ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE DETECCIÓN DE JABAS INCOMPLETAS CON BOTELLA VACÍA

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL

CÉSAR FERNANDO ARAGUILLÍN COLLAGUAZO

JEANNETH DEL PILAR TITUAÑA NAVARRETE

DIRECTOR: DOCTOR LUIS CORRALES

QUITO, DICIEMBRE 2005

DECLARACIÓN

Nosotros, Jeanneth del Pilar Tituaña Navarrete y César Fernando Araguillín Collaguazo, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Jeanneth del Pilar Tituaña Navarrete

César Fernando Araguillín Collaguazo

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Jeanneth del Pilar Tituaña Navarrete y César Fernando Araguillín Collaguazo, bajo nuestra supervisión.

Dr. Luis Corrales
DIRECTOR DEL PROYECTO

Ing. Juan Carlos Cifuentes
CODIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Agradezco a Dios por brindarme la vida y a mis Padres que con su esfuerzo, consejos y confianza han sabido guiarme por el camino adecuado para lograr mis metas y objetivos propuestos.

Agradezco al Ing. Jesús David Rueda y al Ing. Juan Carlos Cifuentes, Gerente y Jefe de Embotellado respectivamente de Cervecería Andina por haberme permitido desarrollar el reciente Proyecto en las instalaciones de la Empresa que atinadamente dirigen.

Agradezco al Ing. Juan Carlos Cifuentes y al Dr. Luís Corrales que con sus consejos, guías y enseñanzas me supieron dirigir a lo largo del presente Proyecto para alcanzar los mejores resultados.

Agradezco al Ing. Byron Moya, Tecnólogo Benito Santillán, Eléctrico Guillermo Pérez, personal de talleres y en general a todas las personas de Cervecería Andina que colaboraron desinteresadamente con la ejecución de este Proyecto.

Agradezco a Pilar Tituaña que se esforzó junto a mí para desarrollar y terminar nuestro Proyecto de Titulación.

Agradezco a mi hermana, abuelos, tíos y demás familiares que con sus ánimos y compañía me impulsaron para terminar este Proyecto.

Agradezco a mis amigos y en especial a Gabriela que con su amistad y apoyo me animaron en instantes buenos y malos a lo largo de mi Carrera Universitaria y en la realización de este proyecto.

Agradezco a la Escuela Politécnica Nacional y a todos los profesores que a lo largo mi permanencia en la Universidad me impartieron los conocimientos necesarios e impulsaron a aprender más y concluir con éxito la Carrera de Ingeniería en Electrónica y Control.

César Araguillín

DEDICATORIA

Dedico este proyecto a mis Padres César y Beatriz, a mi Hermana Pamela, a mis Abuelitos y a mi querida Gabriela que estuvieron conmigo a lo largo de este Proyecto.

César Araguillín

AGRADECIMIENTOS

En primer lugar quiero agradecer a Dios por la vida maravillosa que me ha regalado y a su incondicional apoyo que me ha ayudado a culminar mi carrera.

Agradezco infinitamente al Ing. Juan Carlos Cifuentes por la oportunidad de realizar este proyecto en Cervecería Andina, y abrir nuevas expectativas del campo de trabajo para el cual nos hemos preparado. Por haber transmitido sus conocimientos y consejos que por siempre tendré presente.

Al Ing. Byron Moya, Tecnologo Benito Santillán, Eléctrico Guillermo Pérez, y a todo el personal de Cervecería Andina quienes con sus oportunos consejos y ayuda hicieron de la labor dentro de la Empresa un tiempo enriquecedor, ameno, pero sobre todo de gran confianza y amistad.

Al Doctor Luis Corrales por su amplia disponibilidad y entrega para ayudarnos a solucionar cualquier duda que se presentó en el transcurso de este proyecto.

Quiero agradecer a César por su ayuda y paciencia en los buenos y malos momentos que hemos atravesado y que me han ayudado a ser un mejor.

Agradezco a mis Padres y hermanos por su incansable esfuerzo por ayudarme a terminar la carrera, pero sobre todo por su amor y comprensión en cada momento de mi vida.

Jeanneth Tituaña

DEDICATORIA

Dedico este trabajo a las personas que con su apoyo, confianza y amor lograron que mi carrera se vea realizada, mis Padres Gerardo y Rosa, a mi adorable hermana Lucía, a mi amado esposo Danny y con especial amor a mi hijo Alejandro a quienes amo con todo mi corazón.

Jeanneth Tituaña

CONTENIDO

RESUMEN..... 1
PRESENTACIÓN..... 3

CAPITULO 1..... 5
DESCRIPCIÓN DE LA SECCIÓN DE ENVASE Y DEL PROBLEMA DE CAJAS CON BOTELLA VACÍA..... 5
1.1 Descripción general de la sección de embotellado..... 6
1.1.1 Agua Manantial Línea #2..... 6
1.1.2 Línea #3..... 6
1.2 Línea de embotellado #3..... 6
1.2.1 Despaletizadora y Desencajonadora..... 7
1.2.2 Lavadora de botellas..... 7
1.2.3 Inspectores de Botellas..... 7
1.2.4 Llenadora y tapadora..... 8
1.2.5 Pasteurizador..... 8
1.2.6 Etiquetadora e inspector de nivel de llenado..... 8
1.2.7 Encajonadora y Paletizadora..... 8
1.3 Descripción del problema de cajas con botella vacía (cajas que entrega el distribuidor)..... 11
1.4 Causas del problema..... 11
1.5 Consideraciones técnicas del proceso..... 12
1.6 Posibles soluciones..... 13

CAPITULO 2..... 15
PLANTEAMIENTO Y VALORACIÓN DE MÉTODOS PARA SOLUCIONAR EL PROBLEMA..... 15
2.1 Sensores fotoeléctricos..... 15
2.1.1 Características..... 16
2.1.2 Características del sensor probado..... 17
2.1.3 Prueba de sensores sobre las cajas..... 18
2.1.3.1 Resultados obtenidos con sensores sobre las cajas..... 19
2.1.4 Prueba de sensores y lengüetas de caucho..... 20

2.1.4.1 Resultados obtenidos con el método de las lengüetas..... 20
2.2 Sensores ultrasónicos..... 21
2.2.1 Curvas y características..... 21
2.2.2 Características del sensor probado..... 23
2.2.3 Pruebas de sensores ultrasónicos sobre la caja..... 24
2.2.3.1 Resultados obtenidos con los sensores ultrasónicos..... 25
2.3 Análisis digital de imágenes..... 25
2.3.1 Análisis computarizado con el software de LabView..... 26
2.3.2 Características de los equipos de inspección visual DVT..... 27
2.4 Selección de la mejor alternativa..... 29

CAPITULO 3..... 33
DISEÑO E IMPLEMENTACIÓN DEL SISTEMA Y ELEMENTOS PARA LA DETECCIÓN DE BOTELLA ROTA Y FALTANTE..... 33
3.1 Ubicación del sistema dentro del envase..... 33
3.2 Equipo de inspección visual DVT..... 36
3.2.1 Características de los lentes LTC-25F..... 37
3.2.2 Características de la tarjeta de entradas/salidas (CON-IBOB)..... 40
3.2.3 Características del cable de poder CBL5-D10..... 41
3.3 Requerimientos del sistema..... 41
3.3.1 Transportadora para separación entre Cajas..... 42
3.3.1.1 Descripción de las condiciones originales de la transportadora antes de la ubicación del sistema..... 42
3.3.1.2 Descripción de la sección de transportadora requerida por el sistema de inspección visual..... 44
3.3.2 Rieles para posicionar cajas..... 50
3.3.3 Protección para cámaras digitales..... 50
3.3.4 Sistema de iluminación para cajas..... 51
3.3.5 Sensor trigger para la captura de imágenes..... 52
3.4 Instalación del tablero de control..... 53
3.5 Elección de cable para comunicación..... 54
3.6 Instalación y cableado de los equipos..... 56

CAPITULO 4..... 58
DESARROLLO DE PROGRAMAS DE INSPECCIÓN, Y DE INTERFASE PARA DEL USUARIO..... 58
4.1 Funcionamiento del SmartImage sensor..... 59
4.2 Direccionamiento de las Cámaras..... 59
4.2.1 Topología física usada para conectar las cámaras..... 61
4.2.2 Protocolos MAC..... 62
4.3 Plataforma de programación. FrameWork..... 62
4.3.1 Concepto de SoftSensor. 63
4.3.1.1 Descripción de cada uno de los sensores..... 65
4.3.2 Concepto de un Producto. 68
4.3.3 Niveles de Programación de la Cámara..... 69
4.3.4 Definición de Script..... 70
4.3.4.1 Background Scripts..... 70
4.3.4.2 Foreground Scripts..... 71
4.4 Proceso de Análisis de la Caja..... 71
4.4.1 Bloque: Detección de Caja..... 72
4.4.2 Bloque: Adquisición de Imágenes..... 73
4.4.2.1 Trigger Source..... 74
4.4.2.2 Tiempo de exposición..... 74
4.4.2.3 Tiempo de digitalización..... 75
4.4.2.4 Ganancia del Producto..... 75
4.4.2.5 Ganancia del Sensor..... 75
4.4.2.6 Ventana Parcial de la Imagen..... 75
4.4.3 Bloque: Análisis de Imagen..... 76
4.4.3.1 Aplicación del Producto..... 78
4.4.3.2 Visualización de resultados..... 83
4.4.4 Bloque 4: Toma de Acciones..... 86
4.5 Diseño de la Interfaz con el Usuario..... 87
4.5.1 Pantalla de Presentación..... 87
4.5.2 Pantalla de Conexión..... 89
4.5.3 Ventana de Archivo de Imágenes fallidas..... 94

4.5.4 Pantalla de Internet..... 96

CAPITULO 5..... 99
PRUEBAS Y RESULTADOS..... 99
5.1 Pruebas de Calibración..... 99
5.1.1 Prueba de Calibración de Imágenes..... 99
5.1.2 Prueba de Calibración de Sensores..... 100
5.1.2.1 Prueba 1 de calibración: Sensor de Presencia de Caja..... 101
5.1.2.2 Prueba 2 de calibración: Obtención de Valores para Threshold de Sensores de Botellas..... 104
5.1.2.3 Prueba 3 de calibración: Recalibración de Valores para Threshold de Sensores de Botellas..... 109
5.2 Resultados de la Prueba..... 110
5.2.1 Resultados Prueba de Calibración de Imágenes..... 110
5.2.2 Resultado Prueba 1 de Calibración: Sensor de Detección de Caja... 110
5.2.3 Resultado Prueba 2 de Calibración: Obtención de Valores para Threshold de Sensores de Manchas..... 112
5.2.4 Resultado Prueba 3 de Calibración: Después de la Recalibración de Valores para Threshold de Sensores de Botellas..... 115

CAPITULO 6..... 119
CONCLUSIONES Y RECOMENDACIONES..... 119
6.1 Conclusiones..... 119
6.2 Recomendaciones..... 122

REFERENCIAS BIBLIOGRÁFICAS..... 124
ANEXOS..... 125
Anexo 1: Planos Eléctricos..... 126
Anexo 2: Descripción de Equipo de Inspección Adquirido..... 134
Anexo 3: Manual de Usuario..... 148
Anexo 4: Código Fuente 156

RESUMEN

El objetivo de este proyecto fue implementar un sistema que realice una inspección del estado de cada una de las jabas con botellas vacías que ingresan a la sección de embotellado #3 de Cervecería Andina, e identifique la condición física de cada una de las doce botellas contenidas en la jaba. El sistema de inspección debió hallar dentro de la jaba botellas rotas, ausentes o con anomalías tales como basuras en su interior, botellas cubiertas o tapadas. El registro de estos datos permitirá llevar una estadística diaria de las anomalías que se detecten, y que se traducen en pérdidas económicas para la Cervecería

Para cumplir con este objetivo en primer lugar se analizaron las condiciones de trabajo de la línea de embotellado de la Cervecería, para buscar el lugar adecuado para la ubicación del nuevo sistema, sin que éste interfiera con el proceso normal de envasado.

Para detectar las botellas con problemas se experimentó con varias opciones, desde el uso de simples sensores fotoeléctricos, hasta el empleo de sofisticadas cámaras industriales digitales. Finalmente, la alternativa que se implementó consistió en realizar la inspección mediante el análisis digital de imágenes capturadas con cámaras industriales digitales que son capaces de adquirir, inspeccionar, emitir resultados y ejecutar acciones de control sin la necesidad de estar conectada a una computadora.

El sistema de inspección seleccionado adquiere la imagen de la jaba mientras se halla en movimiento en la transportadora inmediatamente después que un sensor fotoeléctrico de trigger ha detectado que la jaba esta en la posición adecuada.

Con la imagen de la jaba adquirida y digitalizada, la cámara inicia el proceso de inspección de la jaba en busca de las fallas antes descritas evaluando la condición física de cada botella y en base a este análisis emite resultados del estado físico de cada botella, los resultados son almacenados en registros para luego ser enviados a la interfase desarrollada en Visual Basic.

El método usado para la identificación del buen estado o presencia de una botella es la medición de las manchas oscuras que tiene una botella tratando de discernir todas estas manchas de la que corresponde al agujero del pico de una botella sana. Para esto el programa implementado utiliza sensores virtuales sobre la imagen cuyo objetivo es medir parámetros tales como brillos y manchas oscuras de cada botella. Dichos valores son comparados con parámetros obtenidos experimentalmente del tamaño en píxeles de una mancha correspondiente al agujero del pico de la botella y del estado de brillo de las botellas, determinando así el estado físico de cada botella en la jaba, terminando la inspección de esa imagen.

De este análisis se obtienen resultados que indican el número de botellas en buen estado por cada jaba y los datos son acumulados en registros de la cámara y visualizados en una PC gracias a una HMI desarrollada en Visual Basic.

El sistema implementado una vez puesto en marcha, fue probado en producción con 20883 jabas que ingresaron a la línea de embotellado #3, el programa implementado procesó las botella en las jabas y detectó un 0.99% de botellas con falla. Este porcentaje de 0.99% de botellas con falla comprende botellas rotas, ausentes, tapadas, y botellas malas que pasaron por buenas y viceversa.

Es decir el porcentaje de botellas con fallas que arroja el sistema se debe al cumplimiento del objetivo del proyecto, a condiciones indeseables de las botellas tales como diferentes grados de suciedad de las botellas y por último a errores de percepción de la cámara. El porcentaje de botellas rotas y ausentes es de 0.277%, el de botellas tapadas y con basura es de 0.6% y el error que comete la cámara es de 0.13%.

El porcentaje de botellas con fallas y el error que comete la cámara esta dentro de un rango normal para equipos de estas características y con las limitaciones que tiene el sistema desarrollado.

PRESENTACIÓN

El presente proyecto "Diseño y Construcción de un Sistema de Inspección de Jabas Incompletas con Botella Vacía", ha sido realizado en Cervecería Andina S.A. con la asesoría y supervisión del Ing. Juan Carlos Cifuentes, Jefe de Embotellado. El proyecto fue auspiciado en su totalidad por Cervecería Andina S.A.

En la actualidad todas las industrias tienden a la automatización de sus procesos, así como a la optimización de sus recursos, con el fin de incrementar su producción y disminuir sus gastos.

Sin lugar a duda Cervecería Andina S.A. es una empresa de larga trayectoria dedicada a la producción de la tradicional cerveza Pilsener y otros productos como Agua Manantial y Pony Malta. Es por ello que, en su afán de mejorar y automatizar todos los procesos involucrados en la producción de cerveza, se ha propuesto solucionar el problema existente en la sección de envasado, con la llegada de las cajas de cerveza con botellas vacías. Las cajas llegan con botellas rotas o faltantes que se traducen en pérdidas económicas para la Cervecería.

En este proyecto se busca una solución que posibilite la rápida identificación de las botellas con anomalías que redundará en un ahorro para la Cervecería.

Con este objetivo, en el Capítulo 1 se hace una breve descripción del proceso de embotellado, así como de las anomalías que se encuentran en las cajas entrantes a la sección de embotellado. Además, se analiza las posibles soluciones y las consideraciones técnicas del proceso de embotellado, las que no deben ser alteradas una vez que el sistema se haya implementado.

En el Capítulo 2, conscientes de las anomalías que se deben identificar dentro de cada una de las cajas; se procedió a analizar y probar algunas alternativas, como el uso de sensores fotoeléctricos, e incluso la programación de sofisticadas

cámaras digitales. En este Capítulo se valora y compara los resultados obtenidos con cada alternativa y finalmente se escogió aquella que cumple de mejor manera los requerimientos del proyecto y que a su vez satisfaga las necesidades de Cervecería Andina.

En el Capítulo 3 se describe la ubicación física que el sistema ocupó dentro de la línea de embotellado, también se detallan las características de todos los elementos y equipos necesarios para la implementación del sistema, como las cámaras digitales y su funcionamiento, transportadora para cajas, sistema de iluminación, tablero de control, sensores trigger, etc. Finalmente, se muestra la interconexión entre todos los elementos que intervienen en el sistema.

En el Capítulo 4 se describe los pasos que se debe seguir para programar las cámaras digitales, de forma que sean capaces de adquirir imágenes de cada una de las cajas que ingresan a la línea de embotellado y en forma instantánea den un veredicto del estado de cada una de las botellas integrantes de la caja.
Las cámaras digitales empleadas para el objetivo del proyecto son capaces de almacenar todo tipo de datos, y gracias al tipo de comunicación que posee (Ethernet) se implementó un HMI diseñado en Visual Basic, que permite la visualización de las imágenes que las cámaras están adquiriendo y también los registros de cajas y botellas buenas y anómalas que se estén detectando.

En el siguiente Capítulo 5 se hace varias pruebas con el programa implementado en las cámaras, con estas se pudo ver la mejor calibración de las cámaras para que el error por concepto de programación sea lo menor posible, y adicionalmente el error arrojado sea lo mas confiable posible.

Los resultados y conclusiones de este proyecto ayudan a la gestión administrativa de la Cervecería y que le permite llevar una estadística de las pérdidas económicas que sufre por concepto de botellas rotas y faltantes en la sección de entrada del envase.

CAPITULO 1

DESCRIPCIÓN DE LA SECCIÓN DE ENVASE Y DEL PROBLEMA DE CAJAS CON BOTELLA VACÍA

En Cervecería Andina se lleva a cabo la producción de la conocida cerveza Pilsener y otros productos como cerveza Clausen, Dorada, Club, Agua Manantial y refrescos como Pony Malta.

Para la producción de cerveza se empieza con la mezcla y cocción de productos como cebada, lúpulo, agua y otros. Una vez que los productos se han preparado empieza el proceso de fermentación y enfriamiento, aquí de acuerdo al tiempo de fermentación se obtendrán las diferentes variedades de cerveza que luego saldrán al mercado.

Una vez que la cerveza deseada está lista pasa por los ductos correspondientes y llega a la sección de embotellado, en donde la cerveza será envasada en cada uno de los envases correspondientes.

En la Cervecería existe la visión de automatizar y mejorar cada uno de los procesos involucrados en la producción, es así que en la sección de embotellado se ha visto la necesidad de implementar un sistema que permita llevar una estadística continua del estado en que las botellas vacías, dentro de sus respectivas jabas, son entregadas a la Cervecería.

El proyecto que aquí se trata, tiene como objetivo general identificar botellas rotas y ausentes dentro de cada una de las cajas que están llegando a la Cervecería. Por medio de una interfaz (HMI) adecuada se desplegará la información capturada y procesada sobre las botellas buenas y malas que están entrando a la Cervecería, ayudando a la gestión administrativa que registra las pérdidas económicas que la empresa enfrenta por concepto de botellas rotas y ausentes.

Para entender mejor lo realizado en este proyecto, se empieza por hacer una descripción general del proceso de embotellado.

1.1 DESCRIPCIÓN GENERAL DE LA SECCIÓN DE EMBOTELLADO

La sección de embotellado consta de 3 líneas de producción y son:

1.1.1 AGUA MANANTIAL

En la planta de agua manantial se produce agua pura, que cumple con todas las normas exigidas y es apta para el consumo humano.

1.1.2 LÍNEA #2

Esta planta se encarga de envasar cerveza en presentación media, en las marcas Pilsener, Clausen, Dorada y Club; además refrescos como Pony Malta.

1.1.3 LÍNEA #3

Es la que se encarga de envasar cerveza en presentación grande, en las marcas Pilsener, Clausen y Dorada.

El sistema que se propone, se instaló en la Línea #3, por ello a continuación se hace una descripción del proceso de envase en esta sección exclusivamente.

1.2 LÍNEA DE EMBOTELLADO #3

En toda la línea de producción de envase existen dos líneas de transporte paralelas, así como dos máquinas del mismo tipo que agilizan el trabajo. Cada máquina y línea de transporte funciona en forma independiente una de la otra, evitando que cualquier problema que surja a lo largo de cualquier línea de transporte detenga la línea de producción en su totalidad.

1.2.1 DESPALETIZADORA Y DESENCAJONADORA

Los distribuidores llegan hasta la cervecería para entregar cajas con botellas vacías, las cajas vienen apiladas en paletas que llevan setenta y dos cajas, las cajas son guardadas en despacho hasta el momento que se las requiera.

Cuando se requiere envase vacío, las paletas son llevadas, con ayuda de montacargas, hasta la máquina despaletizadora en donde son desagrupadas y ubicadas en dos líneas de transporte que las llevan hasta la máquina desencajonadora.

La máquina despaletizadora utiliza un sistema de chupones que extrae simultáneamente todas las botellas de cada una de las doce cajas que se encuentran dentro de la máquina, para luego ubicar las botellas en la mesa de la desencajonadora. Las cajas vacías son llevadas hasta la lavadora de cajas.

Por otro lado, en la mesa de la desencajonadora las botellas son acumuladas y transportadas mediante cadenas de barajas hasta la máquina lavadora de botellas.

En esta sección es donde se tiene el problema de las cajas que llegan con botellas rotas o faltantes y el sistema implementado trabajará sobre cada una de las cajas, aunque la identificación de anomalías será por cada una de las botellas.

1.2.2 LAVADORA DE BOTELLA

En esta máquina se realiza un trabajo muy importante dentro del proceso de embotellado. Aquí se lavan las botellas y se asegura su total desinfección. Con ayuda de agua y soda cáustica que está a alta temperatura se asegura la limpieza interna y externa de cada una de las botellas, quedando listas para ser llenadas con producto. El proceso de lavado tarda aproximadamente una hora.

1.2.3 INSPECTORES DE BOTELLAS

Una vez que las botellas han sido lavadas, salen a las líneas de transporte en dos carrileras, las botellas deben pasar por los inspectores de botellas, estas

máquinas se encargan del rechazo de todas las botellas que presenten anomalías como: botellas rotas, botellas despostilladas, botellas con residuo de soda cáustica, botellas con residuos sólidos en su interior, botellas mal lavadas, botellas que no pertenezcan al modelo de la cervecería, etc.

Las botellas que han sido aprobadas por los inspectores de botellas continúan su trayectoria hasta las máquinas llenadoras. Cabe indicar que los inspectores de botellas trabajan en forma individual sobre cada una de las botellas.

1.2.4 LLENADORA Y TAPADORA

Las máquinas llenadoras son las encargadas de envasar cerveza en cada una de las botellas y de forma casi simultánea las botellas son tapadas herméticamente.

1.2.5 PASTEURIZADOR

El proceso de pasteurización lleva aproximadamente una hora, este proceso se lo realiza en forma continua. La pasteurización es necesaria para eliminar gérmenes residuales y posibilitan un tiempo largo para el consumo de la cerveza, una vez que las botellas terminan con el proceso en envase.

1.2.6 ETIQUETADORA E INSPECTOR DE NIVEL DE LLENADO

Existen dos máquinas etiquetadotas, (una para cada línea de transporte). Estas máquinas están destinadas a pegar las etiquetas en cada una de las botellas; para dar paso a la acción del "video jet" que se encarga de imprimir sobre la etiqueta datos como la fecha y hora de elaboración.

A continuación pasan a los Inspectores de Nivel de llenado. Estas máquinas tienen como función rechazar las botellas que no tienen su respectiva tapa, así como las botellas que no alcancen el nivel requerido de cerveza.

1.2.7 ENCAJONADORA Y PALETIZADORA

Las botellas avanzan por los transportadores hasta llegar a la mesa de carga de la encajonadora. Aquí las botellas son encarriladas en veinte y cuatro filas. Dicha formación ayuda a que doce cajas vacías sean llenadas en forma simultánea con

la ayuda de un sistema de chupones, propios de la máquina, que toma las botellas necesarias de la mesa de la encajonadora para colocarlas en las cajas.

De la misma manera como se encajonaron las botellas, las jabas son transportadas hasta la máquina paletizadora. Aquí son tomadas y apiladas hasta completar una paleta de setenta y dos cajas en cada paleta y pasan a ser guardadas en el despacho.

Todo el proceso realizado por las máquinas ya descritas es explicado en forma gráfica en la Figura 1.1

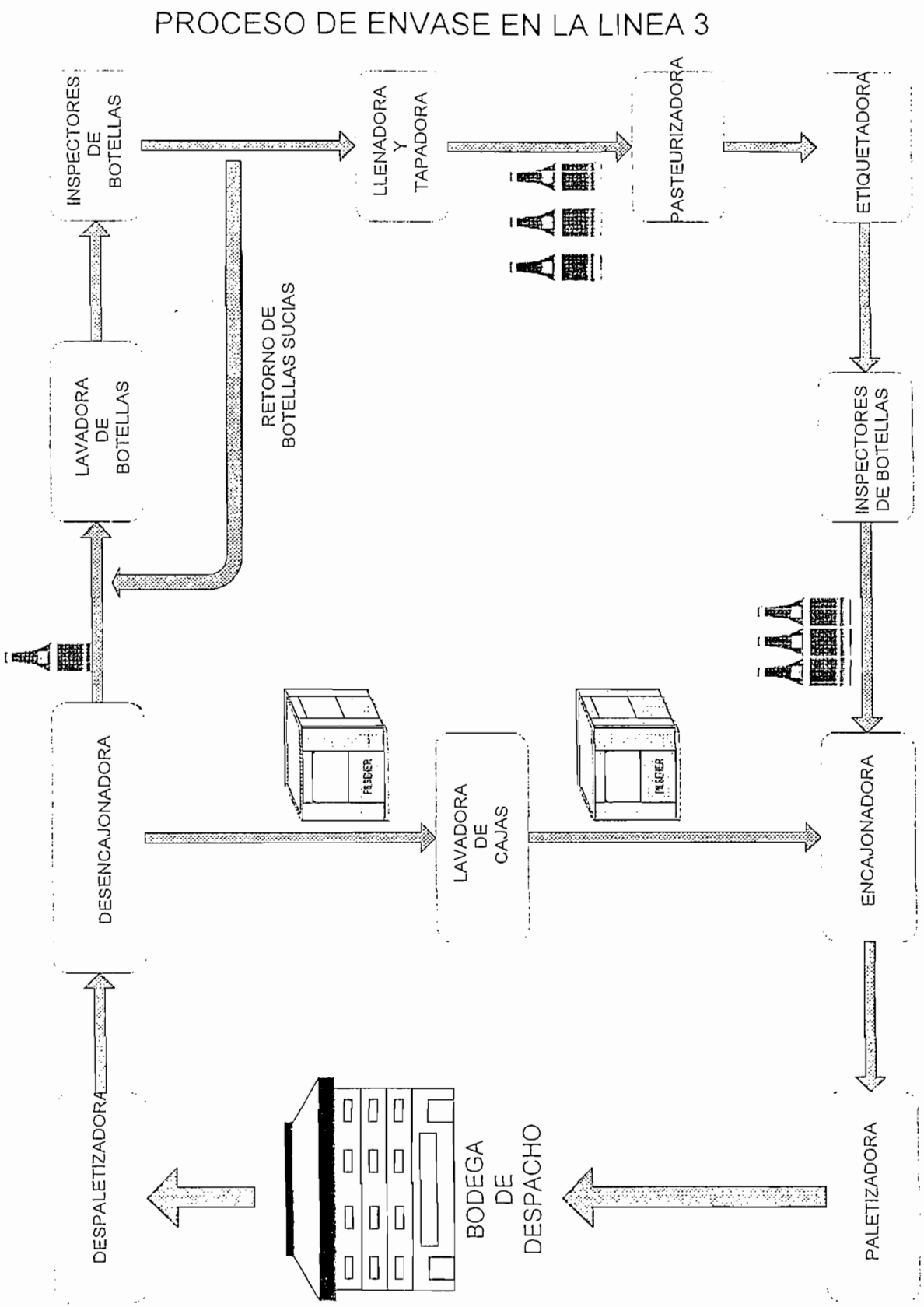


FIGURA 1.1: Proceso de embotellado de la Línea #3.

1.3 DESCRIPCIÓN DEL PROBLEMA DE CAJAS CON BOTELLAS VACÍAS

Las cajas con botellas vacías (cajas que entrega el distribuidor), que llegan a la línea de embotellado pueden presentar las siguientes "anomalías":

a) Botellas rotas.
b) Botellas con tapa.
c) Botellas con vasos encima.
d) Fundas plásticas en las cajas.
e) Botellas acostadas en las cajas.
f) Cajas con botellas faltantes.
g) Botellas de otras marcas.

Todo este tipo de anomalías se trasladan a la máquina desencajonadora. En este punto, el operario que maneja dicha máquina procura retirar las basuras y vasos que se hallan en las cajas; pero las botellas rotas por lo general no son retiradas y las botellas faltantes lógicamente ya no pueden ser suplidas ni contabilizadas.

Debido a que estas anomalías se presentan continuamente en las cajas, la Empresa ha identificado la necesidad de instalar un sistema automático o manual que permita contabilizar el número de botellas con anomalías que están llegando a la Cervecería.

1.4 CAUSAS DEL PROBLEMA

A la entrada de la sección de embotellado llegan los distribuidores con cajas con botellas vacías, las cuales vienen apiladas en grupos (paletas) de setenta y dos jabas e inmediatamente son almacenadas en la bodega de despacho.

Es por demás lógico que el estado de las botellas no puede ser inspeccionado, por la manera en como son entregadas las cajas. Entonces, para inspeccionar el estado de las botellas, es necesario que las cajas estén desagrupadas.

Por ello se pensó que el sistema debía ser implementado en la línea de producción y que pueda monitorear el estado de las botellas vacías dentro de su jaba respectiva.

1.5 CONSIDERACIONES TÉCNICAS DEL PROCESO

El sistema debe cumplir con los siguientes requisitos:

a) Implementar dos sistemas idénticos, uno para cada transportadora de cajas.
b) El sistema implementado debe registrar las botellas rotas o ausentes que se encuentren en cada una de las cajas, que permitan llevar una diaria estadística del estado en que llegan las cajas hasta la Cervecería.
c) Cada sistema debe funcionar independientemente uno del otro, tal como lo hacen las transportadoras o cualquiera de las máquinas dentro del proceso de envase.
d) Evitar crear cuellos de botella en cualquiera de las transportadoras.
e) No provocar retardos en el transcurso normal de producción, para no restar eficiencia al tiempo de embotellado.
f) El sistema implementado debe contar con todas las protecciones necesarias para evitar que los equipos utilizados sufran algún desperfecto en el momento del aseo del envase.
g) Usar todos los recursos humanos y físicos disponibles en la Cervecería que abaraten el costo final del proyecto.
h) El sistema debe cumplir con las normas y requisitos que se exige en la Cervecería.
i) El sistema que se implemente debe provocar la menor cantidad de cambios posibles dentro del proceso, es decir el sistema debe buscar adaptarse a las condiciones ya establecidas dentro del proceso de envase.
j) El sistema debe ser independiente de los transportadores para que en el futuro, frente a un cambio de transportadores, el sistema pueda continuar funcionando, requiriendo solamente algunos ajustes de calibración.

1.6 POSIBLES SOLUCIONES

La solución ideal para evitar estas anomalías sería que los distribuidores entreguen las cajas en forma óptima; es decir, que las cajas contengan doce botellas limpias correspondientes a la marca de Cervecería Andina, destapada, completamente sana y que las cajas o botellas no tengan basura en su interior.

Es poco práctico poner a operarios que se dediquen a chequear el número de botellas faltantes o rotas dentro de cada una de las cajas, no solo por el tiempo que se necesitaría para este fin, sino que se debe tomar en cuenta que cuando los distribuidores llegan a descargar las cajas, éstas se hallan apiladas en múltiples filas y columnas, lo cual, por cierto, ayuda a ocultar las cajas con anomalías antes ya descritas; además, dentro de los objetivos de la cervecería está la automatización total de todos los procesos que se llevan a cabo dentro de la línea de envase.

En la actualidad, los problemas con botellas faltantes o rotas dentro de las cajas no son tomadas en cuenta y todas las botellas sanas y rotas atraviesan los procesos de desencajonamiento y lavado de botellas, lo cual representa pérdida de tiempo y dinero al ocupar recursos en el lavado de botellas que finalmente el "Inspector de Botellas" desechará.

La solución que se pretende implementar se limita a la contabilización de botellas en buen estado frente a botellas rotas o ausentes dentro de las cajas. Se podría pensar que implementar un sistema que se limite a llevar la estadística de botellas rotas o ausentes en las cajas es un gasto innecesario; sin embargo, al implementar un sistema de inspección que permita contabilizar las botellas rotas o faltantes dentro de las cajas permitirá realizar actividades de control que tiendan a minimizar este impacto.

Además, en un futuro próximo los transportadores actuales serán cambiados por otros más modernos, en los cuales existe la posibilidad de expandir el sistema no solo para detectar sino para rechazar las cajas con los problemas ya mencionados.

Una vez que se identificó el problema existente en las jabas con botella vacía y sus causas, se analizaron los requerimientos técnicos del proceso que influyeron en el método seleccionado para solucionar el problema, las alternativas para inspeccionar problemas de cajas con botellas vacías y se describirán en el Capítulo 2.

CAPITULO 2

PLANTEAMIENTO Y VALORACIÓN DE MÉTODOS PARA SOLUCIONAR EL PROBLEMA

En este Capítulo se valoran los posibles métodos para inspeccionar los problemas con botellas vacías descritos en el Capítulo 1. Se probó métodos típicos como el uso de sensores convencionales (sensores fotoeléctricos, etc.) que no dieron el resultado deseado, pero también se analizó el uso de métodos más sofisticados de adquisición de imágenes que permitan identificar las anomalías antes descritas. Al final del capítulo se realiza una comparación entre los métodos propuestos y se escoge la mejor alternativa en base a aspectos tecnológicos, económicos y de fidelidad.

Para resolver el problema de botella rota y ausente en las cajas se deben tomar en cuenta dos puntos clave. El primero consiste en notar que las botellas están vacías y por ende destapadas; en segundo lugar, que las botellas llegan dentro de cajas. Al tratarse de botellas destapadas, desde un principio se tuvo que descartar el uso de algún sensor inductivo; por otro lado, el nivel del pico de las botellas queda muy por debajo de la altura de la jaba, exigiendo como requisito algún sensor con un alcance de por lo menos 8cm.

Para justificar suficientemente la selección final hecha, a continuación se analizan los sensores fotoeléctricos, por su versatilidad en cuanto a modos de sensado y a su amplio alcance.

2.1 SENSORES FOTOELÉCTRICOS

Para determinar si el uso de sensores fotoeléctricos es la tecnología adecuada para el reconocimiento de botellas rotas y ausentes dentro de sus respectivas cajas, se empezó con el estudio de sus características más relevantes.

2.1.1 CARACTERÍSTICAS

Se tomó en cuenta 2 formas de sensado de los sensores fotoeléctricos, Modo Difuso y Retroreflectivo. El modo difuso es aquel en el que el sensor emite un haz de luz, el cual es retornado solo en caso de que un objeto (objeto a sensar) interfiera con el haz de luz; en cambio, en el modo retroreflectivo, el haz de luz es emitido e impacta en un espejo que retorna la señal, pero cuando un objeto interrumpe el haz de luz ya no existe retorno de la señal. [5]

Todos los sensores fotoeléctricos funcionan bajo el mismo principio: existe una fuente de luz emitida por un diodo que es capturada por otro diodo, la luz emitida por los diodos puede ser visible e incluso invisible, por ejemplo infrarroja. [5]

La luz visible de los leds se usa para indicar el estado del sensor, y en aplicaciones reflectivas para alinear el sensor con el reflector. La intensidad de luz de los leds infrarrojos es más grande que la de los leds de luz visible. Esta característica los hace mejores para aplicaciones difusas. [5]

Por otro lado, los sensores fotoeléctricos suelen tener protecciones contra luces externas que causen interferencias y puedan dar señales erróneas. [5]

El receptor es un circuito generalmente hecho a base de un fototransistor, el cual es sensible a la misma frecuencia de la luz que envía el emisor. Los pulsos de luz generados por el fotodiodo son recibidos por el fototransistor y convertidos en señales eléctricas. Estas señales son analizadas para determinar si, estas son el resultado de la luz transmitida. [5]

Los sensores fotoeléctricos proporcionan un bajo porcentaje de lecturas erróneas, debido a que el circuito receptor esta encendido solo en los ciclos de encendido del emisor, entonces hay muy poco tiempo para que una luz errónea de una señal falsa. [5]

2.1.2 CARACTERÍSTICAS DEL SENSOR PROBADO

El sensor que se seleccionó para esta prueba fue provisto por Cervecería Andina y las características más importantes se indican en la Figura 2.1

Marca	SICK
Serie	Compacta
Código	WT-250-S162
Voltaje	10-240 Vdc/24-240Vac
Potencia	< 2 VA
Salida	Relé
Alcance	1-60 cm
Protección	IP 67
f. switcheo	33/s

FIGURA 2.1: Características del Sensor Fotoeléctrico.[1]

Se debe tomar en cuenta que el objetivo del proyecto es detectar ausencia/presencia de botellas y además detectar botellas con pico roto. Esto quiere decir que hay que calibrar al sensor de forma tal que pueda ver solamente la parte superior del pico de la botella y en caso de que la botella no lo tenga, entonces el sensor debe notar esta diferencia.

En la Figura 2.2 se puede ver las curvas bajo las cuales trabaja el sensor. La curva #1 se aplica cuando el objeto a sensar es negro y el fondo es blanco, la curva #2 se aplica a objetos grises con fondo blanco y la curva #3 se aplica a objetos con rasgo blanco y fondo blanco. [6]

[1] SENSICK, Sensors for Automation Catalog

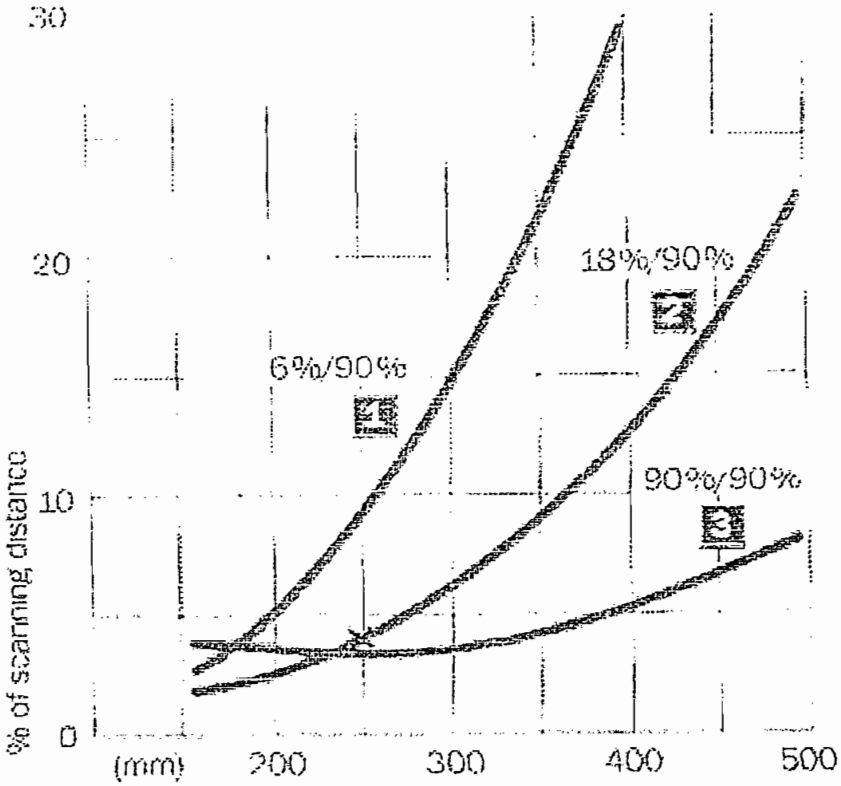


FIGURA 2.2: Curvas de trabajo del sensor fotoeléctrico distancia objeto/sensor (mm.) vs. % de escaneo del sensor.[2]

De acuerdo a la necesidad del sistema se ubicó el punto de trabajo en la curva #2 con la marca de color rojo tal como se ve en la Figura 2.2, considerando que el color de la botella es café y se apoya en cajas color amarillo.

Se realizaron algunas pruebas con los sensores fotoeléctricos, con el fin de determinar su funcionalidad y posibles desventajas. Estas pruebas se describen a continuación.

2.1.3 PRUEBA DE SENSORES SOBRE LAS CAJAS

Este método era el más obvio y se lo puede ver en la Figura 2.3. Puesto que en la caja se cuenta con tres filas, se dispusieron a una altura aproximada de 20cm, sobre la caja, tres sensores centrados sobre cada una de las filas, con la sensibilidad y ajuste apropiados. Dispuestos de esta manera se procedió a desarrollar un método de sensado en conjunto; es decir, para comprobar que

[2] SENSICK, Sensors for Automation Catalog

cada caja esté completa se decidió que cada uno de los sensores debía contar 6 pulsos correspondientes a las cuatro botellas y a los dos bordes de la jaba, que el sensor ve pasar.

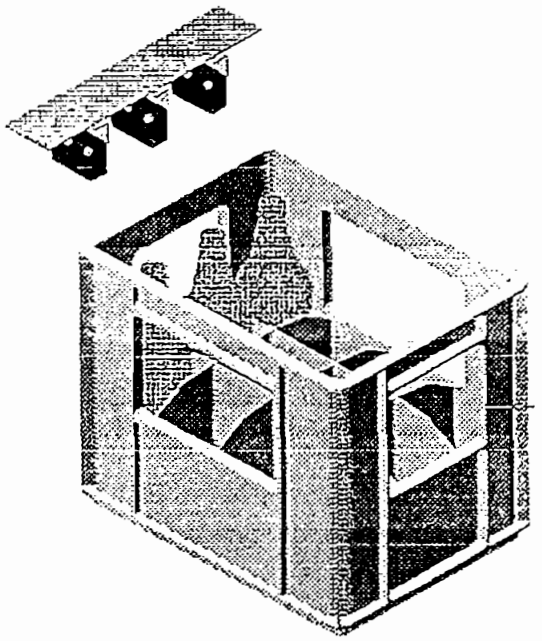


FIGURA 2.3: Sensores fotoeléctricos ubicados sobre las cajas.

2.1.3.1 Resultados obtenidos con sensores sobre las cajas

a) Con este método de sensado sobre las botellas se consiguió ver ausencia y presencia de botellas; sin embargo, no se pudo ver botellas con pico roto.
b) Los sensores no respondieron cuando la jaba pasa rápidamente y en lugar de contar 6 pulsos, correspondientes a los dos bordes de la jaba y a cuatro de los picos de la botella, apenas contaba 4 o 5 pulsos. Se determinó que la confusión se presentaba al momento que pasa el borde de la jaba y la botella contigua, por la velocidad solo se percibía un pulso. Esto se debía a que el cono de sensado es amplio con el se cubre tanto la botella así como el borde de la jaba.
c) Se observó que este método era sensible a las vibraciones, por lo que se requería calibrarlos continuamente.
d) Por otro lado, se notó que existen anormalidades típicas en las cajas, por ejemplo las cajas que vienen en muchas ocasiones rotas, descuadradas,

etc., dando como resultado que las botellas estén viradas y se escapen del rango de sensado del sensor.

2.1.4 PRUEBA DE SENSORES Y LENGÜETAS DE CAUCHO

Este método es de sensado indirecto; es decir, en lugar de que el sensor cuente los picos de las botellas, contará el número de veces que la lengüeta se deflexione. Con esto se intenta lograr que la lengüeta tome en cuenta cada uno de los picos de las botellas así como los bordes de las jabas. Además, para lograr esto, la lengüeta se hizo con una medida tal que al encontrar una botella sin pico; ésta no se deflexione.

Con esta disposición física el sensor tiene que contar 6 pulsos en caso de que la fila que le corresponda esté completa, e irá disminuyendo el número de pulsos si la caja esta incompleta o con botella rota.

En la Figura 2.4 se puede apreciar la forma en que se dispusieron los sensores y las lengüetas.

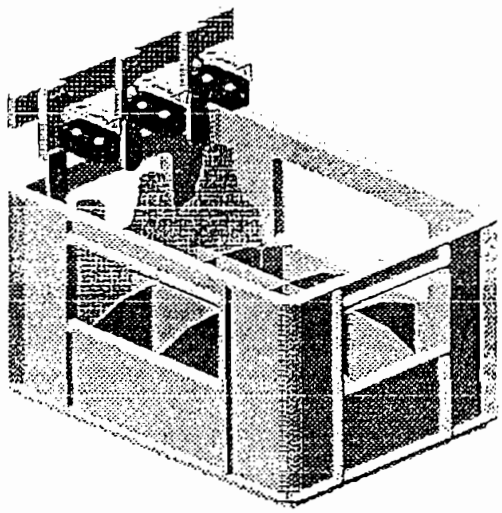


FIGURA 2.4: Sensores Fotoeléctricos y lengüetas de caucho sobre las cajas.

2.1.4.1 Resultados obtenidos con el método de las lengüetas

a) Al usar lengüetas de color negro no se consiguió buena respuesta de los sensores, ya que no reflejan bien la luz del sensor que incide sobre estas.

Fue necesario recubrir las lengüetas con adhesivos blancos para probar el método.

b) La lengüeta se dobla aún cuando la botella esté rota siempre cuando la botella tiene algún resto de pico.
c) Con el método de la lengüeta el sensor es capaz de responder a la presencia de cada una de las botellas dentro de la caja y aún hacer una pequeña diferencia entre el borde de la caja y las botellas contiguas, siempre y cuando la caja transite a una velocidad moderada.
d) El sensor sigue siendo sensible a vibraciones.
e) Se pudo observar que las lengüetas con el pasar del tiempo y el trabajo continuo, terminan por no volver a su estado original (vertical), dando como resultado la obstrucción permanente de la visión del sensor con lo cual este método exige de observación continua para corregir este problema.
f) Cuando la caja transita muy rápido se vuelve al problema del método anterior; es decir, la lengüeta no es capaz de hacer dos deflexiones en el momento que pasa el borde la caja y la botella contigua, en el 100% de las veces.

2.2 SENSORES ULTRASÓNICOS

Se consideró el uso de sensores de sonar por la alta eficiencia que ofrecen, los altos rangos de distancia a la que son capaces de sensar y, además, porque dichos sensores estaban disponibles en la Cervecería.

2.2.1 PRINCIPIOS DE FUNCIONAMIENTO DEL SENSOR ULTRASÓNICO

Los sensores ultrasónicos son usados en aplicaciones de monitoreo en donde el sensor no puede estar en contacto con el objeto sensado. Se usa también en control de nivel, monitoreo de flujo y en bandas transportadoras. [5]

Los sensores ultrasónicos emiten y miden el tiempo de recorrido de un pulso de sonido, para calcular la distancia a la que está el objeto en el que incidió el pulso de sonido. [5]

Los pulsos de sonido son emitidos y recibidos por un diafragma en la cara del transductor tal como se ve en la Figura 2.5.

FIGURA 2.5: Pulso de sonido emitido por Sensor Ultrasónico e incidente en un objeto. [8]

A continuación en la Figura 2.6 se puede ver las zonas de detección de un sensor ultrasónico.

