagleBoard-xM (BB-xM)*, con capacidad de comunicación Modbus RTU, comunicación en red local TCP, interface de usuario local y remota, entradas y salidas digitales.

Se desarrolló el SICC a partir de un sistema base hardware centrado en una *BeagleBoard-xM* que cuenta con módulos hardware de: vídeo, comunicación industrial Modbus RTU, comunicación de red TCP Cliente/Servidor, Interface de usuario local, interface de entradas y salidas digitales. Se desarrolló en SICC un sistema complementario software soportado en un SO Linux *Embedded* que cuenta con módulos software de adquisición de vídeo, procesamiento de vídeo, comunicación industrial Modbus RTU, comunicación de red TCP Cliente/Servidor, interface de usuario local y remota, lectura y escritura de señales digitales. Todos estos conducentes a soportar un sistema óptico-industrial de inspección de características de calidad embebido.

SICC se diseñó con elementos *hardware* variados, como una red Modbus Novus incremental que consta inicialmente de dos módulos DigiRail 2R (dos salidas tipo relé) y 4C (cuatro entradas lógicas a 24v), una cámara de adquisición de imágenes RGB (640x480 pixeles) y de profundidad (11 bits) de Microsoft (Kinect), una pantalla de vídeo HDMI (7 pulgadas), un pulsador (asociado a una salida digital de relé) y dos luces pilotos (status potencia, alarma) y finalmente un puerto de comunicaciones Ethernet TCP. Con el fin de facilitar las tareas de configuración, puesta a punto del sistema SICC y manejo de la interface de usuario se cuenta con un teclado-mouse inalámbrico portátil. Todos los componentes descritos

anteriormente son coordinados por un procesador Cortex A8 arquitectura ARM de una tarjeta BeagleBoard-xM.

Para SICC se desarrollaron 7 módulos software: 1. Módulo de adquisición de vídeo, lo conforma un conjunto de funciones que permite obtener la imagen de profundidad y RGB del sensor de vídeo Kinect. 2. Módulo de comunicación Modbus: lo conforma un conjunto de funciones que permiten escribir/leer los módulos DigiRail Novus. 3. Módulo de comunicación TCP (Servidor en la tarjeta BB-xM): lo conforma un conjunto de funciones que permite crear un servidor usando sockets TCP, el cual envía datos de procesamiento a los clientes remotos. 4. Módulo de comunicación TCP Cliente: lo conforma un conjunto de funciones que se ejecutan en un cliente remoto permitiendo configurar el SICC, además permite tener acceso a datos de procesamiento on-line. 5. Módulo de procesamiento de imágenes: Lo conforma un conjunto de funciones que permite contar objetos segmentados por profundidad y detectar patrones determinados por la técnica de correlación cruzada. 6. SICC cuenta con un módulo de interface de usuario local que se ejecuta en la tarjeta BB-xM. 7. Módulo de interface de usuario remoto que se ejecuta en el cliente TCP, permite monitorear los resultados de procesamiento de forma remota.

Se diseñó un componente software de SICC especializado en procesamiento de imágenes que cuenta con dos funcionalidades: la capacidad de contar objetos segmentados por profundidad usando el sensor de vídeo 3D Kinect y reconocer patrones determinados en un objeto usando la técnica de correlación cruzada. Aunque SICC cuenta con dos aplicaciones particulares: *Counter Objects* y *Pattern Recognition*, la modularidad del componente software de SICC de procesamiento de imágenes permite incluir en trabajos futuros otro tipo de aplicaciones.

Se validó el funcionamiento integral de SICC en un entorno industrial (Línea de envasado de la Industria Licorera del Cauca ILC), durante un periodo de tiempo de treinta (30) días, donde se comprobó el funcionamiento exitoso del sistema,

logrando detectar 2.25% (697 matrices defectuosas) de producto defectuoso del total (30201 matrices inspeccionadas) durante el lapso de tiempo en las diferentes presentaciones del producto inspeccionado (cajas de licor en las presentaciones media, botella y garrafa).

El desarrollo de este sistema de inspección de características de calidad embebido para ambientes industriales establece una base de desarrollo para una línea de investigación que involucre el procesamiento de imágenes en líneas de manufactura encaminados a mejorar la calidad de los productos.

### **5.1. Trabajos futuros**

Con base en la experiencia obtenida en el desarrollo de este proyecto y la relación con la industria local y sus necesidades se hace oportuno mencionar algunas modificaciones, mejoras y complementos para el sistema de inspección de características de calidad desarrollado.

