



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

“Aplicación de Visión con LabVIEW para la Detección de Frascos sin  
Etiqueta”

### **TESIS DE GRADO**

Previa la obtención del Título de:

### **INGENIERO EN ELECTRICIDAD**

Presentado por:

José Gonzalo Vera Isaza

Especialización: Electrónica y Automatización Industrial

Wilson Gabriel Veliz Plua

Especialización: Electrónica y Telecomunicaciones

GUAYAQUIL – ECUADOR

AÑO 2008

# AGRADECIMIENTO

A Dios por su gran amor, a mi familia por su apoyo incondicional, y a todos mis amigos que de alguna u otra forma colaboraron para la culminación de este trabajo.

Wilson Gabriel Veliz Plua.

Agradeciendo al Señor todo poderoso por darme la oportunidad de vivir, de crecer junto a unos padres responsables, amables, cariñosos que luchan por brindarnos un mejor futuro, por darnos un apoyo incondicional, por enseñarnos el deseo de superación, por

el sacrificio que tuvieron que pasar para ver los frutos de nuestro esfuerzo.

Ellos, que ahora se sienten más orgullosos por lograr uno de los sueños de padres: el ver a sus hijos logrando alcanzar el título profesional, que es el primer paso para crecer en el mundo laboral.

Con este logro que alcanzo, espero ser de motivación para mis hermanos y hermanas que también luchan por cumplir sus metas que de pequeños nos propusimos; A Mayi y Kathy les agradezco por soportarme, por comprenderme; a Jorge y Abel les enseño que si uno se propone una meta con esfuerzo y paciencia, lo consigue.

A Mariela, por ser mí mejor amiga, por estar siempre en las buenas y en las

malas, por brindarme tu amor, tus consejos, porque de ti aprendí a valorar más la vida. Además, te agradezco por aportar con un granito de arena en esta tesis.

A mi abuelita Paquita, mi tío Armando y mi primo Michael por aportar en mi desarrollo personal.

Al Ing. Carlos Valdivieso, por su incondicional ayuda y consejos para lograr terminar este proyecto.

Gracias a todos aquellos que no he nombrado pero que saben que están dentro de mi corazón y que siempre lo estarán.

Gonzalo Vera Isaza

# DEDICATORIA

A Dios por ser nuestro padre y por guiarnos en la vida, a mi mami Carolina Plua García, mi papi Wilson Veliz Olvera, mis hermanos y tías, primas y primos, que con su ayuda y demostración de afecto han contribuido a formar una excelente persona y cada día llenan de felicidad mi corazón.

Wilson Gabriel Veliz Plua

Dedico esta tesis a mis padres, hermanos y a Mariela por su apoyo sin límites.

Gonzalo Vera Isaza.

# TRIBUNAL DE GRADUACIÓN



---

Ing. Holguer Cevallos  
Presidente del Tribunal



---

Ing. Carlos Valdivieso  
Director de Tesis



---

Ing. Maria Antonieta  
Álvarez  
Miembro Principal



---

Ing. César Martín Moreno  
Miembro Principal

# DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este trabajo, me corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)



---

Wilson Gabriel Veliz Plua



---

José Gonzalo Vera Isaza

# RESUMEN

El proyecto de tesis de grado tiene como finalidad efectuar el desarrollo e implementación de un programa realizado en LABVIEW 8.2 que permite la detección de frascos sin etiquetas usando el paquete IMAQ Visión de LABVIEW, la aplicación va a utilizar las técnicas de procesamiento de imágenes por lo cual se va a hacer uso de un periférico como lo es una cámara USB para adquirir las imágenes de los frascos y se va a realizar una comparación con una imagen cargada con anterioridad en LabVIEW, luego se enviará una señal al brazo robótico y al empuja frascos comandados por servo motores que están acoplados a una plataforma implementada con bandas transportadoras que van a llevar botellas, tanto el brazo robótico como el empuja frascos estarán ubicados en un punto estratégico para realizar la acción de dejar pasar la botella (si cumple con los parámetros de desarrollo) ó rechazar aquellas botellas que no cumplan con los requisitos establecidos en el programa y serán enviadas a una zona de eliminación.



La banda transportadora y el brazo robótico fueron desarrollados en el tópico **“Plataforma para el desarrollo de proyectos con dsPICs y visión Robótica con LabVIEW”**.

Otro punto a tratar en la realización del proyecto de tesis es la aplicación de los conocimientos adquiridos en las materias del tópico y de lo aprendido en el transcurso de la Universidad para ponerlos en práctica y de dar a conocer las ventajas de trabajar con LabVIEW para la adquisición de imágenes por medio de computadores.