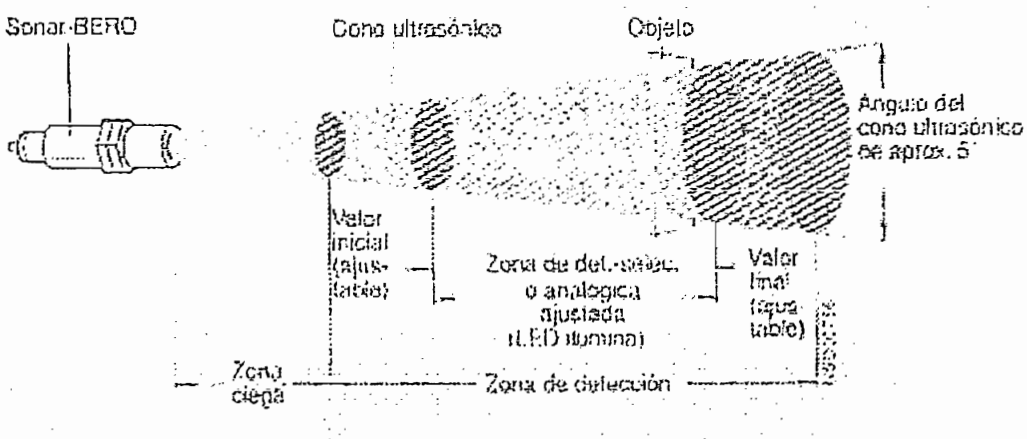


FIGURA 2.6: Zonas de detección de un Sensor Ultrasónico. [8]

ANGULO DE CONO ULTRASÓNICO: se define así al área en la cual un objeto puede ser detectado. [5]

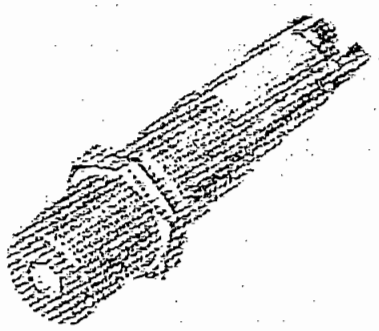
ZONA CIEGA: es la región no detectable por el sensor y define la mínima distancia para detección del objeto. [5]

El buen resultado que pueda arrojar un sensor ultrasónico depende en gran porcentaje del tipo de material que se esté detectando considerándose que un objeto es bueno para ser detectado si tiene una alta densidad ya sea el objeto sólido, líquido o granular. [5]

Los objetos como madera, ropa tienen muy alta absorción del sonido es por ello que con este tipo de material es inapropiado usar sensores de sonar. Además, aunque los líquidos son óptimos para el uso de sensores de sonar, pueden llegar a ser indetectables si la superficie del líquido se halla cubierta de burbujas. Otro factor que hace inapropiado el uso de sensores ultrasónicos es la temperatura a la que se encuentran los objetos a ser sensados, la cual no debe superar los 100 grados centígrados. [5]

2.2.2 CARACTERÍSTICAS DEL SENSOR PROBADO

Las características del sensor ultrasónico utilizado se indican en la Figura 2.7



Marca	SIEMENS
Serie	SONAR-BERO
Código	3RG6012-3AD00
Voltaje	20-30 Vdc
Corriente	< 50mA sin carga
Salida	PNP-NPN
Alcance	6-30 cm
Protección	IP 65
f conmutación	8 Hz
F ultrasónica	400 Khz
Histéresis	1 cm

FIGURA 2.7: Características del Sensor Ultrasónico.[3]

Los sensores utilizados ofrecen la ventaja de dos tipos de conexión, NA y NC para el uso a conveniencia del usuario tal como se ve en la Figura 2.8.

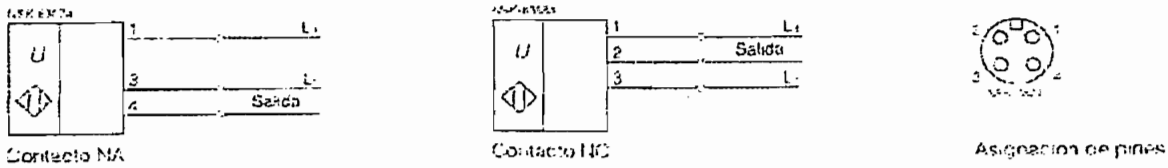


FIGURA 2.8: Tipo de conexiones del Sensor Ultrasónico. [8]

[3] SIEMENS, Control y Distribución Catalogo

2.2.3 PRUEBAS DE SENSORES ULTRASÓNICOS SOBRE LA CAJA

Se realizó la siguiente prueba con los sensores ultrasónicos:

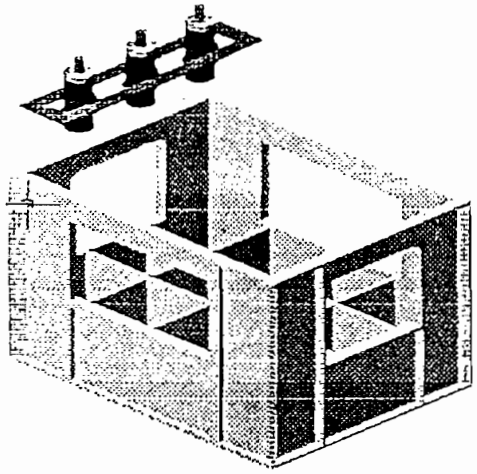


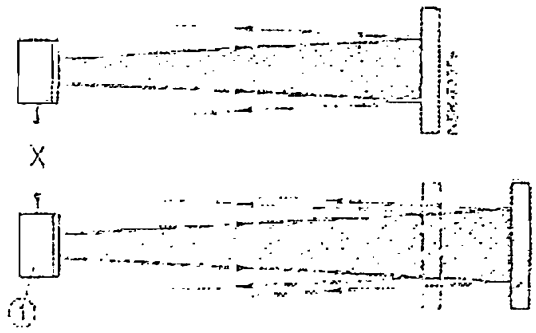
FIGURA 2.9: Sensores Ultrasónicos sobre la caja.

Se dispuso tres sensores ultrasónicos sobre la caja, tal como se muestra en la Figura 2.9. Se ubicaron tres sensores ultrasónicos sobre la caja, a distancias entre 20 y 30cm, distanciados entre sí 8cm, de manera que estén centrados sobre cada una de las filas de la caja.

De la misma forma que con los sensores fotoeléctricos, cada sensor debería contar 6 pulsos en caso de que la caja esté completa.

Se puede prever que con este tipo de sensado solamente se podía detectar presencia y ausencia de botellas en las cajas. Sin embargo la idea primordial era hallar un método que posibilite al sensor detectar los bordes de la caja así como las botellas, siempre y cuando la botella esté completa. Si a una botella le falta una parte del pico entonces el sensor ya no puede detectar esa anomalía.

Al realizar las pruebas se pudo observar que la distancia (X) que se puede apreciar en la Figura 2.10, es la mínima distancia requerida entre sensores para el correcto funcionamiento de los mismos. Para este tipo de sensor esta distancia era de 15cm. Puesto que las dimensiones de la caja no lo permiten se previó el cruce de conos de sensado, aún a costa de originar señales falsas.



FIRURA 2.10: Mínima distancia requerida entre sensores ultrasónicos para su correcto funcionamiento. [8]

2.2.3.1 Resultados obtenidos con los sensores ultrasónicos

a) Los sensores exhibieron un buen alcance lo que permitió ubicarlos a cualquier distancia; pero se debe tomar en cuenta que mientras la distancia entre la botella y el sensor es mayor, los conos de sensado tienen mas posibilidad de chocar entre sí dando como resultado pulsos erróneos.
b) Aún cuando son de fácil calibración, para la aplicación que se desea fue muy difícil conseguir la distancia correcta requerida. Por un lado la sensibilidad debe ser la apropiada para que sense los picos de la botella y a la vez deje pasar por alto las botellas que no tengan pico.
c) Resultaron ser muy sensibles a vibraciones.
d) Tienen una curva de histéresis muy grande, causando que el rango de detección sea más grande de lo que se requería; es decir, no permitió tener precisión en el sensado.
e) Debido al gran cono de sensado, que iba aumentando proporcionalmente con la distancia a la que se ubicaba el sensor, fue difícil conseguir el conteo de los 6 pulsos por jaba, ya que el sensor no dejaba distinguir el borde de la jaba y la botella contigua.

2.3 ANÁLISIS DIGITAL DE IMÁGENES

Los sistemas que se describen a continuación funcionan mediante la adquisición digital de imágenes. Dichas imágenes son adquiridas con cámaras digitales; para

luego, con ayuda de software especializado, almacenarlas y procesarlas hasta obtener los resultados buscados. Se consideró el uso de dos tipos de sistemas que se describen a continuación.

2.3.1 ANÁLISIS COMPUTARIZADO CON EL SOFTWARE DE LABVIEW

Este sistema se basa en la captura de imágenes por medio de una cámara digital, con características apropiadas, imágenes que luego son sometidas a procesamiento mediante el IMAQ de LabView.

Con ayuda del IMAQ se almacena en la computadora imágenes patrón. El principio de funcionamiento consiste en comparar cada una de las imágenes capturadas con las imágenes patrón para llegar a un resultado que indique si las imágenes capturadas tienen alguna relación con las imágenes patrónes almacenadas. En caso que las imágenes capturadas no coincidan con las imágenes patrón el veredicto será de imagen incorrecta. [9]

Adicionalmente, se requiere del diseño de un circuito que, dependiendo de si la imagen analizada es correcta o incorrecta, pueda transformar el resultado en señales eléctricas de suficiente voltaje para realizar alguna acción.

El sistema demostró tener las desventajas que se señalan a continuación:

a) El sistema requiere el uso de una computadora de alta eficiencia y con sistema operativo compatible con el software de LabView.
b) El sistema operativo debe ser estable para que el sistema de adquisición de imágenes funcione correctamente en tiempo real y la PC no se cuelgue mientras esta en ejecución el programa inspector.
c) La computadora requiere de gran capacidad de memoria.
d) Se debe usar cámaras compatibles con el software de LabView.

2.3.2 CARACTERÍSTICAS DE LOS EQUIPOS DE INSPECCIÓN VISUAL DVT

DVT es un sistema de visión artificial muy completo, el cual se basa en la adquisición de imágenes mediante una cámara propia del sistema, para procesar y analizar la imagen con ayuda del procesador interno que posee dicha cámara.

Los sistemas de visión DVT tienen sus propios implementos como cámara, lentes, tarjetas de entrada/salida, cables de poder, cables para comunicación, etc. Un ejemplo de las cámaras que utiliza este sistema se ve en la Figura 2.11 y tienen las siguientes características:

FIGURA 2.11: Cámara digital 530MR.

a) Son usadas a nivel industrial para realizar control de calidad de los productos. [1]
b) Son de tamaño compacto. [1]
c) No requieren de PC o procesador adicional. [1]
d) Luz integrada opcional. [1]
e) Usan comunicación vía ETHERNET. [1]
f) Gracias a la comunicación ETHERNET pueden tener conectividad desde cualquier punto de red local dentro de la planta. [1]
g) Alta velocidad de adquisición de imágenes. [1]
h) Trabaja en tiempo real. [1]
i) Se puede realizar diagnósticos en línea. [1]
j) Se puede realizar pantallas amigables para el usuario. [1]
k) Resolución de hasta 1280x640 píxeles. [1]

l) Existen cámaras con capacidad de adquisición de imágenes en color y en escala de grises. [1]
m) Tienen herramientas capaces de medir, contar, hallar características, comparar, leer caracteres en 1D y 2D y muchas otras opciones más. [1]
n) Las herramientas que usa la cámara son virtuales y hacen la función de sensores ubicadas sobre la imagen adquirida. [1]

Por las características señaladas se puede ver su versatilidad, por ejemplo, se puede ver presencia/ausencia de botellas, así como detectar picos rotos, además el sistema cuenta con múltiples herramientas de programación de fácil aprendizaje para el análisis de imágenes, así como para realizar interfaces para el usuario. Algunas ventajas y desventajas que ofrece el sistema frente a las alternativas mencionadas anteriormente son:

Ventajas:

a) No requiere de una PC o procesador adicional a la cámara.
b) Alta velocidad de trabajo en tiempo real.
c) Permite el diagnóstico en línea.
d) Se puede configurar los parámetros desde cualquier punto de la línea gracias a la conectividad ETHERNET con el software FrameWork.
e) El software Framework esta basado en Windows, el cual es un sistema ya conocido por todos.
f) Se usa en fábricas para llevar a cabo el control de calidad de los productos.
g) Es un sistema que a corto plazo reditúa la inversión inicial.
h) Es de tamaño compacto y puede acceder a cualquier sitio.
i) Es un sistema que se puede adaptar a cualquier aplicación, solo requiere crear nuevos programas de inspección.

Desventajas:

a) Es un sistema que requiere de una alta inversión.
b) Necesita de personal que tenga conocimientos de programación y cableado estructurado para la utilización del sistema.
c) El manejo de la cámara debe ser cuidadoso para no afectarla.

d) El uso de las herramientas virtuales necesita de pruebas y continuas modificaciones hasta hallar los resultados deseados.

2.4 SELECCIÓN DE LA MEJOR ALTERNATIVA

Para llegar a la mejor alternativa se consideraron aspectos económicos y técnicos de cada uno de los sistemas y a cada característica se le dio una valoración de acuerdo a criterios empleados en Cervecería Andina.

- **Precio:** es un aspecto importante pero no decisivo en el momento de la elección del sistema, por ello se le da una valoración máxima de un 10%.
- **Velocidad:** es un aspecto técnico que merece una valoración del 15%, ya que se requiere de un sistema capaz de trabajar a la velocidad que lo hace la línea de embotellado para no crear cuellos de botella.
- **Confiabilidad:** es el aspecto de mayor importancia para Cervecería Andina ya que los resultados deben ser verídicos para que las acciones de control a tomar sean las adecuadas. Por este motivo se le da una valoración de 30%.
- **Pico roto:** este aspecto es indispensable para el buen funcionamiento del sistema, ya que se debe reconocer si la botella dentro de la caja tiene o no el pico completo. Su valoración es del 15%.
- **Botella Ausente:** también resulta esencial reconocer presencia o ausencia de botellas, ya que la ausencia de botella representa pérdidas para Cervecería Andina, además es un requisito que el proyecto debe cumplir; por ello tiene una valoración de 15%.
- **Independencia:** se desea un sistema autónomo que no requiera de elementos adicionales para su óptimo funcionamiento y por ende que no encarezca el costo del sistema ni la complejidad del mismo. Por ello tiene una valoración de 15%.

A continuación en la Tabla 2.1 se presenta un cuadro comparativo entre los sistemas mencionados con sus respectivas valoraciones:

Sistema	CARACTERÍSTICAS						
	Precio 10%	Velocidad 15%	Confiabilidad 30%	Pico roto 15%	Ausencia/Presencia 15%	Independencia 15%	Total 100%
Sensores fotoeléctricos	2%	10%	15%	0%	15%	7%	49%
Sensores fotoeléctricos y lengüetas	2%	9%	12%	0%	10%	7%	40%
Sensores ultrasónicos	3%	5%	10%	0%	15%	7%	40%
LabView	9%	12%	25%	15%	15%	0%	76%
DVT	10%	12%	28%	15%	15%	15%	95%

TABLA 2.1: Cuadro comparativo de los métodos analizados para solucionar el problema de cajas con botellas rotas o faltantes.

Se puede ver de acuerdo a la Tabla 2.1 que las mejores opciones para la aplicación que se requiere son los sistemas LabView y DVT; puesto que el uso de sensores fotoeléctricos y ultrasónicos solamente permiten detectar presencia y ausencia de botellas, pero no son capaces de sensar botellas con pico roto.

Sin embargo usar el sistema de adquisición de imágenes con ayuda de LabView implica la utilización de una computadora de alta eficiencia y buenas características de memoria además de velocidad de procesamiento. También se debe recordar que para cualquier aplicación se necesita entradas y salidas adicionales para el manejo de sensores trigger, señales indicadoras e incluso señales para movimiento de transportadoras, con lo cual el sistema tiene la necesidad de usar circuitos o un PLC adicional para el manejo de dichos elementos.

Esta situación da como resultado el encarecimiento del sistema, así como el uso - de mayor número de elementos para implementar este sistema.

El sistema DVT ofrece grandes ventajas en cuanto a versatilidad de aplicaciones; es decir, si en otro sector del embotellado se requiere hacer otro tipo de análisis; por ejemplo, verificar si las botellas están tapadas, entonces el sistema podría reinstalarse en el lugar adecuado y hacer un nuevo programa de análisis que determine si las botellas están siendo tapadas correctamente.

Para que el sistema DVT funcione correctamente no requiere de un computador, el computador es usado solamente para hacer el programa de análisis de imágenes y transmitirlo a la cámara. También se lo usa para visualizar por medio de un programa "amigable con el usuario" las imágenes adquiridas por la cámara, así como los registros de imágenes correctas e incorrectas que la cámara este hallando

Por estos motivos, se consideró al sistema DVT como la opción adecuada para el objetivo de este proyecto; y, aún cuando el sistema es caro, el uso del mismo

ayudará al control de calidad de productos lo cual es de suma importancia dentro de la planta.

Una vez que se valoró las alternativas propuestas para realizar el proyecto de acuerdo a los Requerimientos de Cervecería Andina y se escogió un método, se pudo empezar a analizar y describir todos los componentes que se debieron incluir para implementar el sistema y que se detallan en el Capítulo 3.

CAPITULO 3

DISEÑO E IMPLEMENTACIÓN DEL HARDWARE

En el presente Capítulo se detallan los pasos necesarios para la implementación del sistema dentro de la línea #3 de embotellado, así como la descripción de los equipos y elementos que fueron usados. Aquí se describen los requerimientos físicos del sistema de acuerdo al método seleccionado y al que se hizo referencia en el Capítulo 2, también se esquematiza la ubicación física del sistema dentro de la planta, así como la interconexión entre los elementos que intervinieron en el sistema de adquisición de imágenes.

3.1 UBICACIÓN DEL SISTEMA DENTRO DEL ENVASE

La selección de un sitio dentro del envase para ubicar el sistema fue una tarea complicada por cuanto existen muchos impedimentos. Por ejemplo, a lo largo de la línea de transporte desde la despaletizadora hasta la desencajonadora, no existe un lugar natural de separación entre las cajas; dicha separación es indispensable para la adquisición de imágenes. La sección de entrada en la mayor parte del tramo es inclinada, provocando acumulación de cajas en varios sectores.

Para eliminar esta aglomeración y asegurar la separación entre las cajas, se probó con frenos dinámicos ubicados en puntos estratégicos de la transportadora; sin embargo, la presión de las cajas que vienen una tras de otra por transportadoras de rodillos locos e inclinados eliminó cualquier posibilidad del uso de dichos frenos.

Otro inconveniente que se tuvo es que otros sectores que pudiesen haber sido escogidos para la instalación del sistema son muy sucios y la cámara digital habría sido expuesta a daños que la perjudiquen. Adicionalmente, el sitio a seleccionar debía poseer cierto rango de amplitud que permita la instalación así como mantenimiento del sistema.

Esto implicó buscar un lugar con mucho espacio libre sobre la transportadora. Es importante notar esta situación ya que por lo menos una cuarta parte de la sección de transporte de entrada de cajas tiene ubicada sobre sí a la sección de transporte de salida de cajas; es decir, dichas partes no favorecían para la instalación del sistema.

En la Figura 3.1 se puede ver la línea de envase #3 y el lugar específico en donde se instaló el sistema de inspección visual. Aquí se puede apreciar que el sistema fue instalado a la salida de la máquina despaletizadora. Para la instalación se requirió de cambios físicos en la transportadora. Para asegurar la separación de cajas se cambió una sección de la línea de transporte para que tenga sus propios motoreductores, sistema de bandas, rodillos ejes, etc. Todos estos puntos de detallan más adelante en el Subcapítulo 3.3.

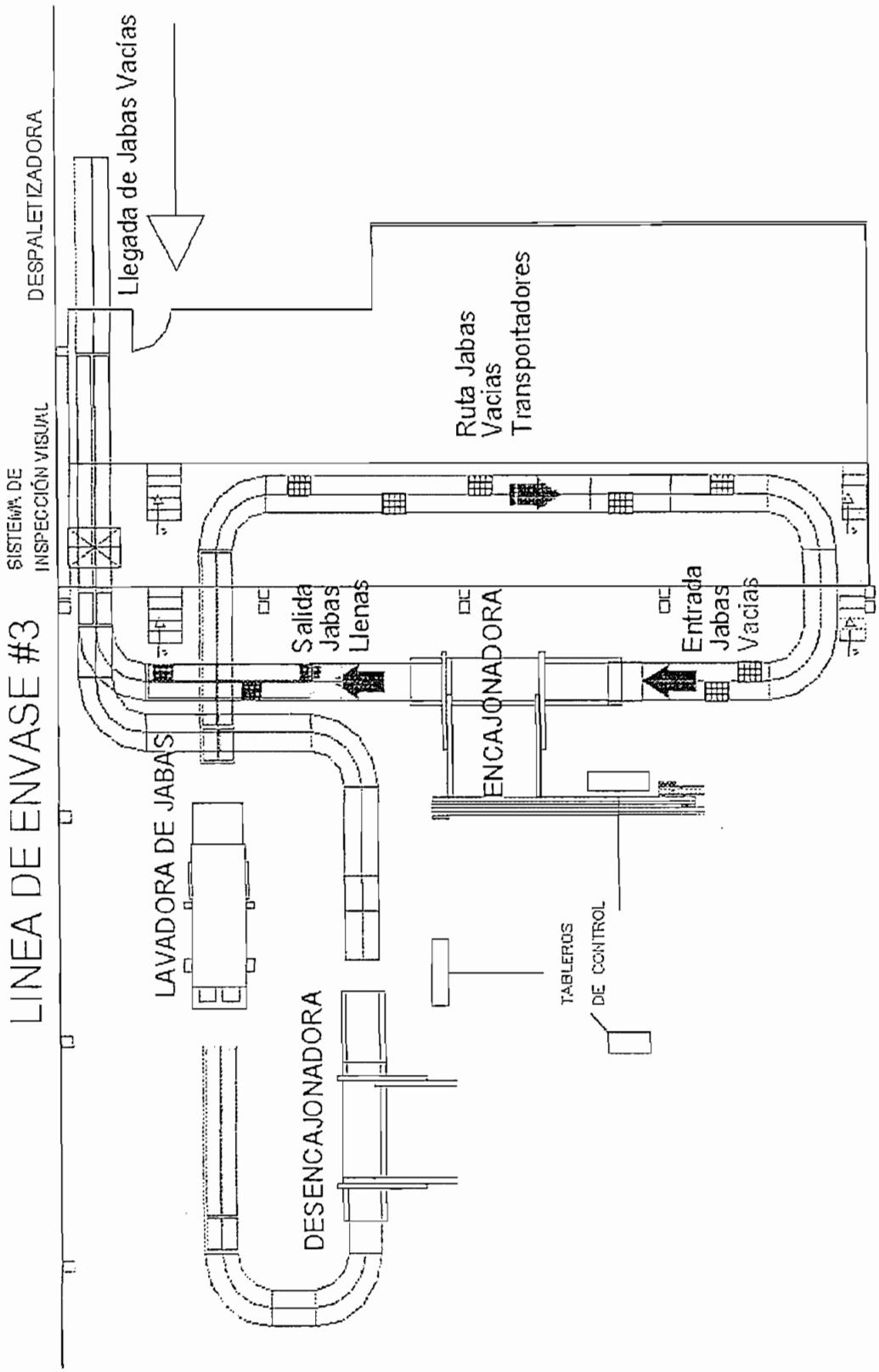


FIGURA 3.1: Ubicación del sistema de inspección visual.

3.2 EQUIPO DE INSPECCIÓN VISUAL DVT

Este sofisticado sistema de inspección visual posee las características que se indican en la Tabla 3.1:

CAMARA 530MR

Procesador	Motorola
Memoria	16Mb de RAM y 4Mb Flash
Tamaño	112mm x 60mm x 34mm (sin incluir lentes)
Montaje	Tornillos M4, 7.8mmm profundidad
Peso	170g (sin lentes o luz integrada)
Potencia	24Vdc, 300mA. Adicional 5W para luz integrada
Temp. de operación	0-45 °C
Sensor de Imagen	4.8mm x 3.6mm (formato 1/3") CCD
Resolución	640x480 píxeles
Cuadros por seg.	75
I/O digitales	24Vdc, 8 entradas/salidas configurables
Certificación	CE certificada
Color	Blanco y negro
Iluminación	Integrada
Puerto Externo	RJ-45 para 10/100Mbps comunicación Ethernet , conector de 10 pines para I/O digitales

TABLA 3.1: Características de la cámara 530MR.[4]

[4] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario

El principio de funcionamiento de la cámara 530MR es capturar y analizar la imagen en tiempo real. Puede almacenar dichas imágenes, así como los resultados de los análisis, dentro de la memoria flash que posee; además, puede transmitir los datos hacia cualquier PC que se conecte con la cámara, por medio del protocolo de comunicación Ethernet.

3.2.1 CARACTERÍSTICAS DE LOS LENTES

En adición a la cámara digital 530MR, es necesario el uso de lentes que permitan visualizar con mayor definición a un determinado objeto.

Para la aplicación deseada se seleccionaron en primera instancia los lentes LTC-25F que se muestran en la Figura 3.2. Estos lentes cuentan con dos tornillos de regulación, el primero permite la disminución o aumento de luz para la imagen; y el segundo tornillo da mayor nitidez y definición a la imagen que se desea capturar. [1]

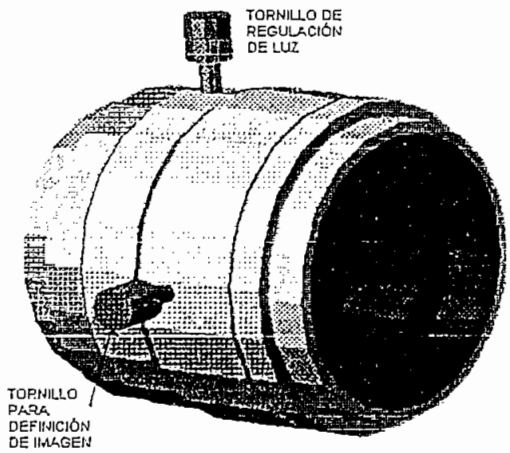


FIGURA 3.2 Lentes LTC-25F.

Una desventaja de la utilización de este lente es la carencia de un tornillo de regulación que permita controlar el zoom, con lo cual se presenta un gran limitante en cuanto a la distancia requerida entre la caja y la cámara; mientras más grande es el objeto que se desea analizar, mayor es la distancia que se requiere entre el lente y dicho objeto.

A continuación en la Figura 3.3 se observa la foto de una caja con botellas vacías, que se adquirió usando el lente LTC-25F. Aquí se ve una caja con botellas faltantes, botellas rotas y botellas en buen estado. Por tanto el uso de este tipo de lente representa un inconveniente físico (altura de la cámara respecto a la caja) y no un limitante en cuanto a identificación de fallas por parte de la cámara. [1]

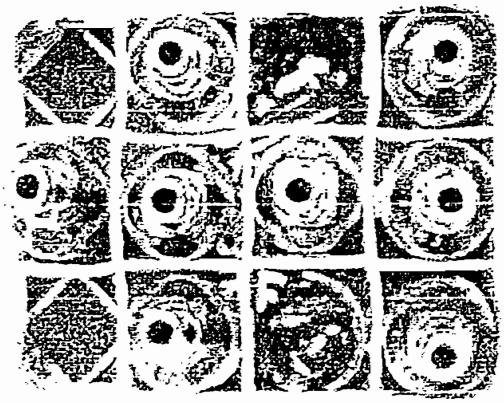


FIGURA 3.3: Imagen obtenida con lente LTC-25F (lente sin zoom).

La imagen de la Figura 3.3 fue tomada a una altura aproximada de 2.30m sobre la caja, y aunque la distancia es extremadamente grande las imágenes obtenidas tienen buena definición, la suficiente como para que el software que utiliza la cámara 530MR sea capaz de identificar las anomalías ya mencionadas. Debido a que el lente no tiene zoom, se puede ver en la Figura 3.3 la similitud entre cada una de las botellas (botellas del centro y esquineras).

El otro tipo de lente que se seleccionó fueron los de la serie LTC-05Z los cuales poseen zoom, una perilla de regulación de luz y una perilla de regulación de la nitidez de la imagen. El uso de un lente que tiene zoom da como resultado una imagen como la que se muestra en la Figura 3.4, dicha foto se adquirió a una altura de 70cm sobre la caja aproximadamente.

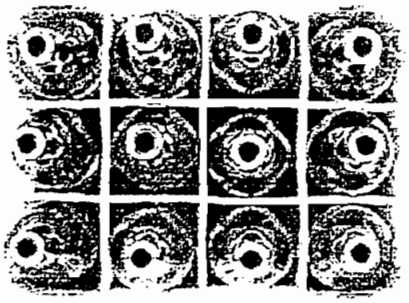


FIGURA 3.4: Imagen obtenida con cámara que tiene zoom LTZ-05Z.

La imagen de la Figura 3.4 permite ver que las botellas en los extremos de la caja, aparecen inclinadas...Esto se debe al uso del zoom, que en definitiva abre la imagen adquirida.

Después de re-evaluar este resultado se pensó que las diferencias entre las dos imágenes adquiridas repercutirían en la programación usada para identificar las anomalías que se hallen dentro de las cajas, y en la falta de estética que el sistema adquiriría dentro de la línea de envase.

Finalmente se escogió el lente con zoom LTC-05Z que permitió al sistema estar mas cerca de la jabas ayudando a la estética del sistema y bajar el nivel de interferencia que produce la luz ambiental que rodea al sistema favoreciendo a la programación de las cámaras. La imagen del lente LTC-05Z se muestra en la Figura 3.5.

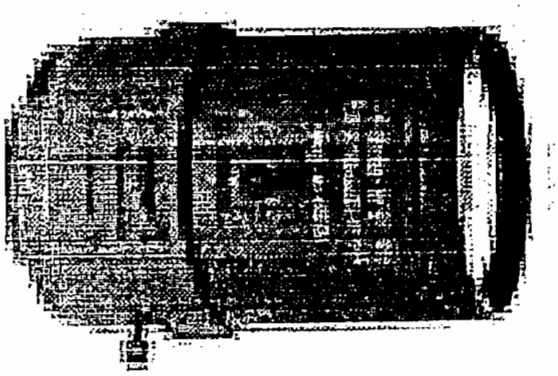


FIGURA 3.5: Lentes LTC-05Z.

3.2.2 CARACTERÍSTICAS DE LA TARJETA DE ENTRADAS/SALIDAS (CON-IBOB)

La cámara 530MR por si sola realiza funciones de adquisición y análisis de imágenes, así como archivo de resultados, pero a parte de su puerto de Ethernet que se halla integrado en la cámara, ella no puede conectarse directamente a una fuente de energía para su funcionamiento, manejar dispositivos actuadores o recibir señales de sensores externos. [1]

Para completar las múltiples funciones que posee la cámara, existe una tarjeta de 8 entradas/salidas configurables que funcionan a 24Vdc y se la puede ver en la Figura 3.6.

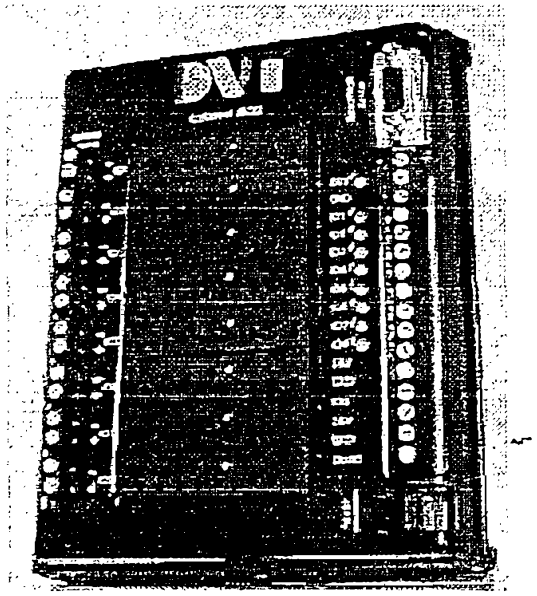


FIGURA 3.6: Tarjeta de entradas/salidas (CON-IBOB).[5]

Las 8 entradas/salidas son configurables y cada pin puede ser configurado como entrada o salida, según le convenga al usuario. Con ayuda de esta tarjeta se pueden conectar varios dispositivos como:

- **Sensores trigger**: que den una señal de presencia o ausencia del objeto que se desea analizar, para que la cámara pueda adquirir la imagen cada

[5] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario

vez que el sensor se lo indique, y para que la imagen sea capturada en la misma posición siempre. El pin preconfigurado para entrada de señal trigger es el pin 1.

- **Señales para PLC:** en muchos casos será necesario sacar señales que se dirijan a un PLC, para que controle el movimiento de algún motor con el fin de parar o dar marcha a algún transportador por el cual estén circulando los objetos que están siendo analizados por la cámara, y que en caso de tener alguna falla serían retirados de la línea de producción.
- **Dispositivos actuadores:** en caso de que la cámara haya encontrado alguna falla en el objeto analizado, ésta puede enviar una señal para activar luces de aviso o elementos actuadores como pistones neumáticos, etc. a través de relés que amplifique la corriente y aíslen la tarjeta.
- **Recepción de señales externas:** tales como pulsantes, sensores o señales desde algún PLC que requiera el sistema.
- **Alimentación:** la tarjeta posee borneras para que se le conecte una fuente de alimentación en 24Vdc que a su vez alimentara de energía a la cámara.

Como se había señalado antes, el proyecto no abarca la discriminación física de cajas con anomalías, pero es importante recalcar que el sistema puede llegar a cubrir estos alcances, si las necesidades de la empresa así lo requieren.

3.2.3 CARACTERÍSTICAS DEL CABLE DE PODER CBL5-D10

El cable de poder constituido por 10 hilos conecta la cámara y la tarjeta de entradas/salidas (CON-IBOB), y tiene como función llevar la alimentación hasta la cámara, así como las señales de trigger para la adquisición de la imagen, y las señales de advertencia o falla que genere la cámara para que la tarjeta ejecute alguna acción. [1]

3.3 REQUERIMIENTOS DEL SISTEMA

Existen varios factores físicos con los que se deben contar para el correcto funcionamiento del sistema de adquisición de imágenes, entre ellos está la separación entre cajas, sistema que ayude a ubicar a las cajas siempre en la

misma posición, protección física para las cámaras e iluminación para adquisición de imágenes. Todos estos puntos se detallan a continuación.

3.3.1 TRANSPORTADORA PARA SEPARACIÓN ENTRE CAJAS

El sistema de inspección visual requiere separación entre cajas, de manera que un sensor trigger pueda advertir a la cámara de la presencia o ausencia de la caja que se vaya a analizar.

3.3.1.1 Descripción de las condiciones originales de la transportadora antes de la ubicación del sistema

Dentro de la línea de envase, a continuación de la máquina despaletizadora, existe una sección de transporte llamada **Banda de rodillo vivo**, tal como se ve en la Figura 3.7; que consta de motores y bandas para dicha sección.

En este diseño, la banda se halla en medio de los rodillos superiores y los rodillos inferiores; es decir, la caja está en contacto directamente con los rodillos superiores.

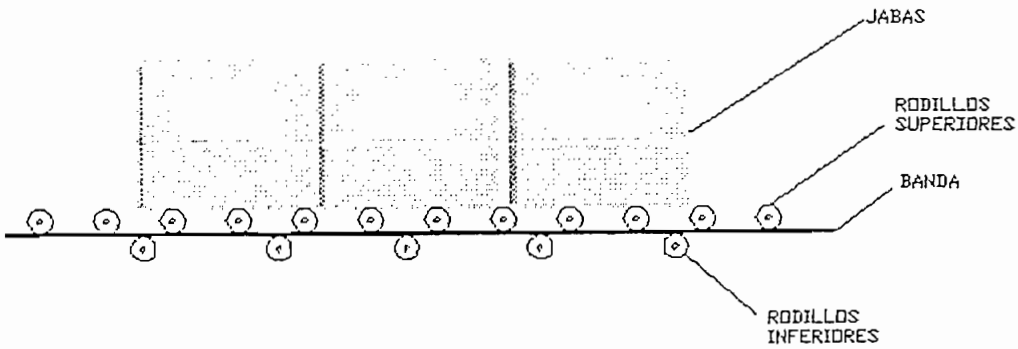


FIGURA 3.7: Sección de Transporte compuesta por rodillo vivo.

En el envase se aprecia la forma física de esta sección de transporte tal como se ve en la Figura 3.8.

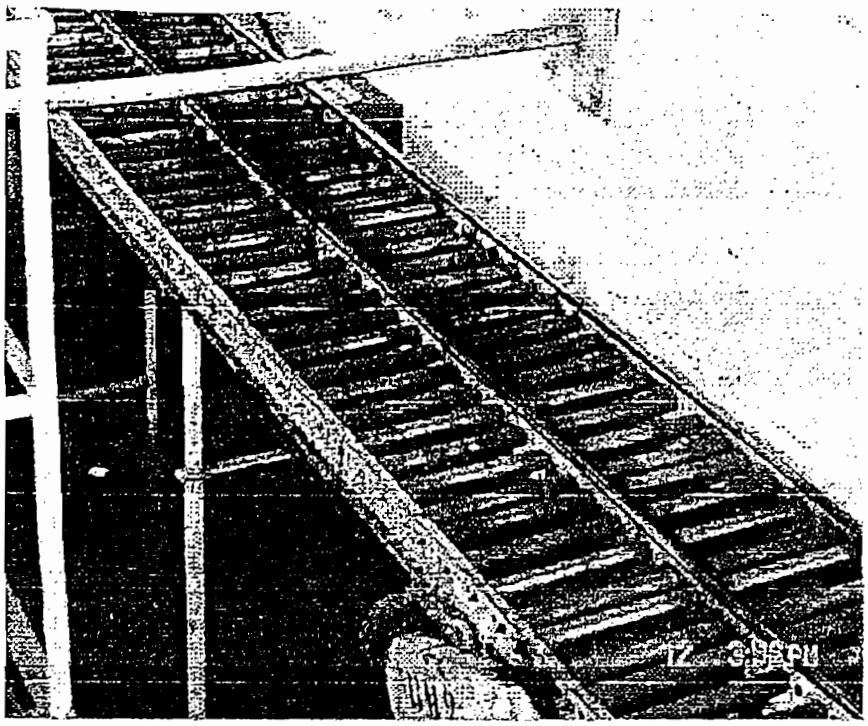


FIGURA 3.8: Sección de Transporte original con Rodillo Vivo.

Como consecuencia de este diseño la caja tiene cierta libertad de movimiento independiente del movimiento de la banda, pudiendo resbalar y juntarse una con otra, dificultando al sistema que se desea implementar la diferenciación entre una y otra caja.

Esta transportadora tenía originalmente los datos mostrados en la Tabla 3.2

Longitud	7m
Ancho por Carril	0.45m
Número de carriles	2
Velocidad de transporte	50.3 RPM

TABLA 3.2: Datos de la transportadora original, Transportadora #1.

Esta sección de transportadora, a la que se denominará Transportadora #1, tiene el fin de frenar las cajas con botellas vacías, las mismas que están siendo transportadas sobre bandas inclinadas (bandas de rodillo loco) y bajan con mucha velocidad, debido a que la máquina despaletizadora las coloca sobre las bandas y

las empuja hacia abajo. Una vez que las cajas llegan hasta la Transportadora #1, éstas adquieren baja velocidad y continúan circulando por la línea de transporte, tal como se muestra en la Figura 3.9.

Nota: Se debe recordar que en la línea de envase existen dos líneas de transporte paralelas, y al referirse a la Transportadora #1, se hace referencia a las dos carrileras.

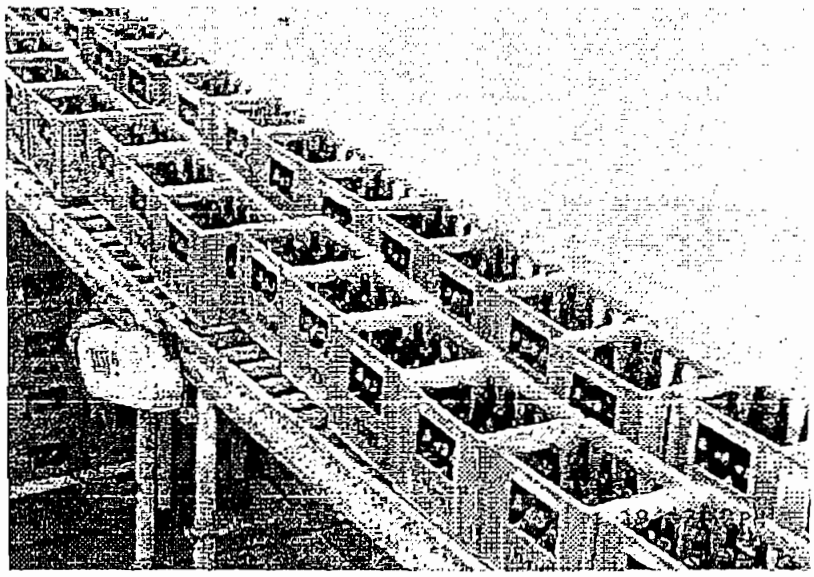


FIGURA 3.9: Sección de Transporte original con cajas circulando Transportadora #1.

Para el funcionamiento del sistema de inspección visual se requería de separación entre cajas, y la Transportadora #1 por si sola no era capaz de producir dicha condición; sin embargo, permitió la construcción de una segunda transportadora a continuación de ésta, para finalmente conseguir la separación entre cajas.

3.3.1.2 Descripción de la sección de transportadora requerida por el sistema de inspección visual

En el sector seleccionado, para poder ubicar el inspector fue necesario recortar la Transportadora #1 original y reemplazar el tramo recortado por una nueva sección de transportadora, a la que se denominará Transportadora #2, que cumpla la

condición de separar las cajas y además no generar ningún cuello de botella en dicho sitio.

Para cumplir estas condiciones, se optó por un diseño de transportadora llamado **Rodillo con Banda,** mostrado en la Figura 3.10. En este diseño, la caja se ubica directamente sobre la banda impidiendo que las cajas resbalen y se junten, como ocurre con el diseño de la trasportadora anterior.

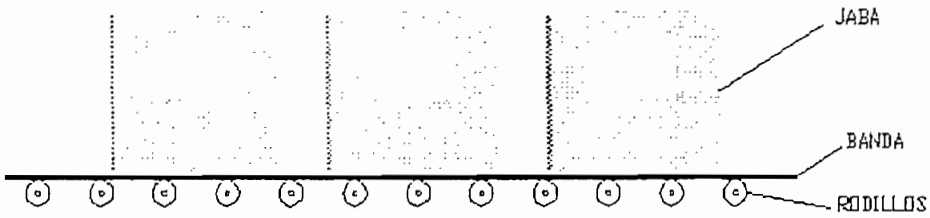


FIGURA 3.10: Sección De Transporte implementada con "Rodillo con Banda".

En la Figura 3.11 se ve la sección de transporte implementada en el envase.

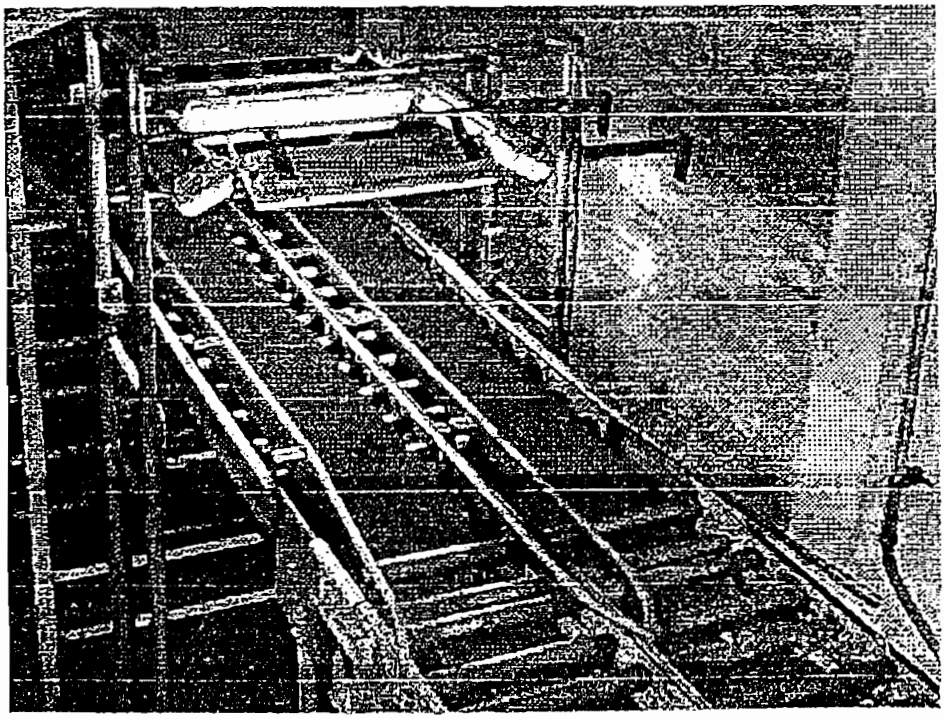


FIGURA 3.11: Nueva sección de transporte con bandas.

Para generar la separación entre cajas y evitar cuellos de botella en ese sitio, la nueva sección de transporte, Transportadora #2, debe tener una velocidad superior a la Transportadora #1, antes ya descrita, pero no excesiva pues generaría cuellos de botella a continuación de la Transportadora #2. El tamaño de la nueva sección de transporte esta indicada en la Tabla 3.3.

Longitud	2m
Ancho por carril	0.45m
Numero de carriles	2

TABLA 3.3: Datos de la nueva sección de transporte. Transportadora #2.

Una vez que la Transportadora #2, mostrada en la Figura 3.12, fue instalada se consiguió la separación entre cajas debido a la diferencia de velocidades entre las Transportadoras #1 y #2.

Nota: Al igual que la Transportadora #1, al referirse a la Transportadora #2 se habla del par de carrileras paralelas que transportan las cajas.

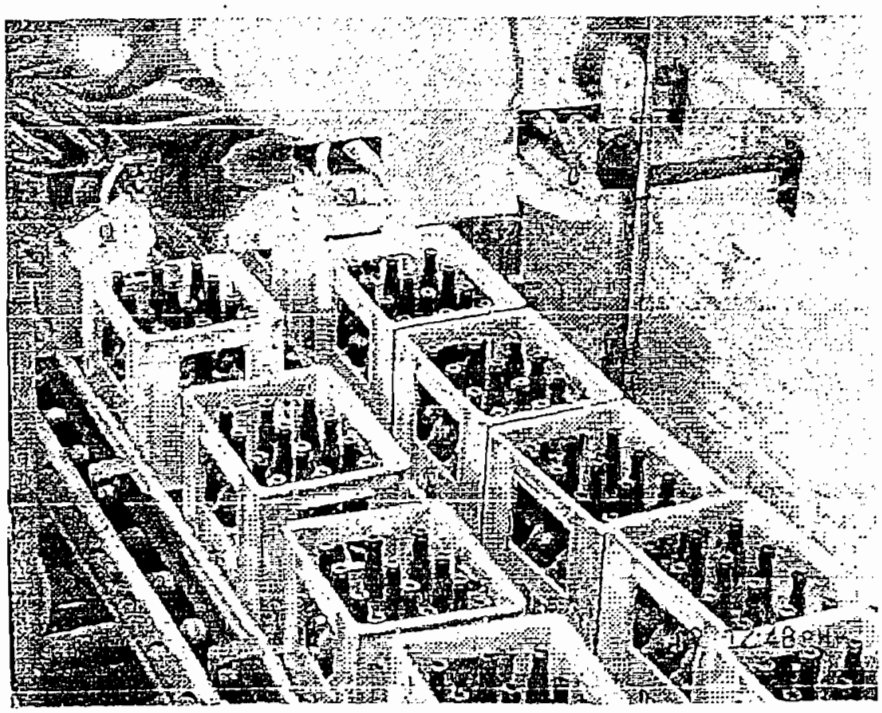


FIGURA 3.12 Nueva sección de Transporte con cajas circulando. Transportadora #2

Como ya se dijo, la Transportadora #2 consta de dos carriles paralelos, por ello fue necesario el uso de un motoreductor para cada carril. Los motoreductores utilizados en la nueva sección de transporte estaban disponibles en Cervecería Andina, y aunque son similares en forma, varían en potencia y relación de reducción de velocidad. Estos datos están indicados en la Tabla 3.4.

Parámetro	Moto reductor 1	Moto reductor 1
Relación de reducción	1 : 12	1 : 24
Velocidad del motor	1710 RPM	1710 RPM
Velocidad salida del reductor	114 RPM	68.4 RPM
Potencia Motor	550 W	750 W
Voltaje	440 V	440 V
Factor de potencia	0.74	0.76
Frecuencia	60 Hz	60 Hz
Factor de protección	65	65

TABLA 3.4: Parámetros de los Motoreductores de la nueva sección de transporte.