1. Desde la perspectiva del diseño hardware es conveniente probar la siguiente versión de la tarjeta Beagle llamada *BeagleBone Black*, ésta cuenta con un desempeño similar a la usada en este proyecto pero tiene ventajas como: menor consumo de potencia, menor tamaño físico, menor costo, etc. Tiene soporte al sistema operativo Linux al igual que la tarjeta BB-xM razón por la que el desarrollo software completamente es transparente a esta nueva modificación.
2. Incluir un sistema de reconocimiento de códigos OCR. En las líneas de producción es muy común encontrar que los sistemas de visión están dedicados a este tipo de tareas. Al pensar en el sistema SICC como un producto comercial, sin duda, para ser competitivo debe contar con este tipo de funcionalidades.

3. Si bien el desarrollo de este proyecto no se enfocó en la generación de un aplicativo software amplio de procesamiento de imágenes para aplicaciones industriales. Se propone realizar una búsqueda de los aplicativos de mayor demanda en las líneas de manufactura usando visión de máquina e incluirlos en el sistema desarrollado, por ejemplo (Detección de objetos particulares, determinación de la orientación de piezas, medida de nivel de líquido en envases, integridad de los objetos, etc.).

## BIBLIOGRAFÍA

- [1] M. Chen, «Development of an Embedded Industrial Vision System,» de *Industrial Electronics And Applications 1st IEEE Conference*, Singapore, 2006.
- [2] R. Lopez, «Aplicación del Sistema Robot Visión Pro Para Operaciones Automáticas de Control de Calidad,» *Ingeniería y Desarrollo*, vol. 9, pp. 88-97, 2001.
- [3] G. Zambrano, «Estación de Control de Calidad por Visión Artificial Para un Centro de Manufactura Integrada por Computador,» *Ingeniería y Universidad*, vol. 11, pp. 33-55, 2007.
- [4] F. Prieto, «An Automated Inspection System,» *The International Journal of Advanced Manufacturing Technology*, 2002.
- [5] Rockwell Automation, [En línea]. Available: <http://ab.rockwellautomation.com/Sensors-Switches/Photoelectric-Vision-Sensors>. [Último acceso: 12 October 2013].
- [6] Victronic, [En línea]. Available: <http://www.vitronic.de/en/logistics/>. [Último acceso: 12 October 2013].
- [7] Cognex, [En línea]. Available: <http://www.cognex.com/make-it-right.aspx?langtype=1034>. [Último acceso: 12 October 2013].
- [8] K. Boyer, «Robust Online Detection of Pipeline Corrosion From Range Data,» *Machine Vision and Applications*, vol. 12, pp. 291-304, 2001.
- [9] Altera Corporation, «Video and Image Processing Design Using FPGAs,» *ALTERA*, vol. 1, nº 1, pp. 1-2, 2007.
- [10] J. Velásquez, «Procesamiento de Video en Tiempo Real Utilizando FPGA,» de *4to Congreso de Computo AGECOMP*, Cuernavaca, Mexico, 2005.
- [11] D. Zuluaga, «Arquitectura de la DSP Blackfin ADSP-BF533 de Analog Devices y su Viabilidad en el Procesamiento Digital de Imágenes,» *Scientia et Technica*, vol. XVI, nº 48, pp. 179-184, 2011.
- [12] J. A. A. Cardona, «Blackfin: DSP para la Era Multimedia,» Medellin, Colombia, 2010.
- [13] B. Martinez, *Arquitecturas de Procesadores ARM de 32 Bits, Circuitos Integrados y Entornos de Desarrollo*, Sangolqui: Escuela Politécnica de Ingeniería Eléctrica Automatización y Control, 2007.
- [14] S. Chon, «Getting Started on TI ARM Embedded Processor Development-The Basics,» *Texas Instruments*, pp. 1-21, 2013.
- [15] T. Instruments, «Texas Instruments Official Page,» [En línea]. Available: <http://www.ti.com/lscds/ti/arm/applications.page>. [Último acceso: 1 October 2013].
- [16] BeagleBoard.org, «BeagleBoard xM rev C System Reference Manual,» Richardson TX, 2010.
- [17] IDS, [En línea]. Available: <http://es.ids-imaging.com/ensenso.html>. [Último