La diferencia de potencias no fue un problema pues, al no ser tan grande la nueva sesión de transporte, la carga que soportan los motoreductores no es muy alta y no es permanente pues las cajas están en movimiento, Además, en el reductor se obtuvo un incremento en el torque pues la velocidad a la salida del reductor es inferior a la velocidad de salida del motor.

Dado que la velocidad de la transportadora original es de 50.3 RPM, se tomo como base la velocidad de salida del motoreductor 2 que es 68.4RPM (Ver Tabla 3.4), pues la velocidad de salida del motoreductor 1 es muy alta, lo que ocasionaría un cuello de botella a continuación de la nueva sección de transporte.

Para transferir la potencia desde el eje del motoreductor hasta el eje motriz de la transportadora se colocó en cada eje un piñon. El número de dientes del piñón se listan en la Tabla 3.5

Piñón al eje del motoreductor 2	15 dientes
Velocidad del motoreductor 2	68.4 RPM
Piñón motriz del transportador carril 2	15 dientes
Velocidad del transportador carril 2	68.4 RPM

TABLA 3.5 Datos del Motoreductor 2

Nota: Se escogió piñones de 15 dientes pues son más fáciles de encontrar que uno de menor número de dientes, y son de tamaño moderado respecto a uno de mayor número de dientes; pues entre mas dientes se incrementa el diámetro del piñón.

Como medio de acople entre los piñones del motoreductor y del eje motriz de la transportadora se utilizó una cadena como se muestra en la Figura 3.13.

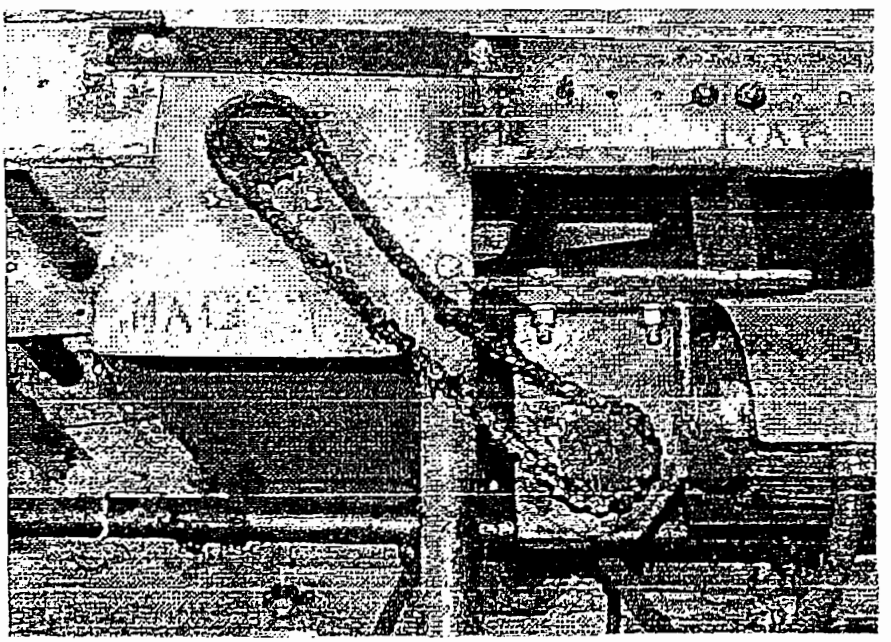


FIGURA 3.13: Cadena de acople del motoreductor y el eje motriz de la transportadora.

Para mantener una velocidad igual en el eje del motoreductor 2 y en el eje motriz de la transportadora carril 2, los piñones correspondientes deben tener igual número de dientes para mantener así una relación de velocidad igual a 1.

Puesto que los motoreductores son diferentes, las velocidades al eje motriz son diferentes, para igualar la velocidad del motoreductor 1 a la del motoreductor 2 se cambió el número de dientes en el piñón del eje motriz del transportador carril 1 en base a los cálculos que se detallan en la Ecuación 3.1

Cálculo de la relación de velocidades.

$$\frac{VelocidadTransportador}{VelocidadMotoreductor} = \frac{PiñonMotoreductor}{PiñonTransportadora}$$

$$PiñonTransportador = \frac{PiñonMotoreductor * VelocidadMotoreductor}{VelocidadTransportador}$$

$$PiñonTransportador = \frac{15 * 114}{68.4} = 25\,Dientes$$

Ec. 3.1

En la Cervecería se disponía de un piñón de 26 dientes. Por tanto la velocidad al transportador es la que indica la Ecuación 3.2

$$\frac{VelocidadTransportador}{VelocidadMotoreductor} = \frac{PiñonMotoreductor}{PiñonTransportador}$$

$$VelocidadTransportador = \frac{PiñonMotoreductor * VelocidadMotoreductor}{PiñonTransportador}$$

$$VelocidadTransportador = \frac{15 * 68.4}{26} = 65.77\,RPM$$

Ec.3.2

Con los cálculos realizados se pudo completar los datos para el Motoreductor 1 que se ven en la Tabla 3.6.

Piñón al eje del motoreductor 1	15 dientes
Velocidad del motoreductor 1	114 rpm
Piñón motriz del transportador carril 1	26 dientes
Velocidad del transportador carril 1	65.77 rpm

TABLA 3.6 Datos del Motoreductor 1

3.3.2 RIELES PARA POSICIONAR CAJAS

Una vez que se consiguió la separación entre cajas era imprescindible lograr que las cajas que están siendo inspeccionadas se ubiquen en la misma posición, ya que la programación de las cámaras digitales así lo requiere; es decir, el objeto a analizar debe tener la menor posibilidad de movimiento rotacional y traslacional para hacer un correcto análisis del objeto.

Para este fin se construyó guías para cada una de las carrileras, ubicadas sobre la Transportadora #2, que permitan posicionar las cajas en la posición requerida y que tengan un juego máximo de 1cm en sentido transversal a su movimiento natural.

3.3.3 PROTECCIÓN PARA LAS CÁMARAS DIGITALES

Las cámaras digitales son dispositivos muy delicados y de alto costo, que no solo requieren de manejo exclusivo de personal calificado, sino que es indispensable protegerlas de la humedad, polvo, gases tóxicos y altas temperaturas. Para ello se construyó campanas de acero inoxidable que ayudan, no solo a la protección contra agentes externos que dañen a la cámara, sino que ayudan a concentrar toda la luz de la cámara hacia un mismo objetivo, en este caso las cajas con botella vacía. En la Figura 3.14 se puede apreciar las campanas.

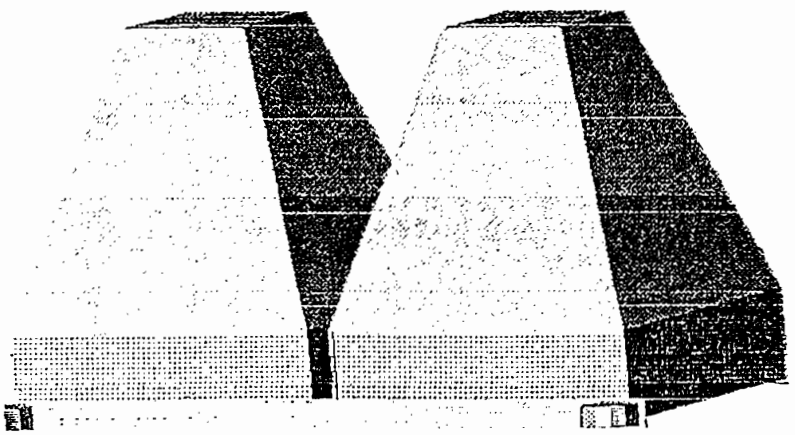


FIGURA 3.14: Campanas de Acero Inoxidable.

Las dimensiones de estas campanas son de 50cm por lado en la base, y de 80cm de altura. La cámara digital 530MR se ubica dentro de cada una de las campanas

y dentro de ellas tienen movilidad tanto en sentido vertical como en sentido horizontal, para posicionarlas correctamente sobre el objetivo, que son las cajas de cerveza.

3.3.4 SISTEMA DE ILUMINACIÓN PARA CAJAS

La iluminación se puede considerar la parte más crítica dentro del sistema de inspección visual. Las cámaras son mucho menos sensibles y versátiles que la visión humana, y las condiciones de iluminación deben ayudar al máximo para que una cámara pueda capturar una imagen que el ojo humano podría distinguir sin necesidad de una iluminación tan especializada. La necesidad de una mejor iluminación se hace mucho más evidente cuando el objeto a analizar presenta formas complejas o superficies muy reflectantes como el caso que se puede ver en la Figura 3.15. [10]

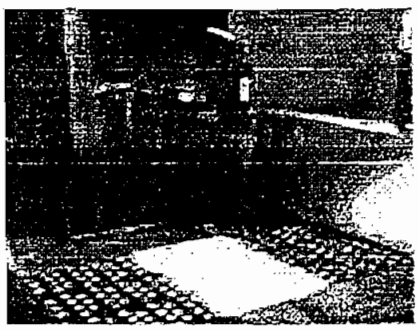


FIGURA 3.15: Sistema de iluminación para empaques de pastillas.

Las cámaras capturan la luz reflejada de los objetos sobre todo si estos son de vidrio. El propósito de la iluminación utilizada en las aplicaciones de visión es controlar la forma en que la cámara ve el objeto. La luz se refleja de forma distinta si se ilumina una bola de acero, que si se ilumina una hoja de papel blanco y el sistema de iluminación por tanto debe ajustarse al objeto a iluminar.

Para la elección de la iluminación que requieren las cajas, se tomó en cuenta los criterios sugeridos por el fabricante de las cámaras (DVT). Así por ejemplo, para iluminación de botellas de vidrio se sugiere luz roja, ya que permite resaltar detalles importantes de la forma y contornos de dicha botella, siempre y cuando sea posible la inspección de la botella, unidad por unidad. También se sugiere el

uso de luz blanca en aplicaciones donde se requiere examinar contenidos de envases semi-transparentes o transparentes. [10]

Para la aplicación que aquí se analiza, el uso de luz roja debería ser el más óptimo, ya que los objetos a inspeccionar son botellas de vidrio; sin embargo, hay que recordar que la inspección, si bien es de cada una de las doce botellas vacías de la caja, no se la realiza de botella en botella, sino mas bien en conjunto; es decir, hay una sola adquisición de imagen por caja, con lo cual la perspectiva del objeto a inspeccionar cambia drásticamente. Por ejemplo, dado que las botellas están dentro de la caja resulta imposible usar una luz de fondo que ayude a resaltar los bordes. [1]

Por ello, el uso de luz roja resultó impráctico ya que físicamente resulta imposible colocar las luces detrás del objeto que se desea inspeccionar. Además, el uso de luz roja implica un alto costo económico que encarece más aplicación. Explorando otras alternativas se pudo observar que el uso de lámparas fluorescentes de luz blanca dio buenos resultados y su costo de compra y mantenimiento es mucho menor que el de lámparas de luz roja.

El uso de luces adicionales a las que tiene la cámara resultó imprescindible, pues así se pudo aislar la luz ambiental con que cuenta el sector en donde se colocó el sistema de inspección visual. Con el aislamiento que se consiguió se logró uniformar la intensidad de luz con la que las imágenes son capturadas y por ende facilitó la programación y procesamiento de dichas imágenes.

3.3.5 SENSOR TRIGGER PARA LA CAPTURA DE IMÁGENES

La cámara 530MR requiere de una señal de disparo que le indique el momento en que debe adquirir la imagen. La señal puede ser programada por el usuario (señal interna) con intervalos de tiempo deseado por el programador, o también puede ser una señal externa proporcionada por un sensor.

Para el sistema implementado se recurrió al uso de un sensor fotoeléctrico cuya única función es detectar la presencia de cajas y dar una señal a la cámara

respectiva para la inmediata adquisición de la imagen. Sus características se describen en la Tabla 3.7.

	Marca	SICK
	Serie	Compacta
	Código	WT-250-S162
	Voltaje	10-240 Vdc/24-240Vac
	Potencia	< 2 VA
	Salida	Relé
	Alcance	1-60 cm
	Protección	IP 67
	f switcheo	33/s

TABLA 3.7: Características del sensor Trigger.[6]

3.4 INSTALACIÓN DEL TABLERO DE CONTROL

Para la implementación del sistema de inspección visual fue necesario instalar un tablero eléctrico cuya disposición y contenido se ven en el esquema de la Figura 3.16.

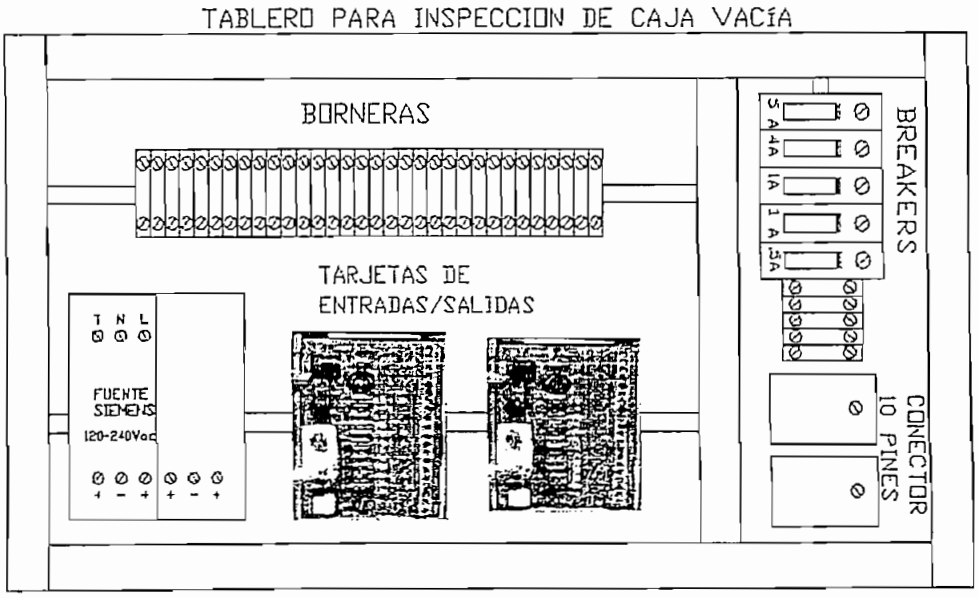


FIGURA 3.16: Tablero de Control ubicado en el envase.

[6] SENSICK, Sensors for Automation Catalog

Luego de la ubicación de los elementos, el tablero quedó con los elementos ubicado tal como se muestra en la foto de la Figura 3.17.

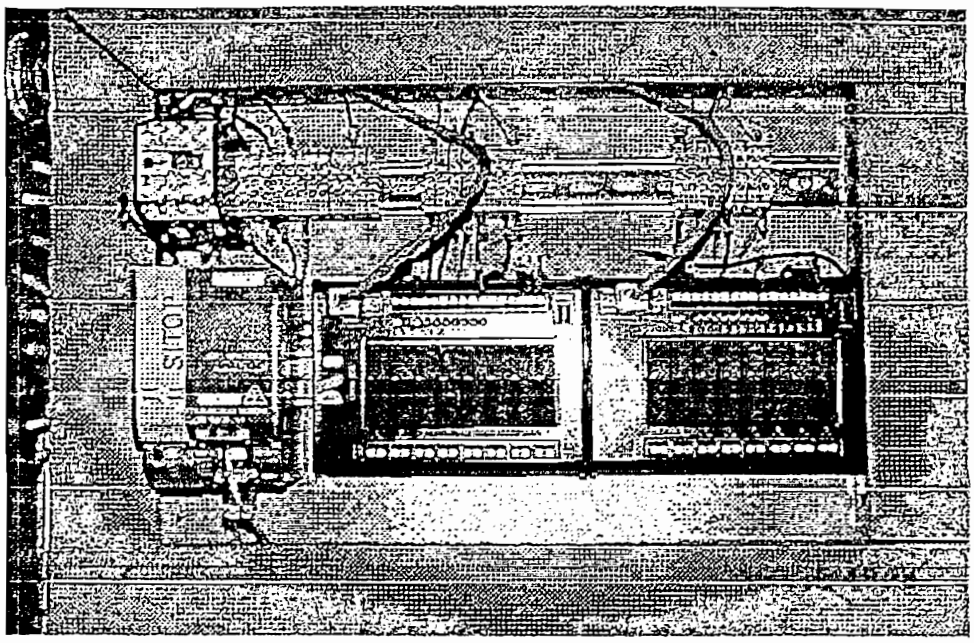


FIGURA 3.17: Ubicación real de los elementos del tablero de control .

3.5 ELECCIÓN DE CABLE PARA COMUNICACIÓN

Para elegir un cable apropiado para la aplicación se debe tener en cuenta las siguientes características físicas de la sección de envase:

a) La distancia estimada entre el punto de red con el que cuenta el envase y la ubicación física donde se ubicarán las cámaras es de 90m aproximadamente.
b) El lugar por donde el cable debe transitar es un ambiente netamente industrial, sometido a: ruido, calor, suciedad, etc.
c) El cable que se use no atravesará la sección de envase en línea recta, ya que tendrá que seguir las rutas ya determinadas por los canales y tuberías existentes, y éstas curvan, suben, bajan en todo el trayecto. Los canales y tuberías están ubicadas sobre los 3.5m de altura sobre el nivel del piso.
d) Los canales existentes tienen líneas de alta voltaje que producen interferencia con cualquier cable de comunicación.

e) Debido a la gran distancia que existen entre el punto de red y las cámaras, así como la altura a la que se requiere ubicar el cable; se ha descartado el uso de tuberías para llevar el cable, siendo necesario buscar un cable que anule todos los problemas antes descritos.

	Marca	BELDEN
	Serie	7921A
	# de pares	4 (blindados)
	Conductor	24AWG
	Aislamiento1	Lámina de Al y cinta de poliester
	Aislamiento2	Cobre estañado
	Protección a	Aceite, ruido
	Norma	NEMA WC-63.1
	Categoría	5e
	Protocolo	Ethernet

TABLA 3.8: Características del Cable Seleccionado.[7]

En la Tabla 3.8 se puede apreciar las características del cable seleccionado (STP), el objetivo de usar un cable con doble blindaje es prescindir del uso de tuberías. El apantallado de cobre elimina la posibilidad de interferencias producidas por las líneas de alta voltaje e interferencias RFI y EMI. Además, como se puede apreciar en las características que se listan a continuación, el cable cumple con las normas para cableado estructurado.

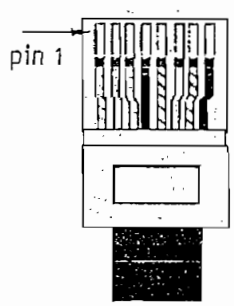
Los estándares que regulan el cableado estructurado en el mundo están definidos por la EIA/TIA (Electronics Industries Association/Telecomuncations Industries Association) que especifica el cableado estructurado sobre cable de par trenzado UTP de categoría 5, el estándar 568A.

- CATEGORÍA 5e.- se refiere a un cable par trenzado que consta de 8 hilos y usa conectores RJ-45. Dependiendo del número de trenzas por unidad

[7] BELDEN ELECTRONICS DIVISION

de longitud, los cables de par trenzado se clasifican en categorías. A mayor número de trenzas mayor es la velocidad de transferencia de datos. La categoría 5 y 5e tienen una velocidad de 1 Gbps. [11]

- **NORMA 568A.-** Se refiere al orden que los 4 pares del cable UTP deben ocupar en el momento de ser insertados en el conector RJ-45. Al usar la Norma 568A se puede construir únicamente cables directos. Además la distancia máxima entre dos dispositivos usando esta norma es de 100m. El cable UTP tiene 8 cables y su disposición se muestra en la Tabla 3.9. [11]



# PIN	COLOR
1	Blanco/Verde
2	Verde
3	Blanco/Naranja
4	Azul
5	Blanco/Azul
6	Naranja
7	Blanco/Café
8	Café

TABLA 3.9: Distribución de pares del cable UTP en el conector RJ-45.[8]

3.6 INSTALACIÓN Y CABLEADO DE LOS EQUIPOS

La Figura 3.18 es un diagrama unifilar de la conexión entre todos los elementos utilizados para la implementación del sistema. Aquí se puede ver el uso de dos cámaras digitales, cada una tiene su tarjeta de entradas/salidas que está alimentada por una fuente de 24Vdc. Cada tarjeta de entradas/salidas usa una entrada para el sensor trigger que indica la presencia de jaba en cada una de las líneas de transportación.

Adicionalmente, las cámaras digitales se conectan con la computadora de envase, esto con el fin de transmitir datos desde la cámara hasta la PC, para ello

[8] www.eveliux.com

fue necesario el uso de un switch. El switch permitió poner en red a las cámaras digitales, gracias al protocolo Ethernet con que cuentan las cámaras.

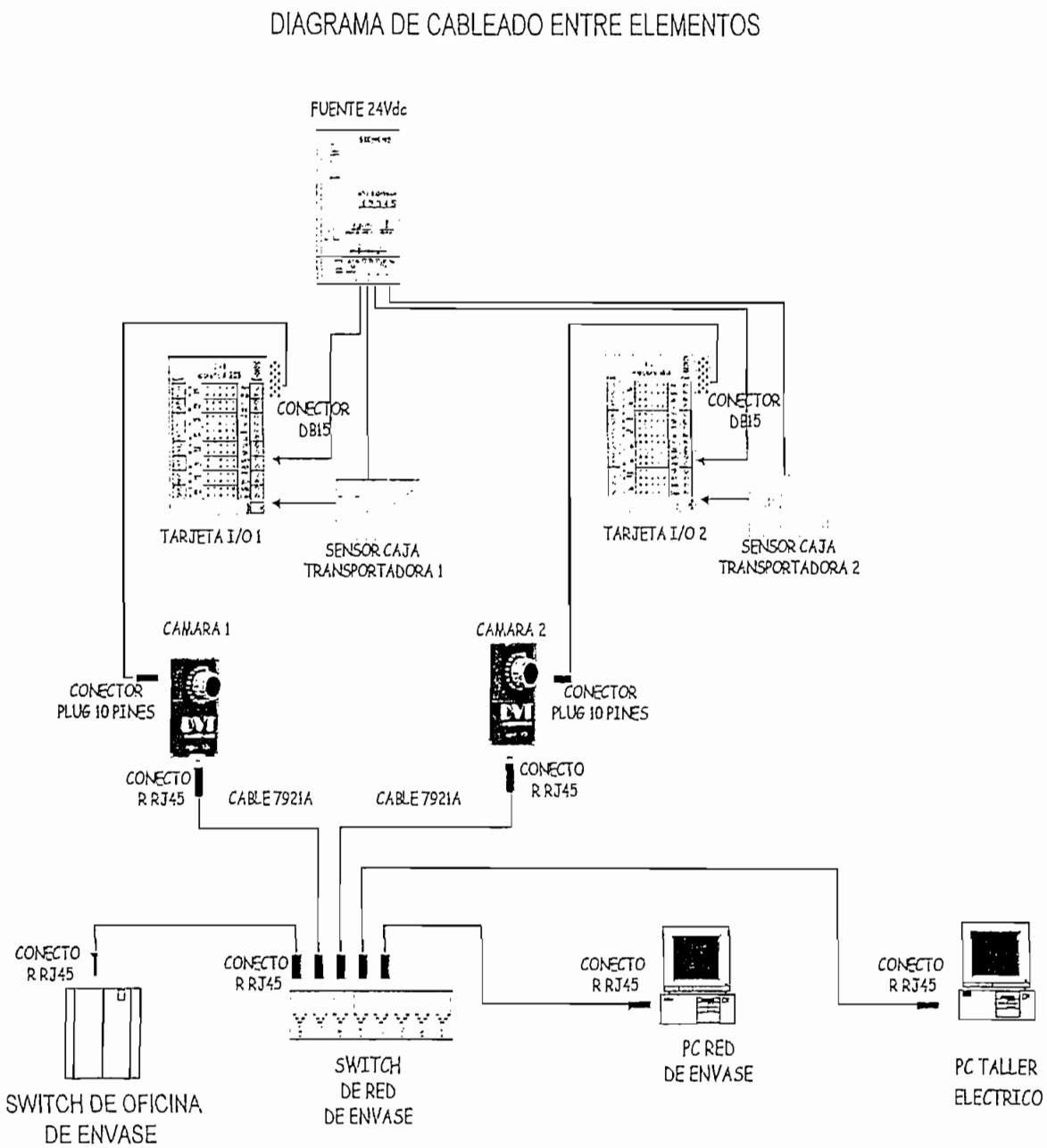


FIGURA 3.18: Diagrama unifilar de conexión entre elementos del sistema.

Una vez que se ha descrito el hardware que requirió el método seleccionado, en el Capitulo 4 se describirá el software que se desarrollo para conseguir el objetivo de desarrollar un sistema de detección de jabas incompletas con botellas vacías.

CAPITULO 4

DESARROLLO DE PROGRAMAS DE INSPECCIÓN, Y DE INTERFASE PARA DEL USUARIO

En este capítulo se describen los pasos que se siguen para diseñar los programas de inspección para jabas con botellas vacías, así como la interfase en Visual Basic. Se explican los conceptos y términos con que trabaja el sistema de inspección DVT, así como los parámetros necesarios para configurar cada una de las cámaras.

Como ya se señaló en el Capítulo 3, el problema de botella rota y ausente se resolvió usando un sistema de inspección visual. Para ello se usa cámaras industriales especializadas, las cuales cuentan con su propia plataforma de programación. En el esquema de la Figura 4.1 se puede apreciar el flujo de programación que se usó para el análisis de imágenes.

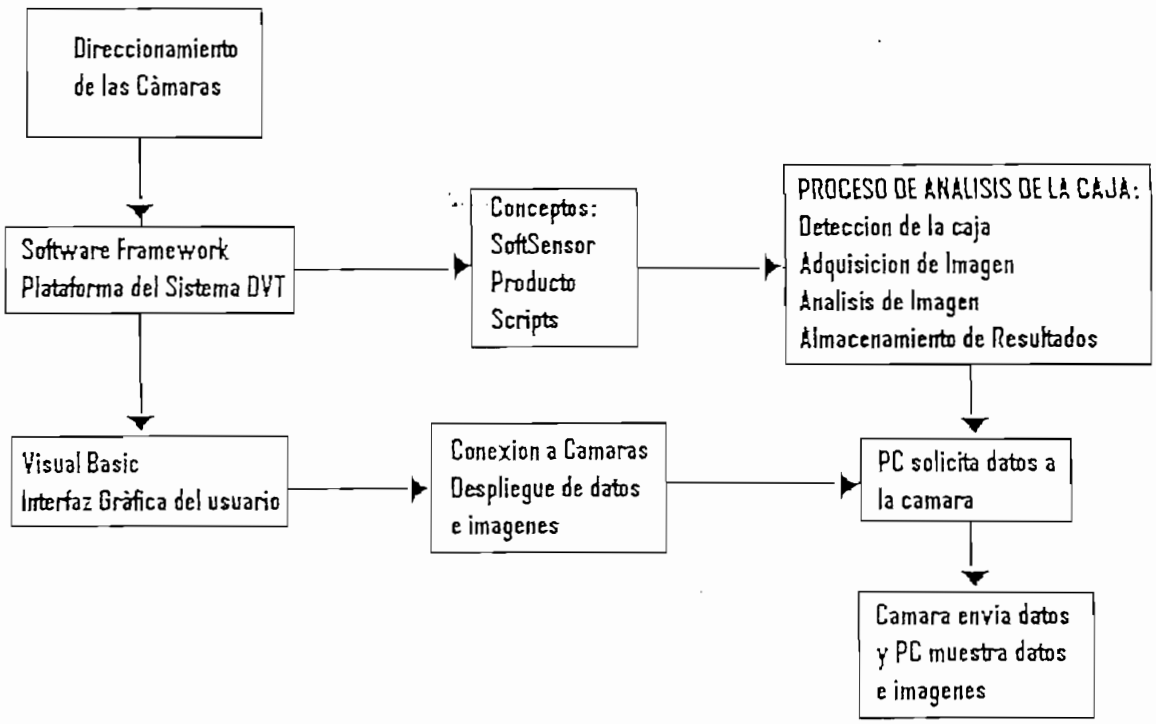


FIGURA 4.1: Esquema General de programación de las cámaras 530MR.

4.1 FUNCIÓN DEL SMARTIMAGE SENSOR

El sensor SmartImage no es más que la cámara 530MR antes descrita. La información que ofrece esta herramienta puede ser enviada a un PLC o cualquier computadora que se encuentre en red con la cámara. La información toma la forma de datos que pueden ser manejados para el Control Estadístico de Procesos tal como sucede en el caso del presente proyecto. Los datos pueden ser simples señales de PASO o FALLA, pero también pueden ser datos complejos de posiciones, niveles de intensidad, etc. [1]

Para acceder al manejo de esta versátil herramienta se cuenta con FrameWork que es la plataforma del programa para realizar cualquier tarea de inspección.

4.2 DIRECCIONAMIENTO DE LAS CÁMARAS

Las cámaras digitales con que se trabajó tienen el mismo tipo de direcciones IP que cualquier computadora, son tratadas como otro dispositivo conectado a la red local de Cervecería Andina.

Existen 3 clases de direcciones IP que ARIN (Registro Americano de Números de Internet) se encarga de designar a cada solicitante, y son las siguientes: [4]

	8bits	8bits	8bits	8bits
Clase A:	Network	Host	Host	Host
Clase B:	Network	Net	Host	Host
Clase C:	Network	Net	Net	Host

Las direcciones Clase A son destinadas para los gobiernos de todo el mundo, las Clase B para medianas empresas y las Clase C para el resto de solicitantes. Las direcciones IP con las que las cámaras trabajan son de Clase B; es decir con las que la Cervecería trabaja. [4]

Para direccionar las cámaras fue necesario usar una computadora que se halle compartiendo la misma red que las cámaras. Para conectar una PC y el

SmartImage Sensor se recurre al protocolo TCP/IP (Protocolo de control de Transmisión/Protocolo Internet), que se basa en la identificación de las estaciones de una red con los números IP. Para que dos dispositivos se puedan comunicar deben estar en la misma Red de área Local (LAN).

Para asignar una dirección IP a las cámaras y así permitir que sean parte de la red con la que está trabajando la sección de envase, se deben seguir los pasos que se describen a continuación:

a) Iniciar Framework (Framework es el paquete con el que trabajan las cámaras y se instala en cualquier computadora que el usuario vaya a usar). [1]
b) Escoger la Opción PC Comunicación de la Barra Principal de FrameWork. [1]
c) Abrir el "Network neighborhood". Este permite identificar las cámaras en la red. La computadora buscará las cámaras en la red local. Adicionalmente muestra el nombre y dirección IP de la computadora usada, como se muestra en la Figura 4.2. [1]

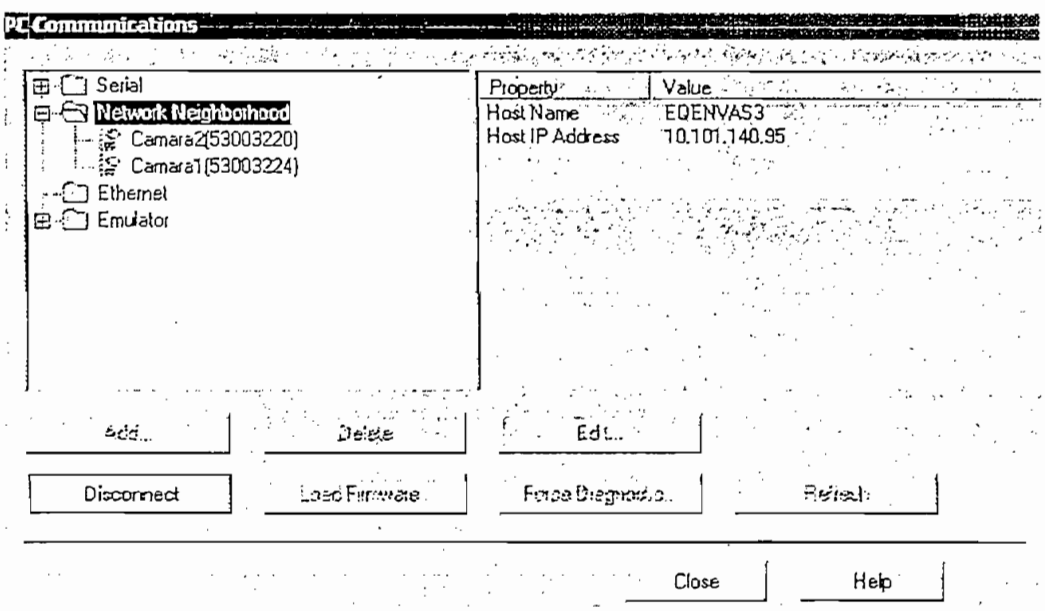


FIGURA 4.2: Reconocimiento de las cámaras desde la PC.

d) Una vez que han sido identificadas las cámaras se puede editar los datos de cada una con el número IP y nombre que se va a asignar a la cámara, tal como se ve en la Figura 4.3. [1]

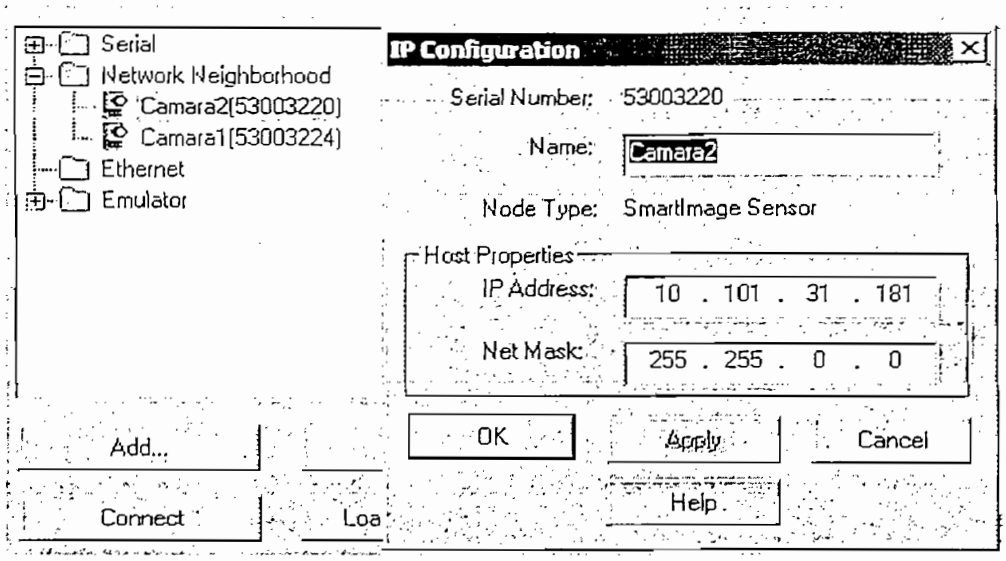


FIGURA 4.3: Asignación de direcciones IP para las cámaras.

e) También es necesario escribir la dirección de máscara de red. Si la dirección IP asignada ya esta siendo usada por otro dispositivo entonces aparecerá un mensaje de alerta para que se vuelva a intentar con otra dirección. [1]

4.2.1 TOPOLOGIA FISICA USADA PARA CONECTAR LAS CÁMARAS

La topología que se utiliza para conectar las cámaras es del tipo estrella. En la Figura 4.4 se puede apreciar las estaciones conectadas a un concentrador o HUB con un cable por computadora. Para el caso del presente proyecto, cada estación representa cada una de las cámaras, así como a las computadoras pertenecientes a la red. Como concentrador se seleccionó un Switch. La ventaja de esta topología es la ausencia de colisiones en la transmisión y dialogo directo de cada estación con el servidor; además, la caída de una estación no interrumpe la red. [4]

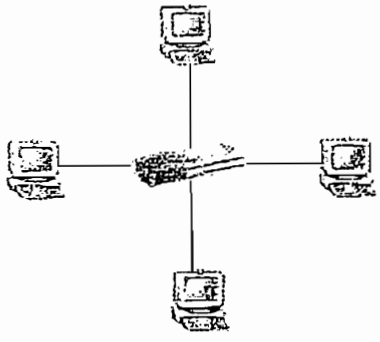


FIGURA 4.4: Topología Estrella usada para conectar las cámaras a un concentrador. [9]

4.2.2 PROTOCOLOS MAC

Cuando varios dispositivos se conectan a un medio físico compartido (switch, hub, etc.), se requiere establecer varias normas de comunicación para utilizar dicho dispositivo, dichas normas se identifican como protocolo MAC. El protocolo empleado para comunicar las cámaras y las computadoras en la red es: **CSMA/CD** *(Acceso Múltiple con detección de portadora y detección de colisiones).* Se trata de un protocolo no determinístico en donde el que primero llega es el primero en servirse; en este sistema sencillo todos los dispositivos que comparten la red (en este caso las cámaras) esperan que el medio esté libre para poder transmitir, pero si dos dispositivos transmiten al mismo tiempo entonces habrá una colisión de datos y la transmisión deberá repetirse mas tarde. Al ser detectada la colisión por todos los dispositivos, esperan hasta que la falla concluya para volver a transmitir. [4]

El método de acceso CSMA/CD es el adoptado por las redes Ethernet, y tienen por característica que son de tipo "broadcast", es decir un mensaje generado por un dispositivo es escuchado por el resto para ver si este le corresponde.

4.3 PLATAFORMA DE PROGRAMACIÓN FRAMEWORK

La plataforma del programa FrameWork consta de 3 componentes principales:

---

[9] DR. LUÍS CORRALES, folleto de interfases

- ***Interfase de usuario*** de FrameWork (pantalla en donde se ven las imágenes y se analiza la imagen) que se corre en una computadora y usa un sistema amigable de Windows. Es una aplicación que presenta una pantalla amigable al usuario, aquí se puede agregar cualquier tipo de sensor virtual para realizar el análisis de cualquier imagen capturada por la cámara. [1]
- ***Firmware*** de FrameWork, que es la parte del programa que corre en el sensor SmartImage (cámara 530MR). Es el que se encarga de procesar todas las imágenes y determina los resultados de acuerdo a los sensores virtuales usados. Firmware es una aplicación independiente; es decir, no requiere que la interfase de usuario este abierta para que un Sensor SmartImage inspeccione las piezas, determine resultados y los comunique al dispositivo que el usuario lo requiera. El Firmware procesa en la cámara todos los programas que fueron desarrollados en el FrameWork.[1]
- ***Emulador de Firmware*** que en ausencia del sensor SmartImage (cámara 530MR) imita el equipo completo en la computadora. [1]

Para entender la plataforma de programación del SmartImage que es el Framework y las funciones que realiza, a continuación se describe algunos conceptos básicos que el usuario debe conocer.

4.3.1 CONCEPTO DE SOFTSENSOR

Los SoftSensor o sensores virtuales son sub-elementos de un Producto, son herramientas o sensores virtuales ubicados sobre una imagen de alguna pieza que se desea inspeccionar, y realizan tareas específicas como ver el tamaño de un objeto, ver su posición, la forma del objeto, etc. [1]

Para inspeccionar ciertas características de una pieza se pueden crear varios productos para dicho objetivo, y cada producto puede tener varios SoftSensors, solo se debe tomar en cuenta que aumentar el número de sensores o productos aumenta el tiempo de inspección para cada imagen.

A continuación se ve en la Figura 4.5 la pantalla de FrameWork en donde se puede apreciar la barra de SoftSensors y todas las herramientas con que cuenta.

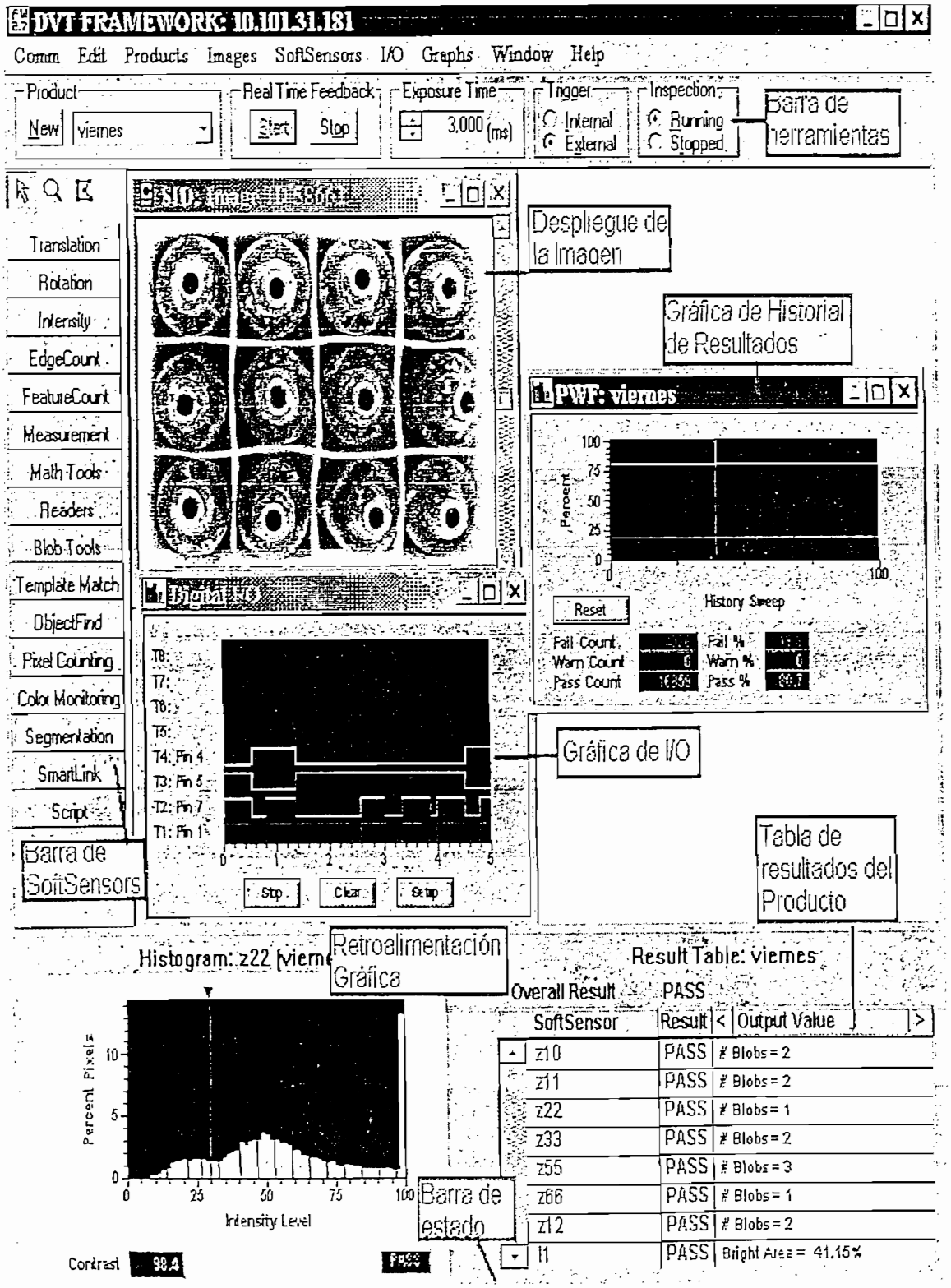


FIGURA 4.5: Pantalla de Framework. Sensores Virtuales disponibles.

Como se puede apreciar en la Figura 4.5, la pantalla de Framework cuenta con:

a) ***Barra de SoftSensors*** son todos los sensores virtuales disponibles que se encargan de identificar características específicas dentro de la imagen analizada. [1]
b) ***Barra de Herramientas*** es la barra donde están los botones para iniciar o detener la captura de imágenes, para determinar si la captura de imágenes será con un trigger interno o externo, para iniciar o detener el programa creado por el programador, para crear un nuevo programa, etc. [1]
c) ***Despliegue de Imagen*** es la pantalla donde aparecen las imágenes que la cámara esta capturando y sobre la cual se ubican los sensores virtuales para realizar una tarea que el programador les asigne. [1]
d) ***Grafica de entradas/salidas*** es la gráfica en donde se puede ver la asignación que el usuario le dio a cada pin y el tiempo que le lleva a cada uno realizar determinada acción. [1]
e) ***Gráfica de Historial de Resultados*** es la pantalla que recopila el porcentaje de imágenes con y sin falla respecto al total de imágenes, de este veredicto se encarga el producto creado por el programador. [1]
f) ***Tabla de Resultados*** aquí de muestra cada SoftSensor usado con su nombre respectivo y el veredicto que cada sensor está dando, que puede ser PASS o FAIL, estos valores van cambiando conforme la imagen adquirida lo va haciendo también. [1]
g) ***Retroalimentación Gráfica*** esta gráfica es dirigida a cada uno de los sensores y muestra el nivel de intensidad, en función del total de píxeles encontrados por el sensor sobre la imagen que se este analizando. [1]

4.3.1.1 Descripción de cada uno de los sensores

Framework cuenta con varios sensores que pueden tener usos múltiples, como por ejemplo dar las coordenadas de la ubicación de un objeto examinado, identificar la forma de un objeto comparándola con una imagen aprendida, hacer conteo de píxeles para verificar irregularidades en la superficie de objetos, etc.

Cada uno de los sensores que se explican a continuación genera retroalimentación gráfica y aparece automáticamente en la parte izquierda del panel de Resultados una vez que el sensor ha sido activado.

a) **Traslación y Rotación** *(Traslation y Rotation):* estos dos tipos de sensores están diseñados para encontrar la ubicación de un borde en una imagen. Muestran la posición del borde (Traslación) o el ángulo de rotación (Rotación). Sabiendo la ubicación exacta del borde de un objeto examinado, se puede dar tolerancias de movimiento (max/mín) que sean admisibles para que las condiciones de Pasa y Falla sean activadas. [1]
b) **Intensidad** *(Intensity):* Este grupo de sensores pueden ser lineales o de área; es decir, examinan el número de píxeles a lo largo de una línea o dentro de un área, y hacen una comparación del número de píxeles brillantes existentes frente al total de píxeles analizados. Se pueden programar las condiciones de Pasa y Falla según el contraste en la región inspeccionada y el porcentaje de píxeles brillantes. [1]
c) **Conteo de Bordes** *(EdgeCount):* esta herramienta cuenta el número de veces que el SoftSensor cruza los bordes de una imagen (se considera una borde como la transición entre píxeles obscuros y brillantes). [1]
d) **Conteo de Características** *(FeatureCount):* esta herramienta cuenta el número de características que halla en un objeto (se llama característica a un grupo de píxeles obscuros o brillantes que son evaluados en base al tamaño que posean, diagnosticando Pasa o Falla de acuerdo a la necesidad del usuario). [1]
e) **Mediciones** *(Measurement):* esta herramienta es muy sofisticada e incluye cálculos matemáticos, usa calibradores que ayudan a medir distancia a través de una línea o de un área, e incluso es capaz de localizar círculos de un determinado radio establecido por la necesidad del usuario. Todos estos valores pueden ser usados posteriormente en una programación que no es de tipo gráfica para obtener resultados más precisos de las dimensiones del objeto analizado. [1]
f) **Herramientas Matemáticas** *(MathTools):* las herramientas matemáticas permiten hallar distancias, ángulos y otros parámetros. Para encontrar

estos valores se ayudan de otros SoftSensors que les permitan ubicar los bordes de un objeto. Adicionalmente, la herramienta de Script que se explicará más adelante ofrece la opción de manipular estos valores y usarlos en aplicaciones particulares. [1]

g) **Lectores** *(Readers):* esta herramienta muy sofisticada permite aprender códigos de matrices de datos (una versión bi-dimensional del código de barras) y códigos en una dimensión. También tiene un soporte beta para el Reconocimiento Ópticos de Caracteres (OCR). [1]

h) **Herramienta de Figuras** *(BlobTools):* esta herramienta analiza la imagen buscando formas y calculando el tamaño de las formas halladas. Dan la facilidad de seleccionar las formas encontradas en la imagen, y descartarlas o tomarlas en cuenta de acuerdo al tamaño que convenga al usuario. [1]

i) **Igualación de Patrones** *(Template Match):* esta herramienta muy compleja, está diseñada para aprender automáticamente una porción de una imagen y buscar dicha imagen en inspecciones posteriores, además arroja los valores del error respecto a la imagen aprendida. [1]

j) **Busca Objetos** *(Object Find)* :esta herramienta aprende una forma definida por el usuario. Esta forma puede ser generada a partir de la agrupación de píxeles claros y obscuros. Luego esta forma será buscada en cada una de las imágenes inspeccionadas. [1]

4.3.2 CONCEPTO DE UN PRODUCTO

Un producto es el nivel de organización más alto de FrameWork. Se dice que un producto es la colección de todos los parámetros de inspección y configuración de las entradas/salidas para una inspección en particular, entendiéndose por parámetros de inspección todos los sensores virtuales y sus respectivos valores de calibración. [1]

Por ejemplo, en la línea de producción de Cervecería Andina se puede crear dos productos: con el primero se puede dar una señal de PASO o FALLA si la botella inspeccionada no corresponde al tamaño Standard que requiere la compañía; por otro lado, un segundo producto puede dar una señal de PASO o FALLA en caso

que la misma botella inspeccionada esté rota o no. Los resultados arrojados por cada uno de los productos pueden ser mostrados en dos o cuatro salidas digitales de acuerdo a la necesidad del usuario.

4.3.3 NIVELES DE PROGRAMACIÓN DE LA CÁMARA

Dentro de los SmartImage Sensors existen tres niveles de programación: nivel del sistema, nivel del producto y nivel del SoftSensor, niveles que se pueden apreciar en la Figura 4.6: [3]

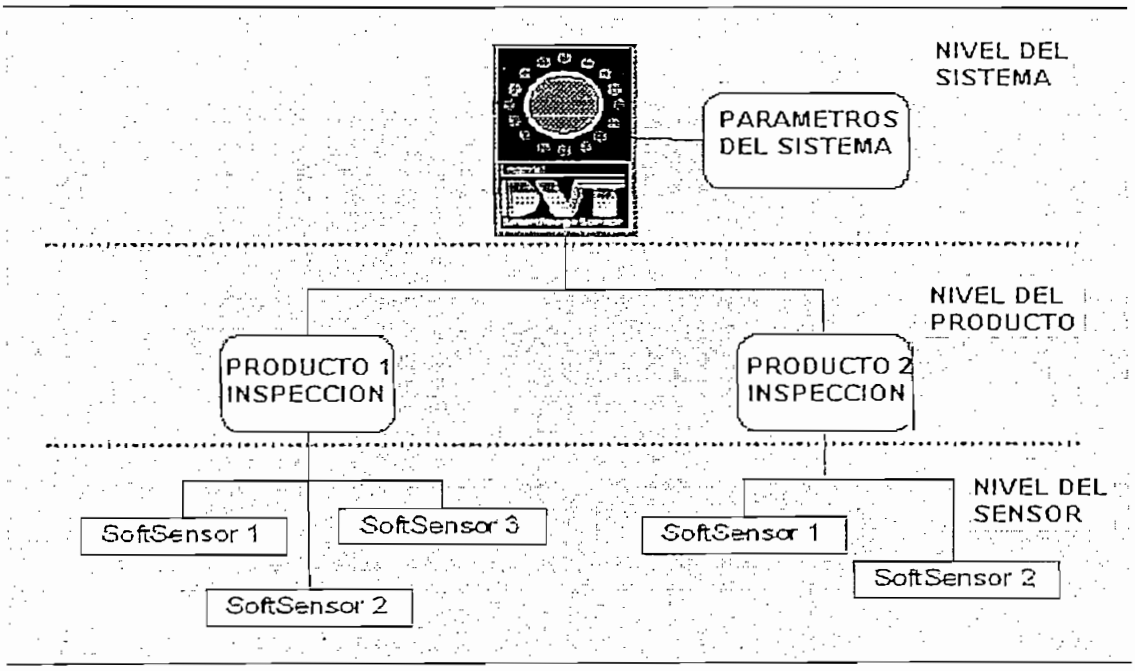


FIGURA 4.6: Niveles de programación de un SmartImage.[10]

Estos tres niveles son totalmente independientes uno del otro.

El nivel del Sistema contiene parámetros que afectan el funcionamiento del SmartImage (cámara 530MR). Parámetros como modo de disparo (externo o interno) y valores para comunicación externa. Dichos parámetros no cambian de una inspección a la otra; es decir, una vez que se han fijado se mantienen hasta que el usuario desee cambiarlos nuevamente.[3]

[10] DVT CORPORATION, DVT Script Reference Manual

El segundo nivel abarca la inspección de los productos, los cuales son específicos para cada inspección. Usualmente los SoftSensors usados realizan una variedad de inspecciones de una misma parte, y la inspección de un producto es necesaria para cada inspección realizada. [3]

En el tercer nivel están los SoftSensor que el usuario desee usar y son los encargados de realizar las tareas de inspección. Cada inspección es definida por la inspección que se desea realizar del producto. Así, por ejemplo, para verificar la presencia o ausencia de un objeto en la imagen adquirida se puede usar un SoftSensor de Intensidad, incluso alguno de aprendizaje de objetos; además, verificar si la pieza presente tiene la forma requerida. Con la ayuda de los productos se podría hacer que mientras un producto realiza la inspección de presencia/ausencia, el otro realiza la identificación de la forma correcta de un objeto.[3]

Este tipo de estructura interna permite al usuario realizar pequeñas tareas con varios productos, y permiten la solución de estos en forma independiente. Dentro de la Inspección de un Producto se puede tener control de sus propios SoftSensors y no a los SoftSensors de otros productos. Una vez que la inspección ha finalizado, los datos usados para la inspección desaparecen del sistema. Para muchas aplicaciones, la independencia entre productos limita el poder de las herramientas de los SmartImage. Es para este tipo de limitaciones que existe una herramienta llamada DVT Scripts, con su ayuda se puede acceder a valores y parámetros de SoftSensors de otros productos que no sean del que se este tratando en ese instante.[3]

Existen dos tipos de Scripts: los Background y Foreground scripts. La diferencia entre estas dos herramientas es que el uso de los Background es independiente de las inspecciones que se efectúen y funciona a nivel del sistema.

Usando cualquiera de los dos tipos de scripts se puede acceder a escribir en la memoria y dichos datos pueden ser compartidos por todos los scripts existentes, puesto que la memoria es única y puede ser leída por cualquier Script.

4.3.4 DEFINICIÓN DE SCRIPT

Básicamente es una herramienta programable usada en los SmartImage Sensors (cámara 530MR). Cada script es como un programa que puede contener un número de funciones definidas por el usuario.

Los scripts pueden realizar un número de tareas pero no accederán datos que recoge un SoftSensor. Pueden acceder y modificar un producto y el parámetro de cada uno de los SoftSensors que lo conforman, establecer comunicación con dispositivos externos, preprocesar imágenes antes que el SoftSensor las analice, realizar cálculos matemáticos y más. En la Figura 4.7 se observa el nivel que ocupa cada tipo de Script dentro de la programación de la cámara. [3]

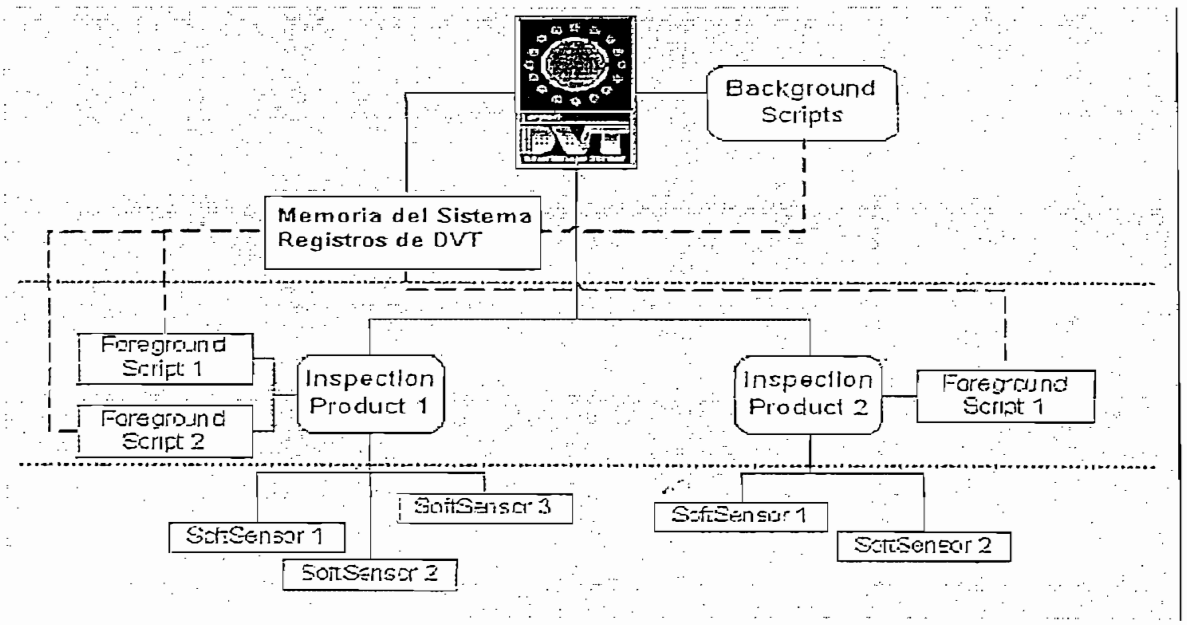


FIGURA 4.7: Backgrounds y Foregrounds.[11]

4.3.4.1 Background Scripts

Los Background scripts son creados a nivel del sistema. Cuando éste se inicializa se crea una clase llamada main(); aquí residen la mayoría de los script lógicos creados. Estos no se asocian con ninguna Inspección de Producto; es decir, son independientes de las inspecciones. No se ejecutan cada vez que una inspección tiene lugar, sino que pueden estar funcionando una vez que la cámara es

[11] DVT CORPORATION, DVT Script Reference Manual

activada; también pueden ser activados con una señal cuando el usuario configura alguna entrada para activar dicho background, e incluso manualmente cada vez que el usuario lo requiera. El último método se usa en simulaciones para propósitos de prueba. [3]

Puesto que los Background son creados a nivel del sistema, estos tienen acceso al sistema y parámetros del producto. Con ayuda de los background se puede establecer comunicación con dispositivos externos como un PLC.

4.3.4.2 Foreground Scripts

Los Foreground Scripts son creados a nivel del producto, y son diseñados para trabajar específicamente con las inspecciones. Cuando un Foreground es ejecutado, se inicia una clase llamada inspect(). Un Foreground corre cada vez que se hace una inspección del Producto. Un Foreground es creado como cualquier otro SoftSensor y se lo halla junto a éstos en el panel de FrameWork. La diferencia con el resto de SoftSensors es que estos pueden acceder a datos de los SoftSensors en la misma Inspección del producto. Además, se puede realizar cómputos matemáticos y lógicos, compartir información con otras áreas vía registros del sistema, y formación de caracteres para enviarlos fuera del sistema para compartirlos con otros SmartImage. [3]

4.4 PROCESO DE ANÁLISIS DE LA CAJA

Para analizar el estado de las botellas de cada una de las cajas que ingresan a la línea de embotellado es necesario seguir los siguientes pasos que se pueden apreciar en el flujograma de la Figura 4.8.

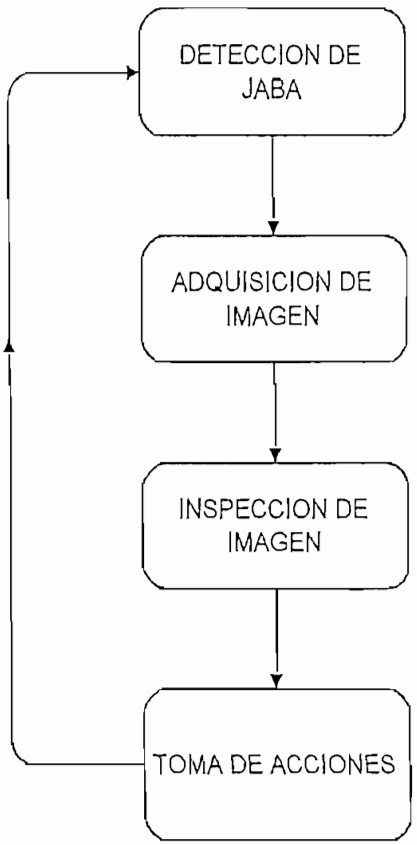


FIGURA 4.8: Secuencia para análisis de estado de las botellas.

4.4.1 BLOQUE: DETECCIÓN DE CAJA

Para detectar la presencia de una caja, se recurrió al uso de un sensor trigger, ya descrito en el Capitulo 3. La función del sensor consiste en emitir una señal eléctrica a la tarjeta de entradas (CON-IBOB), a la cual esta conectada, una vez que ha detectado la presencia de una caja.

El sensor, por supuesto, está ubicado en un sitio estratégico para la captura de la imagen de la caja. A la vez que la tarjeta de entradas recibe la señal del sensor, ésta indica a la cámara digital que es el momento de adquirir la imagen que esté presente en ese momento. Entonces las acciones que se ejecutan en este bloque son:

- Aproximación de jaba a inspector.
- Detección de jaba por parte del sensor trigger fotoeléctrico.
- Información a la tarjeta Con-Ibob por parte del sensor trigger fotoeléctrico.

4.4.2 BLOQUE: ADQUISICIÓN DE IMÁGENES

El ojo del Sensor SmartImage (Cámara 530MR) es un dispositivo tipo CCD (Charged-Coupled Device). Un CCD es un dispositivo que está dividido en pequeñas foto celdas llamados píxeles. Cada píxel mide la intensidad de la luz y la traduce a voltaje eléctrico. La resolución de una imagen se obtiene por el número de píxeles horizontales y verticales en un CCD. El sensor SmartImage Serie 530MR tiene una resolución de 640x480 píxeles; es decir posee 307200 píxeles por imagen. [1]

El cerebro del Sensor SmartImage (Cámara 530MR) es el microprocesador que se encarga de digitalizar los datos que captó píxel del CCD adquiriendo así una imagen para ser inspeccionada. [1}

Dentro del FrameWork existen varios parámetros configurables para que la imagen adquirida sea de mejor definición, con mayor o menor cantidad de luz, etc. Todos los parámetros que se describen a continuación se los puede programar usando la opción parámetros de imagen del menú principal, desplegándose una pantalla como la que se muestra en la Figura 4.9. [1]

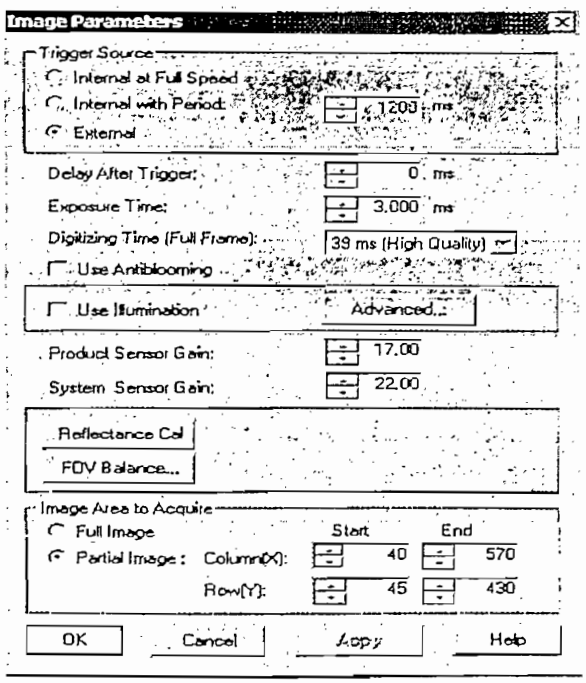


FIGURA 4.9: Parámetros para control de Imagen. [12]

[12] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario

4.4.2.1 Trigger Source

Se refiere al origen de la señal de trigger que recibe la cámara para adquirir una imagen, puede ser interna con intervalos de tiempo establecidos por el usuario o a máxima velocidad determinada por el procesador de la cámara y disparos externos generados por un sensor trigger externo, como es en este caso, esperando la señal del sensor fotoeléctrico de trigger que detecta la presencia de jabas. [1]

4.4.2.2 Tiempo de Exposición

El tiempo de exposición hace referencia al tiempo que el CCD localizado en la cabeza del sensor está expuesto a la luz durante la adquisición de la imagen. Mientras más largo es el tiempo de exposición, más luz entrará al CCD. Existen tres factores para escoger el tiempo de exposición. [1}

- **Velocidad de las partes Inspeccionadas:** Para partes de alta velocidad, se necesita de un periodo corto de exposición para minimizar imágenes borrosas.
- **Tiempo de Inspección:** Tomando en cuenta el número de partes por minuto a inspeccionar, se debe tomar en cuenta que para una alta frecuencia, se debe tratar de reducir el tiempo de exposición.
- **Luz disponible:** El tiempo de exposición se debe calibrar de acuerdo a las condiciones de luz existentes, condiciones influenciadas tanto por luces de fondo o luces propias de la cámara. Para reducir el tiempo de exposición es aconsejable incrementar la iluminación en el área de inspección, incrementar la ganancia en el CCD (con lo cual el CCD es más sensible a la luz), o ajustar el lente de apertura para dejar entrar mayor o menor cantidad de luz al lente.

Nota: al incrementar el tiempo de exposición, se hace mas lento el proceso de adquisición de la imagen, pudiendo dar origen a imágenes borrosas en inspecciones en movimiento.

4.4.2.3 Tiempo de Digitalización

Es el tiempo que la cámara utiliza para transformar todos los niveles de voltaje medidos en cada una de las celdas del CCD, en píxeles. Cada píxel se interpreta como una palabra digital de 8 bits (0-255); siendo el valor 0 para negro y el valor 255 para blanco. Sin embargo, los valores digitales obtenidos luego serán transformados a una nueva escala, con valores de 0 a 100 (0 para negro y 100 para blanco). Esta nueva escala se debe exclusivamente al modo de trabajo propio del Sistema de la cámara. [1}

4.4.2.4 Ganancia del Producto

Este control permite ajustar la ganancia de la imagen capturada y hace que el CCD de la cámara 530MR sea más o menos sensible a la luz. El valor que el usuario fije para éste parámetro será aplicado exclusivamente para el producto sobre el cual esté trabajando en ese momento. Si existiesen varios productos ejecutándose en la cámara; entonces, cada uno de ellos puede tener diferente valor en cuanto a la ganancia del Producto. [1]

4.4.2.5 Ganancia del Sensor

Este control permite ajustar la ganancia de la imagen capturada y hace que el CCD de la cámara 530MR sea más o menos sensible a la luz. El ajuste efectivo de ganancias le permite minimizar imágenes borrosas debido a largos periodos de exposición. Es recomendable cuando las partes analizadas estén en movimiento, incrementar la ganancia y reducir el tiempo de exposición para eliminar imágenes borrosas. El valor que el usuario fije para éste parámetro será aplicado para todos los productos que se estén ejecutando en la cámara. [1}

4.4.2.6 Ventana Parcial de la Imagen

Haciendo uso de esta herramienta se puede agilizar el tiempo de adquisición de una determinada imagen, ya que al programar al sensor 530MR para adquirir solo una determinada porción de píxeles de la imagen, se limita el tamaño de la misma. El tiempo que toma la adquisición de una imagen cuya resolución es de 640x480 píxeles es de 40ms, aproximadamente, hasta transferirla a la memoria de la cámara. Este tiempo es proporcional al número de píxeles que se adquieren.

Por tanto, si se reduce la ventana de adquisición a la mitad, el tiempo de adquisición también se reducirá en este caso a la cuarta parte. [1}

En base a los parámetros mencionados se puede resumir el funcionamiento de este bloque en las siguientes acciones:

- Tarjeta CON-IBOB informa a cámara mediante señal trigger de jaba en posición para adquisición de imagen.
- Cámara inicia tiempo de exposición de CCD a imagen de jaba.
- Cámara inicia tiempo de digitalización de imagen de jaba.
- Cámara recorta imagen para tener una ventana parcial de la imagen de la jaba.

4.4.3 BLOQUE: ANÁLISIS DE IMAGEN

Para analizar el estado de las botellas en las imágenes adquiridas se procedió a crear algunas versiones de programas que sirvieron para optar por un método mas eficiente para inspección de botellas vacías en sus respectivas jabas. Las versiones de programa que se desarrollaron se describen a continuación:

- **Versión 1.-**

  Esta versión de programa fue realizada exclusivamente con sensores virtuales de intensidad. Estos sensores miden el nivel de brillo y contraste que posee el area de imagen que se halla bajo ellos y en base a estos datos pueden generar un resultado.

  Al crear un programa con ellos se determinó que no servirían para ser utilizados de forma permanente pues al haber jabas que ingresan con botellas en diferentes niveles de suciedad el programa se vuelve inútil pues en ciertas ocasiones medía bien y en otras no; además es incapaz de reconocer botellas rotas. La ventaja que ofrece este programa es que fue rápido.

- **Versión 2.-**

Esta versión de programa utilizó sensores de virtuales busca objetos. Estos sensores son capaces de aprender una forma que el usuario determine y buscarla en cada inspección que se realice dando en base a esta búsqueda una respuesta.

El programa que se creó con ellos determinó que serían de difícil calibración pues en ciertas ocasiones habiendo una botella correcta daba un resultado de Falla y además fue de mayor importancia que este programa volvía lenta a la cámara ocupando mas tiempo del que se disponía que es de alrededor de 700ms.

- **Versión 3.-**

Esta versión de programa utilizó sensores virtuales de conteo de píxeles los cuales miden porcentajes de píxeles claros y oscuros contabilizando la cantidad de píxeles que posee en cada nivel de intensidad. En base a esos datos el sensor arroja resultados.

Al igual que la versión 1, esta versión de programa tampoco funcionó por la variación en el nivel de suciedad que poseen las botellas.

- **Versión 4.-**

Esta versión de programa utilizó sensores de manchas, los cuales agrupan píxeles claros de oscuros y generan manchas que posteriormente serán medidas y contadas para generar un resultado.

Este programa hubiera funcionado bien si las botellas fueran uniformes siempre, pero el inconveniente que presento es que con cada variación en la intensidad de las botellas este sensor debía ser recalibrado para que vuelva a funcionar. Además este programa era relativamente rápido.

NOTA: Se referirá a Botellas en Mal estado, a las botellas rotas o faltantes. Y botellas en buen estado, a las que no estén rotas. Se refiere a Cajas Buenas a aquellas en las que no existe Botellas en Mal estado.

Para conseguir los ítems señalados, se aplicó un producto que costa de los siguientes sensores:

a) 1 Sensor para Presencia de Caja.
b) 12 sensores de Intensidad para cada botella.
c) 1 Sensor Script para calibrar Thresholds de Sensores de Blobs.
d) 12 sensores para verificar buen estado de las botellas.
e) 1 Sensor Script para acumular datos de los sensores anteriores.

Los sensores cumplen la tarea de examinar si la imagen es correcta o no, y se reúnen en una secuencia de programación como se muestra en el flujograma de la Figura 4.10.

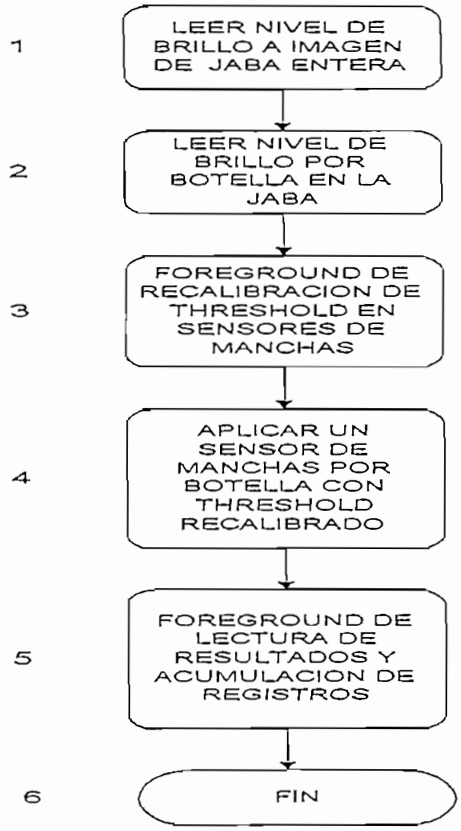


FIGURA 4.10: Secuencia para análisis de imagen capturada.

- **En el primer bloque:**
  - Aplicar un ***sensor de intensidad*** a toda la jaba para asegurar presencia de jaba en caso de falsa señal de trigger.
  - Medir porcentaje de brillo de la imagen con el sensor de intensidad.
  - Si el resultado del porcentaje de brillo de la jaba esta bajo el nivel calibrado dar resultado PASS de lo contrario dar FAIL.
- **En el segundo bloque:**
  - Aplicar **12** ***sensores de intensidad*** (uno para cada botella).
  - Medir el porcentaje de brillo de las botellas con su sensor de intensidad correspondiente. Esto pues cada botella llega a la línea de embotellado con diferentes niveles de suciedad.
- **En el tercer bloque:**
  - Foreground Script que toma los porcentajes de Brillo de cada uno de los 12 sensores de intensidad ya mencionados, y calibra automáticamente el valor de % de ***Thresholds*** de cada uno de los ***Sensores de Blobs*** siguientes para darles uniformidad en su análisis. Se explica este bloque con el flujograma de la Figura 4.11a.
- **En el cuarto bloque:**
  - Aplicar **12** ***sensores de manchas*** uno por botella con threshold recalibrado.
  - Mostrar manchas oscuras que tengan el tamaño precalibrado para el pico de las botellas.
  - Contar el número de manchas mostradas.
  - Si el numero de manchas esta en el rango precalibrado dar PASS de lo contrario dar FAIL.
- **En el quinto bloque:**
  - Foreground Script de contabilización de resultados y almacenamiento de registros. Se explica este bloque con el flujograma de la Figura 4.11b.
- **En el sexto bloque:**
  - Fin de la inspección

Todos los valores de intensidad y su valor correspondiente de Threshold fueron encontrados en forma experimental; es decir, viendo el comportamiento de cada

uno de los sensores y cambiando manualmente los valores de Threshold. La secuencia del flujograma de la Figura 4.10 se ejecuta con cada adquisición de imagen.

En los flujogramas de la Figura 4.11 se puede ver la secuencia que sigue el Bloque 3 y el Bloque 5 del flujograma de la Figura 4.10.

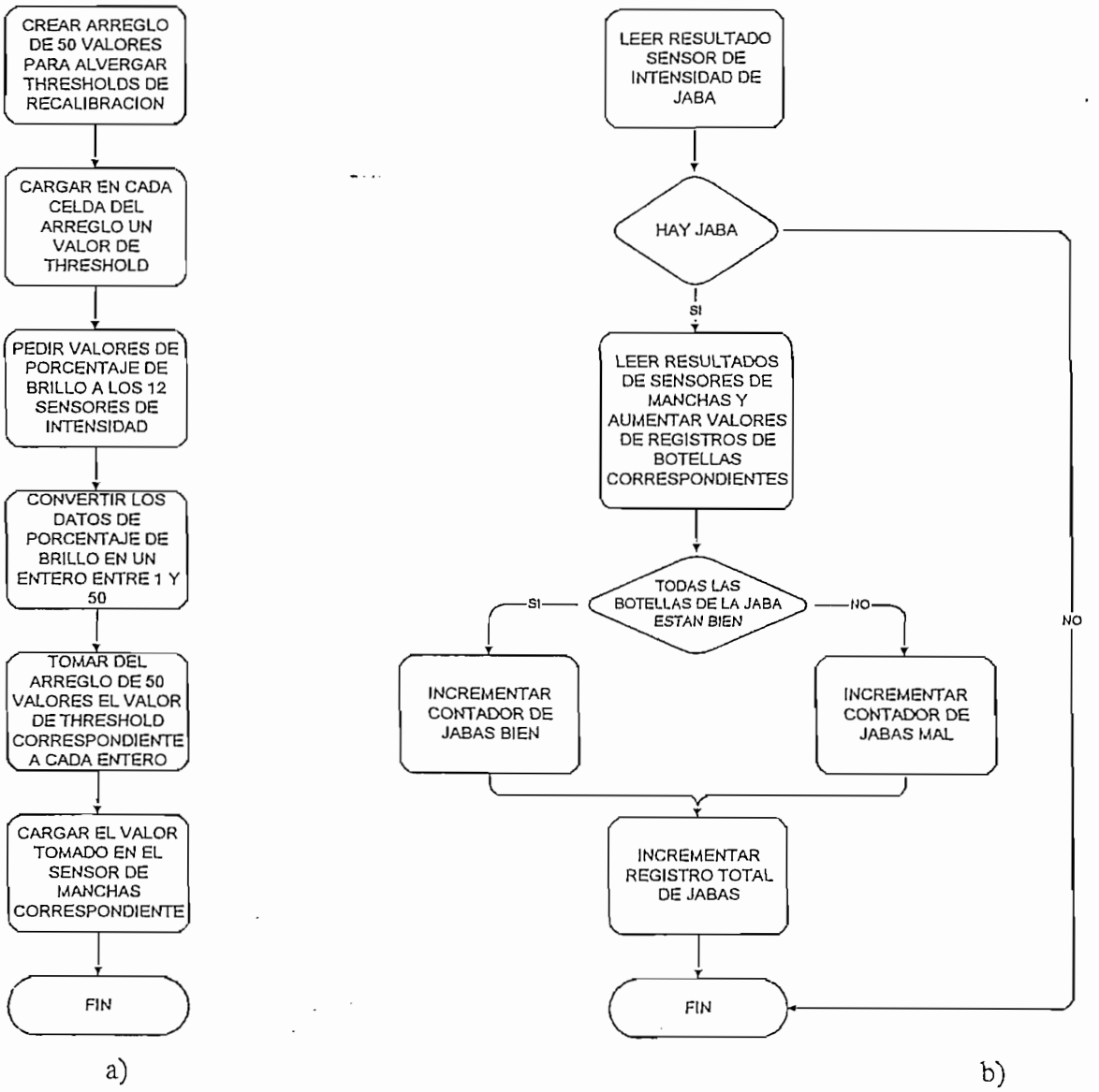


FIGURA 4.11: a)Foreground de recalibración de Thresholds,
b) Foreground de conteo y acumulación de datos.

A continuación se describe la secuencia de acciones que se realiza en el flujograma de la Figura 4.11a.

- Crear un arreglo de 50 valores.
- En cada celda del arreglo se carga un valor de threshold que se utilizara en la recalibración de los sensores de manchas.
- Se lee el porcentaje de brillo que midieron los 12 sensores de intensidad ubicados sobre cada botella.
- Cada valor del porcentaje de brillo es convertido en un entero entre 1 y 50.
- Con los enteros obtenidos se consulta en el arreglo creado antes y se toma el valor que corresponda a cada entero.
- Cada dato es cargado a su respectivo sensor de manchas para su inspección.
- Termina la subrutina del Foreground de recalibración de Thresholds.

La siguiente secuencia de acciones corresponde al flujograma de la Figura 4.11b.

- Leer el resultado del sensor de intensidad de jabas
- Si el resultado fue PASS continua de lo contrario termina la subrutina del Foreground para contabilización y almacenamiento de datos.
- Con PASS en el sensor de intensidad de jabas: lee el resultado de los sensores de manchas
- Incrementa contador de botellas buenas si existieran.
- Incrementa contador de botellas malas si existieran.
- Si en la jaba todas las botellas están correctas, incrementa contador de jabas bien.
- Si en la jaba existiera por lo menos una botella con falla, incrementa contador de jabas mal.
- Incrementa contador total de jabas.
- Termina subrutina de Foreground para contabilización y almacenamiento de datos.

4.4.3.2 Visualización de Resultados

Existen varias formas de visualizar los resultados arrojados por el producto aplicado. Por ejemplo, una vez que un Producto es ejecutado arroja una Tabla de Resultados como el de la Tabla 4.12. [1]

Result Table: DetecciónBotellas1

Overall Result	PASS	
SoftSensor	Result	Output Value
DetectorJaba	PASS	Bright Area = 70.42%
ControlJaba	PASS	
cb1	PASS	Bright Area = 17.90%
cb2	PASS	Bright Area = 47.34%
cb3	PASS	Bright Area = 59.12%
cb4	PASS	Bright Area = 26.70%
cb5	PASS	Bright Area = 70.38%
cb6	PASS	Bright Area = 81.49%
cb7	PASS	Bright Area = 55.70%
cb8	PASS	Bright Area = 50.04%
cb9	PASS	Bright Area = 49.43%
cb10	PASS	Bright Area = 59.76%
cb11	PASS	Bright Area = 37.63%
cb12	PASS	Bright Area = 25.23%
ControlThresholdCa	PASS	
sb1	PASS	# Blobs = 2
sb2	PASS	# Blobs = 1
sb3	PASS	# Blobs = 2
sb4	PASS	# Blobs = 2
sb5	PASS	# Blobs = 2
sb6	PASS	# Blobs = 1
sb7	PASS	# Blobs = 2
sb8	PASS	# Blobs = 2
sb9	PASS	# Blobs = 1
sb10	PASS	# Blobs = 1
sb11	PASS	# Blobs = 1
sb12	PASS	# Blobs = 1
AnalisisBotellasCam	PASS	

TABLA 4.12: Tabla de Resultados.[13]

En la Tabla de Resultados 4.12 aparece cada uno de los sensores que pertenecen al Producto, y cada vez que una imagen es analizada los valores de los sensores van cambiando. El orden en que cada uno de los sensores aparece en la tabla de resultados corresponde al orden en el que el usuario los creó. En la

[13] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario

tabla de resultados aparece información como: el nombre de cada sensor (en el lado izquierdo de la tabla); además, los resultados Pasa/Falla de cada sensor y detalles como número de objetos encontrados, número de manchas halladas, porcentaje de Brillo, etc., de acuerdo a la característica propia del sensor que se esté usando.

Para este proyecto se usó sensores de Blob (manchas), por ello en la tabla de resultados aparece # de manchas halladas por cada sensor.

También se puede desplegar una pantalla que hace un histórico de los resultados del Producto aplicado, tal como se muestra en la Figura 4.12. Aquí aparece el número total de Imágenes correctas **"Pass Count"**, número total de Imágenes fallidas **"Fail Count"**, y todos estos valores traducidos en porcentajes.

NOTA: Se refiere a Imagen Correcta a aquella en la que todos los sensores de producto dieron como veredicto PASA. Mientras que para Imagen Fallida se refiere a aquella en que uno o más sensores del Producto dieron como veredicto FALLA.

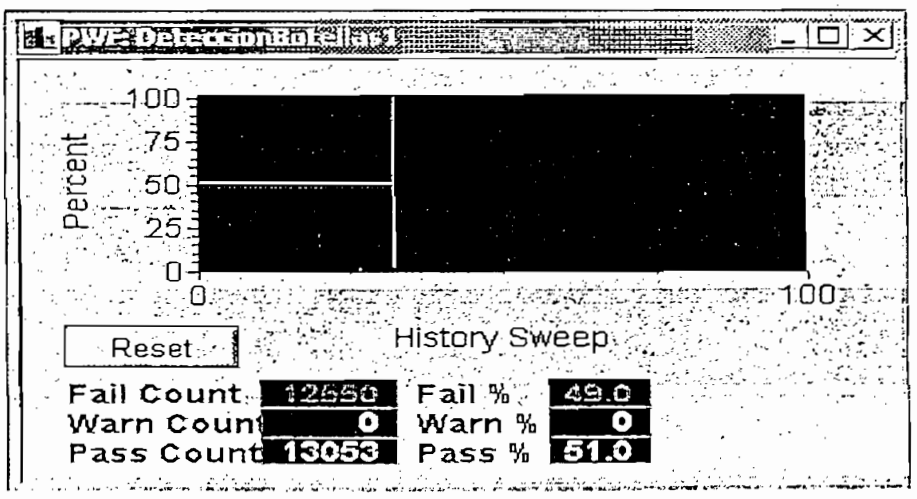
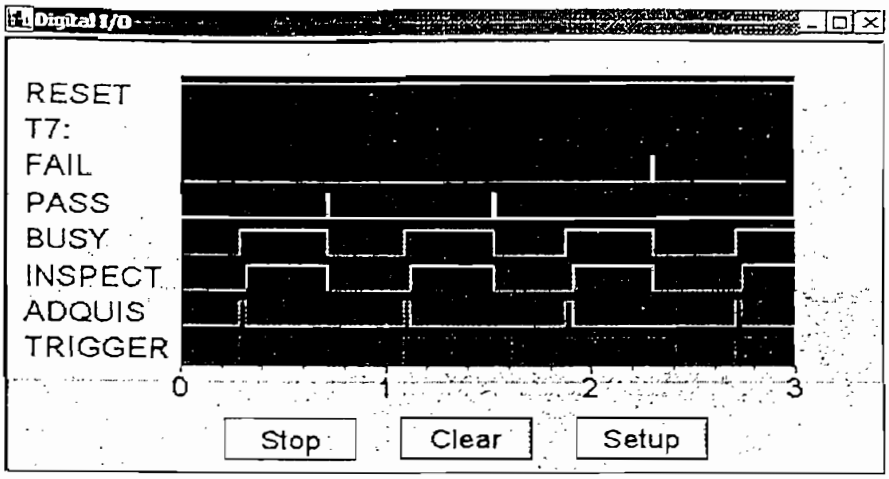


FIGURA 4.12: Gráfica de Estadísticas de Pasa/Alerta/Falla.[14]

Existe además una pantalla de I/O en función del tiempo que como ya se dijo, los 8 canales se pueden configurar según la necesidad del usuario. En la Figura 4.13 se puede apreciar el uso para cada uno de los canales.

[14] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario



[15]

FIGURA 4.13: Diagrama de tiempos de los canales de la tarjeta de I/O (CON-IBOB).

En el esquema de la Figura 4.13 se ve la función que se asignó a cada uno de los canales y que se señalan en la Tabla 4.1

CANAL	FUNCIÓN
1	Señal de Trigger
2	Tiempo de Adquisición
3	Tiempo de Inspección
4	Tiempo de ocupado (busy)
5	Señal de Pass
6	Señal de Falla
7	No tiene función
8	Reset (reseteo de contadores)

TABLA 4.1: Función de cada uno de los pines de la tarjeta de I/O.

Nota: El tiempo de Ocupado es el resultado del tiempo de inspección más el tiempo de adquisición.

[15] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario

La Figura 4.13 es de mucha importancia para el programador, ya que con este diagrama de tiempos se aprecia el tiempo que el Producto utilizado está ocupando, respecto al tiempo que el sensor Trigger está activo. Si el tiempo de Busy (tiempo de ejecución del producto) es mayor que el tiempo que el Trigger está activo; entonces el Producto y sus sensores deben ser cambiados de tal forma que esta situación adversa con los tiempos, se invierta. Caso contrario, aún cuando el sensor advierta a la cámara de la presencia de otra caja, la imagen advertida pasará por alto y no será analizada.

Se cuenta además con otro gráfico de tiempos, que se muestra en la Figura 4.14. En esta gráfica se puede acceder a datos como tiempo máx./mín. de inspección del Producto, así como datos de tiempos máx./mín. entre las inspecciones realizadas.

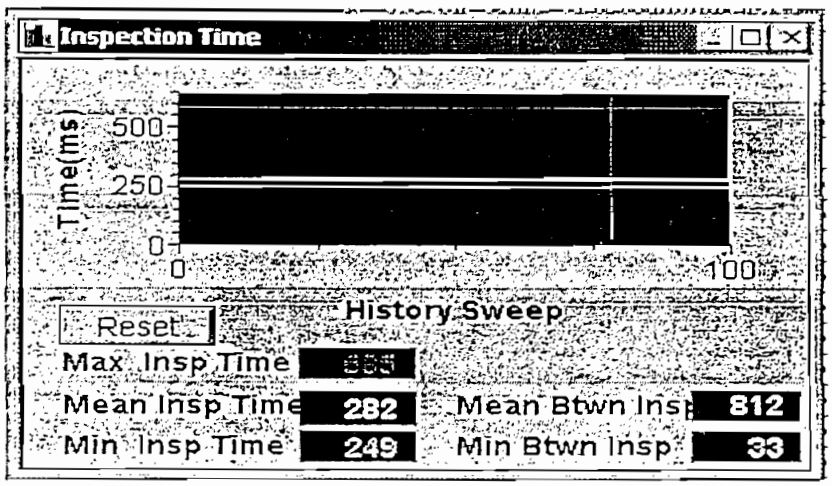


FIGURA 4.14: Gráfica de tiempos empleados por el producto activo.[16]

4.4.4 BLOQUE 4 : TOMA DE ACCIONES

Una vez que se realiza la inspección, la toma de acciones que se ejecutó para el proyecto es el almacenamiento de datos para que sean visualizados desde las interfaces que se describen a continuación. Estas interfaces son las que piden los datos a las cámaras en el instante que se lo requiera..

[16] DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario

4.5 DISEÑO DE LA INTERFAZ CON EL USUARIO

Se recurrió al programa Visual Basic para desarrollar la interfaz gráfica para que el usuario tenga acceso a los datos almacenados por la cámara 530MR, así como a las imágenes fallidas guardadas por la misma.

Para ello se diseñaron y desarrollaron 3 pantallas. La primera es la pantalla de presentación del proyecto, tal como se ve en la Figura 4.15, la segunda es una pantalla de acceso a datos almacenados por la cámara; datos como # de botellas en buen estado, # de botellas malas, # cajas buenas, # cajas malas, tal como se ve en la Figura 4.17 y la tercera es una pantalla que se despliega solo en caso que el usuario quiera acceder a las imágenes fallidas que la cámara haya almacenado, tal como se ve en la Figura 4.25.

Para entender mejor las pantallas creadas, a continuación se describen los diagramas de flujo de las pantallas desarrolladas.

4.5.1 PANTALLA DE PRESENTACIÓN

La pantalla de la Figura 4.15 es explicada con el diagrama de flujo de la Figura 4.16.

FIGURA 4.15: Pantalla de presentación del proyecto.

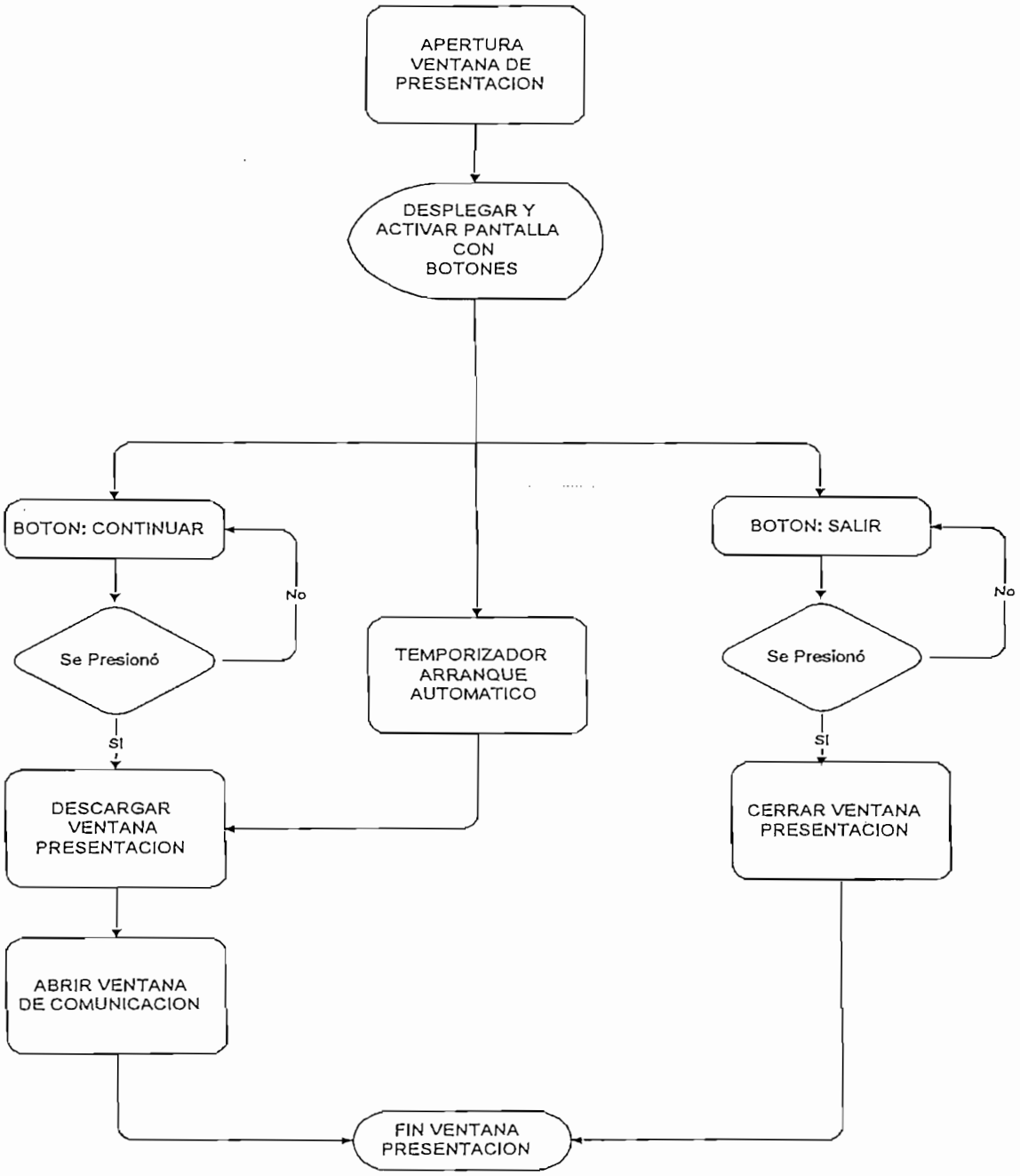


FIGURA 4.16: Diagrama de flujo de la pantalla de la Figura 4.15

Secuencia de acciones que se ejecutan en esta ventana:

- **Cuando se carga la ventana.**
  - Activar botones.
  - Especificar rutas para los botones
  - Activar configuraciones de la ventana.

- Desplegar ventana
- Activar temporizador de continuación automática.

- **Si se presiona Botón Continuar**
  - Cargar la pantalla de comunicación mostrada en la Figura. 4.17
  - Finalizar la pantalla de presentación mostrada en la Figura 4.15
- **Si se presiona Botón Salir**
  - Terminar la pantalla de presentación mostrada en la Figura. 4.15

4.5.2 PANTALLA DE CONEXIÓN

La pantalla de la Figura 4.17 es explicada con el diagrama de flujo de la Figura 4.18

NOTA: Cada cámara funciona en forma independiente, así que los datos e imágenes que el usuario requiera también lo son. Por tanto, de aquí en adelante la explicación de la programación para una de las cámaras es aplicable a la otra.

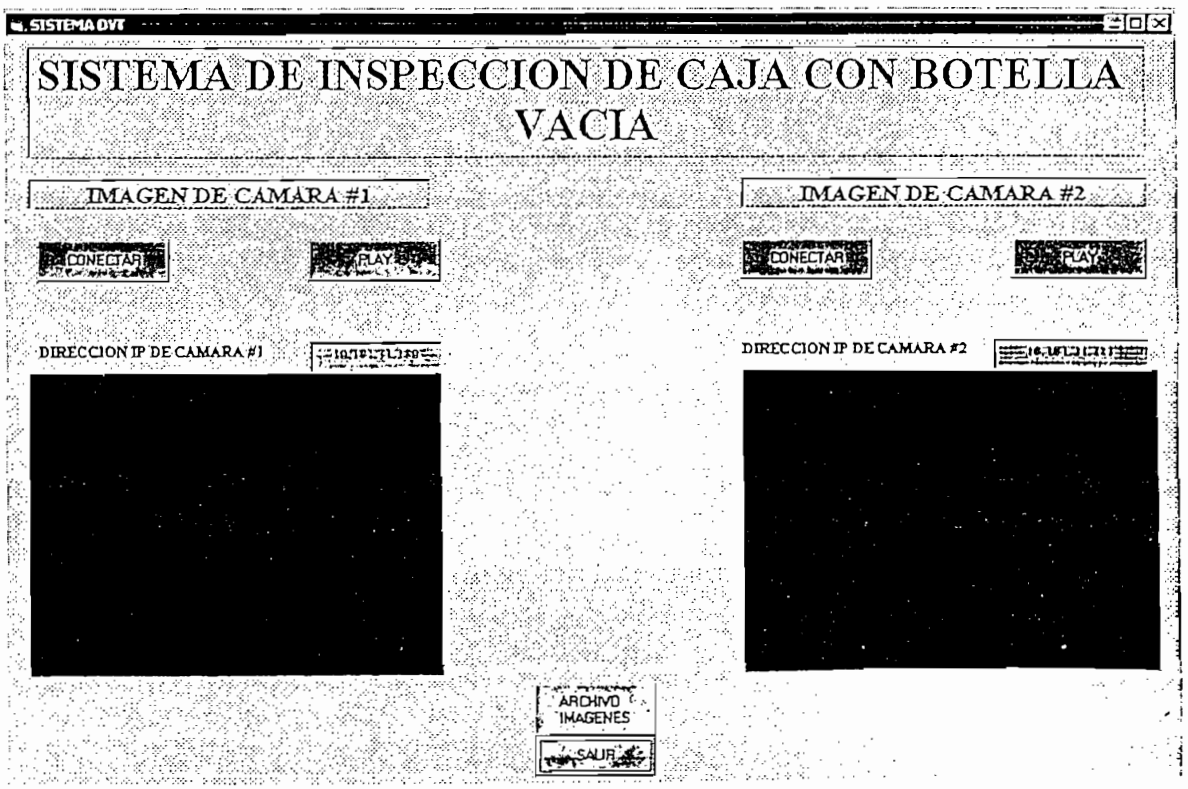


FIGURA 4.17: Descarga de la ventana de Conexión.