- acceso: 20 October 2013].
- [18] Vision Components, [En línea]. Available: <http://www.vision-components.com/en/products/smart-cameras/vc-nano-3d/>. [Último acceso: 20 October 2013].
  - [19] Photon Focus, [En línea]. Available: <http://www.photonfocus.com/html/eng/products/products.php?prodId=41>. [Último acceso: 20 October 2013].
  - [20] H. Gonzalez, «Metrological Evaluation of Microsoft Kinect and Asus Xtion Sensors,» *Measurement*, vol. 46, pp. 1800-1806, 2013.
  - [21] M. T. F. D. G. Sansoni, «State-of-the-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine and Criminal Investigation,» Sensors, Brescia, 2009.
  - [22] Microsoft Kinect, «Sensor Work Shop at ITP,» Microsoft, [En línea]. Available: <http://itp.nyu.edu/physcomp/sensors/Reports/MicrosoftKinect>. [Último acceso: 10 October 2013].
  - [23] T. J. M. Andersen, «Kinect Depth Sensor Evaluation for Computer Vision Applications,» Aarhus University, Denmark, 2012.
  - [24] C. Pillajo, «Human Machine Interface HMI Using Kinect Sensor to Control a Scara Robot,» de *Communications and Computing IEEE*, Medellin, 2013.
  - [25] N. Rafibakhsh, «Applying Fuzzy Logic for Optimal Placement of XBOX Kinect Sensors for Industrial Applications,» *Advanced Materials Research*, vol. 628, pp. 433-439, 2012.
  - [26] M. Tölgyessy, «The Kinect Sensor in Robotic Educaation,» de *Proceedings of 2nd International Conference on Robotics in Education*, Vienna, 2011.
  - [27] H. Zimmerman, «OSI Reference Model - The ISO Model of Architecture for Open System Interconnection,» *Transactions And Communications*, vol. 28, nº 4, pp. 425-432, 1980.
  - [28] MODICON, «Modbus Protocol Reference Guide,» Modicon, Inc., Massachusetts, 1996.
  - [29] E. Malamas, «Survey on Industrial Vision Systems, Applications and Tools,» *Image and Vision Computing*, vol. 21, nº 2, 2003.
  - [30] M. Magee, «An Industrial Model Based ComputerVision System,» *Journal of Manufacturing Systems*, vol. 14, pp. 169-186, 1995.
  - [31] R. Hunsicker, «Automating Vision Inspection and Measurement System for External Screw Threads,» *Journal of Manufacturing Systems*, vol. 13, pp. 370-383, 1994.
  - [32] C. Bahlmann, «Artificial Neural Networks for Automated Quality Control of Textile Seams,» *Pattern Recognition*, vol. 32, pp. 1049-1060, 1999.
  - [33] S. Bhandarkar, «A System for Detection and Renderingn of Internal Log Defects Using Computer Tomography,» *Machine Vision And Applications*, vol. 11, pp. 171-190, 1999.
  - [34] A. Jimenez, «Automatic Fruit Recognition:A Survey And New Results Using Range/Attenuation Image,» *Pattern Recognition*, vol. 32, pp. 1719-1736,

1999.

- [35] A. Mital, «A Comparison Between Manual And Hybrid Methods in Parts Inspection,» *Integrated Manufacturing Systems*, vol. 9, pp. 344-349, 1998.
- [36] H. Golnabi, «Design And Application of Industrial Machine Vision Systems,» *Robotics And Computer-Integrated Manufacturing*, vol. 23, pp. 630-637, 2007.
- [37] C. Wei, «A Design of Vision-Based Location Control System for Steel Rolling Mill,» de *Control Conference* , Hunan, 2007.
- [38] P. Drake, «A Decision Tree of Neural Networks for Clasifying Images of Wood Veener,» *International Journal of Advanced Manufacturing Technology*, vol. 14, pp. 280-285, 1998.
- [39] T. Cootes, «Statistical Grey-Level Models for Object Location and Identification,» *Image and Video Computing*, vol. 14, pp. 533-540, 1996.
- [40] K. Kim, «Automatic Visual Inspection System to Detect Wrongly Attached Components,» de *International Conference on Signal Processing Applications and Technology*, Toronto, 1998.
- [41] J. Jeng, «Gap Inspection and Alignment Using a Vision Technique for Laser Butt Joint Welding,» *International Journal of Advanced Manufacturing Technology*, vol. 16, pp. 2012-2016, 2000.
- [42] M. Moreira, «Image Classification for the Quality Control of Watches,» *Journal of Intelligent and Fuzzy Systems*, vol. 7, pp. 151-158, 1999.
- [43] Cognex, «In-Sight Vision Systems Product Guide,» Cognex Corporation, USA, 2012.
- [44] Rockwell Automation, «Multi Sight 48MS User Manual,» Rockwell Automation, USA, 2006.
- [45] Vitronic Vision, «VICAMSnap Data Sheet,» Vitronic, USA, 2012.
- [46] Vitronic Vision, «VICAMssi2 Data Sheet,» Vitronic, USA, 2009.
- [47] R. Lopez, «Aplicación del Sistema Robot Vision Pro para Operaciones Automáticas de Control de Calidad,» *Ingeniería y Desarrollo*, vol. 9, pp. 88-97, 2001.
- [48] L. Pencue, «Detección y Clasificación de Defectos en Frutas Mediante el Procesamiento Digiital de Imagenes,» *Revista Colombiana de Fisica*, vol. 35, nº 1, pp. 148-151, 2003.
- [49] Financiamiento Mendoza Productiva, «Diseño de Productos y Procesos de Producción,» 2007. [En línea]. Available: <http://www.portalpymes.mendoza.gov.ar//ArchivosPdf/Diseniodeproductosyprocesos2.pdf>. [Último acceso: 23 August 2013].
- [50] R. Lopez, «Aplicación del Sistema Robot Visión Pro Para Operaciones Automáticas de Control de Calidad,» *Ingeniería y Desarrollo*, vol. 9, pp. 88-97, 2001.
- [51] R. Salazar, *Modbus RTU, Implementación del Protocolo en Microcontrolador*, Bucaramanga, 2006.