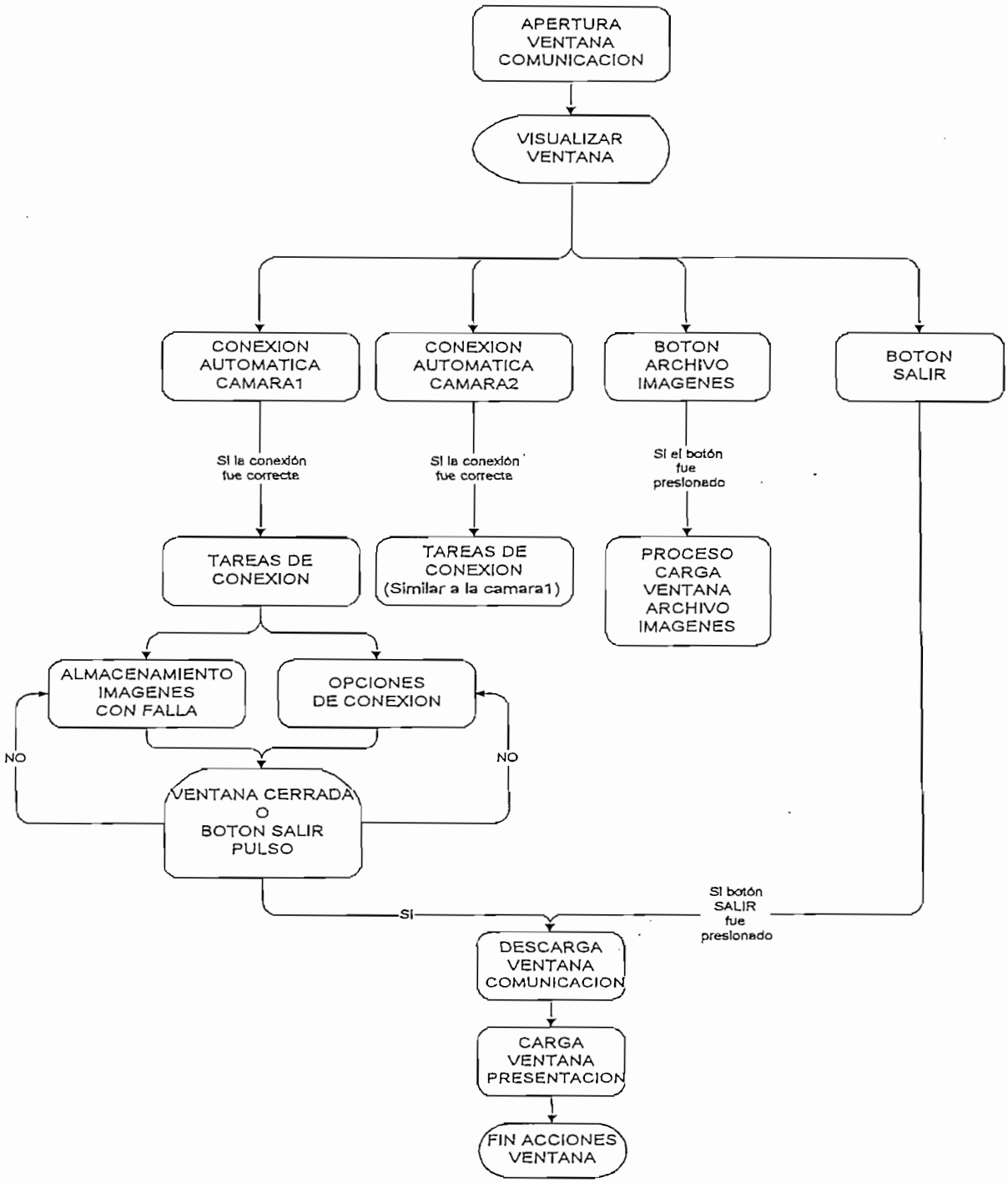


FIGURA 4.18: Diagrama de flujo para la pantalla de la Figura 4.17

La secuencia de acciones que tiene esta pantalla se describe a continuación:

- **Cuando se carga la ventana.**
  - Desactivar botones: conectar, play.

- Activar botones: archivo de imágenes, salir.
- Especificar rutas para los botones.
- Ocultar cuadros de visualización de contadores.
- Activar configuraciones de la ventana.
- Desplegar ventana
- Activar temporizador de conexión automática.

- **Bloque de conexión automática de cámara.**
  - Se describe con el flujograma de la Figura 4.19.

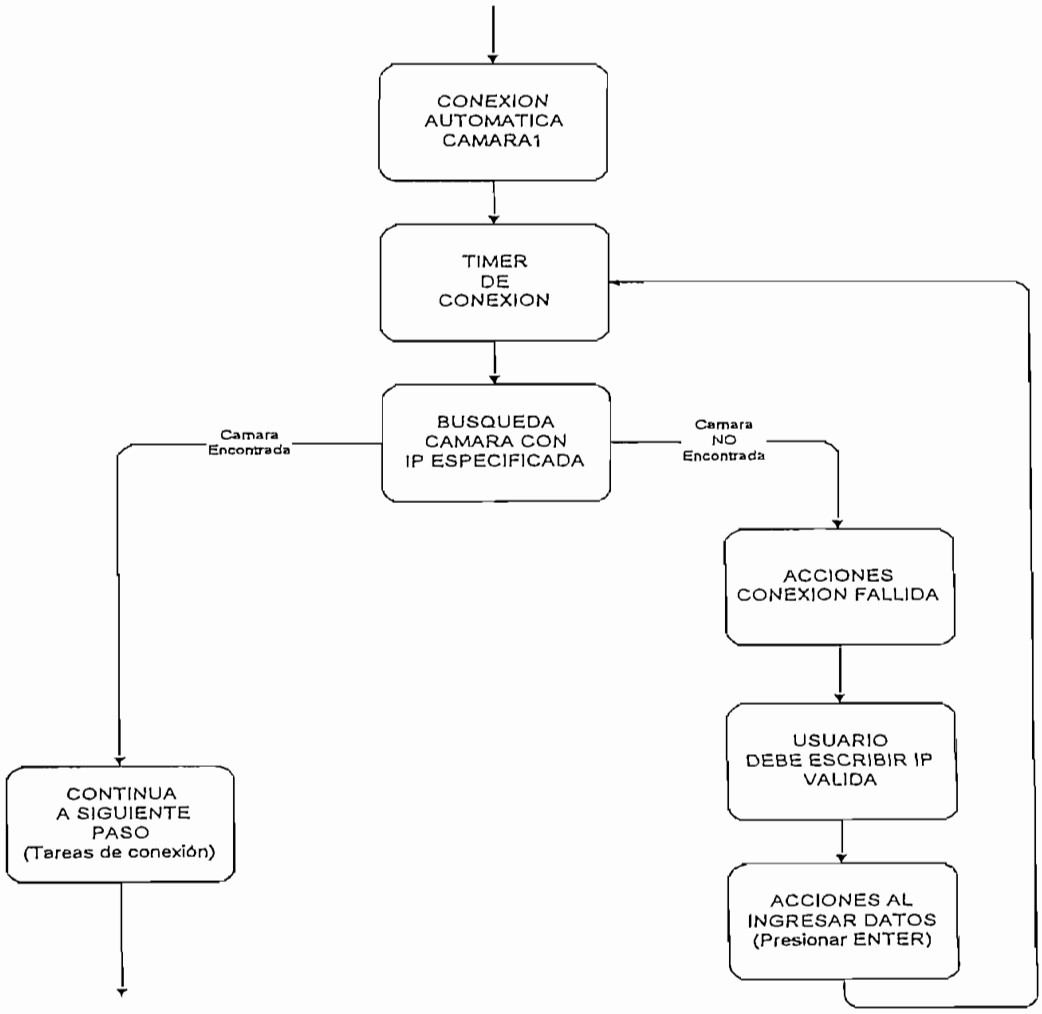


FIGURA 4.19: **Proceso de conexión automática de las cámaras.**

- Con el temporizador de conexión automática activo
- Configurar socket de conexión a redes ethernet.
- Ingresar a la red ethernet de Cervecería Andina.

- Buscar la cámara con la dirección IP preconfigurada.
- Si la cámara no es hallada:
  - Mostrar mensaje de conexión Fallida
  - Activar cuadro de ingreso de dirección IP
  - Escribir una IP correcta y presionar Enter
  - Se vuelve a ejecutar el bloque de conexión automática descrito en la Figura 4.19
- **Si la cámara fue hallada:**
  - Activar botones: conectar, play.
  - Continúa con el proceso a bloque Tareas de conexión.

- **Bloque Tareas de Conexión.**
  - Se describe con el flujograma de la Figura 4.20

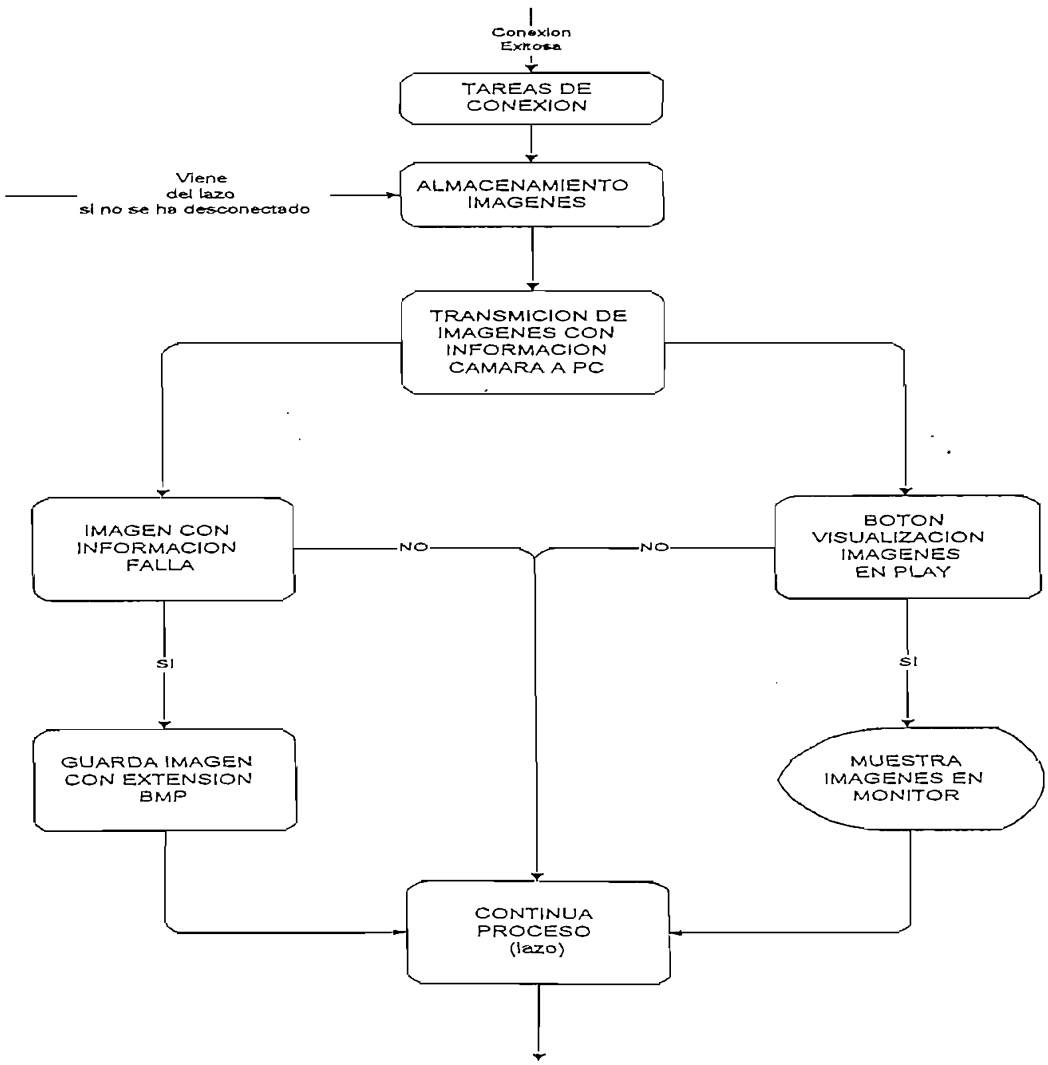


FIGURA 4.20: Proceso de Transmisión de imágenes.

- Programa de visual pide a cámara que envié imágenes con información.
- Cámara transmite imágenes.
- Identificar imágenes con falla.
- Si imagen tiene falla almacenar con extensión BMP.
- Si se presiona botón Play:
  - Cambia letrero de botón Play a Stop
  - Recibir imágenes que llegan en el visor de imágenes.
  - Desplegar imágenes.
  - Si se presiona el botón ahora con letrero Stop se detiene la visualización de imágenes
- Si se presiona botón Conectar:
  - Cambia letrero de botón Conectar a Desconectar
  - Visualizar cuadros de contadores.
  - Pedir a cámaras el envió de registros de contadores.
  - Recibir datos y ubicarlos en los cuadros de contadores.
  - Desplegar cuadros de contadores con datos.
  - Esperar 1 segundo y repetir el ciclo de pedido de registros de contadores a cámaras.
  - Si se presiona el botón de desconectar se ocultan los cuadros de contadores y se detiene el pedido de registros de contadores a cámara.

- **Cuando se presiona botón Archivo de Imágenes.**
  - Se describe en el Subcapítulo 4.5.3
- **Si se presiona Botón Salir.**
  - Cerrar todos los sockets de conexiones.
  - Cerrar las cuadros, botones y visores.
  - Detener las peticiones de datos a la cámara.
  - Descargar la ventana de Comunicaciones.
  - Cargar la ventana de Presentación.

Una vez que se ha descargado la ventana de comunicaciones se carga la venta de Presentación y se repite el flujograma de la Figura 4.16

4.5.3 VENTANA DE ARCHIVO DE IMÁGENES

La Figura 4.21 corresponde a la ventana de Archivo de Imágenes y es descrito con el flujograma de la Figura 4.22.

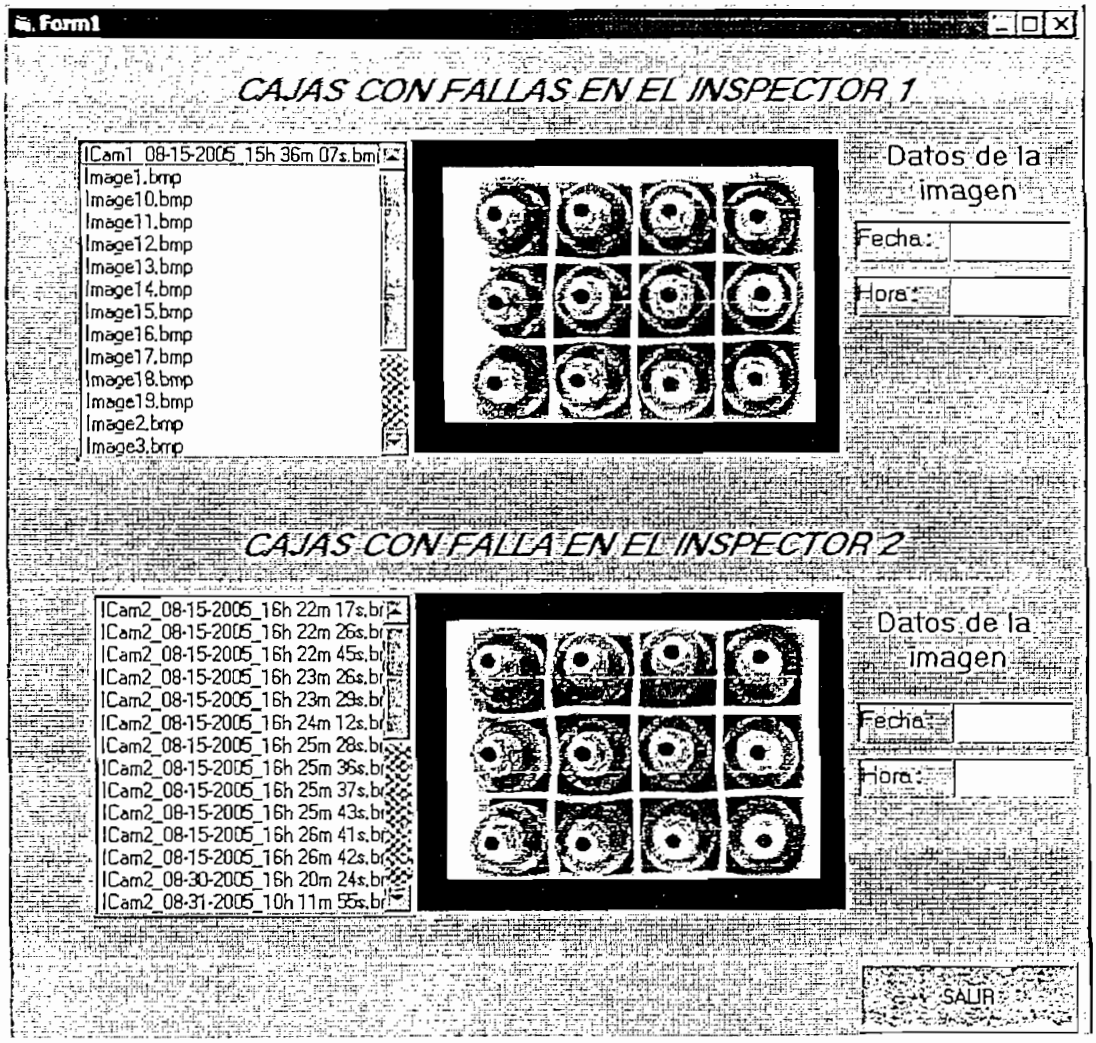


FIGURA 4.21: Ventana de Visualización de Imágenes Fallidas.

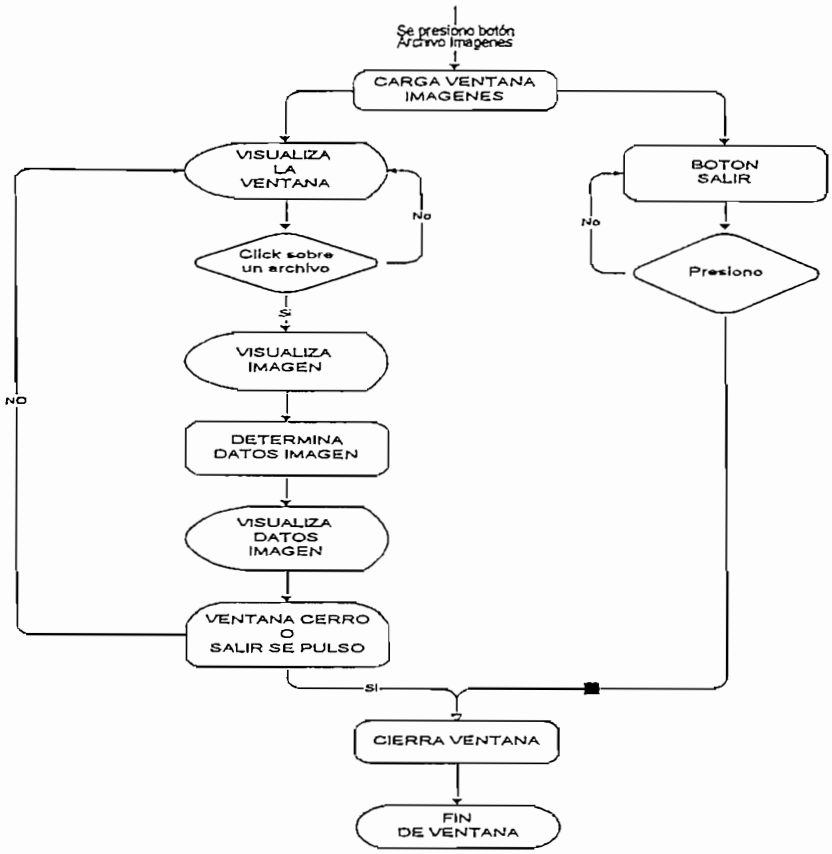


FIGURA 4.22: Diagrama de flujo de la pantalla de la Figura 4.21

- Poner en segundo plano la ventana de comunicaciones.
- Activar botones, visores de información de la imagen, visor de archivos y visor de imágenes.
- Leer el archivo de imágenes en el disco duro.
- Extraer los registros y visualizarlos en el visor de archivos.
- Desplegar ventana
- Si se selecciona un archivo:
  - Localizar archivo.
  - Visualizar en el visor de imágenes.
  - Leer la información de la imagen.
  - Ordenar la información.
  - Mostrar la información en los cuadros de información de hora y fecha de grabación de la imagen.
- Si se presiona el botón Salir:
  - Descargar todos los botones y cuadros de la ventana.
  - Cerrar ventana de Archivo de Imágenes.

- Activar ventana de comunicaciones.

A continuación se ve una pantalla que se puede visualizar desde cualquier computadora que pertenezca a la red de Cervecería Andina. A diferencia de las pantallas de Visual Basic que requieren de una computadora donde se halle instalado la HMI desarrollada, esta ventana no requiere de una PC especifica sino que la PC que solicita la información tenga instalado el Protocolo TCP/IP y el software del Internet Explorer.

4.5.4 PANTALLA DE INTERNET

Tal como se ve en la Figura 4.23, ésta es una página de Internet en donde se despliegan datos registrados por las cámara, como por ejemplo: # botellas en buen estado, # botellas en mal estado, # cajas buenas y malas. [10]

Para explicar como se desarrolló la pantalla de la Figura 4.23 se recurre al esquema de la Figura 4.24

FIGURA 4.23: Página de Internet con datos de la cámara.[17]

[17] www.dvtsensors.com

FIGURA 4.24: Programación para descargar página de Internet.

- PC solicita información de cámara con IP digitada en el Internet Explorer
- PC busca la dirección en la red local.
- Cuando halla la dirección halló la cámara.
- PC notifica a cámara que entre en conexión.
- Cámara empieza la conexión.
- Cámara recoge datos de contadores.
- Cámara configura pagina HTML con datos de contadores.
- Cámara envía la pagina HTML a PC solicitante.

- PC recibe página y la visualiza.
- Cámara cierra la conexión.

Con el software descrito se procedió a calibrar el programa con el fin de que funcione óptimamente. Estas calibraciones se explican en el Capitulo 5.

CAPITULO 5
PRUEBAS Y RESULTADOS

Este capitulo muestra los pasos que se tomaron para realizar las pruebas de adquisición de imágenes y de las pruebas que se hicieron para obtener los valores para recalibración de los Thresholds de los sensores de manchas que se encargan de dictaminar si la botella se encuentra correcta, rota o ausente. Además se indican los resultados que se obtuvieron de dichas pruebas.

5.1 PRUEBAS DE CALIBRACIÓN

Las pruebas que se realizaron a continuación están enfocadas a encontrar valores de calibración que permitan a la cámara adquirir una imagen correcta así como la calibración adecuada del programa inspector de botellas. Todas las pruebas realizadas son de tipo experimental.

5.1.1 PRUEBA DE CALIBRACIÓN DE IMÁGENES

Esta prueba fue ejecutada bajo la condición que la línea de envase se halle en funcionamiento. Esa condición fue necesaria, para reflejar en las pruebas una situación real y así al estar en movimiento las jabas, encontrar valores reales que permitirán disminuir el ruido en la imagen adquirida debido al movimiento. De esta forma se obtendría un programa inspector que funcione de manera sobre todo confiable.

Para cumplir la condición anterior los pasos que se tomaron para realizar esta prueba son:

- Iniciar el programa de configuración de la cámara (FrameWork) en modo de conexión, esto con el fin de adquirir las imágenes en tiempo real para su configuración.
- Abrir la ventana de configuración de imágenes del programa.
- Manipular los parámetros de esta ventana observando la imagen que se fue adquiriendo hasta obtener imágenes claras, libres de ruido debido al movimiento de las jabas.

En la Tabla 5.1 se indican los cambios más relevantes desde los valores iniciales hasta los valores con los que se obtuvo buenos resultados.

Tiempo de Exposición [ms]	Tiempo de digitalización [ms]	Ganancia del Producto [u]	Ganancia del Sensor [u]	Observación
8.3	26	2.01	2	Parámetros originales
5	26	2.01	2	Parámetros De Prueba
3	39	2.01	2	
2	39	2.01	5	
2	39	5.01	5	
....	....	....	....	
2	39	5	10	
2	39	9.11	10	
2	39	9.11	14	Parámetros Finales

TABLA 5.1: Valores de calibración de imagen

5.1.2 PRUEBAS DE CALIBRACIÓN DE SENSORES

Las pruebas de calibración de sensores se realizaron en modo de simulación y con imágenes grabadas, esto con el fin de tener tiempo para colectar datos, modificar valores y repetir las imágenes.

Una vez que se obtuvieron imágenes nítidas para inspeccionar se procedió a grabar dos secuencias de imágenes para las dos pruebas de calibración que se diseñaron.

La primera secuencia de 100 cuadros contenía imágenes con y sin jabas a un mismo nivel de intensidad de luz. Esta secuencia nos serviría para realizar la prueba que se describe en el literal 5.1.2.1. La segunda secuencia de 500 cuadros contenía imágenes con jabas a diferente nivel de intensidad de luz. Esta secuencia nos serviría para realizar la prueba que se describe en el literal 5.1.2.2

Como se describió en el Capítulo 4, Sección 4.4.3.1, el programa de inspección está compuesto por seis secciones que se ejecutan secuencialmente en el orden descrito a continuación:

a) 1 Sensor para Presencia de Caja.
b) 12 Sensores de Intensidad para cada botella.
c) 1 Sensor Script para calibrar Thresholds de Sensores de Blobs.
d) 12 Sensores de manchas para verificar buen estado de las botellas.
e) 1 Sensor Script para acumular datos de los sensores anteriores.

Las dos pruebas de calibración que se diseñaron están dirigidas a obtener los mejores valores para el Sensor de Detección de Jaba y para los Sensores de manchas de los literales a) y d) respectivamente.

5.1.2.1 Prueba 1 de calibración: Sensor de Presencia de Caja

Esta prueba se la realizó para determinar si la imagen que adquirió la cámara contiene una jaba o hubo una falsa detección debida a las vibraciones de la transportadora.

Para esta prueba se usó un sensor de intensidad que mide dos parámetros de la imagen que se halla bajo el sensor. En base a cualquiera de esos dos parámetros, o a ambos, el sensor puede dar un resultado de PASO o FALLA. Las dos características que puede medir este sensor son:

- Medida del porcentaje de contraste
- Medida del porcentaje de brillo

La prueba utilizó la respuesta: medida del porcentaje de brillo de la imagen bajo el sensor. Esa medida permite obtener un estado de presencia (PASS) o ausencia (FAIL) de la jaba en la imagen adquirida.

Para proceder a la prueba de calibración del sensor de detección de jaba, se ejecutaron los siguientes pasos:

- Iniciar el programa de configuración de la cámara en modo de simulación.
- Cargar en el simulador la primera secuencia de imágenes grabadas.
- Asignar al sensor de Presencia de Caja, las configuraciones fijas que se muestra en la Tabla 5.2.
- Hacer correr la imágenes conjuntamente con el sensor aplicado
- Recoger datos del sensor para cada imagen en la Tabla 5.3
- Obtener en Excel el valor máximo para calibrar imágenes con jaba y el mínimo para calibrar imágenes sin jaba.
- Calcular un promedio entre el máximo y el mínimo, para luego cargar dicho valor al sensor. Los valores se hallan en la Tabla 5.4

Tipo de Threshold	Intensity: Fixed Value
Nivel del Threshold	50
Porcentaje del nivel de llenado	100

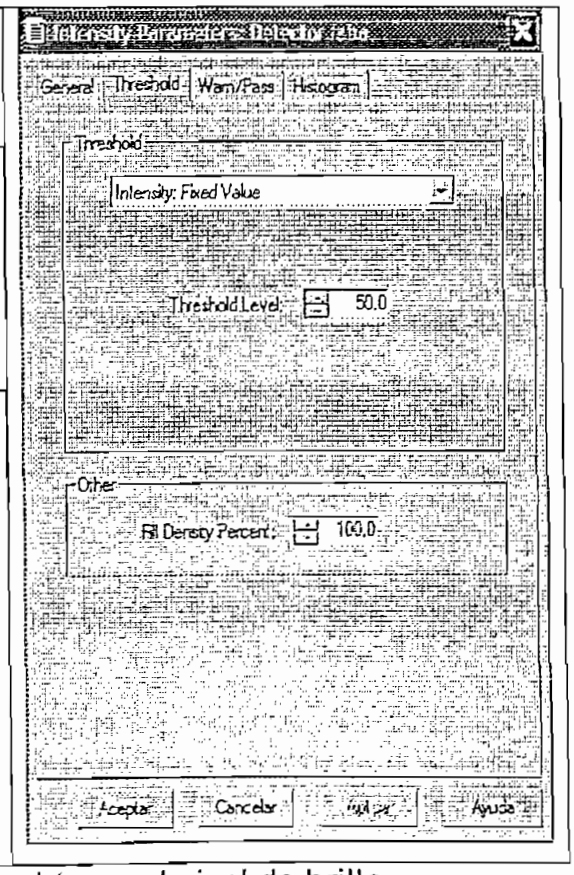


TABLA 5.2: Calibraciones para obtener el nivel de brillo.

Nro. Imagen	Porcentaje de Brillo con Jaba	Porcentaje de Brillo sin Jaba
1	48.62	—
2	46.85	—
3	—	100
4	54.32	—
5	—	97.56
6	65.09	—
7	—	100
8	79.13	—
9	—	99.42
10	80.04	—
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....
86	—	99.88
87	93.32	—
88	89.56	—
89	—	97.71
90	94.36	—
91	—	100
92	76.39	—
93	—	100
94	92.87	—
95	93.12	—
96	—	100
97	94.25	—
98	—	100
99	—	100
100	88.78	—

TABLA 5.3: Valores de porcentajes de brillo obtenidos.

Valor máx del porcentaje de brillo para imágenes con Jabas	94.96
Valor mín del porcentaje de brillo para imágenes sin jabas o incompletas	97.51
Promedio	96

TABLA 5.4: Valores Máximo y Mínimo.

En la Figura 5.1 se observa la asignación del valor calculado al sensor de intensidad: Presencia de Caja.

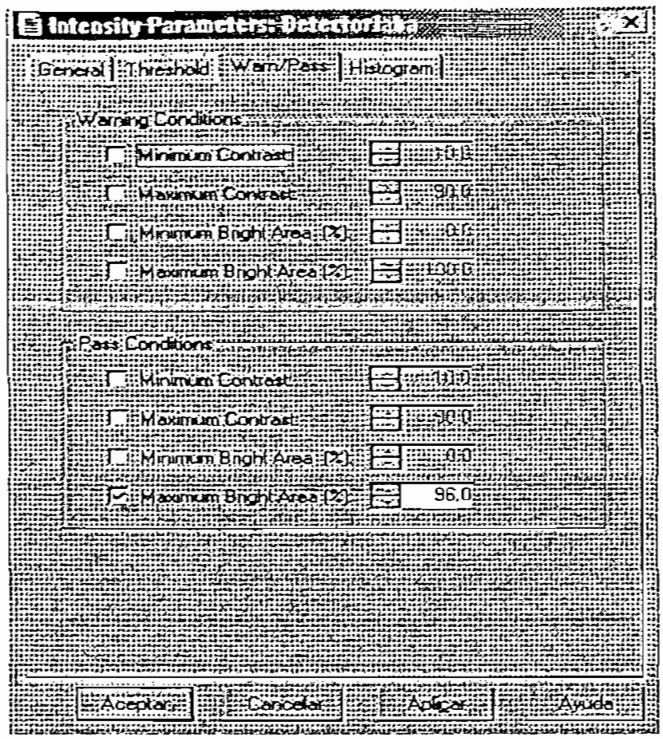


FIGURA 5.1: Asignación del valor en el sensor Presencia de Caja

5.1.2.2 Prueba 2 de Calibración: Obtención de Valores para Threshold de Sensores de Botellas

Esta prueba fue muy importante ya que permitió que cada uno de los sensores de manchas se encargue de discriminar botella buena, rota o ausente dentro de las jabas. Debido a las variaciones de intensidad de luz a las que las jabas están sometidas e incluso a condiciones de la botella a la entrada del envase (limpia/sucia), los sensores aplicados no podían tener valores de Threshold fijos, ya que habrían dado veredictos erróneos al confundirse debido a las variaciones en las intensidades de luz. Esto debido a que cuando el día obscurece todo tiende a hacerse color negro, y si el día se aclara entonces todo tiende a hacerse blanco.

Por ello fue necesario que los valores de Threshold de cada uno de los 12 sensores de manchas vayan cambiando con cada adquisición de imagen. Dichos valores se basaron en el nivel de brillo que presente cada una de las botellas. Es decir, cada sensor de mancha cuenta con un sensor auxiliar de intensidad que le advierta del nivel de intensidad de luz con el que se cuenta.

Los pasos que se siguieron para ejecutar esta prueba se describen a continuación:

- Iniciar el programa de configuración de la cámara en modo de simulación
- Cargar la segunda secuencia de imágenes y el programa de inspección
- Asignar al sensor de intensidad de la botella 1: cb1, configuraciones iniciales que se indican en la Tabla 5.5. Este sensor no se utilizó para dar una respuesta sino para medir el porcentaje de brillo de la botella.

No. de Sensor	Tipo de Threshold	Nivel de Threshold	Porcentaje de Llenado	
Cb1	Percent of Path of Contrast	80	100	

TABLA 5.5: Calibraciones iniciales sensor cb1.

- Asignar al sensor de botella: sb1, una configuración inicial que se indica en la Tabla 5.6 y Figura 5.2

Threshold	Color de la Mancha	Morfología	Filtro	Tamaño de Mancha		Condiciones de Paso	
				Mín No. Píxeles	Máx No. Píxeles	Mín No. Manchas	Máx No. Manchas
Percent of Path of Contrast	Dark	None	None	206	750	1	6

TABLA 5.6: Configuraciones Iniciales Sensor de Botella: sb1.

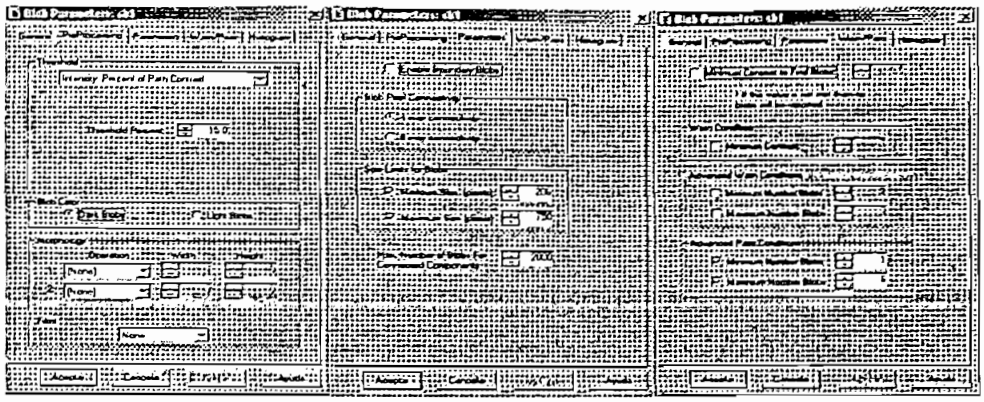


FIGURA 5.2: Calibración de valores iniciales en sensor de botella: sb1.

- Correr las imágenes anotando las variaciones de porcentaje de brillo del sensor cb1 y el valor que se asigno al threshold del sensor sb1, de manera que el sensor sb1 discrimine una botella en buen o mal estado. Estos datos están en la Tabla 5.6
- Calcular una media para los Thresholds obtenidos en cada rango de porcentaje de brillo, ver Tabla 5.6, y los valores de las medias se introdujeron en la tabla de recalibración usada para el sensor del literal c) o Script: ControlThreshold. El sensor del literal c) es un sensor programable que reasigna un valor de threshold a cada uno de los 12 sensores de manchas aplicados, dependiendo del nivel de brillo de la botella correspondiente.
- Puesto que la misma tabla de recalibración debía funcionar para todos los sensores de manchas, se calibró a los sensores de intensidad del cb2 al cb12 con los parámetros indicados en la Tabla 5.7, y de igual manera se calibró a los sensores de botellas del sb2 al sb12 con los parámetros indicados en la Tabla 5.8. Esos parámetros fueron hallados experimentalmente.

Nota: Los sensores de intensidad de botellas son 12, uno por cada botella y de igual manera los sensores de manchas son 12 sensores uno por cada botella. En ambos casos los sensores están ubicados sobre la imagen en la secuencia mostrada en la Figura 5.3.

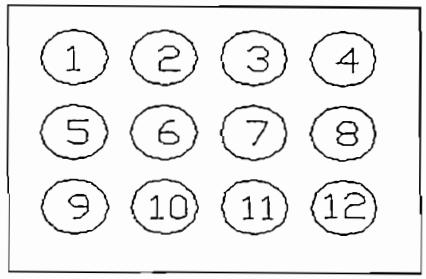


FIGURA 5.3: Disposición de los sensores en la imagen

Porcentaje de brillo CB1		Valores de Calibración de Threshold Sensor de Manchas sb1					Promedio de calibración Threshold Sb1
		1ro	2do	3ro	4to	5to	
de 0	a 2	18	17	16	16	18	17
de 2	a 4	12	10	20	22	21	17
de 4	a 6	15	25	22	23	21	21
de 6	a 8	17	29	15	21	24	21
de 8	a 10	40	30	26	28	26	30
de 10	a 12	40	35	20	24	25	29
de 12	a 14	40	40	43	25	30	36
de 14	a 16	40	42	45	41	43	42
de 16	a 18	40	42	38	43	42	41
de 18	a 20	38	40	43	36	40	39
de 20	a 22	40	35	38	41	35	38
de 22	a 24	40	34	35	36	40	37
de 24	a 26	35	42	43	42	51	43
....	....	...	....	....	....	....	
....	....	...	....	....	....	....	
de 80	a 82	60	60	62	59	59	60
de 82	a 84	60	56	58	61	56	58
de 84	a 86	55	56	59	56	65	58
de 86	a 88	60	70	65	60	55	62
de 88	a 90	62	70	68	60	70	66
de 90	a 92	65	66	67	60	65	65
de 92	a 94	70	67	62	59	62	64
de 94	a 96	70	69	63	68	73	69
de 96	a 98	75	70	73	73	70	72
de 98	a 100	75	74	73	76	76	75

TABLA 5.6: Tabla de calibración de Sensores de Botella: sb1

No de Sensor	Tipo de Threshold	Nivel de Threshold	Porcentaje de Llenado
Cb2	Percent of Path of Contrast	74	100
Cb3	Percent of Path of Contrast	66	100
Cb4	Percent of Path of Contrast	70	100
Cb5	Percent of Path of Contrast	65	100
Cb6	Percent of Path of Contrast	65	100
Cb7	Percent of Path of Contrast	69	100
Cb8	Percent of Path of Contrast	66	100
Cb9	Percent of Path of Contrast	74	100
Cb10	Percent of Path of Contrast	74	100
Cb11	Percent of Path of Contrast	75	100
Cb12	Percent of Path of Contrast	62	100

TABLA 5.7: Calibración de los Sensores de Intensidad de Botellas

No. Sensor	Threshold	Tamaño de Mancha		Condiciones de Paso	
		Mín No. Píxeles	Máx No. Píxeles	Mín No. Manchas	Máx No. Manchas
sb2	Percent of Path of Contrast	200	700	1	6
sb3	Percent of Path of Contrast	250	800	1	6
sb4	Percent of Path of Contrast	225	725	1	6
sb5	Percent of Path of Contrast	250	800	1	6
sb6	Percent of Path of Contrast	250	850	1	6
sb7	Percent of Path of Contrast	250	800	1	6
sb8	Percent of Path of Contrast	200	700	1	6
sb9	Percent of Path of Contrast	200	750	1	6
sb10	Percent of Path of Contrast	215	750	1	6
sb11	Percent of Path of Contrast	250	700	1	6
sb12	Percent of Path of Contrast	250	700	1	6
Color de la mancha para todos los sensores					DARK
Tipo de filtro para todos los sensores					Ninguno
Tipo de morfología para todos los sensores					Ninguno

TABLA 5.8: Calibración de parámetros en los Sensores de Botellas

5.1.2.3 Prueba 3 de Calibración: Recalibración de Valores para Threshold de Sensores de Botellas.

Esta prueba se realizó debido a los altos errores que cometió el programa con las calibraciones previamente obtenidas. Estos errores se indican en la sección 5.2.3

Esta prueba se realizó mientras la sala de envase 3 estaba en operación esto con el objetivo de que la recalibración se ejecute con imágenes que se iban adquiriendo en ese momento, y para ver como los cambios que se realizaban en la tabla se iban aplicando y si ellos servían para la mayoría de las imágenes sin dar muchos problemas. La nueva tabla de calibración de threshold de los sensores de botellas se indica en la Tabla 5.9.

Rangos de Intensidad		Recalibración	Rangos de Intensidad		Recalibración
de 0	a 2	17	de 50	a 52	53
de 2	a 4	17	de 52	a 54	50
de 4	a 6	21	de 54	a 56	50
de 6	a 8	21	de 56	a 58	60
de 8	a 10	30	de 58	a 60	50
de 10	a 12	30	de 60	a 62	50
de 12	a 14	36	de 62	a 64	50
de 14	a 16	40	de 64	a 66	53
de 16	a 18	40	de 66	a 68	50
de 18	a 20	40	de 68	a 70	50
de 20	a 22	39	de 70	a 72	50
de 22	a 24	39	de 72	a 74	53
de 24	a 26	40	de 74	a 76	55
de 26	a 28	42	de 76	a 78	55
de 28	a 30	40	de 78	a 80	57
de 30	a 32	43	de 80	a 82	59
de 32	a 34	43	de 82	a 84	60
de 34	a 36	45	de 84	a 86	56
de 36	a 38	45	de 86	a 88	55
de 38	a 40	45	de 88	a 90	70
de 40	a 42	50	de 90	a 92	62
de 42	a 44	50	de 92	a 94	65
de 44	a 46	50	de 94	a 96	70
de 46	a 48	50	de 96	a 98	70
de 48	a 50	53	de 98	a 100	75

TABLA 5.9: Recalibración de Thresholds para sensores de manchas

5.2 RESULTADOS DE LA PRUEBAS

Los resultados de las pruebas se obtuvieron al probar las calibraciones del programa inspector en tiempo real, es decir el programa fue puesto a prueba mientras la línea de embotellado estaba en funcionamiento.

5.2.1 RESULTADOS PRUEBA DE CALIBRACIÓN DE IMÁGENES

Los resultados de la prueba efectuada en el numeral 5.1.1 fueron las imágenes que se obtuvieron hasta llegar a una imagen clara. La secuencia de imágenes muestra la variación de cada foto a medida que se cambiaban los valores. En la Figura 5.4 se ven ocho imágenes y sus variaciones más importantes.

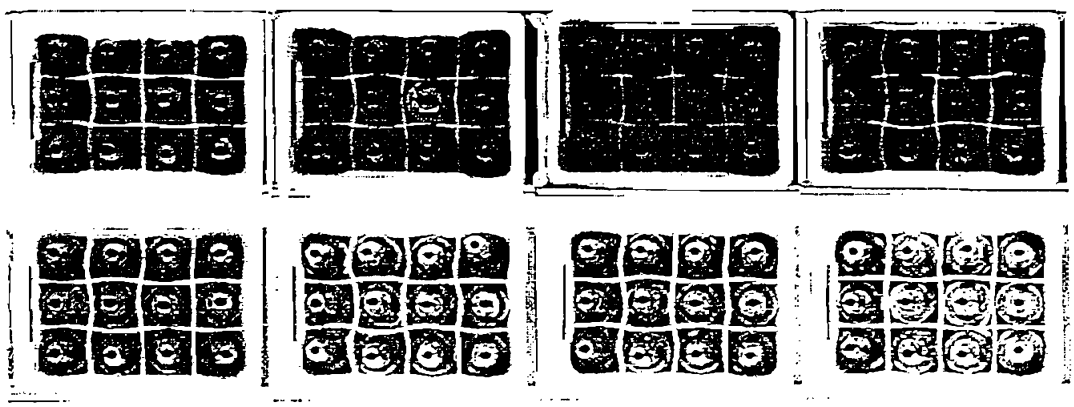


FIGURA 5.4: Muestra de imágenes durante la Calibración de Imágenes.

5.2.2 RESULTADO PRUEBA 1 DE CALIBRACIÓN: SENSOR DE DETECCIÓN DE JABA

Los resultados de la prueba efectuada en el numeral 5.1.2.1 se hicieron solamente con el sensor de presencia de caja y los otros sensores fueron desactivados para que no interfieran en la respuesta.

Una vez hecho esto la cámara inspeccionó presencia/ausencia de jaba exclusivamente y se obtuvo los resultados mostrados en la Figura 5.5

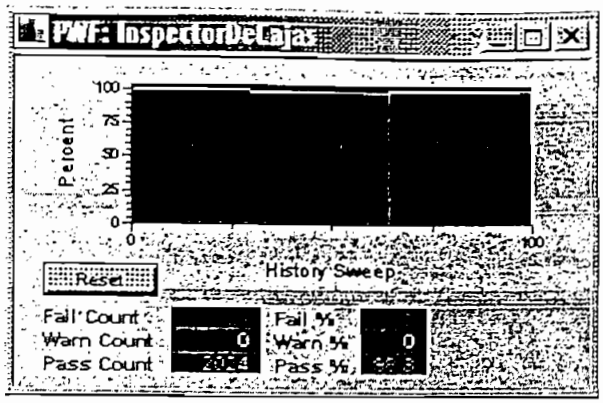


FIGURA 5.5: Cuadro de resultado para el Sensor de Detección de Jabas

De las 2048 imágenes que se utilizaron para la prueba, hubo un 1.2% o 24 imágenes que tenían alguna falla. Este porcentaje de falla se debió exclusivamente a que la jaba no estaba presente o la jaba no estuvo bien posicionada y por tanto la imagen apareció con solo una parte de la jaba.

Las fallas que detectó el Sensor de Detección de Jabas fueron atribuidas a vibraciones propias de las líneas de transporte que generan detecciones falsas en el sensor trigger que dispara a la cámara.

Cabe señalar que en esta prueba no existieron fallas que se debieran a alguna situación causada por la programación de la cámara; es decir las fallas obtenidas solo se debieron a problemas de vibraciones que se originan en la transportadora en movimiento.

El porcentaje de imágenes correctas así como el detalle de las imágenes con falla se indica en la Tabla 5.10

Tipo de Imagen	Tipo de Falla	#imágenes	Porcentaje	Imagen Muestra
Correcta	Ninguna	2024	98.8	
Con Falla	Jaba ausente en la imagen	19	0.93	
	Jaba incompleta en la imagen	5	0.27	
Total		2048	100	1 2 3 1. Imágenes Correctas 2. Imágenes sin Jabas 3. Imágenes con Jabas Incompletas

TABLA 5.10: Resultado de la Prueba 1: Detección de Jaba.

5.2.3 RESULTADO PRUEBA 2 DE CALIBRACIÓN: OBTENCIÓN DE VALORES PARA THRESHOLD DE SENSORES DE MANCHAS.

Para obtener los resultados de la prueba del numeral 5.1.2.2 se bloqueó el sensor de presencia de caja para que no interfiera en la respuesta y los otros sensores fueron activados.

Los porcentajes que se obtuvieron para estos resultados están relacionados con la cantidad de botellas buenas frente a botellas fallosas, entendiéndose como

fallosas, botellas rotas o ausentes. Pero cabe mencionar que dentro de las botellas fallosas se tomó en cuenta a mas botellas rotas o ausentes, botellas tapadas o con alguna suciedad, botellas que estando bien se contabilizaron como fallosas y botellas que estando mal se contabilizaron como correctas.

Entonces se tomó como una falla ajena a la cámara botellas rotas, ausentes, tapadas o con alguna suciedad y como errores propios de la cámara botellas buenas contabilizadas como fallosas y botellas rotas o ausentes contabilizadas como buenas.

Una vez hecho esto, la cámara pudo hacer el análisis correspondiente para diferenciar entre botellas correctas, rotas o ausentes. Estos datos se encuentran en la Figura 5.6; este es un cuadro de resultados que se generó a través de un Script. Con el Script se contabilizó botellas buenas y botellas con fallas. En la Figura 5.7 se muestra un cuadro de resultados generado por el propio programa de configuración de la cámara y que está relacionado con la cantidad de jabas que entraron sin anormalidades en su contenido y la cantidad de jabas que tenían alguna anormalidad en las botellas que se hallaban en su interior.

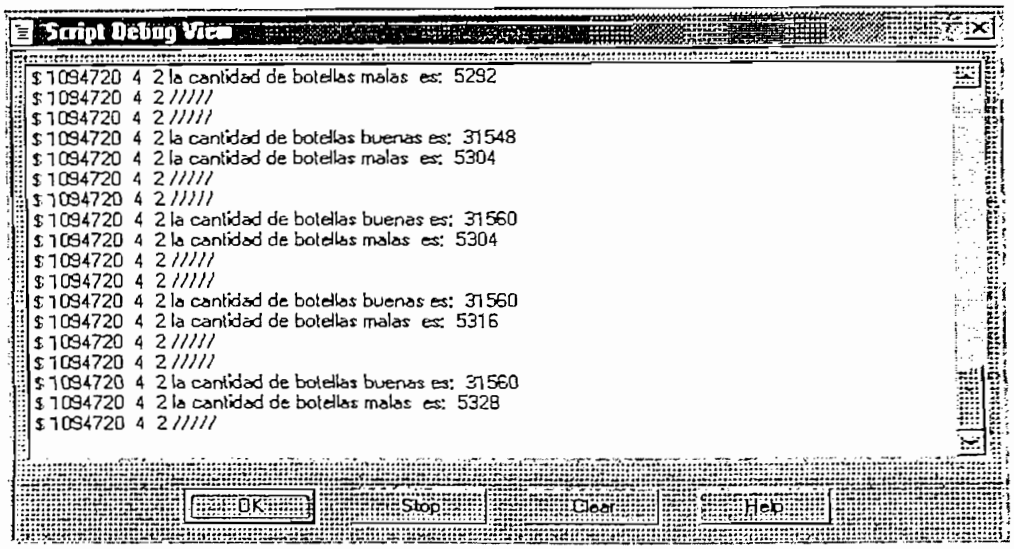


FIGURA 5.6: Script contabilizador de botellas

Cabe mencionar que la cantidad de botellas con fallas que contabilizo el Script incluye todas las fallas descritas antes.

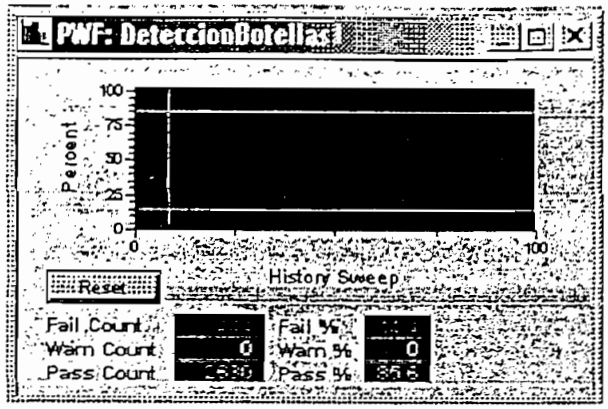


FIGURA 5.7: Cuadro de la estadística general del programa

De la Figura 5.6 y después de haber analizado cada imagen de la secuencia de imágenes que se utilizó para obtener estos resultados se obtuvo lo porcentajes que se muestran en la Tabla 5.11

Tipo de Botellas	Tipo de Fallas	Cantidades	Porcentajes
Buenas	Botellas buenas	31546	85.518
	Botellas malas captadas como buenas	14	0.038
Fallosas	Botellas rotas, ausentes, tapadas o con alguna basura	136	0.369
	Botellas buenas captadas como malas	5192	14.075
Total		36888	100
Error cometido por la equipo			14.113

TABLA 5.11: Errores cometidos por el programa de inspección

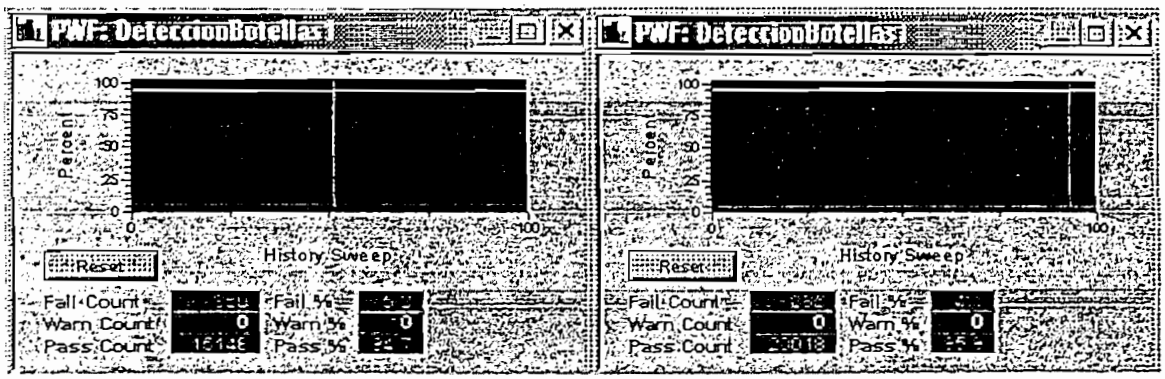
De los resultados obtenidos en la Tabla 5.11 se notó que el error que cometía el equipo de inspeccion fue demasiado alto razón por la cual se elaboró la prueba número 3 descrita en la Sección 5.1.2.3.

5.2.4 RESULTADO PRUEBA 3 DE CALIBRACIÓN: DESPUES DE LA RECALIBRACIÓN DE VALORES PARA THRESHOLD DE SENSORES DE BOTELLAS

Con los cambios que se realizaron en la Tabla 5.9, los resultados que se obtuvieron fueron mucho mejores que con los datos de la Tabla 5.6. Es así que en la Figura 5.8 y la Figura 5.9 se puede apreciar el máximo y mínimo error que el programa comete debido a las anomalías que se detallan en la Tabla 5.12

Al igual que en los resultados anteriores se considera como fallas ajenas al equipo las botellas que estan rotas, ausentes, tapadas o con alguna basura y como errores propios de la camara botellas buenas contabilizadas como fallosas y botellas rotas o ausentes contabilizadas como buenas.

La Figura 5.8 esta compuesta por dos cuadros en los que se hallan los limites de Porcentaje de falla con los que el programa trabaja y que esta comprendido entre 4.1 y 5.3 % para jabas que contienen algun tipo de falla en las botellas.



a) b)

FIGURA 5.8: Cuadros de las estadísticas generales del Programa Inspector para: Limite superior de Falla, b) Limite inferior de Falla

En la Figura 5.9 se ven los valores registrados por el Script que contabiliza la cantidad de botellas buenas y la cantidad de botellas fallosas. Los valores registrados son para la pantalla de la Figura 5.8, literal a) en donde el porcentaje de falla es del 5.3%.

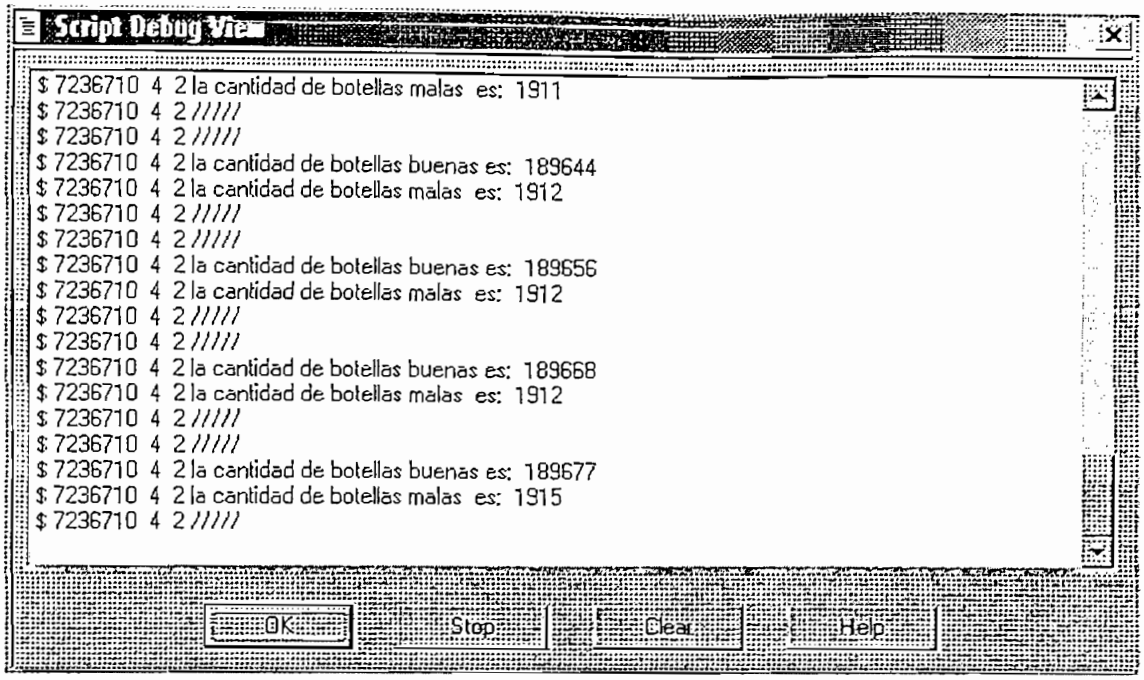


FIGURA 5.9: Contabilizador de botellas .

Para elaborar la Tabla 5.12 de porcentajes se tomaron los resultados del script de la Figura 5.12 y de las imágenes que fueron grabadas mientras el programa estaba inspeccionando con la planta en funcionamiento.

Estas imágenes que se grabaron fueron utilizadas para constatar los resultados que se obtuvieron y para desglosar las fallas propias de las jabas de los errores que comete el programa de inspección.

Tipo de Botella	Características	Cantidades	Porcentajes	Imágenes muestra
Buenas	Ninguna	189662	98.993	
	Botellas malas captadas como buenas	15	0.008	
Fallosas	Ausentes o Rotas	530	0.277	
	Botellas tapadas o con algún tipo de basura	1150	0.600	
	Botella buenas captadas como malas	235	0.122	
Total		191592	100 %	
Error total cometido por el programa			0.13 % del total de botellas contabilizadas	

TABLA 5.12: Detalle de los errores cometidos por el programa de inspección

De los resultados obtenidos se ve que el error cometido por programación fue bajo, apenas corresponde a un 0.13%. Este error se debe a la percepción equivocada de la cámara cuando al pasar botella buena la detecta como mala.
En cambio el porcentaje de botellas rotas y ausentes que la cámara es capaz de detectar corresponde al 0.3% aproximadamente, este es el porcentaje mas importante, ya que corresponde al objetivo del proyecto y es el que permite llevar la gestión administrativa a la Empresa.
Por otro lado existe un error del 0.6% debido a condiciones indeseables en las botellas de entrada, condiciones como botellas tapadas, con vasos sobre las botellas o con basura. Este es un factor externo ajeno tanto a la programación como a la estadística que se desea realizar con los datos arrojados por el programa implementado.

De las pruebas y resultados obtenidos podemos concluir algunos puntos que se describen en el siguiente Capitulo 6.

CAPITULO 6
CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

1. Dado que el error por programación es de apenas un 0.13%, se concluye que el programa implementado cumple de manera satisfactoria con los objetivos concernientes a este proyecto.

2. Las estadísticas arrojan un resultado de un 0.6% de botellas rotas y ausentes que ingresan a la línea de envase #3, y corresponde a una cantidad de 1150 botellas, de un total de 191592 botellas analizadas.

3. La distancia desde la ubicación del sistema hasta el punto mas cercano de red existe 80m aproximadamente, distancia que cumple con las normas de cableado estructurado; sin embargo, el cable de red no pudo ser transportado por tuberías debido a la dificultad física que ofrece la línea de envase, por ello fue necesario seleccionar un cable de red especializado (STP) que tiene doble blindaje y que ayudó en la instalación de la red.

4. Para el funcionamiento del sistema fue necesario separar las cajas unas de otras, dicha condición no existía en la línea de envase, siendo necesario el reemplazo de la sección de transportadora en donde se colocaría el sistema.

5. En sistemas de inspección basados en análisis digital de imágenes es de suma importancia la forma y tipo de iluminación que se usa para capturar una imagen, ya que ligeras variaciones de luz incidentes sobre el objeto puede generar grandes cambios en los resultados de la inspección. Para el proyecto se usó lámparas fluorescentes capaces de realzar bordes de las botellas, que ayuden a reconocer dichos bordes en el momento de programar las cámaras.

6. Para obtener imágenes nítidas, fue necesario calibrar parámetros físicos y de programación; en cuanto a los físicos las perillas del lente ayudaron al ajuste del enfoque, zoom y a calibrar la cantidad de la luz que ingresaría a la cámara. En cuanto a programación los parámetros de tiempo de adquisición de la imagen y ganancia del sistema ayudaron a eliminar el ruido producido por el movimiento de las jabas y se le dio mayor nitidez a la imagen.

7. Las jabas con botella vacía que ingresan a la línea 3 de envase de Cervecería Andina no tienen uniformidad en su aspecto, esto independientemente de que lleguen con botellas rotas o ausentes. Las jabas llegan con mucha suciedad dentro de ellas: plásticos, vasos, papeles, fundas, objetos ajenos a la jaba, etc. Además las botellas vacías tienen en su interior servilletas, sobrantes de cerveza, están tapadas, cubiertas con vasos, etc. Algunas de estas situaciones se trataron de aislar mediante la programación de las cámaras, sin embargo no se eliminaron totalmente.

8. Debido a las situaciones mencionadas en el párrafo anterior el error que se comete en el proyecto es alto, pues el sistema es incapaz de diferenciar entre botellas ausentes y botellas con vaso plástico ò botellas con liquido en su interior, ya que las cámaras se guían en variaciones de intensidad de luz para dictaminar si una botella esta presente o no; y para los casos mencionados la cámara se confunde fácilmente.

9. Para el proyecto se usaron varios programas que usaban ciertas herramientas de programación propias de la cámara como sensores busca objetos, preprocesamiento de la imagen, aprendizaje de características, etc. Estos programas funcionaron solo en situaciones simuladas, pues en tiempo real el programa fue incapaz de procesar todas las imágenes capturadas, dejando pasar por alto por lo menos un 20% del total de imágenes que en realidad se debieron analizar.

10. Debido a la velocidad de transportación de las cajas, (60 cajas/segundo) el programa de inspección aplicado se vio limitado en cuanto a herramientas que se podían usar, pues mientras más compleja es la herramienta usada, mas lento se torna el programa. Por tanto el programa usado para el sistema se basa en herramientas simples capaces de ejecutarse en el tiempo real de transporte de las cajas.

11. Se debe anotar que el equipo de inspección DVT con el que se llevo a cabo el sistema, se aplica en la industria a inspecciones de unidades; es decir, existe varias herramientas que se pueden aplicar para analizar un solo objeto por imagen adquirida. Pero para este proyecto se requiere aplicar por lo menos 12 sensores a cada imagen adquirida con lo cual se asegura un largo tiempo de procesamiento para el programa aplicado.

12. El método de inspección que se escogió fue seleccionado porque utiliza sensores que pueden trabajar rápido y para evitar la recalibración de sensores se coloco un sensor Script que se encargaría de esto en forma automática en cuanto se perciba un cambio de tonalidad de luz extrema, para ello se uso una tabla precalibrada que se calculo en base a datos experimentales.

13. De las pruebas realizadas se puede ver que el sensor de de jabas trabaja eficientemente porque analiza a la jaba y no de sección en sección como lo hace el resto del programa. Además no tiene el problema en discernir las variaciones que presentan las jabas como sucede con el resto de sensores para las botellas. Este sensor da como respuesta PASS cuando las calibraciones miden la presencia de una jaba entera o da FAIL cuando las mismas calibraciones ven la banda transportadora o la mitad de una jaba.

14. Los sensores de botella cometen varios errores, pues tienen que discriminar botella buena de aquella que tiene basuras en su interior, que esta tapada, cubierta con un vaso, etc. Esto hace que los sensores de

botellas se vuelvan más sensibles a cometer fallas. Sin embargo, es el método que mejor funciono de todas las versiones de programa que se diseñaron.

15. Se aprendió a configurar y manejar los sensores de imagen inteligente DVT, manipularlos, programarlos y controlar la mayor parte de características de estos equipos que permiten realizar análisis digital de imágenes, también a efectuar acciones de control con ellos pues su conectividad en ethernet permite transmitir la información a través de redes Ethernet Industriales hacia otros dispositivos de campo que manejen este protocolo, directamente a una red administrativa o enviar los datos hacia una pantalla de visualización en alguna PC a través de sus Actives X o una pagina de Internet como se ejecuto en el proyecto realizado.

RECOMENDACIONES

1. Para que el sistema sea menos vulnerable a fallas, se recomienda aumentar una cámara mas por cada línea de transportación, esto con el fin de distribuir la inspección de la caja en dos secciones, al conseguir esta situación se podría usar una programación más compleja (que requiere más tiempo para su ejecución); sin embargo, podría detectar características más importantes que ayuden a reducir el error que el sistema tal cual como se encuentra en la actualidad posee, y que no se han podido eliminar por la velocidad de transportación de las cajas.

2. Si se desea mejorar el sistema sin utilizar otra cámara a cada línea es necesario que se reemplace la existente por la siguiente serie de cámaras que es mucho más rápida en su procesamiento.

3. Para mejorar la confiabilidad del sistema, es recomendable diseñar un sistema extractor de basuras que se lo ubicaría antes de la posición actual del sistema de inspección que trata este proyecto, así al eliminar todo tipo

de basuras la programación aplicada actualmente a la cámara sería más efectiva. limitaría a diferenciar entre e retirarían basuras que dificultan el análisis y además se ayudaría a evitar daños en la maquina desencajonadota que por motivos de cuerpo ajenos dentro de la jaba se dañan los dispositivos de sujeción de las botellas.

4. Para mejorar el funcionamiento del sistema se debe instalar un mejor sistema de iluminación que resalte de mejor manera los bordes de los picos de las botellas. Incluso el sistema de iluminación debería usar luz roja, según las recomendaciones para sistemas de adquisición de imágenes en blanco y negro, que no se las usó para el sistema por el alto costo.

REFERENCIAS BIBLIOGRÁFICAS

1) DVT CORPORATION, DVT SmartImage Sensor Manual de Usuario, 7ma edición, 2003.

2) DVT CORPORATION, DVT SmartImage Sensor Training Example Book, 7ma edición, 2003.

3) DVT CORPORATION, DVT Script Reference Manual, 4ta edición, diciembre 2002.

4) DR. LUÍS CORRALES, folleto de interfases.

5) PEPPERL+FUCHS, Factory Automation North American Catalog, Edición 2004-2005.

6) SENSICK, Sensors for Automation Catalog, Edición 2001.

7) BELDEN ELECTRONICS DIVISION, Master Catalog, Edición 2003.

8) SIEMENS, Control y Distribución Catalogo, Edición 2000-2001.

9) NATIONAL INSTRUMENTS, IMAQ Vision Manual, IMAQ Vision 6.0, Octubre 2001.

10) www.dvtsensors.com

11) www.eveliux.com

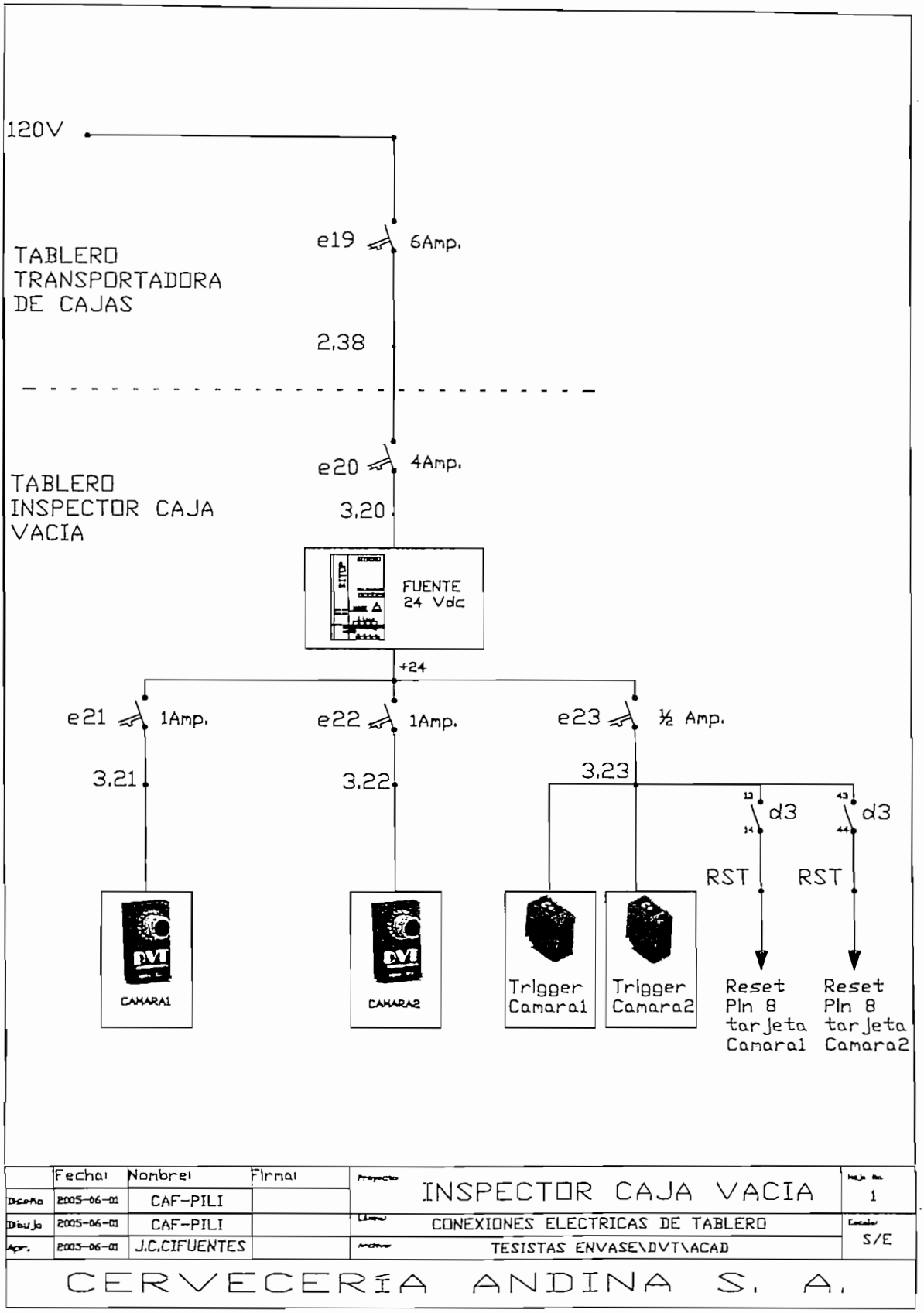
12) www.belden.com

ANEXOS

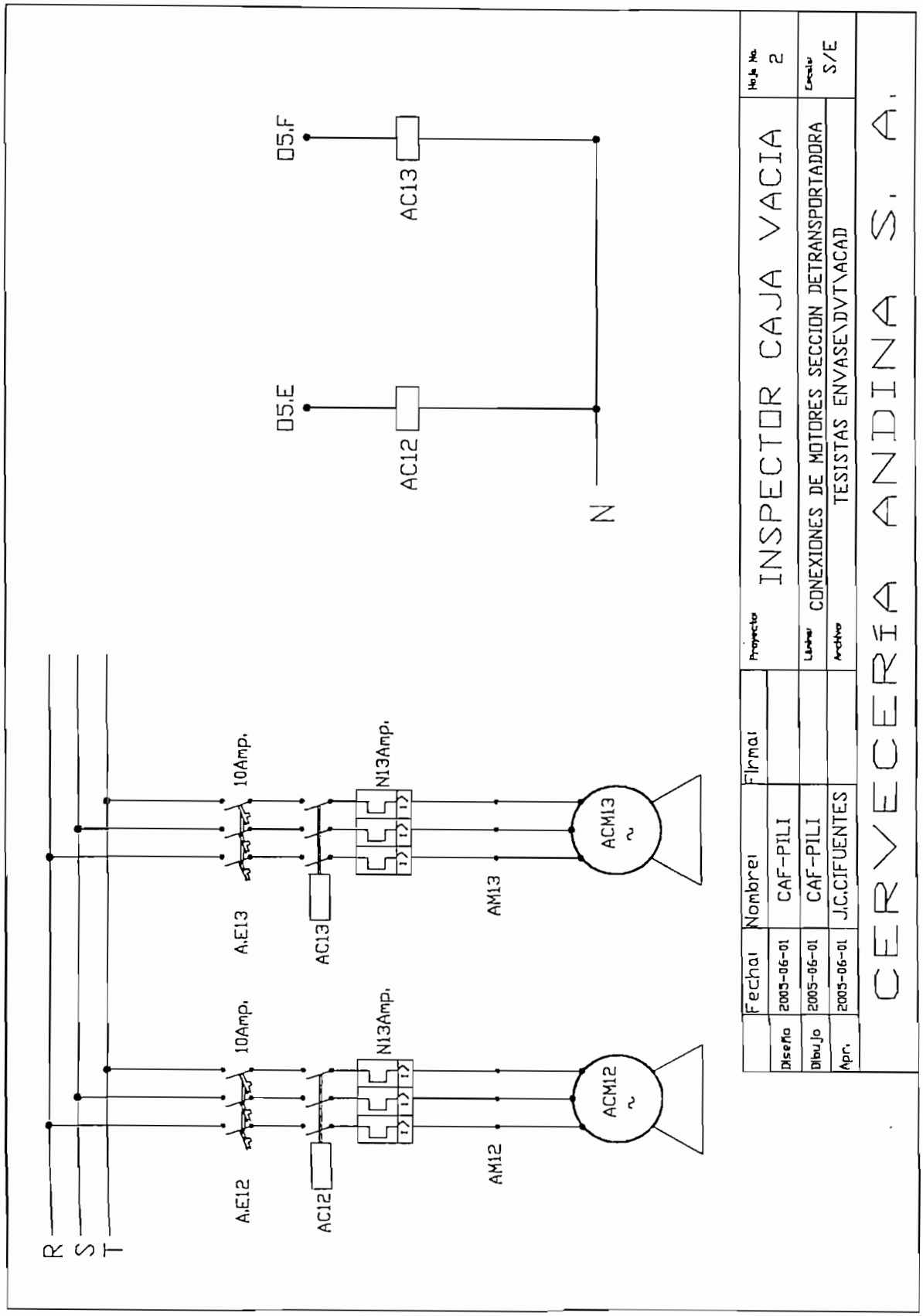
ANEXO 1

PLANOS ELÉCTRICOS

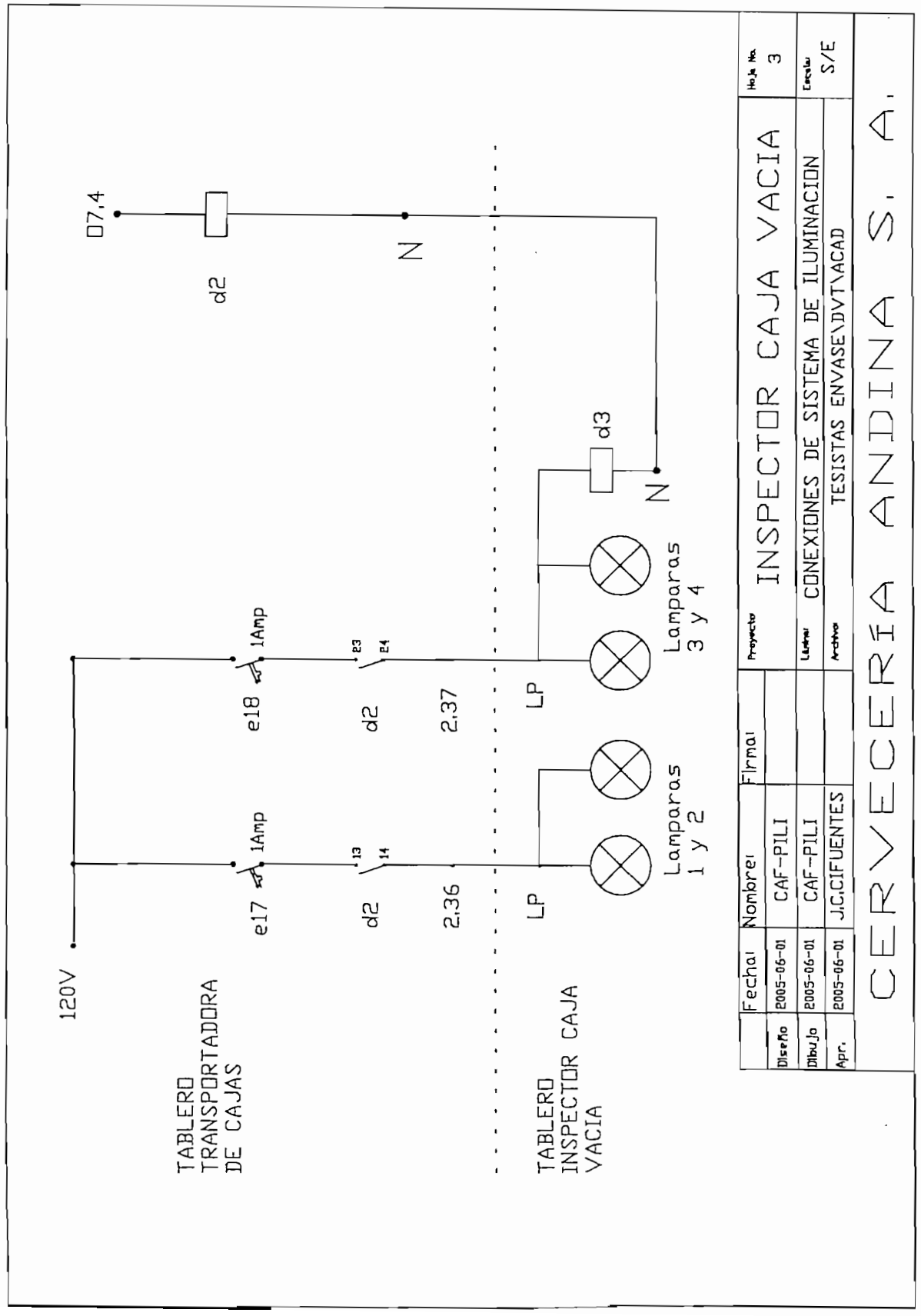
ANEXO 1.1: CONEXIONES ELÉCTRICAS TABLERO



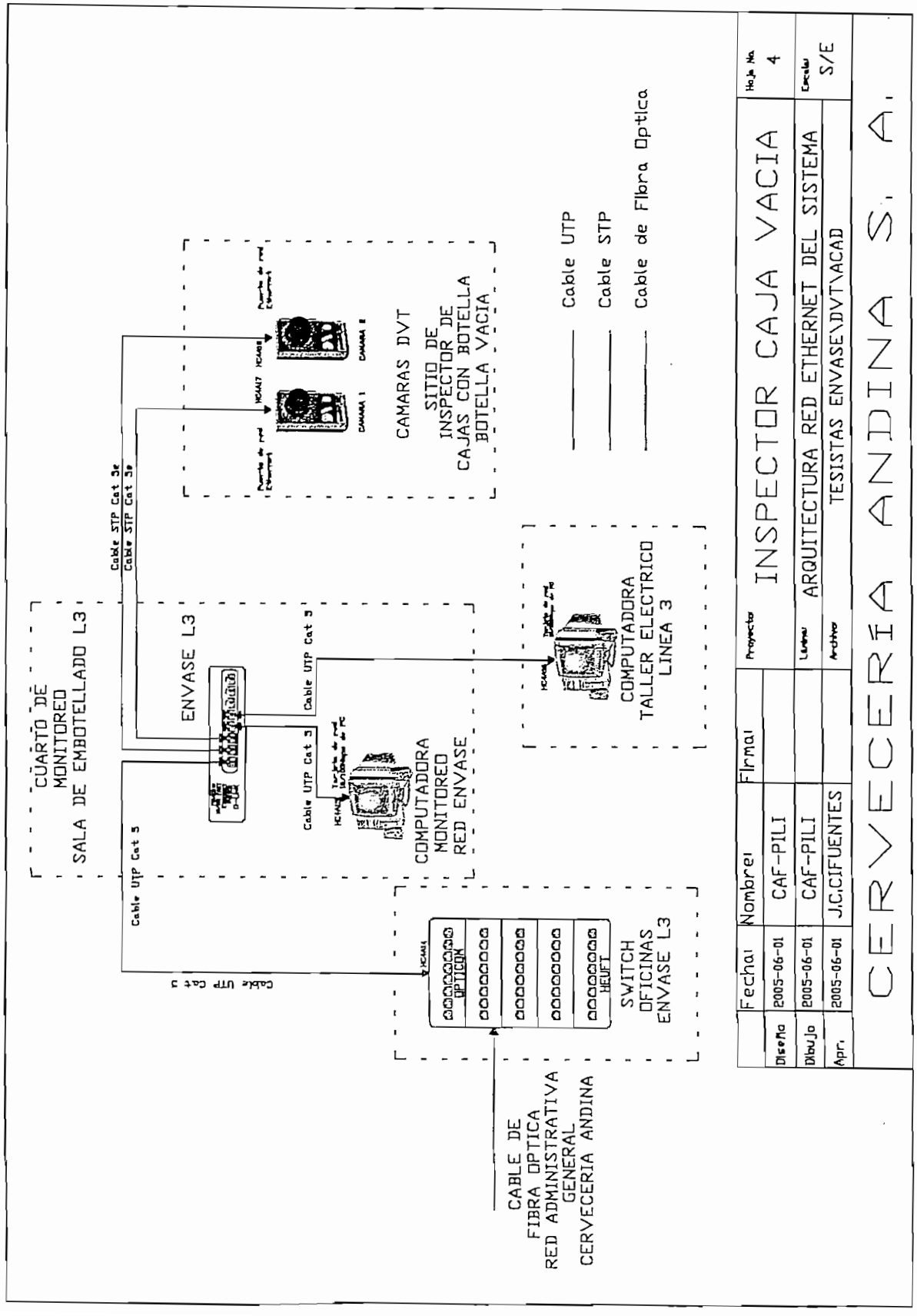
ANEXO 1.2: CONEXIONES MOTORES SECCIÓN DE TRANSPORTADORA PARA INSPECTOR



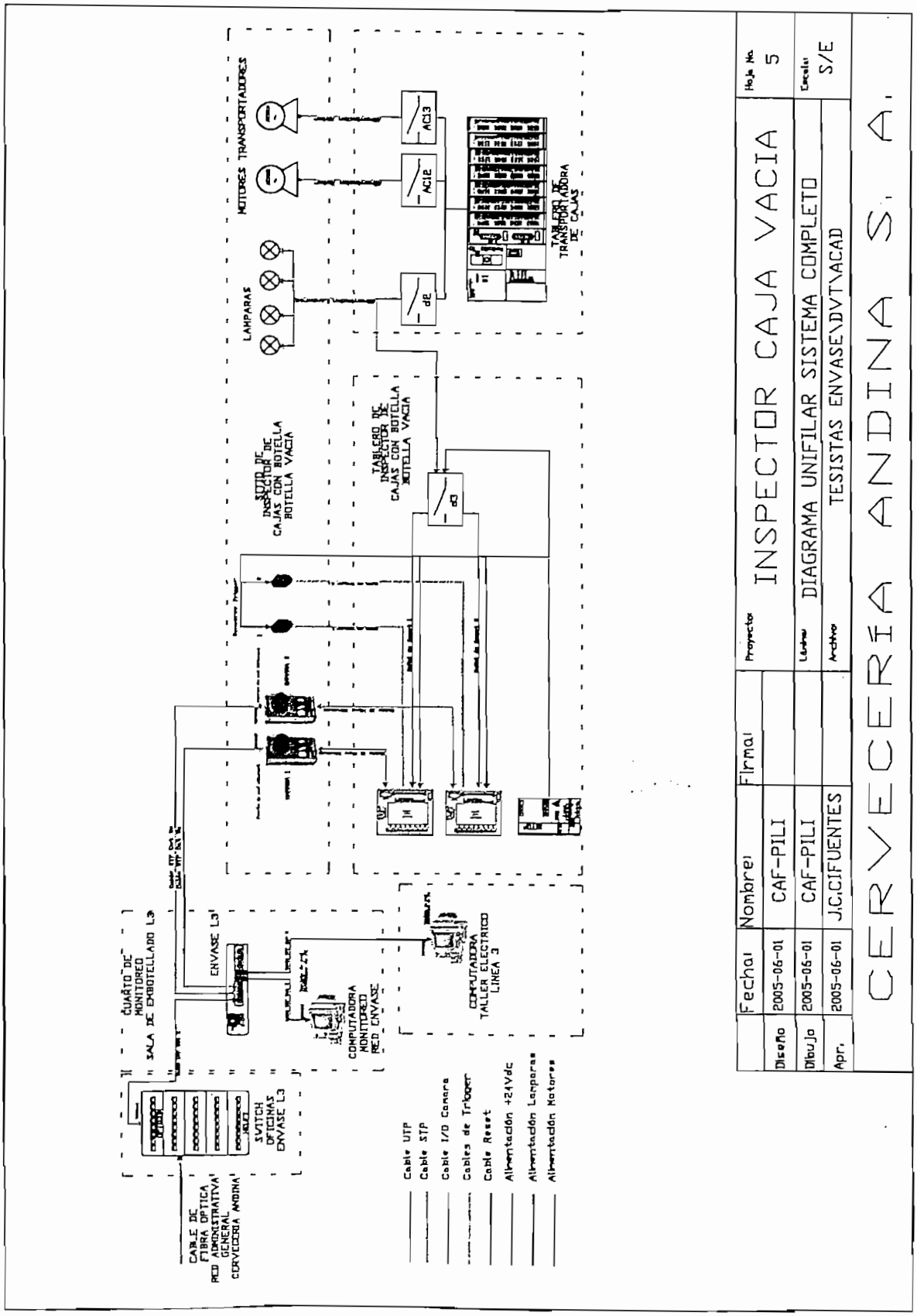
ANEXO 1.3: CONEXIONES SISTEMA DE ILUMINACIÓN DE INSPECTOR



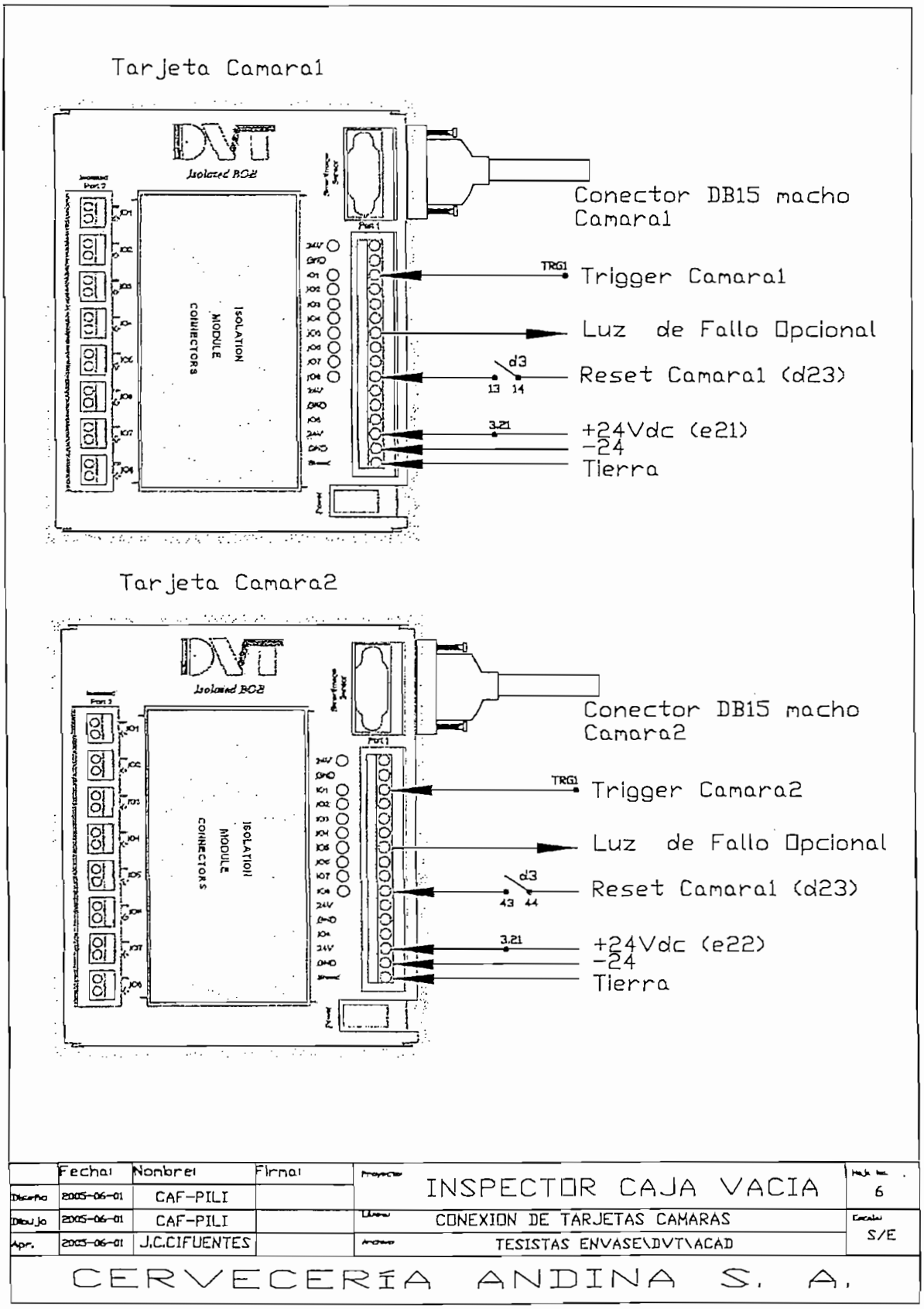
ANEXO 1.4: ARQUITECTURA DE LA RED ETHERNET PARA EL INSPECTOR



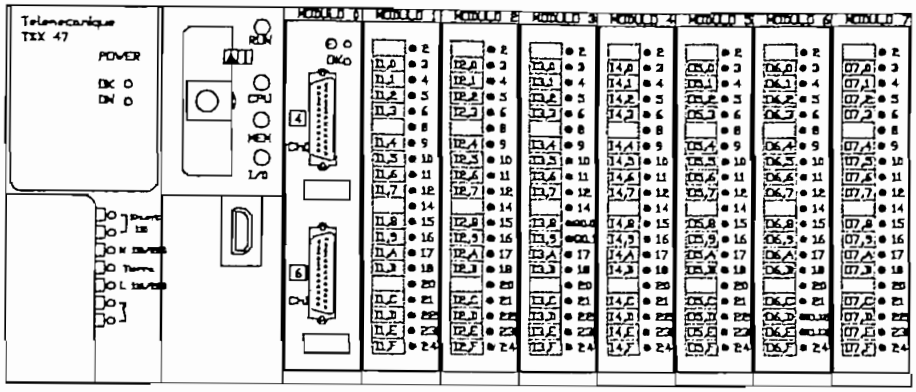
ANEXO 1.5: DIAGRAMA UNIFILAR SISTEMA COMPLETO

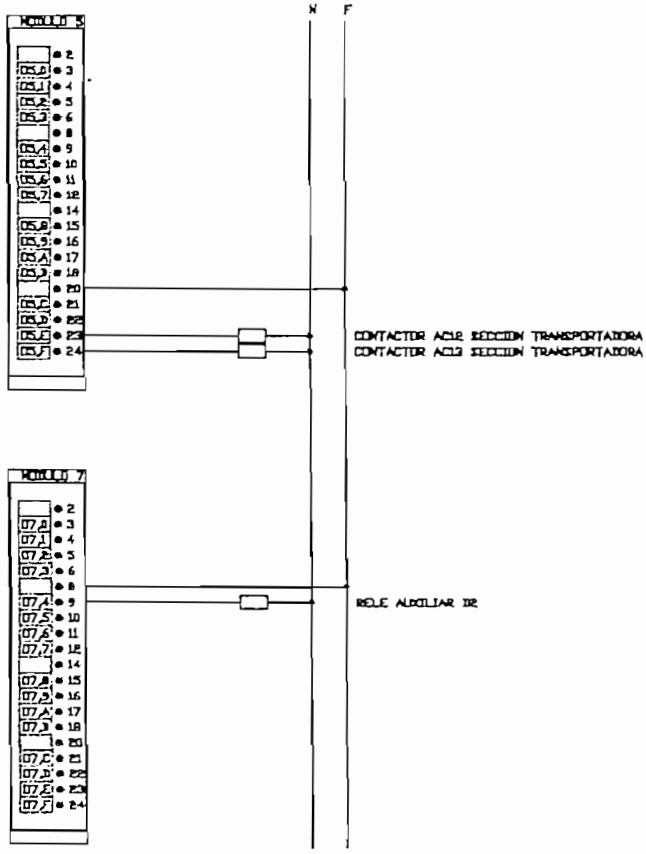


ANEXO 1.6: CONEXIONES EN LAS TARJETAS DE LAS CÁMARAS



ANEXO 1.7: CONEXIONES DE MOTORES Y LÁMPARAS EN PLC TABLERO DE TRANSPORTADORA DE CAJAS





	Fecha:	Nombre:	Firma:			
Diseño	2005-06-01	CAF-PILI		Proyecto	INSPECTOR CAJA VACIA	Hoja No. 7
Dibujo	2005-06-01	CAF-PILI		Título	CONEXION DE LAMPARAS Y MOTORES	Escala S/E
Apr.	2005-06-01	J.C.CIFUENTES		Archivo	TESISTAS ENVASE\DVT\ACAD	

CERVECERÍA ANDINA S. A.

ANEXO 2

DESCRIPCIÓN DEL EQUIPO DE INSPECCION ADQUIRIDO

DESCRIPCIÓN DEL EQUIPO

El equipo que se utiliza para realizar las inspecciones es fabricado por la Corporación DVT y esta conformado por dos partes que son:

a) Hardware
b) Software

HARDWARE

El hardware que se seleccionó para la ejecución del proyecto esta compuesto por las siguientes partes que se describen a continuación:

a. CÁMARA

La cámara es la parte del hardware que se encarga de adquirir y analizar la imagen y en base a este análisis puede ejecutar acciones de control. La serie de la cámara es: Smart Image Sensor Legend 530MR y se indica en la Figura 1

FIGURA 1: Cámara Smart Image Sensor Legend 530MR

Las principales características de este equipo son:

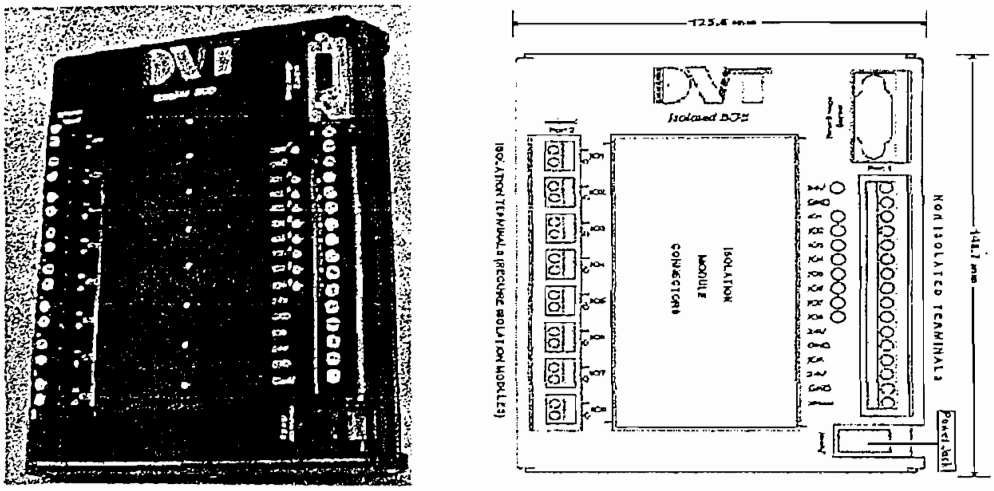
- Procesamiento y tratamiento completo de las imágenes.
- Comunicación desde y hacia dispositivos de campo a través de su puerto de Red Ethernet 10/100 Mbps o mediante su tarjeta de entradas o salidas
- Comunicación hacia redes administrativas por su puerto de Red Ethernet 10100 Mbps o hacía interfaces mediante sus Actives X que pueden ser programados en algún lenguaje de programación como Visual Basic
- Imágenes en tamaño 640 x 480 píxeles en escalas de grises
- Manejo hasta 4 luces externas
- Tamaño reducido
- Alimentación de 24Vdc.
- Manejo de lentes del tipo CS de conexión
- Mas información en el Anexo de Catálogos

b. TARJETA DE ENTRADAS Y SALIDAS

La tarjeta de conexiones mostrada en la Figura 2 se utiliza para suministrar a la cámara la alimentación de 24 Vdc a través de sus borneras de alimentación. La tarjeta no es la fuente de alimentación.