- [52] G. Zambrano, «Estación de Control de Calidad por Visión Artificial para un Centro de Manufactura Integrada por Computador,» *Ingeniería y Universidad*, vol. 11, pp. 33-55, 2007.
- [53] F. Prieto, «An Automated Inspection System,» *International Journal of Advanced Manufacturing Technology*, vol. 19, pp. 917-925, 2002.

## **ANEXOS**

## **Anexo A - RS485 TO USB CONEVERTER NOVUS**

The converter can be configured for four-wire (Full Duplex) RS422 and RS485 or two-wire (HalfDuplex) networks. When operating in two-wire RS485 the data transfer control is automatically done by the converter. Two independent and isolated RS485 networks can be supported by one module thus duplicating the possible number of remote devices.

\_Operational system virtual serial port driver.

\_Supports Windows 7/Vista/XP/2008Server/ 2003Server/98/ME/2000/CE, MAC and Linux 2.4.20or above.64 bit options for latest operating systems.

\_Field Interfaces: RS485 Half Duplex (dual buses),RS485 Full Duplex or RS422.Jumper selected RS485 / RS422.

\_Automatic flow control for RS485 Half Duplex.120 Ohms internal resistors termination enabled by jumpers.

\_Transmission rate: from 300 bps to 250 kbps.

\_MaximumRS485/RS422 cable length : 1200 m

\_Maximum number of devices in the RS485 network:

\_Half Duplex: 2 x 32 devices

\_Full Duplex: 32 devices

\_Data transmission and reception LED indicators

\_Power: from the USB port.

\_Consumption: <100 mA.

\_Isolation: 1500 Vdc from USB interface and the RS485/RS422 interface.

RS485/422 bus protection:  $\pm 60$  Vdc, 15 kV ESD.

\_USB connection: Mini-B connector.

\_A 1.5 m cable with plugs mini-B and A is provided with the module.

\_RS485/422 connector: screw terminal type accepting 1.5 mm<sup>2</sup> (16 AWG) wires.

\_ABS enclosure: 70 x 60 x 18 mm

\_Operating environment: 0 to 70°C, 10 to 90% relative humidity, non-condensing.

## **Anexo B - CONFIGURACION SDCARD PARA BEAGLEBOARD-XM**

- En primer lugar descargar la imagen del sistema operativo.

```
$wget http://rcn-ee.net/deb/rootfs/quantal/ubuntu-12.10-console-armhf-2013-02-16.tar.xz
```

```
$wget http://ynezz.ibawizard.net/beagleboard/quantal/ubuntu-12.10-console-armhf-2013-02-16.tar.xz
```

- Verificar la integridad de la imagen.

```
$md5sum ubuntu-12.10-console-armhf-2013-02-16.tar.xz
```

- Descomprimir la imagen

```
sudo tar -xj ubuntu-12.10-console-armhf-2013-02-16
cd ubuntu-12.10-console-armhf-2013-02-16
```

- Instalar la imagen usando el script descargado setup\_sdcard.sh.

```
$sudo ./setup_sdcard.sh --mmc /dev/sdX --uboot board
```

"board" options:

```
BeagleBoard Ax/Bx - beagle_bx
BeagleBoard Cx/Dx - beagle_cx
BeagleBoard xM - beagle_xm
BeagleBone Ax - bone
BeagleBone Ax - bone_dtb (testing v3.8-rc7 based kernel)
PandaBoard Ax - panda
PandaBoard ES - panda_es
```

En particular para la tarjeta BB-xM.

```
$sudo ./setup_sdcard.sh --mmc /dev/sdX --uboot beagle_xm
```

Additional Options

```
--rootfs <ext4 default>
--swap_file <swap file size in MB's>
--addon pico <ti pico projector>
--svideo-ntsc <use ntsc over dvi for video>
--svideo-pal <use pal over dvi for video>
```

- Desmontar la tarjeta microSD y montar en la tarjeta BB-xM.

Al dar inicio a la tarjeta el Sistema solicita un password y un usuario, por defecto el usuario es ubuntu y el password tempPWD. Configurar la red Ethernet de internet. Actualizar e instalar la interface de usuario del Sistema operativo.

```
$sudo ifconfig -a
$sudo dpkg --configure -a
$$sudo apt-get dist-upgrade
$sudo apt-get install lxde
$sudo apt-get update
```

- Luego instalar las librerías de opencv y asociadas.

```
$sudo apt-get install gcc-multilib libusb-1.0.0-dev git-core build-essential
$sudo apt-get install doxygen graphviz default-jdk freeglut3-dev libopencv-dev
$ sudo apt-get install build-essential libavformat-dev ffmpeg
$ sudo apt-get install libcv2.1 libcvaux2.1 libhighgui2.1 opencv-doc
$ sudo apt-get install libcv-dev libhighgui-dev libcvaux-dev
$sudo apt-get install git-core cmake
$ sudo apt-get install cmake libglut3-dev libxmu-dev libusb-1.0-0-dev pkg-config
libxi-dev
```

- Instalar las librerías para el manejo del sensor KINECT

```
$sudo apt-get install freenect
$sudo apt-get install libxmu-dev libxi-dev
```

- Descargar los archivos fuentes de las librerías para el sensor KINECT

```
$wget
http://ecse.monash.edu.au/twiki/bin/view/WSRNLab/BeagleBoardConfigurationFor
Kinect
```

- Compilar las librerías para el sensor KINECT

```
$ cd Desktop/libfreenect
$ mkdir build
$ cd build
$ cmake
$ make
$ sudo make install
```

- Generar un archivo .rules para el sensor KINECT

```
$ sudo gedit /etc/udev/rules.d/66-kinect.rules
```

```
SYSFS{idVendor}=="045e", SYSFS{idProduct}=="02ae",  
MODE="0660",GROUP="video"
```

```
SYSFS{idVendor}=="045e", SYSFS{idProduct}=="02ad",  
MODE="0660",GROUP="video"
```

```
SYSFS{idVendor}=="045e", SYSFS{idProduct}=="02b0",  
MODE="0660",GROUP="video"
```

- Finalmente probar el programa demo de manejo de KINECT y OpenCV.

```
$ cd libfreenet/build/bin  
$ sudo ./cvdemo
```

- Instalar el IDE de desarrollo CodeBlocks y realizar el link de librerías

```
$ sudo apt-get install codeblokcs
```

```
$ sudo apt-get install libcv-dev libcvaux-dev libhighgui-dev libcv2.1 libhighgui2.1  
libcvaux2.1 opencv-doc
```

- Luego se prepara CodeBlocks para que pueda compilar programas que incluyan las librerías de OpenCV. Abrir el IDE y seleccionar en la barra de herramientas: Settings--> Compiler and Debugger. En la pestaña Linker Settings agregar los enlaces: -lml -lcvaux -lhighgui -lxcvcore -lcv. Y en la pestaña Search Directories->Compilers agregar la ruta de la carpeta include de OpenCV: /usr/include/opencv.
- Hacer una copia de seguridad de libfreenect, Sobre la copia borrar la carpeta build. Copiar los archivos de:

```
copia/wrappers/opencv (Todos)  
copia/wrappers/c_sync (libfreenect_sync.c, libfreenect_sync.h)  
copia/include (libfreenect.h)
```

Al destino:

```
copia/examples (Borrar lo que hay en la carpeta)
```

- Compilar de nuevo:

```
$cd copia/  
$mkdir build  
$cd build/  
$cmake  
$make
```

```
$sudo make install
$ldconfig /usr/local/lib64/
```

- Como prueba, en la carpeta build, ejecutar en la terminal

```
$sudo cvdemo
```

- Si al compilar y correr el código se produce el siguiente error

```
libusb couldn't open USB device /dev/bus/usb/001/012: Permission denied (Donde 001/012 pueden variar)
```

Entonces se debe revisar que ID corresponde al dispositivos de Microsoft digitando en la terminal `lsusb`

```
$lsusb
```

y aparecerá algo como esto:

```
Bus 002 Device 020: ID 045e:02ae Microsoft Corp.
Bus 002 Device 019: ID 045e:02ad Microsoft Corp.
Bus 002 Device 018: ID 045e:02b0 Microsoft Corp.
Bus 002 Device 017: ID 0409:005c NEC Corp. HighSpeedHub
```