Además posee 8 entradas/salidas que son configurables por el usuario y se utilizan para ingresar o sacar señales de la cámara. Estas entradas/salidas pueden ser aisladas de la cámara a través de módulos de aislamiento que se pueden instalar en la tarjeta de conexiones.

El pin 1 de las entradas/salidas esta preconfigurado como estrada de señal de disparo externo. Los otros 7 pines e inclusive el pin 1 se lo puede configurar a la necesidad del usuario.



a) b)

FIGURA 2: Tarjeta de conexiones: a) Imagen de la Tarjeta, b) Ubicación de bornes de conexión

c. LENTES

Los lentes son lo que le permiten a la cámara poder captar imágenes de objetos de diversos tamaños desde pastillas hasta objetos grandes como una jaba e incluso mas grandes.

El lente utilizado para el proyecto se indica en la Figura 3. Este lente posee 3 controles los cuales le permiten enfocar la imagen, regular la cantidad de luz que entra a la cámara y variar el área de la imagen adquirida por medio del zoom. El lente tiene una longitud focal de 5 a 50mm.

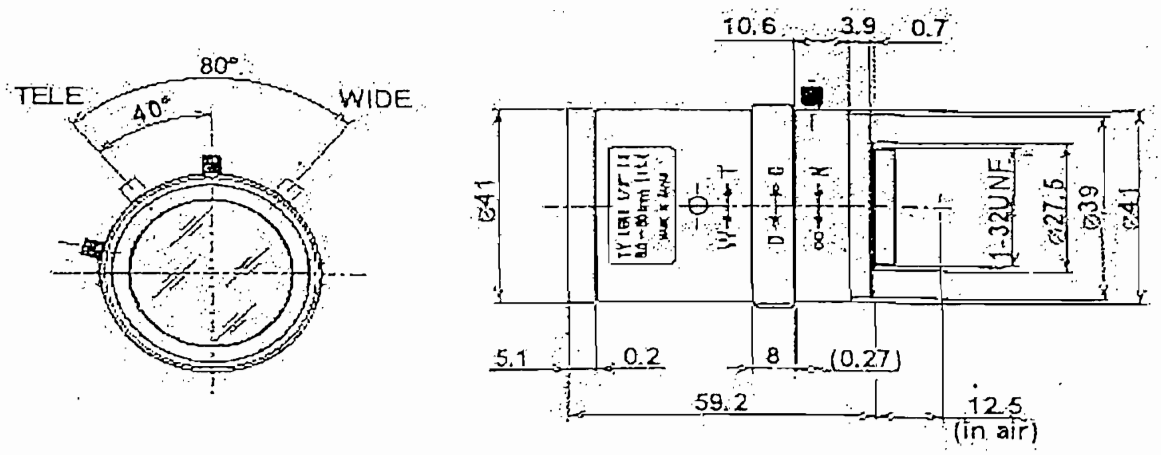


FIGURA 3: Lente LTZ 05Z.

Dependiendo del tipo de montaje del lente el cual puede ser C o CS se utiliza un adaptador de C a CS pues la cámara viene provista para montaje de lentes tipo CS.

SOFTWARE

El software es el medio que permite al usuario crear o realizar modificaciones en la programación de la cámara, configurar la comunicación en red, manejar la tarjeta de entradas/salidas, manipular los sensores virtuales, etc. El software esta compuesto de dos partes importantes:

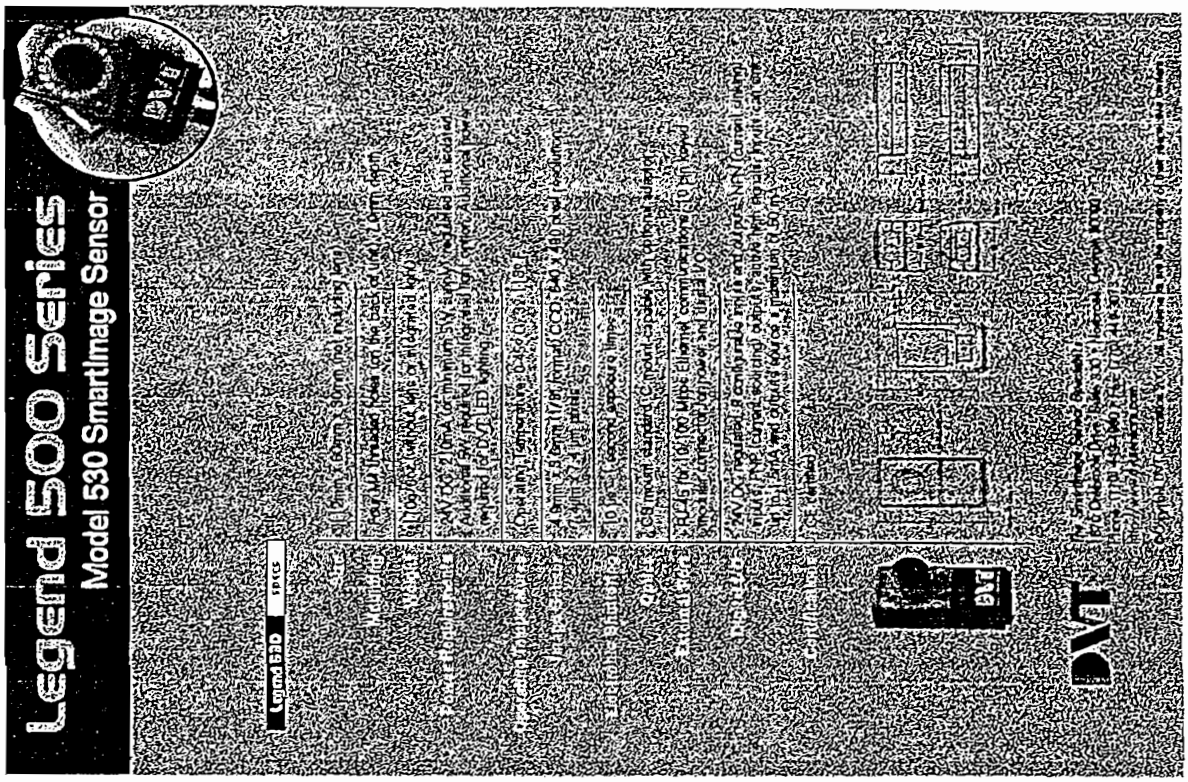
a. El Framework : es la interfase de la cámara en la PC y permite realizar las modificaciones en las configuraciones de la cámara así como crear o modificar los programas que se ejecuten en la cámara pues posee todos los sensores virtuales así como los Scripts de programación; además, permite configurar la tarjeta de entradas o salidas, la comunicación mediante el puerto de Red Ethernet, es decir el FrameWork permite al usuario manipular todas las habilidades de la cámara,

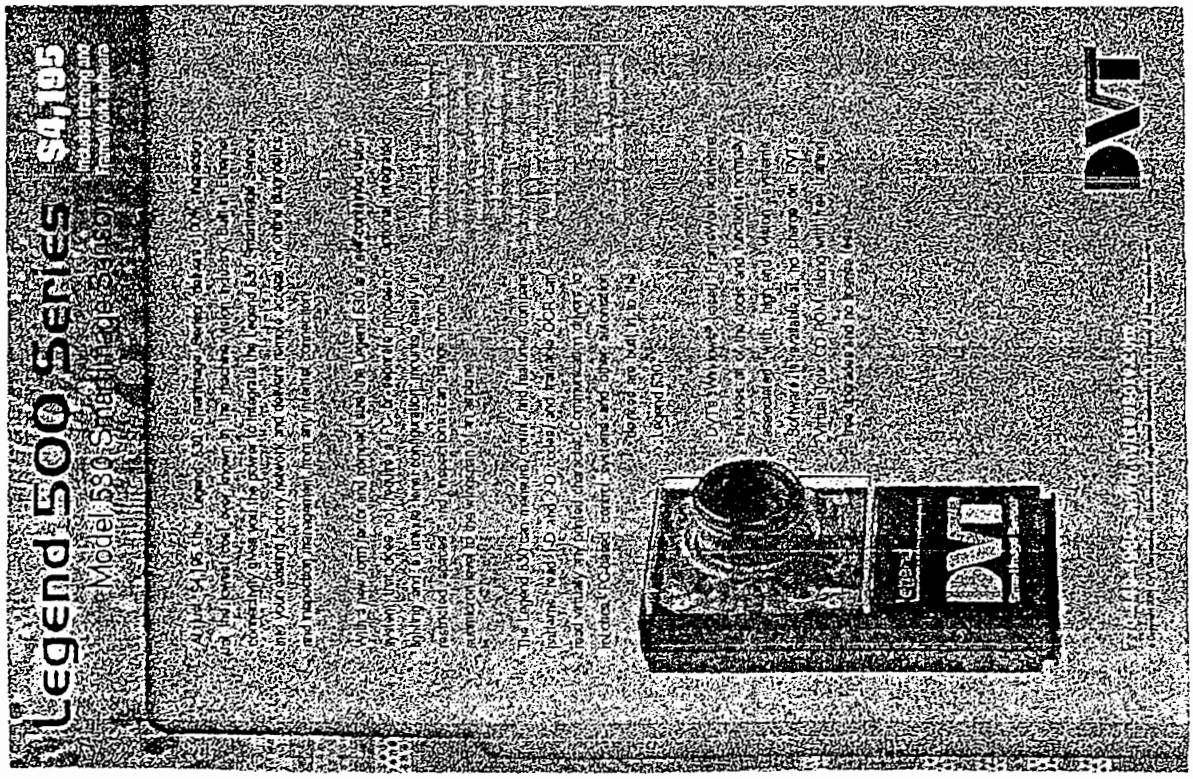
b. El Firmware: al haber desarrollado una aplicación, esta se descarga a la cámara donde el Firmware recoge todas las instrucciones desarrolladas y las interpreta para que la cámara pueda operar en la manera que el usuario programó.

Tanto el FrameWork como el FirmWare son software gratuito que el fabricante, DVT, proporciona gratuitamente a través de su página de Internet: www.dvtsensors.com .

CATÁLOGOS DEL EQUIPO

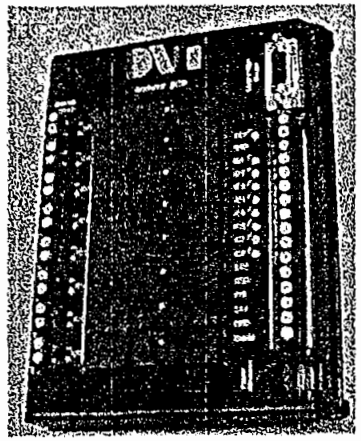
- Cámara 530MR





DVT Isolation Breakout Board

This document provides information about the new DVT Isolation Breakout Board (DVT part number CON-IBOB) available for use with the Legend Series SmartImage Sensors. An image of the isolation breakout board is shown below.



Features

The isolation breakout board for the Legend Series not only provides a convenient method to connect digital I/O, power, and strobe illumination lines; but it also allows for the use of isolation modules. It contains two sets of terminals: one that is directly connected to the SmartImage Sensor, and one that allows for the use of isolation modules.

The port on the right (see diagram) includes I/O, strobe and power screw-down terminals. The I/O lines on this port are to be used when isolation is not needed. Typical examples of the use of this port are the strobe lights. Not only strobe lights do not require any isolation from the system but they require very accurate timing. The use of isolation modules adds a very small delay to the signals (in the order of the microseconds depending on the specific module used) which in most cases is irrelevant (i.e. inspection results). However, the strobe lights must be on exactly while exposure lasts. This is why DVT recommends that strobe lights be connected to the non-isolated terminals. This set of terminals is always enabled, regardless of the use of isolation modules. The output line labeled IO8, corresponding to the SmartImage Sensor I/O 8, is mapped to two different pins because this is usually the output selected for strobe lights. By using two different pins for this I/O line users can connect 2 strobe lights that would respond to the same signal. The terminals that are repeated (24V, GND) are interconnected so whether we provide the power via the jack, or either set of terminals, the functionality will be the same and the remaining terminals become available for use as a power source. See the diagrams at the end of this document for information on sinking/sourcing connections using this set of terminals.

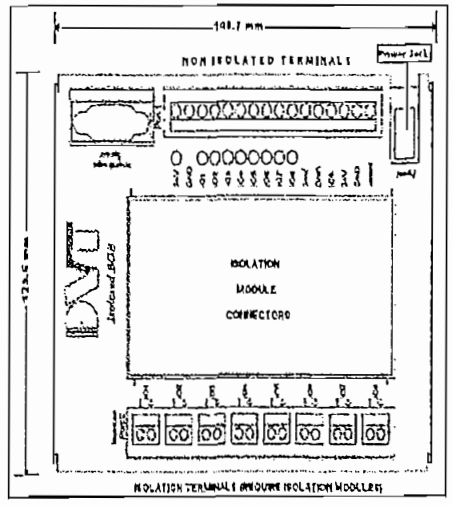
The second port, the one on the left side (see diagram) has the I/O connections to be used in conjunction with isolation modules. These terminals are inactive unless isolation modules are used. Users should decide which lines need isolation and install the corresponding modules (input or output) for those lines only. These modules can be used to provide power isolation when

DVT
SmartImage Sensor Specialists
Applied Engineering Department

working with devices that are using a separate power supply. The optional isolation modules can be acquired from third-party companies.

Isolation Modules

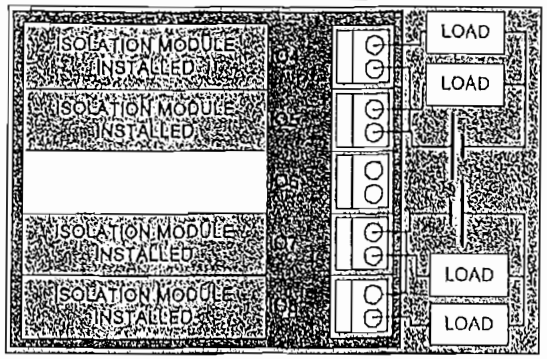
The isolation modules should be installed in the center region of the breakout board (see diagram). The breakout board is designed for G-4 type isolation modules. Specific part numbers depend on the power supply of the device to connect to. For instance, to interface with an external device running on 24 V, the OPTO 22 G4 IDC5 module can be used as an input isolation module and the Grayhill 70G-ODC5 can be used as an output isolation module. The first one is capable of handling voltages between 10V and 32V, and outputs a 5V logic signal to the isolation breakout board which is converted to the appropriate voltage and sent as an input to the SmartImage Sensor. The second one takes in the 5V logic signal from the breakout board and is rated at 60VDC and 3.5A on the output side, so a device running on 24 V DC can be connected on that end. These are just examples of isolation modules; all major electronics distributors carry many of these products.



Isolation Breakout Board Diagram indicating the size of the board.

Connections to Isolation Terminals

The connection between the isolation terminals of the breakout board and the external device is very simple and independent of the type of input/output required for the external device (that is NPN or PNP). The diagram below illustrates how the external devices should be connected to the breakout board. The diagram shows part of an isolated breakout board, which contains only four isolation modules at terminals IO4, IO5, IO7, and IO8. Notice that the external loads with their power supply can be connected in either of the ways shown in the diagram.



NOTE: the number of loads connected using each one of the sources in the diagram is just an example.

Specifications

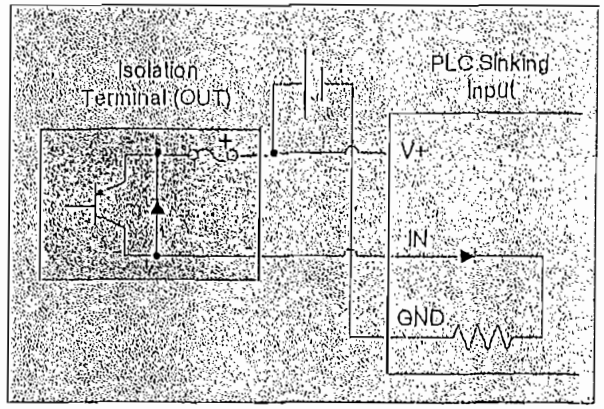
The board is housed in a metal/plastic DIN Rail-mountable case with powder coating on top. The breakout board is approximately 126mm wide by 149mm high by 45mm deep (including all necessary cable clearances). The table below summarizes the features available from the isolation breakout board and their purpose.

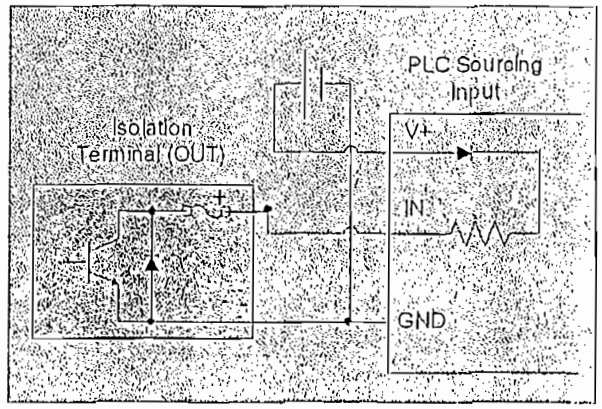
Features	Purpose
Power Jack	Included for the use of a 24V (1.42 A) wall-mount power supply. It should only be used for demonstration and laboratory purposes.
Power Terminals	There are three main power terminals (V+, GND, Shield) providing connections for an industrial power supply. There are two additional pairs of V+ and GND terminals.
Non-Isolated I/O Terminals	Located on the right side of the board there are screw-down terminals for the 8 I/O lines available on the Legend Series. There is an extra terminal connected to I/O line 8 intended to provide one more output for a strobe illumination. This output will be activated during image exposure only if it is configured in FrameWork as the Strobe output.
Isolated I/O Terminals	Located on the left side of the board, these are to be used in combination with isolation modules.
Isolation modules slots	There are 8 screw-down slots for input or output isolation modules.
15 pin high density D-sub plug	This provides connection to the I/O and power lines of the Legend Series systems which use a 10-pin keyed RJ-45 port.

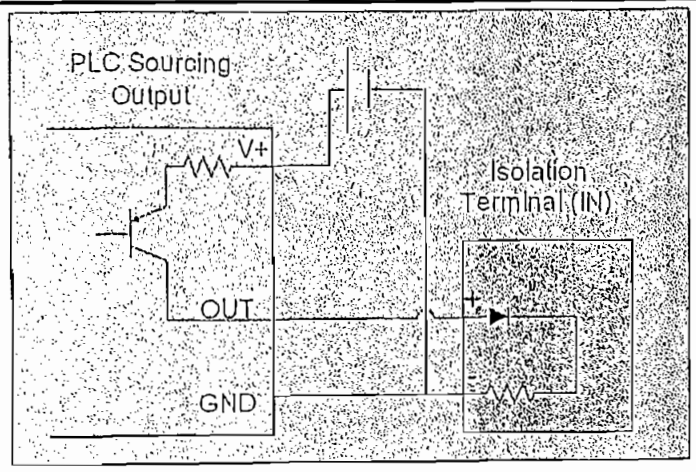
DVT
Smartimage Sensor Specialists
Applied Engineering Department

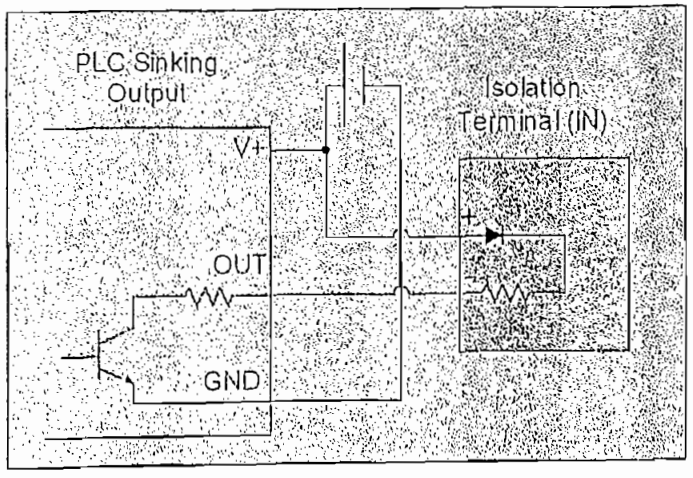
Connection diagrams for the Isolation terminals

The following diagrams illustrate four different methods to connect the SmartImage Sensor to external devices using the set of terminals that can be isolated by installing isolation modules in each. These setups are just suggestions, refer to the documentation of the external device you are using and that of the isolation modules you are using to ensure compatibility.







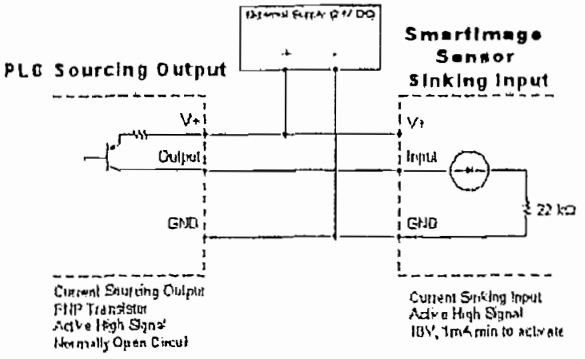


DVT
Smartimage Sensor Specialists
Applied Engineering Department

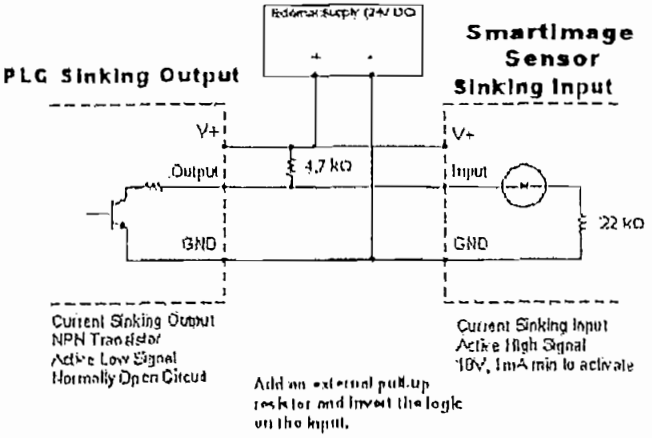
Connection Diagrams for the non Isolation terminals

The following diagrams illustrate different methods to connect the Smartimage Sensor to external devices using the set of terminals without isolation.

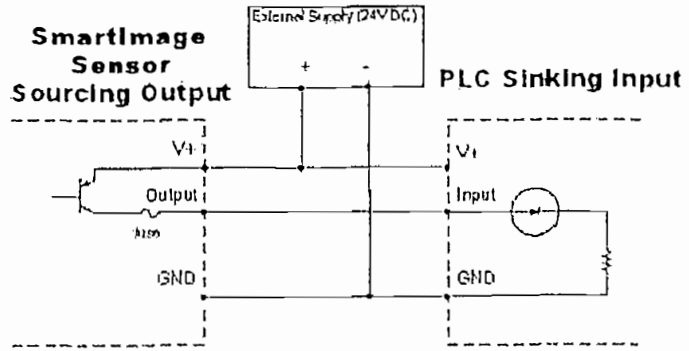
PLC Sourcing Output – Smartimage Sensor Sinking Input



PLC Sinking Output – Smartimage Sensor Sinking Input



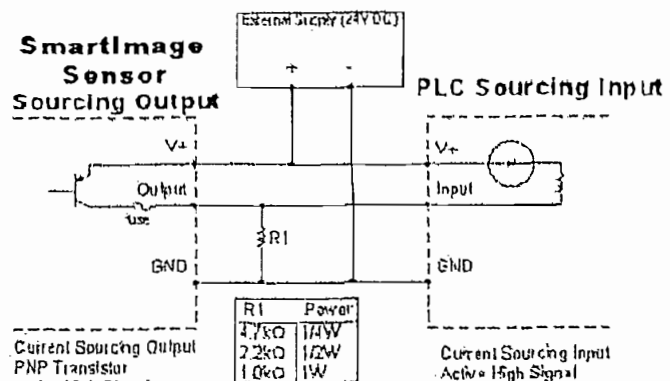
SmartImage Sensor Sourcing Output – PLC Sinking Input



Current Sourcing Output
PNP Transistor
Active High Signal
Normally Open Circuit
Maximum 100mA sourced

Current Sinking Input
Active High Signal

SmartImage Sensor Sourcing Output – PLC Sourcing Input



R1	Power
4.7kΩ	1/4W
2.2kΩ	1/2W
1.0kΩ	1W

Current Sourcing Output
PNP Transistor
Active High Signal
Normally Open Circuit
Maximum 100mA sourced

Add an external pull-down resistor and invert the logic on the output.

Current Sourcing Input
Active High Signal

- Lente LTC 05Z

13VM550

SHARPER BY DESIGN

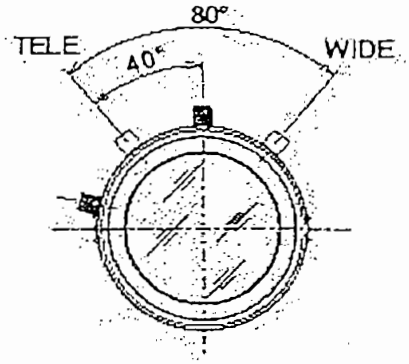
TAMRON

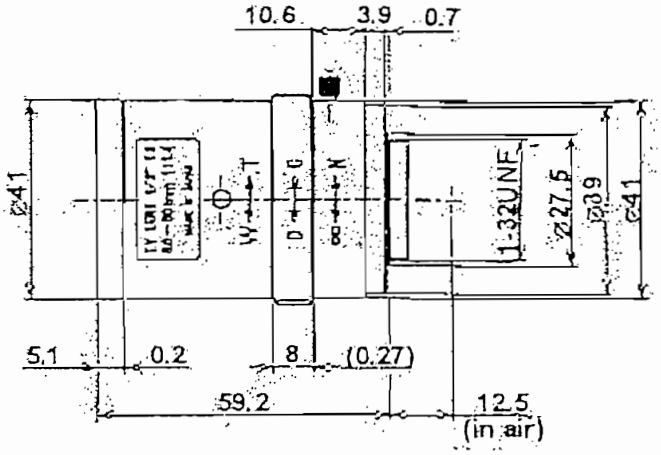
DVT PART NUMBER: LTC-05Z

VARIFOCAL

1/3" 5-50mm F/1.4

MANUAL





Specifications

	Focal Length	5-50mm
	Iris Range	F/1.4 to Close
	Iris Operation	Manual
	Focus Operation	Manual
	Clear Aperture of Front Lens	Ø29.5mm
	Clear Aperture of Rear Lens	Ø9.0mm
	Minimum Object Distance	23.0 in. (0.7m)
	Back Focus	10.05mm
	Filter Size	---
	Maximum Diameter	ø1.7 in. (ø43mm)
Lens Info	Overall Length	2.3 in. (59.2mm)
	Weight	---

For more information please call

1-800-Belden1

See Part-ups and Colors

Color Code Chart : No. 5 for DataTwist Cables (Modified Western Electric).pdf

Description:

4 twisted paired pairs, shielded, 24 AWG solid bare copper conductors insulated polyolefin insul.,O/A lateral foil shield, spiral tinned copper drain wire, tinned copper braid, industrial grade oil resistant PVC jacket. Seq. marking at 2 ft. intervals.

SUITABLE APPLICATIONS:

Suitable Applications: Industrial Ethernet, Harsh Environments, Category 5e, Gigabit Ethernet, 100BaseTX, 100BaseVG ANYLAN, 155ATM, 622ATM, NTSC/PAL Component or Composite Video, AES/EBU Digital Video, RS-422, Noisy Environments, CMX - Outdoor, EtherNetIP™

PHYSICAL CHARACTERISTICS:

CONDUCTOR:

Number of Pairs: 4

Total Number of Conductors: 8

AWG: 24

Stranding: Solid

Conductor Material: BC - Bare Copper

INSULATION:

Insulation Material: PO - Polyolefin

Pair Color Code Chart :

Number	Color	Number	Color
1	White-Blue Stripe & Blue	3	White-Green Stripe & Green
2	White-Orange Stripe & Orange	4	White-Brown Stripe & Brown

OUTER SHIELD:

Outer Shield Material Trade Name: Beldfoil ®

Outer Shield Type: Braid/Tape

Outer Shield Material :

Layer Number	Material Trade Name	Type	Material	% Coverage (%)
1	Beldfoil ®	Tape	Aluminum Foil-Polyester Tape	100
2		Braid	TC - Tinned Copper	70

OUTER SHIELD DRAIN WIRE:

Outer Shield Drain Wire AWG: 24

Outer Shield Drain Wire Stranding: 7x32

Outer Shield Drain Wire Conductor Material: TC - Tinned Copper

Outer Shield Diameter: .239 in.

OUTER JACKET:

Outer Jacket Material: Industrial Grade PVC - Polyvinyl Chloride

OVERALL NOMINAL DIAMETER:

Overall Nominal Diameter: .330 in.

MECHANICAL CHARACTERISTICS:

Operating Temperature Range: -40°C To +75°C

Installation Temperature Range: -25°C To +75°C

Bulk Cable Weight: 50 lbs/1000 ft.

Max. Recommended Pulling Tension: 75 lbs

Min. Bend Radius (Install): 1.32 in

APPLICABLE SPECIFICATIONS AND AGENCY COMPLIANCE:

APPLICABLE STANDARDS:

NEC/(UL) Specification: CMR, CMX-OUTDOOR, UL444

CEC/(UL) Specification: CMR

IEC Specification: IS11801 Category 5

TIA/EIA Specification: 568B.2 Category 5e

Other Specification: NEMA WC-63.1 Category 5e, EtherNetIP™ Compliant

FLAME TEST:

UL Flame Test: UL1666 Riser

CSA Flame Test: FT4

SUITABILITY:

Suitability - Outdoor: Yes

Sunlight Resistance: Yes

Oil Resistance: Yes

PLENUM/NON-PLENUM:

Plenum (Y/N): N

BELDENCable™

7921A Paired - DataTuff® Category 5e

ELECTRICAL CHARACTERISTICS:

Nom. Mutual Capacitance @ 1 KHz	15 pF/ft
Maximum Capacitance Unbalance (pF/100 m)	330 pF/100 m
Nominal Velocity of Propagation	70 %
Maximum Delay (ns/100 m)	538 @ 100MHz ns/100 m
Maximum Delay Skew (ns/100m)	45 ns/100 m
Maximum Conductor DC Resistance @ 20 Deg. C	9.38 Ohms/100 m
Maximum DCR Unbalance @ 20 Deg. C	3 %
Max. Operating Voltage - UL	300 V RMS

ELECTRICAL CHARACTERISTICS - PREMISE:

Premise Cable Electricals Table 1 :

Frequency (MHz)	Max. Attenuation (dB/100 m)	Min. NEXT (dB)	Min. PSNEXT (dB)	Min. ACR (dB)	Min. PSACR (dB)	Min. Return Loss (dB)	Min. Structural Return Loss (dB)
1	2.0	65.3	62.3	63.0	60.0	20.0	25
4	4.1	56.3	53.3	51.0	49.0	23.6	23.0
8	5.8	51.8	48.8	46.0	43.0	25.4	24.5
10	6.5	50.3	47.3	43.0	41.0	26.0	23.0
16	8.2	47.3	44.3	39.0	36.0	26.0	23.0
20	9.3	45.8	42.8	36.5	33.5	26.0	23.0
25	10.4	44.3	41.3	33.9	30.9	25.5	24.3
31.25	11.7	42.9	39.9	31.0	28.0	25.0	23.6
62.5	17.0	38.4	35.4	22.0	19.0	23.5	21.5
100	22.0	35.3	32.3	14.0	11.0	22.5	20.1

Premise Cable Electricals Table 2 :

Frequency (MHz)	Input (Unfitted) Impedance (Ohms)	Fitted Impedance (Ohms)	Min. ELFEXT (dB)	Min. PSELFEXT (dB)
1	100 ± 15	95-110	63.8	60.8
4	100 ± 15	95-110	51.7	48.7
8	100 ± 15	95-107	45.7	42.7
10	100 ± 15	95-107	43.8	40.8
16	100 ± 15	95-107	39.7	36.7
20	100 ± 15	95-107	37.7	34.7
25	100 ± 15	95-107	35.8	32.8
31.25	100 ± 15	95-107	33.9	30.9
62.5	100 ± 15	95-107	27.8	24.8
100	100 ± 15	95-107	23.8	20.8

NOTES:

Notes: US Patent #'s 5,606,151; 5,734,126. EtherNet/IP is a trademark of ControlNet International, Ltd. under license by Open DeviceNet Vendor Association, Inc. Operating temperatures are subject to length de-rating. Cable passes -40C Cold Bend per UL 1581.

BELDENCable™

7921A Paired - DataTuff® Category 5e

PUT-UPS AND COLORS:

Item	Description	Put-Up (ft.)	Ship Weight (lbs.)	Jacket Color	Notes
7921A 0101000	4 PR 24 PP SH PVC	1000	55	BLACK	C
7921A 0102000	4 PR 24 PP SH PVC	2000	106	BLACK	C

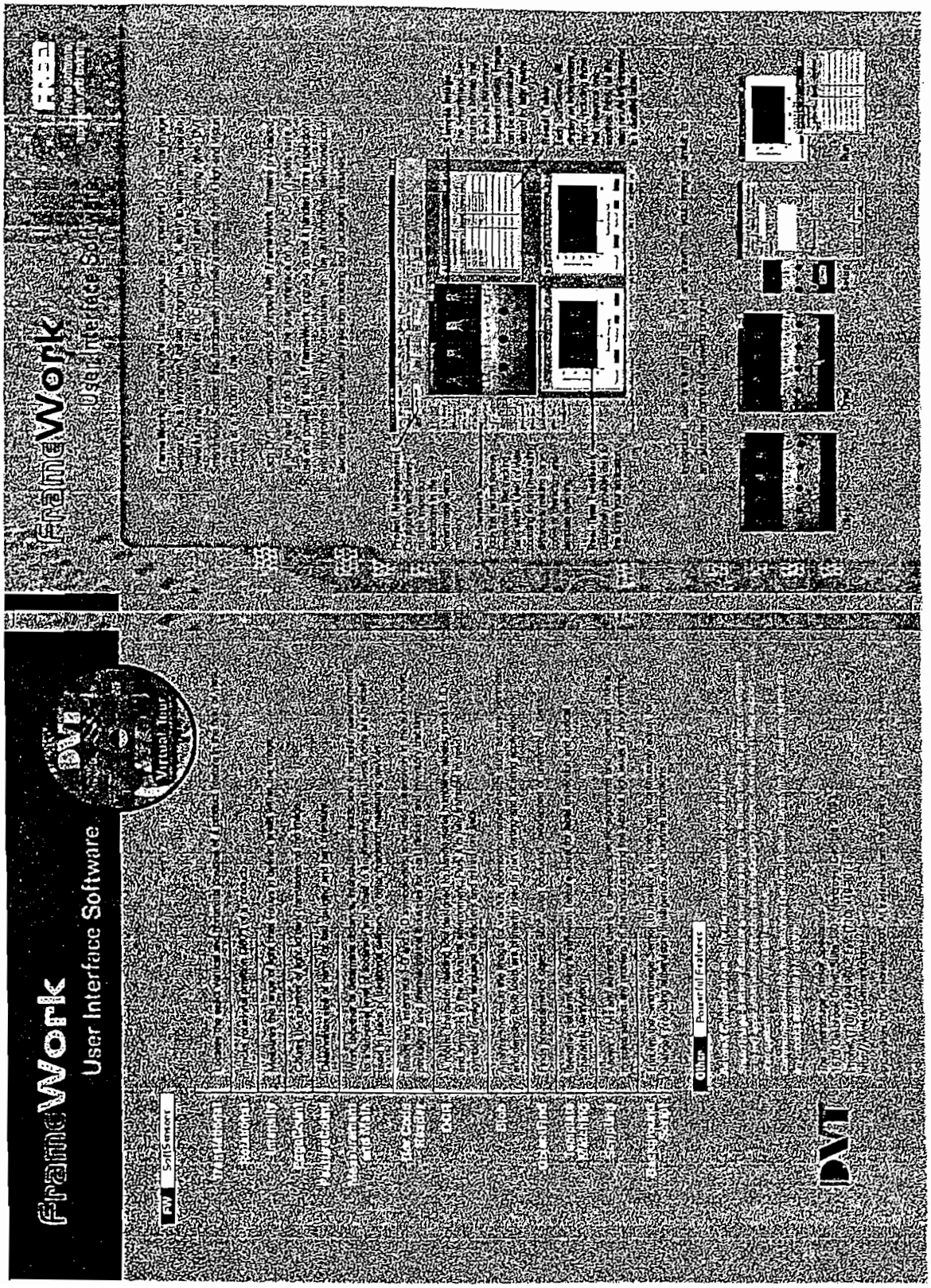
C - CRATE REEL PUT-UP.

Revision Number: 5 Revision Date: 12-20-2003

© 2003 Belden Wire & Cable Company
All Rights Reserved.

Although Belden Electronics Division ("Belden") makes every reasonable effort to ensure their accuracy at the time of this publication, information and specifications described herein are subject to error or omission and to change without notice, and the listing of such information and specifications does not ensure product availability.
Belden provides the information and specifications herein on an "AS IS" basis, with no representations or warranties, whether express, statutory or implied. In no event will Belden be liable for any damages (including consequential, indirect, incidental, special, punitive, or exemplary damages) whatsoever, even if Belden has been advised of the possibility of such damages, whether in an action under contract, negligence or any other theory, arising out of or in connection with the use, or inability to use, the information or specifications described herein.
All sales of Belden products are subject to Belden's standard terms and conditions of sale.

- FrameWork PDF



SoftSensors

ActiveX
Serves custom output time for OEMs and machine builders. With drag-and-drop ActiveX controls for creating a custom interface.

Bar Code Reader
Reads and interprets 1-D bar codes and 2-D data matrix symbologies now (and tomorrow) for inventory and part tracking in a variety of industries.

Blob Analysis
Analyzes part shapes within the SmartImage Sensor's field of View for results concerning characteristics like size, parameters and centroid, often used for part sorting and location.

Coordinate Transformation
Designed to take a pair of coordinates output from another SoftSensor and convert the units from pixels to real-world unit systems. Often used with motion control devices.

SmartImage Sensor Emulator
Mimics a DVT camera by running previously saved images through FrameWork on line. Great for training, presentations, application development and support from any location worldwide.

Intensity
Counts the number of bright and dark pixels in a region of interest. Used for defect detection and part presence or absence.

Lens Distortion Correction
Corrects for non-linear barrel distortion, and is commonly used with short focal length lenses.

Multi Tools and Precision Measurement
Computes dimensions of a variety of part and shapes such as hole diameter, distance, angles and more.

Object Find
Learns the shape of a part so the SmartImage Sensor can inspect, measure, partially occluded and overlapping parts in a wide range of lighting conditions.

OCR
Reads and interprets barcodes, serial numbers, labels, product IDs and symbols, handwritten scripts and rotated characters, and is trainable to read symbols and foreign alphabets.

PDF 417
Accurate software tool easily reads stacked bar codes commonly used in the automotive industry, packaging and for part tracking.

Script Tool
Utilizes tools to develop specific algorithms and inspection criteria to control processes and monitor status. Errors routed by to solve any problem on the manufacturing line without complex programming.

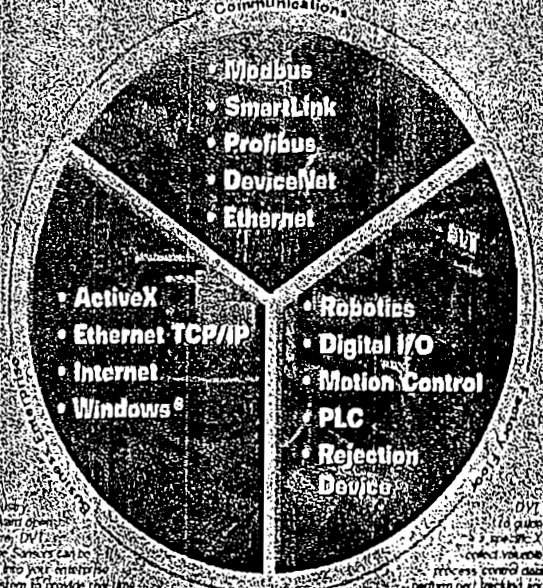
SmartLink
Allows operator to monitor up to 10 linked SmartImage Sensors from one display link various PC's on the manufacturing floor. Offers less complexity and hardware costs.

Thresholding
Permit hot keys and SoftSensors triggering sequences (tools) based on gradient or intensity.

All **FrameWork** software is Backward Compatible, Upward Migratable and Free to all Users

DVT SmartImage Sensors are simply focused on a network that captures information to increase productivity, efficiency, and accuracy in your enterprise. With FrameWork, DVT products add complete connectivity and seamless integration from the factory floor to the front office.

Versatile communications and networking options give you the flexibility of delivering data to a variety of control devices.



As the industry leader, DVT SmartImage Sensors can be integrated into your enterprise control system to provide real-time overview of information throughout the plant floor and across your business enterprise. Vital control information can be viewed via Ethernet by any department on the network and be accessed from any location around the world.

DVT allows you to guide a robot to a specific X-Y location, collect valuable statistical process control data, and even perform part tracking and inventory control functions.

ANEXO 3

MANUAL DE USUARIO

El manual de usuario describe la forma de utilizar la interfase hecha en Visual Basic, así como la Página de Internet que muestra el registro que llevan los contadores. El manual también ayudará a recalibrar el programa de inspección, si fuese necesario.

a. INTERFASE EN VISUAL BASIC

Esta interfase permite que operadores o el administrador del sistema visualicen el estado de los contadores, además se podrá acceder a los registros de imágenes con falla que la cámara haya capturado.

El programa de visualización de registros inicia automáticamente en el momento en el que la PC de envase, que es la que contiene el programa de interfase o HMI arranca. Si por alguna razón el programa fue cerrado se lo puede iniciar siguiendo los siguientes pasos:

- Buscar el icono del programa en el escritorio. Ver Figura 1.
- Dar doble clic sobre el icono
- El programa de interfase se ejecuta solo. La ejecución automática ocurre únicamente si la interfase no estaba trabajando antes o si la computadora esta iniciando, pero si estando en la pantalla de comunicación se presiono el botón salir el programa se pone en la pantalla de presentación y se queda allí hasta que alguien lo vuelva a poner en el modo de comunicación

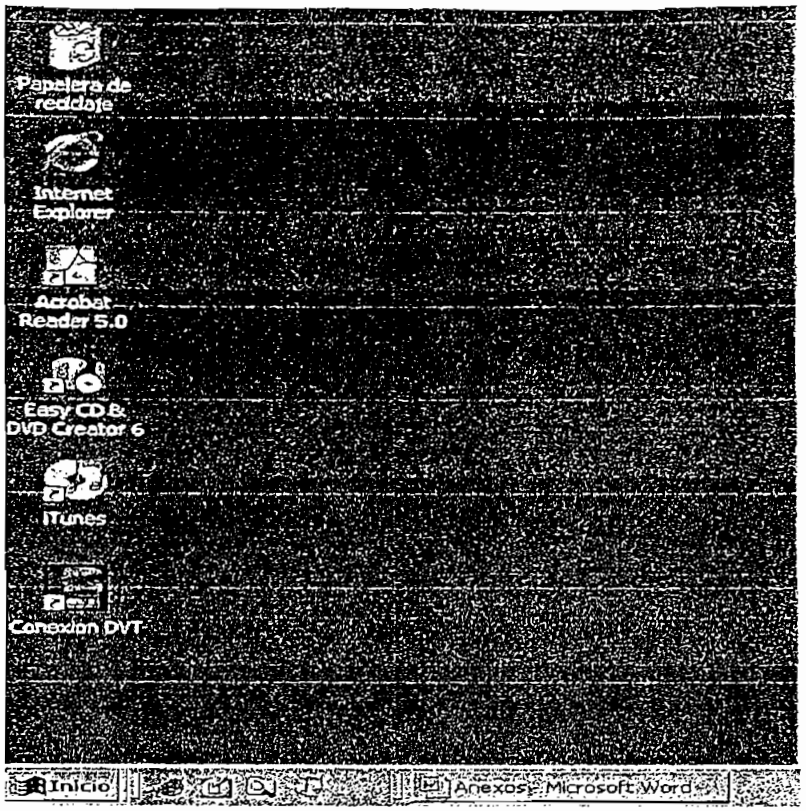


FIGURA 1: Acceso directo desde el escritorio a Conexión DVT.

Una vez que se ha iniciado el programa, la primera ventana que tenemos es la mostrada en la Figura 2. Si esta ventana se carga desde el inicio; automáticamente después de 10 segundos se carga la pantalla de comunicación, ver Figura 3. Si, la ventana de la Figura 2 está activa luego de que el usuario cerró la ventana de comunicación; entonces se puede hacer uso de cualquiera de los dos botones que hay en la pantalla y son:

- Boton Continuar.- este boton carga la ventana de comunicaciones
- Boton Salir.- cierra completamente la interfase

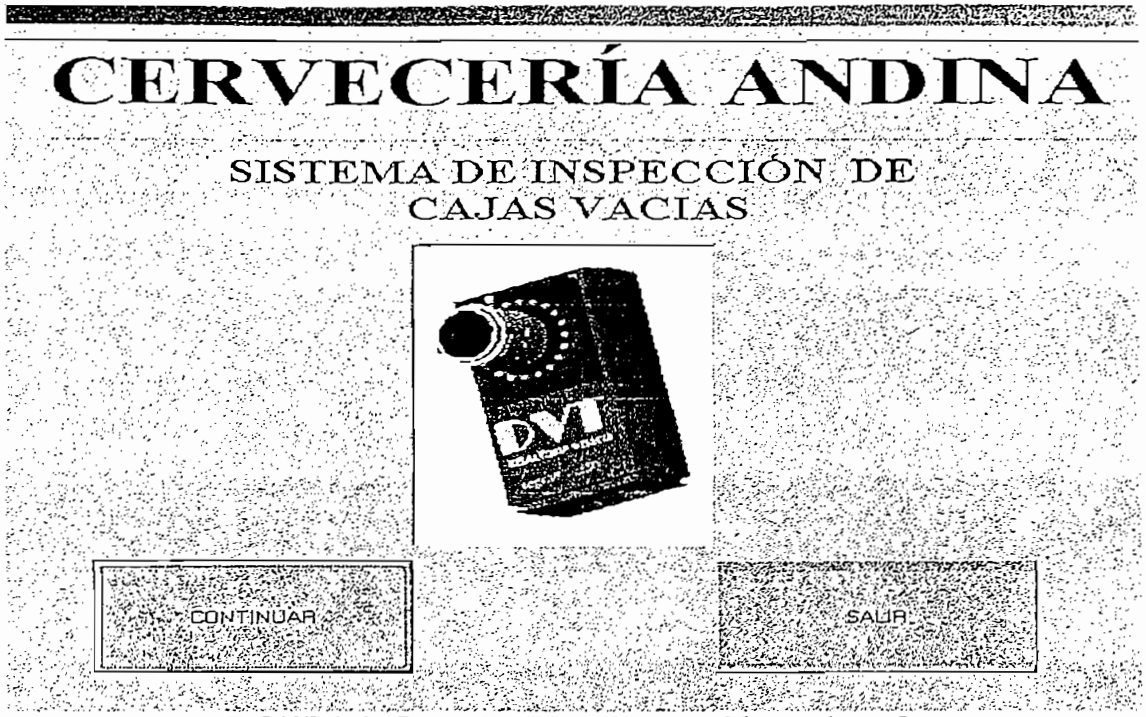


FIGURA 2: Pantalla de presentación de Interfase

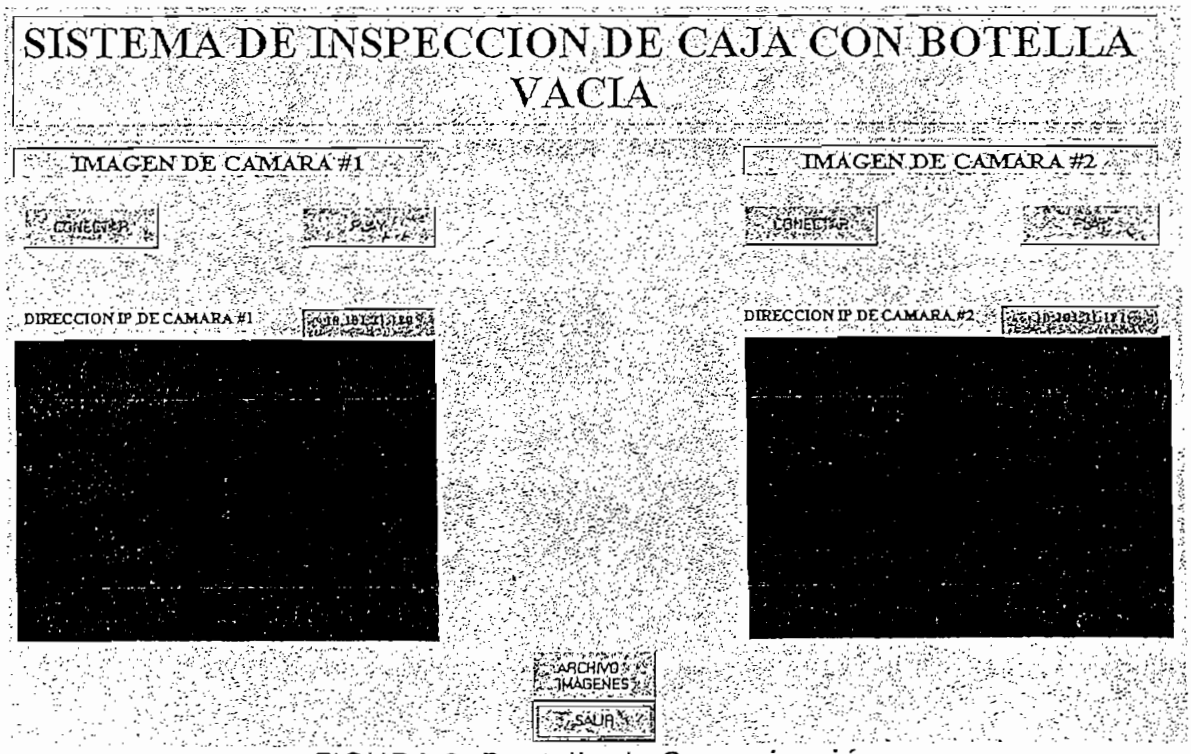


FIGURA 3: Pantalla de Comunicación

Cuando la pantalla de comunicaciones se ejecuta, existe un tiempo de 10 segundos en el que la PC procede buscar en la red a las cámaras. *Si por alguna*

razón la dirección IP de alguna cámara fue cambiada el programa solicita al operador ingresar una nueva dirección IP para intentar conectarse nuevamente; o puede optar por cerrar la interfase.