Para solucionar el problema y poder acceder al dispositivo, digitar en la terminal:

```
$chmod a+rw /dev/bus/usb/002/020
```

Nótese que el 002 es el Bus y 020 Device. Repetir el procedimiento para los 3 ID's de Microsoft:

```
$chmod a+rw /dev/bus/usb/002/019
$chmod a+rw /dev/bus/usb/002/018
```

- Configurar la red de internet

```
$sudo apt-get install network-manager
$sudo nano /etc/network/interfaces
```

- Agregar las siguientes líneas.

```
auto eht0
iface eth0 inet dhcp
hwaddress ether DE:AD:BE:EF:CA:FE
```

- Configurar Autologin para el sistema operativo

```
$sudo nano /etc/lxdm/lxdm.conf
```

```
uncomment autologin=user    (ubuntu)  
uncomment timeout=10  
uncomment session=/usr/bin/startlxde
```

- Generar el autostart de la aplicación

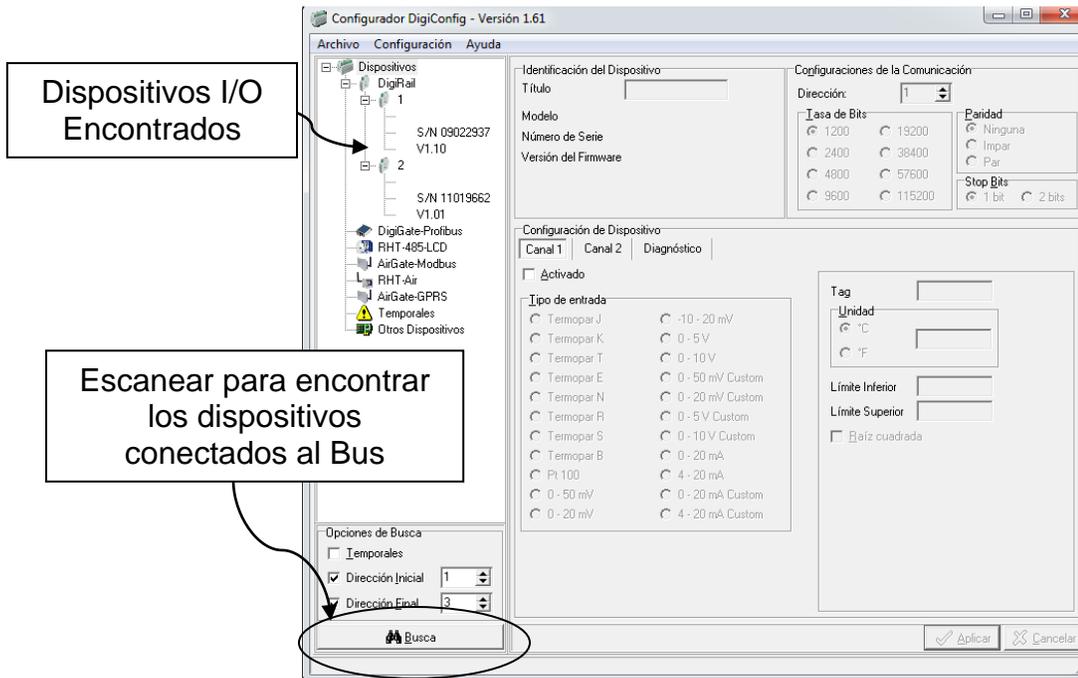
```
sudo nano /usr/share/applications/VQCS.desktop
```

```
[Desktop Entry]  
Name=VQCS  
Exec=/home/...../INTERV2  
Type=Application  
Hidden=false  
NoDisplay=false  
Terminal=false
```

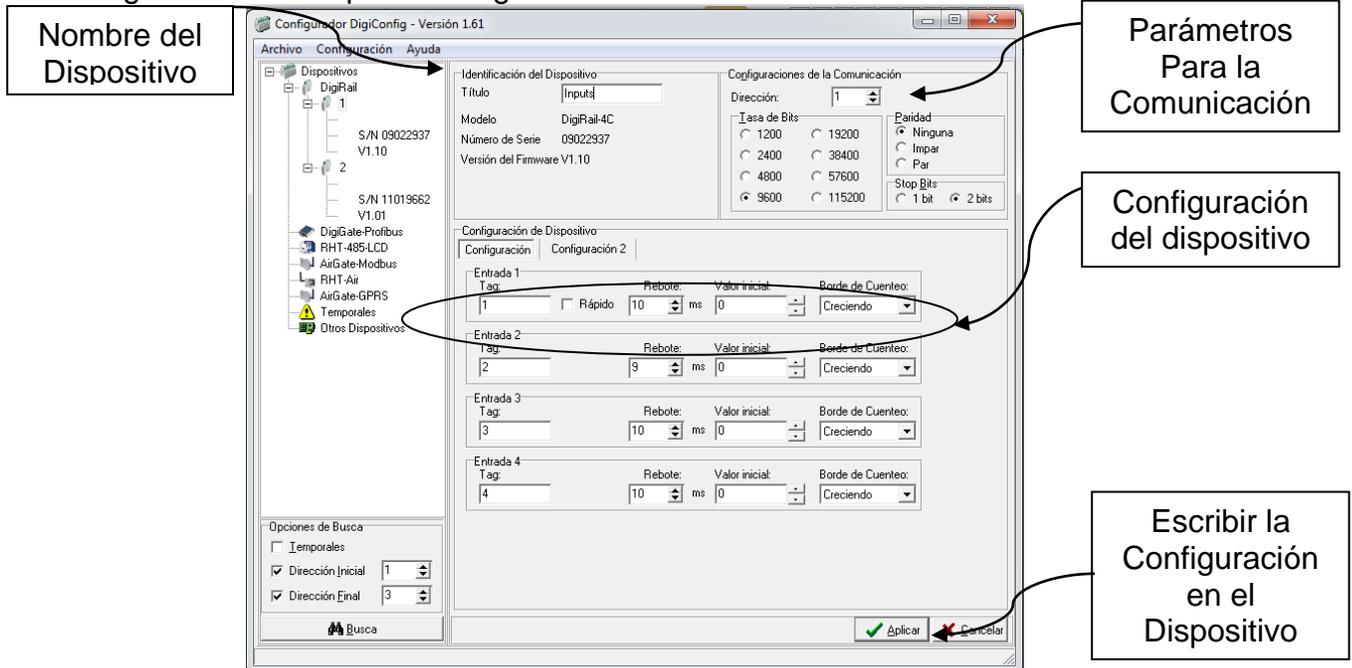
```
$sudo cp /usr/share/applications/INTERV2.desktop /etc/xdg/autostart/
```

## Anexo C - CONFIGURACIÓN DIGIRAIL'S USANDO DIGICONFIG

Instalar el software DigiConfig y descargar el driver apropiado del conversor usb-rs485. Ejecutar el software y escanear para la detección de digirails en el bus. Finalmente sobre la interface configurar parámetros de comunicación: baudrate, stop bits, paridad, etc.



### Configuración del dispositivo DigiRail 4C



## Configuración del Dispositivo DigiRail 2R.

**Nombre del Dispositivo**

**Parámetros Para la Comunicación**

**Configuración del dispositivo**

**Escribir la Configuración en el Dispositivo**

Configurador DigiConfig - Versión 1.61

Archivo Configuración Ayuda

Dispositivos

- DigiRail
  - 1
    - S/N 09022937 V1.10
    - 2
  - 2
    - S/N 11019662 V1.01
  - DigiGate-Profibus
  - RHT-485-LCD
  - AirGate-Modbus
  - RHT-Air
  - AirGate-GPRS
  - Temporales
  - Otros Dispositivos

Identificación del Dispositivo

Título: [Outputs]

Modelo: DigiRail-2R

Número de Serie: 11019662

Versión del Firmware V1.01

Configuraciones de la Comunicación

Dirección: [2]

Tasa de Bits:  
 1200  19200  
 2400  38400  
 4800  57600  
 9600  115200

Paridad:  
 Ninguna  
 Impar  
 Par

Stop Bits:  
 1 bit  2 bits

Configuración de Dispositivo

Configuración | Diagnóstico

Relé 1

Tag: [1]

Temporización del Relé: [2] x 0.01 segundos

Relé 2

Tag: [2]

Temporización del Relé: [2] x 0.01 segundos

Opciones de Busca

Temporales

Dirección Inicial: [1]

Dirección Final: [3]