En la Figura 4 se ve la ventana de comunicación cuando se ha conectado correctamente y en la Figura 5 se ve la ventana de comunicación cuando ha ocurrido una falla en la conexión.

FIGURA 4: Pantalla de Comunicación cuando se conectó correctamente

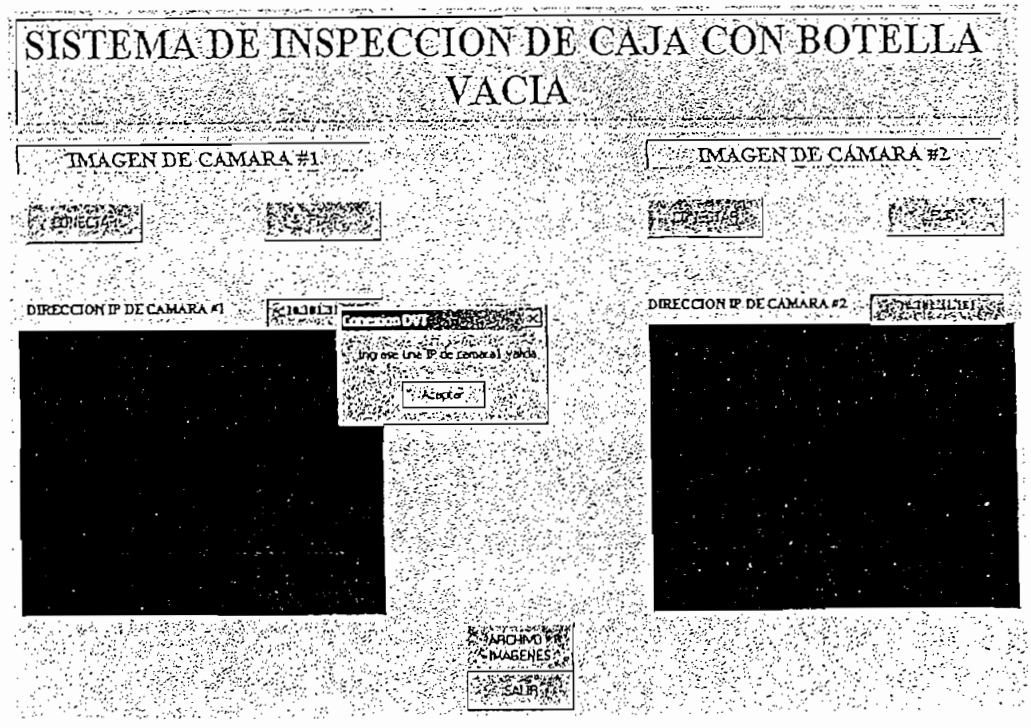


FIGURA 5: Pantalla de Comunicación cuando la dirección IP es incorrecta

Si las conexiones fueron exitosas el programa empieza a grabar las imágenes fallosas que la cámara está capturando y se activan dos botones por cada visor de imágen que son:

- ***Boton Conectar:*** permite visualizar los contadores. Figura 6
- ***Boton Play:*** permite visualizar las imágenes que se van adquiriendo a medida que pasa una jaba. Figura 7

FIGURA 6 y 7: Despliegue de imágenes en tiempo real

Independientemente de si la conexión fue exitosa se tiene disponible dos botones de uso general que son:

- ***Boton Archivo de Imágenes:*** este boton abre la ventana del archivo de imágenes, en la que el usuario puede ver las imágenes de las jabas que contenían alguna falla así como la fecha y hora en la que ingreso a la línea de envase. Esta ventana esta mostrada en la Figura 8 y para seleccionar una imagen solo se hace un clic sobre un archivo que esta mostrado en el visor de archivo de imágenes. Existen dos visores que corresponden a cada cámara.

FIGURA 8: Pantalla de Archivo de Imágenes Fallidas

- *Boton Salir:* este botón cierra la ventana de conexiones y conduce a la ventana de presentación. Una vez que regresa a la ventana de presentación no se reconecta solo, por tanto el usuario debe presionar el botón continuar si desea tener acceso nuevamente a los datos registrados por las cámaras.

b. INTERFASE PARA INGRESAR A LA PAGINA DE INTERNET

Con ayuda de esta interfase se logró ubicar en la red los datos registrados por cada una de las cámaras, datos como: número de botellas en buen y mal estado, así como registros de jabas en buen y mal estado.

La ventaja de tener una interfase en la red en modo de una página de Internet tal como se muestra en la Figura 9, es la facilidad de acceder a los registros de las cámaras digitales sin necesidad de tener acceso a la computadora de envase, que es la única que cuenta con el acceso directo tal como se vio en la Figura 1.

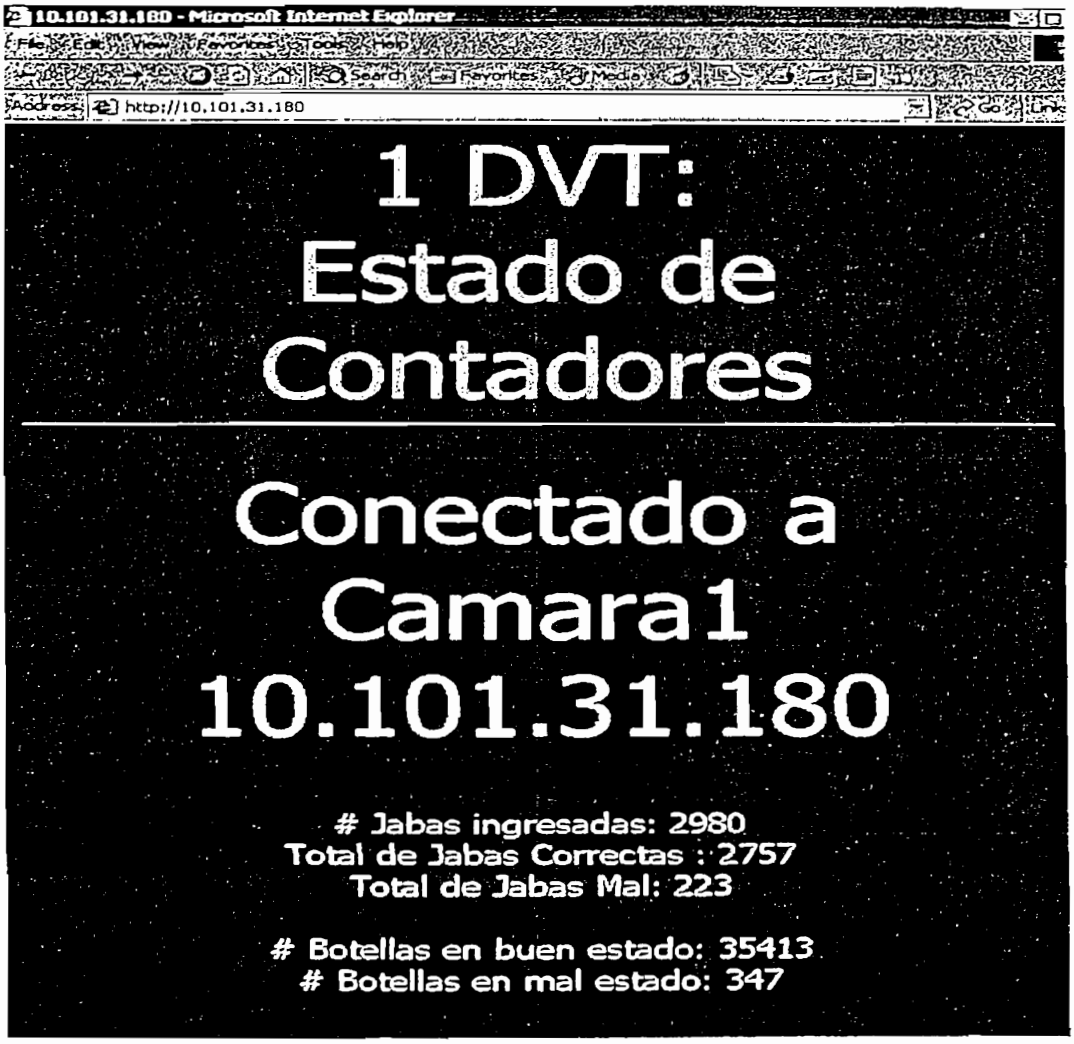


FIGURA 9: Página de Internet con datos de la cámara

Tal como se aprecia en la Figura 9, esta pantalla corresponde a una página de Internet, en la cual se visualiza los registros de una de las cámaras, también se ve que en el espacio de direcciones se ha ubicado la dirección IP de la cámara que es el forma para que los datos puedan ser visualizados.

Si el usuario desea acceder a los registros de la otra cámara, tan solo debe digitar en la barra la dirección IP de la otra cámara e inmediatamente se muestran los datos respectivos.

Por medio de esta página de Internet, los datos visualizados corresponden a los capturados en el momento de la conexión, es decir no se actualizan a medida que la página está conectada a la cámara.

NOTA: La diferencia entre la interfase en Visual Basic y la interfase en Internet, es que la primera recibe periódicamente los valores actuales de los registros de cada una de las cámaras, siempre y cuando la conexión haya sido exitosa. Por supuesto, eso no sucede con la página de Internet, ya que muestra un solo valor de los registros y corresponden a los datos capturados en el momento de la conexión.

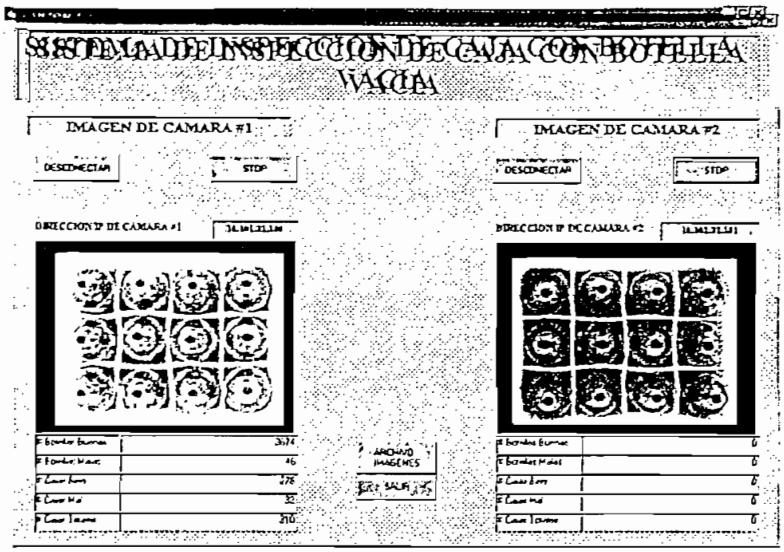
Gracias a la página de Internet, estos datos pueden ser vistos desde cualquier computadora de Cervecería Andina.

ANEXO 4
CÓDIGO FUENTE

El código fuente que se presenta es el que pertenece a la interfase desarrollada en Visual Basic y para los scripts de la cámara de Reset Automático, de Código HTML, de la Tabla de Recalibraciones, Programa de Asignación de Threshold a Sensores de Manchas y Programa de Análisis de resultados y almacenamiento de Registros.

- Código para las Pantallas de Visual Basic

```
Private Sub Command1_Click()
Pasauto.Enabled = False
frmhoja1.Hide
frmhoja2.Show 1

End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()

End Sub

Private Sub Pasauto_Timer()
Command1_Click
End Sub
```




```
'acciones cuando se carga esta ventana
Private Sub Form_Load()
 'acciones de camara1
   cnx1 = 1
   enable1 = 0
   estado1 = 0
   pl1 = 0
   retf1.Enabled = False

   AllCam1.Enabled = False
   FailCam1.Enabled = False

   ConnCam1.Caption = "CONECTAR"
   ConnCam1.Enabled = False
   ConnCam1.Visible = True

   AccionCam1.Caption = "PLAY"
   AccionCam1.Enabled = False
   AccionCam1.Visible = True

   BbienCam1.Visible = False
   ValBbienCam1.Visible = False
   BmalCam1.Visible = False
   ValBmalCam1.Visible = False
   CajasCam1.Visible = False
   ValCajasCam1.Visible = False
   CajasbienCam1.Visible = False
   ValCajasbienCam1.Visible = False
   CajasmalCam1.Visible = False
   ValCajasmalCam1.Visible = False

   IP_txt_cam1.Enabled = False
   tconn1.Enabled = True

   Camara1.Visible = True


   Timer1.Enabled = False

  'acciones de la camara 2
```

```
    cnx2 = 1
    enable2 = 0
    estado2 = 0
    pl2 = 0
    retf2.Enabled = False

    AllCam2.Enabled = False
    FailCam2.Enabled = False

    ConnCam2.Caption = "CONECTAR"
    ConnCam2.Enabled = False
    ConnCam2.Visible = True

    AccionCam2.Caption = "PLAY"
    AccionCam2.Enabled = False
    AccionCam2.Visible = True


    BbienCam2.Visible = False
    ValBbienCam2.Visible = False
    BmalCam2.Visible = False
    ValBmalCam2.Visible = False
    CajasCam2.Visible = False
    ValCajasCam2.Visible = False
    CajasbienCam2.Visible = False
    ValCajasbienCam2.Visible = False
    CajasmalCam2.Visible = False
    ValCajasmalCam2.Visible = False

    IP_txt_cam2.Enabled = False
    tconn2.Enabled = True

    Camara2.Visible = True


    Timer2.Enabled = False

End Sub

Private Sub IP_txt_cam1_KeyPress(KeyAscii1 As Integer)

If KeyAscii1 = 13 Then
Debug.Print "llego1"
tconn1.Enabled = True
End If

End Sub

Private Sub IP_txt_cam2_KeyPress(KeyAscii2 As Integer)

If KeyAscii2 = 13 Then
Debug.Print "llego2"
tconn2.Enabled = True
End If

End Sub

Private Sub retf1_Timer()
retf1.Enabled = False
```

```
FailCam1.Value = 1
End Sub

Private Sub retf2_Timer()
retf2.Enabled = False
FailCam2.Value = 1
End Sub

Private Sub SalirWin2_Click()

enable1 = 0
enable2 = 0
estado1 = 0
estado2 = 0
pl1 = 0
pl2 = 0
If cnx1 = 100 Then
Camara1.ClearImage
Camara1.StopImages
Camara1.Disconnect
End If
If cnx2 = 100 Then
Camara2.ClearImage
Camara2.StopImages
Camara2.Disconnect
End If
cnx1 = 1
cnx2 = 1
SysTerm1.Close
SysTerm2.Close
Unload Me
frmhoja1.Show 1

End Sub

Private Sub SysTerm1_DataArrival(ByVal bytesTotal As Long)
   Dim sysData1 As String
   Dim resultado1 As Integer
   SysTerm1.GetData sysData1, vbString
   buffer1 = buffer1 & sysData1
   'Checking for the command prompt arrival
   If (InStr(buffer1, "?") <> 0 Or InStr(buffer1, "!") <> 0) Then
      resultado1 = InStr(buffer1, "%")
      If (resultado1 <> 0) Then
         If (CInt(Mid(buffer1, resultado1 + 2, 1)) = 0) Then
            If (operacion1 = "R") Then
               If (i1 = 0) Then
                  ValBbienCam1.Caption = Mid(buffer1, 4, InStr(4, buffer1, vbCrLf) - 4)
                  i1 = i1 + 1
                  VerCont1_Click
               ElseIf (i1 = 1) Then
                  ValBmalCam1.Caption = Mid(buffer1, 4, InStr(4, buffer1, vbCrLf) - 4)
                  i1 = i1 + 1
                  VerCont1_Click
               ElseIf (i1 = 2) Then
                  ValCajasCam1.Caption = Mid(buffer1, 4, InStr(4, buffer1, vbCrLf) - 4)
                  i1 = i1 + 1
                  VerCont1_Click
               ElseIf (i1 = 3) Then
```

```
            ValCajasbienCam1.Caption = Mid(buffer1, 4, InStr(4, buffer1, vbCrLf) - 4)
            i1 = i1 + 1
            VerCont1_Click
          ElseIf (i1 = 4) Then
            ValCajasmalCam1.Caption = Mid(buffer1, 4, InStr(4, buffer1, vbCrLf) - 4)
            i1 = 0
          End If

        End If
      Else
        MsgBox "Unsuccessful Register Operation"
      End If
    End If
    buffer1 = ""
  End If


End Sub



Private Sub SysTerm2_DataArrival(ByVal bytesTotal As Long)

  Dim sysData2 As String
  Dim resultado2 As Integer
  SysTerm2.GetData sysData2, vbString
  buffer2 = buffer2 & sysData2
  'Chequeo comandos de arrivo de datos
  If (InStr(buffer2, "?") <> 0 Or InStr(buffer2, "!") <> 0) Then
    resultado2 = InStr(buffer2, "%")
    If (resultado2 <> 0) Then
      If (CInt(Mid(buffer2, resultado2 + 2, 1)) = 0) Then
        If (operacion2 = "R") Then
          If (i2 = 0) Then
            ValBbienCam2.Caption = Mid(buffer2, 4, InStr(4, buffer2, vbCrLf) - 4)
            i2 = i2 + 1
            VerCont2_Click
          ElseIf (i2 = 1) Then
            ValBmalCam2.Caption = Mid(buffer2, 4, InStr(4, buffer2, vbCrLf) - 4)
            i2 = i2 + 1
            VerCont2_Click
          ElseIf (i2 = 2) Then
            ValCajasCam2.Caption = Mid(buffer2, 4, InStr(4, buffer2, vbCrLf) - 4)
            i2 = i2 + 1
            VerCont2_Click
          ElseIf (i2 = 3) Then
            ValCajasbienCam2.Caption = Mid(buffer2, 4, InStr(4, buffer2, vbCrLf) - 4)
            i2 = i2 + 1
            VerCont2_Click
          ElseIf (i2 = 4) Then
            ValCajasmalCam2.Caption = Mid(buffer2, 4, InStr(4, buffer2, vbCrLf) - 4)
            i2 = 0
          End If

        End If
      Else
        MsgBox "Unsuccessful Register Operation"
      End If
    End If
```

```
        buffer2 = ""
    End If

End Sub

Private Sub tconn1_Timer()

On Error GoTo Error11

    Debug.Print "nada1"
        tconn1.Enabled = False
        IP_txt_cam1.Enabled = False
        Camara1.RemoteHost = IP_txt_cam1.Text
        Camara1.Connect
        retf1.Enabled = False
        cnx1 = 100
        enable1 = 123
        estado1 = 123
        AllCam1.Enabled = True
        FailCam1.Enabled = True
        'Camara1.PlayImages
        ConnCam1.Enabled = True
        AccionCam1.Enabled = True
        AccionCam1_Click

        Exit Sub
Error11:
        retf1.Enabled = False
        cnx1 = 1
        IP_txt_cam1.Enabled = True
        tconn1.Enabled = False
        MsgBox "Ingrese una IP de camara1 valida"


End Sub

Private Sub tconn2_Timer()

On Error GoTo Error22

    Debug.Print "nada2"
        tconn2.Enabled = False
        IP_txt_cam2.Enabled = False
        Camara2.RemoteHost = IP_txt_cam2.Text
        Camara2.Connect
        retf2.Enabled = False
        cnx2 = 100
        enable2 = 123
        estado2 = 123
        AllCam2.Enabled = True
        FailCam2.Enabled = True
        'Camara2.PlayImages
        ConnCam2.Enabled = True
        AccionCam2.Enabled = True
        AccionCam2_Click

        Exit Sub
Error22:
```

```
        retf2.Enabled = False
        cnx2 = 1
        IP_txt_cam2.Enabled = True
        tconn2.Enabled = False
        MsgBox "Ingrese una IP de camara2 valida"

End Sub

Private Sub Timer1_Timer()

VerCont1_Click

End Sub

Private Sub Timer2_Timer()

VerCont2_Click

End Sub

Private Sub VerCont1_Click()

Dim str1 As String
   operacion1 = "R"

   If i1 = 0 Then

        str1 = "#Rq" & "1000" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm1.SendData (str1)

   ElseIf i1 = 1 Then

        str1 = "#Rq" & "1004" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm1.SendData (str1)

   ElseIf i1 = 2 Then

        str1 = "#Rq" & "1008" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm1.SendData (str1)

   ElseIf i1 = 3 Then

        str1 = "#Rq" & "1012" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm1.SendData (str1)

   ElseIf i1 = 4 Then

        str1 = "#Rq" & "1016" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm1.SendData (str1)

   End If

End Sub

Private Sub VerCont2_Click()

Dim str2 As String
   operacion2 = "R"
```

```
    If i2 = 0 Then

        str2 = "#Rq" & "1000" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm2.SendData (str2)

    ElseIf i2 = 1 Then

        str2 = "#Rq" & "1004" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm2.SendData (str2)

    ElseIf i2 = 2 Then

        str2 = "#Rq" & "1008" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm2.SendData (str2)

    ElseIf i2 = 3 Then

        str2 = "#Rq" & "1012" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm2.SendData (str2)

    ElseIf i2 = 4 Then

        str2 = "#Rq" & "1016" & "|" & vbCrLf 'Crea los comandos de envio
        SysTerm2.SendData (str2)

    End If

End Sub

Private Sub VisorImagen_Click()
VisorImagenes.Show 1

End Sub
```

```
Private Sub ExitVI_Click()
Unload Me
```

```
End Sub

Private Sub FileCam1_Click()
Dim direccion1 As String
Dim caja1 As Picture
Dim Ruta0 As String
Dim Ruta1 As String
Ruta0 = FileCam1.FileName
Ruta1 = FileCam1.Path
direccion1 = Ruta1 & "\" & Ruta0
Set caja1 = LoadPicture(direccion1)
ImageCam1.Picture = caja1
valf1.Caption = Mid(Ruta0, 7, 10)
valh1.Caption = Mid(Ruta0, 18, 11)

End Sub

Private Sub FileCam2_Click()
Dim direccion2 As String
Dim caja2 As Picture
Dim Ruta2 As String
Dim Ruta3 As String
Ruta2 = FileCam2.FileName
Ruta3 = FileCam2.Path
direccion2 = Ruta3 & "\" & Ruta2
Set caja2 = LoadPicture(direccion2)
ImageCam2.Picture = caja2
valf2.Caption = Mid(Ruta2, 7, 10)
valh2.Caption = Mid(Ruta2, 18, 11)

End Sub

Private Sub Form_Load()
FileCam1.Path = "c:\InspectorCajaVacia\ImagenesFailCam1"
FileCam2.Path = "c:\InspectorCajaVacia\ImagenesFailCam2"

End Sub
```

- Script de Reset Automático de las Cámaras

```
class ResetRegistrosCam1
{
   public static void main()
   {
//Variable generales de este programa
            int         bit = 1;                    //bit para selecionar una entrada
            int         tiempo = 0;                 //almacenamiento del
tiempo parado por la linea
            int         temp;
            int         estado;                     //estado de la espera por
arranque o parada de la linea
            byte    encero;                         //registro para control de
encerado,registro 1040
            byte    condicion;                      //condicion para salir lazo while off
            byte    luz;                            //estado de la parada por corte luz

//Configuracion del bloque de memoria flash a usar
            Flash   restmem = new Flash();
```

```
        restmem.RestoreRegs();                          //recupero registros previamente
guardados en flash


//Recuperando registros y en caso corte luz encerado de ellos
        encero          = RegisterReadByte(1040);

        if (encero != 176)                              //con 176 no resetea
        {
                RegisterWriteInteger(1000,0);   //Encero registro variable botellas bien
                RegisterWriteInteger(1004,0);   //Encero registro variable botellas
   mal
                RegisterWriteInteger(1008,0);   //Encero registro variable total jabas
                RegisterWriteInteger(1012,0);   //Encero registro variable jabas ok
                RegisterWriteInteger(1016,0);   //Encero registro variable jabas mal
        }

//Programa de ejecucion principal
        while(true)
   //Lazo de ejecucion continua
        {
   //espero arranque de linea
                DebugPrint("aun no entra");
                estado  = WaitOnInput(18,-1);           //detecta que arranco la
produccion para resetear las variables

                if (estado == 0)
                {
                        if (encero != 176)
                        {
                        RegisterWriteInteger(1000,0);           //botella bien
                        RegisterWriteInteger(1004,0);           //botella mal
                        RegisterWriteInteger(1008,0);           //jabas total
                        RegisterWriteInteger(1012,0);           //jabas ok
                        RegisterWriteInteger(1016,0);           //jabas mal
                        }
                encero = 176;
   //con 176 no se resetea
                RegisterWriteByte(1040,encero);
                restmem.SaveRegs(1000,1100);

                }
                sleep(300);
   //precaucion por rebotes

   //Espero por apagado linea
                bit             = 1;
                DebugPrint("aun no sale");
                estado  = WaitOnAnyInput(0,bit << 18, -1);      //detecta que la linea se
paro
                tiempo = 0;
                if (estado == 0)
                {
                        encero = 176;
   //con 176 no se resetea
                        RegisterWriteByte(1040,encero);
                        restmem.SaveRegs(1000,1100);
                        sleep(300);
```

```
//inicio de lazo que detecta parada total o corta
        do
        {
                bit             = 1;
                if ((GetInputs() & (bit<<18)) != 0)
                {
                        condicion = 0;
                }
                else
                {
                        sleep(448);
                        tiempo = tiempo + 1;
                        condicion = 255;
                }

                if (tiempo > 7200)
                {
                //Transfiero registros globales a generales en caso de que
la parada es total
                        temp    = RegisterReadInteger(1000);  //lee      el
registro botella bien
                        RegisterWriteInteger(1020,temp);
    //guarda el registro global de produccion botella bien
                        temp    = RegisterReadInteger(1004);  //lee      el
registro botella mal
                        RegisterWriteInteger(1024,temp);
    //guarda el registro global de produccion botella mal
                        temp    = RegisterReadInteger(1008);  //lee      el
registro jaba total
                        RegisterWriteInteger(1028,temp);
    //guarda el registro global de produccion jaba total
                        temp    = RegisterReadInteger(1012);  //lee      el
registro jaba ok
                        RegisterWriteInteger(1032,temp);
    //guarda el registro global de produccion jaba ok
                        temp    = RegisterReadInteger(1016);  //lee      el
registro jaba mal
                        RegisterWriteInteger(1036,temp);
    //guarda el registro global de produccion jaba mal

                        encero = 000;
                        condicion = 0;
                        RegisterWriteByte(1040,encero);
                        RegisterWriteByte(1041,luz);
                        restmem.SaveRegs(1000,1100);
                }
                else
                {
                        encero = 176;
    //con 176 no se resetea
                        RegisterWriteByte(1040,encero);
                        RegisterWriteByte(1041,213);          //palabra
para visual grabe en excel
                }

        } while ( condicion == 255);

}                                                     //fin     lazo     de
deteccion de paro
```

```
            }                                                    //fin de lazo
permanente

    }
}
```

- **Script de Código HTML**

```
class HTMLcam1
{
    public static void main()
    {

            float    porcentaje1;
            float    porcentaje2;
            Socket Sock;
            Socket CSock;

//Configuracion para el llamado de linea CRLF
            String crlf ;
            byte cr[] = new byte[1];
            byte lf[] = new byte[1];
            cr[0] = 13;
            lf[0] = 10;
            crlf = String(cr) + String(lf);

//Constantes

            int Conecta = 300000;
            int EnlaceSock = 80;

            String IPDir = "10.101.31.180";
            String LogoCam2 = "c:/Tesistas Envase/ImageDVT/530MR.bmp";
            String LogoCASA = "c:/Tesistas Envase/ImageDVT/CASA.bmp";
            String LogoDVT = "c:/Tesistas Envase/ImageDVT/LogoDVT.bmp";



//Variable declarations
            long Result, Resultado1, Resultado2;
            String Respuesta = "";

            byte in[] = new byte[1];
            Product P;


//Camara1 WebServer script
            while (true)
            {
                    Sock = Socket();
                    CSock = Socket();

                    Result = Sock.Bind(EnlaceSock);
                    //DebugPrint("Bind:  " + Result);
                    Result = Sock.Listen();
                    //DebugPrint("Listen:  " + Result);

                    do
```

```
{
        Result = Sock.Accept(CSock);
} while (Result != 0);

DebugPrint("Conección Aceptada.");

CSock.RecvTO = Conecta;
Respuesta = "";
in[0] = 0;


//HTML paginas

String HTMLFalla = "<html><head><title>404 Not Found</title></head><body><h1>Not Found</h1><p>The requested URL was not found on this server.</p><hr size=5><address>&copy2002 DVT Corporation</address><div align='center'><a href='http://www.dvtsensors.com/'><img src='" + LogoDVT + "' alt='DVT Corporation Website' border=0></a></div></body></html>";
String HTML = "<html><head><title>" + IPDir + "</title></head><body bgcolor='black'><div align='center'><font face='Tahoma' size='7' color='Yellow'>INSPECTOR TRANSPORTADORA 1 <br>Estado de Contadores</font><hr><br><font size='6' face='Tahoma' color='white'>Conectado a Camara1 " + IPDir + "</font><font face='Tahoma' color='white'><br><p>";

{

porcentaje1 = (100*RegisterReadInteger(1004));
porcentaje1 = porcentaje1/(RegisterReadInteger(1004)+RegisterReadInteger(1000));
porcentaje2 = (100*RegisterReadInteger(1024));
porcentaje2 = porcentaje2/(RegisterReadInteger(1024)+RegisterReadInteger(1020));
HTML = HTML + "<div align='center'><font face='Tahoma' size='4' color='White'>REGISTROS DE LA PRODUCCION ACTUAL<br>";
HTML = HTML + "# Jabas ingresadas: "+ RegisterReadInteger(1008) + "<br>";
HTML = HTML + "Total de Jabas Correctas : " + RegisterReadInteger(1012) + "<br>";
HTML = HTML + "Total de Jabas Mal: " + RegisterReadInteger(1016) + "<br>"+ crlf + crlf;
HTML = HTML + "# Botellas en buen estado: "+ RegisterReadInteger(1000) + "<br>";
HTML = HTML + "# Botellas en mal estado: "+ RegisterReadInteger(1004) + "<br>" + crlf + crlf;
HTML = HTML + "% Botellas en mal estado: "+ porcentaje1 + "<br>" + crlf + crlf;
HTML = HTML + "<br>";
HTML = HTML + "<br>";
HTML = HTML + "REGISTROS DE LA PRODUCCION ANTERIOR<br>";
HTML = HTML + "# Jabas ingresadas: "+ RegisterReadInteger(1028) + "<br>";
HTML = HTML + "Total de Jabas Correctas : " + RegisterReadInteger(1032) + "<br>";
HTML = HTML + "Total de Jabas Mal: " + RegisterReadInteger(1036) + "<br>"+ crlf + crlf;
HTML = HTML + "# Botellas en buen estado: "+ RegisterReadInteger(1020) + "<br>";
HTML = HTML + "# Botellas en mal estado: "+ RegisterReadInteger(1024) + "<br>" + crlf + crlf;
```

```
                    HTML = HTML + "% Botellas en mal estado:  "+ porcentaje2 + "<br>" + crlf
+ crlf;
    }
                    HTML = HTML + "</font><br><br>";
                    HTML = HTML + "<hr><div align='center'><a
href='http://10.101.31.181'><img src='" + LogoCam2 + "' alt='Enlace con Camara 2'
border=1></a></div></body></html>" ;
                    HTML = HTML + "<hr><div align='center'><a
href='http://www.cerveceriaandina.com/'><img src='" + LogoCASA + "' alt='Website de
CERVECERIA ANDINA' border=1></a></div></body></html>"+ crlf + crlf ;
                    HTML = HTML + "<hr><div align='center'><a
href='http://www.dvtsensors.com/'><img src='" + LogoDVT + "' alt='DVT Corporation Website'
border=1></a></div></body></html>"+ crlf + crlf ;



                    String HTMLRetorno = "GET / HTTP/1.1" + crlf;


                    do
                    {
                            Resultado2 = CSock.Recv(in);
                            Respuesta = Respuesta + String(in);
                    } while (in[0] != 10);

                    if (Respuesta.compareTo(HTMLRetorno) == 0)
                    {
                            Resultado1 = CSock.Send(HTML.toByteArray());
                    }

                    else
                    {
                            Resultado1 = CSock.Send(HTMLFalla.toByteArray());
                    }



                    CSock.Close();
                    Sock.Close();
            }


    }
}
```

- Script de la Tabla de Recalibración

```
class TablaCalibracionSB1Cam1
{
   public static void main()
   {
     while(true)
            {
            int     datasb1;
            short   positionsb1;
            short   registersb1;

//Para el sensor sb1
```

```
        positionsb1             = 0;
    for (positionsb1 = 0;positionsb1 <= 10;positionsb1 += 1)
        {
                datasb1                 =       2500;
                registersb1     =       positionsb1;
                registersb1     =       (registersb1 * 4) + 2000;
                RegisterWriteInteger(registersb1,datasb1);
        //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
    }
        for (positionsb1 = 11;positionsb1 <= 14;positionsb1 += 1)
        {
                datasb1                 =       3000;
                registersb1     =       positionsb1;
                registersb1     =       (registersb1 * 4) + 2000;
                RegisterWriteInteger(registersb1,datasb1);
        //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
    }

        for (positionsb1 = 15;positionsb1 <= 43;positionsb1 += 1)
        {
                datasb1                 =       4000;
                registersb1     =       positionsb1;
                registersb1     =       (registersb1 * 4) + 2000;
                RegisterWriteInteger(registersb1,datasb1);
        //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
    }

        for (positionsb1 = 44;positionsb1 <= 57;positionsb1 += 1)
        {
                datasb1                 =       4500;
                registersb1     =       positionsb1;
                registersb1     =       (registersb1 * 4) + 2000;
                RegisterWriteInteger(registersb1,datasb1);
        //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
    }

        for (positionsb1 = 58;positionsb1 <= 65;positionsb1 += 1)
        {
                datasb1                 =       5500;
                registersb1     =       positionsb1;
                registersb1     =       (registersb1 * 4) + 2000;
                RegisterWriteInteger(registersb1,datasb1);
        //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
    }

        for (positionsb1 = 66;positionsb1 <= 76;positionsb1 += 1)
        {
                datasb1                 =       6000;
                registersb1     =       positionsb1;
                registersb1     =       (registersb1 * 4) + 2000;
                RegisterWriteInteger(registersb1,datasb1);
        //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
    }
```

```
            for (positionsb1 = 77;positionsb1 <= 92;positionsb1 += 1)
            {
                    datasb1                 =       7000;
                    registersb1     =       positionsb1;
                    registersb1     =       (registersb1 * 4) + 2000;
                    RegisterWriteInteger(registersb1,datasb1);
            //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
      }

            for (positionsb1 = 93;positionsb1 <= 100;positionsb1 += 1)
            {
                    datasb1                 =       8000;
                    registersb1     =       positionsb1;
                    registersb1     =       (registersb1 * 4) + 2000;
                    RegisterWriteInteger(registersb1,datasb1);
            //      DebugPrint      ("registersb1:                  "+registersb1+"         valor:
"+RegisterReadInteger(registersb1));
      }


            DebugPrint("espero");
            sleep(7200000);
            DebugPrint("paso");
            }

   }
}
```

- Script de Asignación de Threshold a sensores de Manchas

```
class ControlThresholdCam1
{
   public void inspect()
   {

            Sensor intensidad;
            Sensor inspector;
            //byte   ejecucion;
            //ejecucion       = RegisterReadByte(500);
            //DebugPrint ("permi:  "+ejecucion);
            //if (ejecucion == 255)
            //{
                    String  comando;
                    int             valor;
                    double  brillo;
                    float   pbrillo;


            // brillo en cb1
                    brillo          = 10000 - cb1.Percent;
                    pbrillo         = brillo / 100;
                    valor           = pbrillo ;
                    valor           = (valor * 4) + 2000;
                    comando         = "251 0 "+RegisterReadInteger(valor);
                    inspector       = GetSensorByName("sb1");
                    inspector.SetParams(comando);
```

```
        //DebugPrint ("respuesta:  "+n1);
        //DebugPrint ("cargo:  "+cargo);
        //DebugPrint ("brillo:  "+brillo);
        //DebugPrint ("pbrillo:  "+pbrillo);
        //DebugPrint ("valor:  "+valor);
        //DebugPrint ("comando:  "+comando);


// brillo en cb2
        brillo          = 10000 - cb2.Percent;
        pbrillo         = brillo / 100;
        valor           = pbrillo ;
        valor           = (valor * 4) + 2404;
        comando         = "251 0 "+RegisterReadInteger(valor);
        inspector       = GetSensorByName("sb2");
        inspector.SetParams(comando);

// brillo en cb3
        brillo          = 10000 - cb3.Percent;
        pbrillo         = brillo / 100;
        valor           = pbrillo ;
        valor           = (valor * 4) + 2808;
        comando         = "251 0 "+RegisterReadInteger(valor);
        inspector       = GetSensorByName("sb3");
        inspector.SetParams(comando);

// brillo en cb4
        brillo          = 10000 - cb4.Percent;
        pbrillo         = brillo / 100;
        valor           = pbrillo ;
        valor           = (valor * 4) + 3212;
        comando         = "251 0 "+RegisterReadInteger(valor);
        inspector       = GetSensorByName("sb4");
        inspector.SetParams(comando);

// brillo en cb5
        brillo          = 10000 - cb5.Percent;
        pbrillo         = brillo / 100;
        valor           = pbrillo ;
        valor           = (valor * 4) + 3616;
        comando         = "251 0 "+RegisterReadInteger(valor);
        inspector       = GetSensorByName("sb5");
        inspector.SetParams(comando);

// brillo en cb6
        brillo          = 10000 - cb6.Percent;
        pbrillo         = brillo / 100;
        valor           = pbrillo ;
        valor           = (valor * 4) + 4020;
        comando         = "251 0 "+RegisterReadInteger(valor);
        inspector       = GetSensorByName("sb6");
        inspector.SetParams(comando);

// brillo en cb7
        brillo          = 10000 - cb7.Percent;
        pbrillo         = brillo / 100;
        valor           = pbrillo ;
        valor           = (valor * 4) + 4424;
        comando         = "251 0 "+RegisterReadInteger(valor);
```

```
            inspector       = GetSensorByName("sb7");
            inspector.SetParams(comando);

        // brillo en cb8
            brillo          = 10000 - cb8.Percent;
            pbrillo         = brillo / 100;
            valor           = pbrillo ;
            valor           = (valor * 4) + 4828;
            comando         = "251 0 "+RegisterReadInteger(valor);
            inspector       = GetSensorByName("sb8");
            inspector.SetParams(comando);

        // brillo en cb9
            brillo          = 10000 - cb9.Percent;
            pbrillo         = brillo / 100;
            valor           = pbrillo ;
            valor           = (valor * 4) + 5232;
            comando         = "251 0 "+RegisterReadInteger(valor);
            inspector       = GetSensorByName("sb9");
            inspector.SetParams(comando);

        // brillo en cb10
            brillo          = 10000 - cb10.Percent;
            pbrillo         = brillo / 100;
            valor           = pbrillo ;
            valor           = (valor * 4) + 5636;
            comando         = "251 0 "+RegisterReadInteger(valor);
            inspector       = GetSensorByName("sb10");
            inspector.SetParams(comando);

        // brillo en cb11
            brillo          = 10000 - cb11.Percent;
            pbrillo         = brillo / 100;
            valor           = pbrillo ;
            valor           = (valor * 4) + 6040;
            comando         = "251 0 "+RegisterReadInteger(valor);
            inspector       = GetSensorByName("sb11");
            inspector.SetParams(comando);

        // brillo en cb12
            brillo          = 10000 - cb12.Percent;
            pbrillo         = brillo / 100;
            valor           = pbrillo ;
            valor           = (valor * 4) + 6444;
            comando         = "251 0 "+RegisterReadInteger(valor);
            inspector       = GetSensorByName("sb12");
            inspector.SetParams(comando);

        //}
    }
}
```

- Script de Análisis de Resultados y Almacenamiento de Registros

```
class Analisis
{
    public void inspect()
    {
```

```
//Variables para la transportadora 1
        byte    jaba    = 0;        //examina si las jabas vinieron con todas las botellas
        int     jabaok1 = 0;        //contador de jabas ok
        int     jababad1 = 0;       //contador jaba mal
        int             jabatot1 = 0;       //contador total jabas
        int     bmal1   = 0;        //contador de botellas malas transp1
        int     bbien1  = 0;        //contador de botellas buenas transp1
        Sensor SV;

//Determino si detecto una jaba
    SV      = GetSensorByName("J2");
    if (J2 != null)
    {
    if ((J1.Result==PASS)&&(J2.Result==PASS)&&(DetectorJaba.Result==PASS))
    {
//Inicio de adquisicion de datos de sensor 1

        SV      = GetSensorByName("sb1");

        if (SV != null)
        {
                if ((sb1.Result==PASS)||(A1.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

        }

//Inicio de adquisicion de datos de sensor 2

        SV      = GetSensorByName("sb2");

        if (SV != null)
        {
                if ((sb2.Result==PASS)||(A2.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
```

```
                }

            }

//Inicio de adquisicion de datos de sensor 3

            SV      = GetSensorByName("sb3");

            if (SV != null)
            {
                if ((sb3.Result==PASS)||(A3.Result==PASS))
                {
                    bbien1 = RegisterReadInteger(1000);
                    bbien1 = bbien1 + 1     ;
                    RegisterWriteInteger(1000,bbien1);
                    jaba   = jaba + 1;
                    //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                    bmal1 = RegisterReadInteger(1004);
                    bmal1 = bmal1 + 1;
                    RegisterWriteInteger(1004,bmal1);
                    //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

            }

//Inicio de adquisicion de datos de sensor 4

            SV      = GetSensorByName("sb4");

            if (SV != null)
            {
                if ((sb4.Result==PASS)||(A4.Result==PASS))
                {
                    bbien1 = RegisterReadInteger(1000);
                    bbien1 = bbien1 + 1     ;
                    RegisterWriteInteger(1000,bbien1);
                    jaba   = jaba + 1;
                    //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                    bmal1 = RegisterReadInteger(1004);
                    bmal1 = bmal1 + 1;
                    RegisterWriteInteger(1004,bmal1);
                    //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

            }

//Inicio de adquisicion de datos de sensor 5

            SV      = GetSensorByName("sb5");

            if (SV != null)
            {
                if ((sb5.Result==PASS)||(A5.Result==PASS))
```

```
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

        }

//Inicio de adquisicion de datos de sensor 6

        SV      = GetSensorByName("sb6");

        if (SV != null)
        {
                if ((sb6.Result==PASS)||(A6.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

        }

//Inicio de adquisicion de datos de sensor 7

        SV      = GetSensorByName("sb7");

        if (SV != null)
        {
                if ((sb7.Result==PASS)||(A7.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
```

```
                RegisterWriteInteger(1004,bmal1);
                //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
            }

        }

//Inicio de adquisicion de datos de sensor 8

        SV      = GetSensorByName("sb8");

        if (SV != null)
        {
            if ((sb8.Result==PASS)||(A8.Result==PASS))
            {
                bbien1 = RegisterReadInteger(1000);
                bbien1 = bbien1 + 1     ;
                RegisterWriteInteger(1000,bbien1);
                jaba   = jaba + 1;
                //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
            }
            else
            {
                bmal1 = RegisterReadInteger(1004);
                bmal1 = bmal1 + 1;
                RegisterWriteInteger(1004,bmal1);
                //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
            }

        }

//Inicio de adquisicion de datos de sensor 9

        SV      = GetSensorByName("sb9");

        if (SV != null)
        {
            if ((sb9.Result==PASS)||(A9.Result==PASS))
            {
                bbien1 = RegisterReadInteger(1000);
                bbien1 = bbien1 + 1     ;
                RegisterWriteInteger(1000,bbien1);
                jaba   = jaba + 1;
                //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
            }
            else
            {
                bmal1 = RegisterReadInteger(1004);
                bmal1 = bmal1 + 1;
                RegisterWriteInteger(1004,bmal1);
                //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
            }

        }

//Inicio de adquisicion de datos de sensor 10

        SV      = GetSensorByName("sb10");

        if (SV != null)
```

```
        {
                if ((sb10.Result==PASS)||(A10.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

        }

//Inicio de adquisicion de datos de sensor 11

        SV      = GetSensorByName("sb11");

        if (SV != null)
        {
                if ((sb11.Result==PASS)||(A11.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

        }

//Inicio de adquisicion de datos de sensor 12

        SV      = GetSensorByName("sb12");

        if (SV != null)
        {
                if ((sb12.Result==PASS)||(A12.Result==PASS))
                {
                        bbien1 = RegisterReadInteger(1000);
                        bbien1 = bbien1 + 1     ;
                        RegisterWriteInteger(1000,bbien1);
                        jaba   = jaba + 1;
                        //DebugPrint("botellasbien    "+RegisterReadInteger(1000));
                }
                else
                {
```

```
                        bmal1 = RegisterReadInteger(1004);
                        bmal1 = bmal1 + 1;
                        RegisterWriteInteger(1004,bmal1);
                        //DebugPrint("botellasmal    "+RegisterReadInteger(1004));
                }

        }

        if ( jaba == 12)
        {
                Analisis.Result = PASS;
                jabaok1= RegisterReadInteger(1012);
                jabaok1 += 1;
                RegisterWriteInteger(1012,jabaok1);

        }
        else
        {
                Analisis.Result = FAIL;
                jababad1        = RegisterReadInteger(1016);
                jababad1 += 1;
                RegisterWriteInteger(1016,jababad1);
        }

        jaba = 0;
        jabatot1 = RegisterReadInteger(1012) + RegisterReadInteger(1016);
        RegisterWriteInteger(1008,jabatot1);
    }
    else
    {
        Analisis.Result = PASS;
    }

    }
  }
}
```