Busca

Aplicar Cancelar

## Anexo D - MODULO SOFTWARE MODBUS (SOURCE CODE)

Método para calcular el valor de control de errores CRC desarrollado en C.

```
unsigned int CRC16 (unsigned char *nData, unsigned int wLength)
{
static const unsigned int wCRCTable[] = {
0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241,
0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481, 0X0440,
0XCC01, 0X0CC0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40,
0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841,
0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81, 0X1A40,
0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41,
0X1400, 0XD4C1, 0XD581, 0X1540, 0XD701, 0X17C0, 0X1680, 0XD641,
0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040,
0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240,
0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480, 0XF441,
0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41,
0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840,
0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,
0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40,
0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681, 0X2640,
0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041,
0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240,
0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441,
0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41,
0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840,
0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,
0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40,
0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,
0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,
0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,
0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,
0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,
0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,
0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,
0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,
0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,
0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040 };

unsigned char nTemp;
unsigned int wCRCWord = 0xFFFF;

while (wLength--)
{
nTemp = *nData++ ^ wCRCWord;
```

```

wCRCWord>>= 8;
wCRCWord ^= wCRCTable[nTemp];
}
return wCRCWord;

}

```

Método para leer el módulo DigiRail 4C.

```

unsigned int Read4CMoudbusDevice(int serial_handle, unsigned char Device,
unsigned char *Device4CRead, int SizeReg, int timeout, unsigned char*
InputRegister4C)
{
    unsigned char Device4C[8]={Device, 0x03, 0x00, 0x00, 0x00, 0x11,0x00,0x00};
    // 0x01 Direction
    // 0x03 Function Read Register
    // 0x00 0x00 First Register
    // 0x00 0x11 Last Register
    // The last two bytes are CRC bytes next to be calculated
    unsigned char Device4C_2[6]={Device, 0x03, 0x00, 0x00, 0x00, 0x11};

    unsigned char HSBCRC;
    unsigned char LSBCRC;
    unsigned int CRC;
    unsigned int status;

    CRC=CRC16(Device4C_2,sizeof(Device4C_2));
    LSBCRC=CRC;
    HSBCRC=CRC>>8;

    Device4C[sizeof(Device4C_2)]=LSBCRC;
    Device4C[sizeof(Device4C_2)+1]=HSBCRC;

    serial_send(serial_handle, Device4C, 8);

    status=serial_read(serial_handle,Device4CRead,SizeReg,TIMEOUT);
    fflush(stdout);

    //Asignación de los valores leídos al vector de lectura

    InputRegister4C[0]=Device4CRead[20];
    InputRegister4C[1]=Device4CRead[22];
    InputRegister4C[2]=Device4CRead[24];
    InputRegister4C[3]=Device4CRead[26];

    return status;
}

```

Método para escribir el módulo DigiRail 2R.

```
unsigned int Write2RModbusDevice(int serial_handle, unsigned char Device,
unsigned char *Device2RRead, int SizeReg, int timeout, unsigned char Value,
unsigned char SubDevice)
{
    unsigned char Device2R[8]={Device, 0x06, 0x00, SubDevice + 0x08, 0x00, Value,
0x00,0x00};
    // 0x02 Direction
    // 0x06 Function Write Register
    // 0x00 0x08 Register Address
    // 0x00 0x00 Register Value
    // The last two bytes are CRC bytes next to be calculated
    unsigned char Device2R_2[6]={Device, 0x06, 0x00,SubDevice + 0x08, 0x00,
Value};

    unsigned char HSBCRC;
    unsigned char LSBCRC;
    unsigned int CRC;
    unsigned int status;

    CRC=CRC16(Device2R_2,sizeof(Device2R_2));
    LSBCRC=CRC;
    HSBCRC=CRC>>8;

    //printf("CRC:%x\n",CRC);

    Device2R[sizeof(Device2R_2)]=LSBCRC;
    Device2R[sizeof(Device2R_2)+1]=HSBCRC;

    //-- Enviando el comando al puerto serie
    serial_send(serial_handle, Device2R, sizeof(Device2R));

    status=serial_read(serial_handle,Device2RRead,SizeReg,timeout);
    fflush(stdout);

    return status;
}
```

## Anexo E – INTERACCIÓN MÓDULOS SOFTWARE

Descripción: La arquitectura software diseñada cuenta con 3 hilos de proceso para la aplicación local. 3 hilos se ejecutan en la tarjeta BB-xM: 1. Hilo de interface de Usuario: Maneja los eventos de la interface de usuario principal y de aplicación. 2. Hilode de Kinect y procesamiento: Maneja la adquisición de video desde el sensor Kinect y el proceamiento digital de imágenes. 3. Hilo Módulo Modbus RTU: maneja los eventos de entrada y salida que ocurren en el bus Modbus. En adición existe el módulo TCP Servidor que se basa en eventos de una clase que se genera desde la interface de usuario. La interface remota cuenta con un hilo de proceso asociado al manejo de la interface de usuario remota y un módulo basado en eventos para el manejo de sockets TCP. Las aplicaciones remota y local se comunican a través de sockets mediante tramas definidas de lectura y escritura.

Diagrama módulos software desarrollado