# INDICE GENERAL

RESUMEN.....	VIII
INDICE GENERAL.....	X
ABREVIATURAS.....	XX
INDICE DE IMÁGENES.....	XXIII
INDICE DE TABLAS.....	XXXI
INTRODUCCIÓN .....	1
1. Especificación del tema .....	4
1.1 Descripción del problema .....	4
1.2 Antecedentes, causales y efectos del tema planteado.....	5
1.2.1 Antecedentes .....	5
1.2.2 Causales .....	9

1.2.3	Efectos .....	11
1.3	Alcance del proyecto .....	12
1.4	Planteamiento de solución.....	13
2.	Visión por computadora.....	15
2.1	Introducción .....	15
2.1.1	Imágenes.....	15
2.1.2	Imágenes digitales.....	17
2.1.3	Definición de visión por computadora.....	19
2.1.4	Descripción de la visión por computadora.....	19
2.1.5	Comparación entre el sistema de visión humano y el sistema de visión artificial .....	21
2.1.5.1	Sistema de visión humano.....	21
2.1.5.2	Ventajas de la visión humana .....	22
2.1.5.3	Ventajas de la visión artificial.....	23
2.1.6	Esquema general de un sistema de visión por computadora ..	24
2.1.6.1	Escena 3D .....	24

2.1.6.2	Iluminación.....	25
2.1.6.3	Adquisición de imagen.....	26
2.1.6.4	Digitalizador .....	26
2.1.6.5	Preprocesamiento.....	27
2.1.6.6	Segmentación.....	27
2.1.6.7	Descripción .....	28
2.1.6.8	Reconocimiento e interpretación.....	28
2.2	Procesamiento digital de imágenes.....	29
2.2.1	Transformaciones de imágenes. ....	30
2.2.2	Operaciones Puntuales .....	30
2.2.2.1	Operaciones locales .....	33
2.2.2.2	Operaciones globales .....	33
2.2.2.3	Operaciones geométricas .....	34
2.2.3	Detección de bordes.....	35
2.2.4	Segmentación .....	38
2.2.4.1	Umbralización .....	39

2.3	Aplicaciones de visión artificial .....	40
2.3.1	Aplicaciones en Medicina .....	41
2.3.2	Aplicaciones a nivel de Identificación .....	42
2.3.3	Aplicaciones en el área militar.....	43
2.3.4	Aplicaciones en la robótica.....	44
2.3.5	Aplicaciones en el control de calidad en las industrias.....	45
2.3.6	Aplicación en industrias.....	47
2.3.7	Aplicación en otros campos.....	48
2.4	Software utilizado para la adquisición de imágenes .....	50
2.4.1	Requerimientos del software .....	50
2.4.1.1	Selección de la cámara.....	50
2.4.1.2	Flexibilidad del hardware .....	51
2.4.1.3	Precisión y rendimiento del algoritmo .....	51
2.4.1.4	Facilidad al utilizar .....	52
2.4.1.5	Nivel de integración con otros dispositivos .....	52
2.4.1.6	Costo del software .....	53

2.4.2	Tipos de software .....	53
2.4.2.1	Pyvison .....	53
2.4.2.2	I+Solex.....	54
2.4.2.3	MvImpact .....	55
2.4.2.4	OpenCV .....	56
2.4.2.5	LabVIEW.....	56
2.5	Hardware implementados.....	57
2.5.1	LightPix AE20.....	58
2.5.2	Iris access 3000.....	59
2.5.3	SIMATIC VS100 .....	60
3.	Plataforma de desarrollo con dsPICS.....	61
3.1	Programación en LabVIEW .....	61
3.1.1	Introducción.....	61
3.1.2	Área de aplicación.....	63
3.1.3	Entorno de LabVIEW.....	64
3.1.3.1	Panel Frontal.....	65

3.1.3.2	Diagrama de Bloques .....	65
3.1.3.3	Paletas .....	67
3.2	Manejo de los componentes de la librería IMAQ VISION .....	69
3.2.1	IMAQ USB Init: .....	71
3.2.2	IMAQ USB Snap.....	71
3.2.3	IMAQ USB Grab Setup.....	71
3.2.4	IMAQ USB Grab Acquire.....	72
3.2.5	IMAQ USB Close.....	72
3.3	Captura de imágenes mediante LabVIEW.....	73
3.4	Protocolos de comunicación con periféricos.....	76
3.4.1	Comunicación serial .....	76
3.4.1.1	RS-232.....	77
3.4.1.2	USB .....	78
3.4.1.3	Periféricos .....	80
3.5	Interfaces de manejo y control.....	81
3.6	Características generales de los dsPICS .....	83

3.7	Arquitectura, funcionamiento e interfaces.....	85
3.7.1	Arquitectura Harvard .....	85
3.7.2	Motor DSP.....	86
3.7.3	Memoria y contador del programa.....	87
3.7.4	Memoria de datos.....	88
3.7.5	Registro de trabajo .....	88
3.7.6	Grupo de instrucciones.....	89
3.7.7	Interrupciones.....	90
3.7.8	Periféricos .....	90
3.8	Aplicaciones prácticas de los dsPICs .....	91
4.	Diseño e implementación .....	95
4.1	Requerimiento de implementación .....	96
4.1.1	Requerimientos del computador.....	96
4.1.2	Requerimientos del Software .....	96
4.1.3	Requerimientos de la cámara.....	97
4.1.4	Requerimientos de iluminación .....	98



4.2	Consideraciones de diseño .....	99
4.3	Diagrama de flujo .....	101
4.4	Código realizado en LabVIEW para captura de imágenes .....	108
4.4.1	SubVIs creados .....	108
4.4.1.1	Inicio_sesion.vi.....	108
4.4.1.2	Ruta.vi.....	109
4.4.1.3	Detecta_Botella.vi.....	109
4.4.1.4	Punto_interés.vi.....	110
4.4.1.5	Función_bordes.vi.....	110
4.4.1.6	Girar_Botella_detecta_etiqueta.vi.....	111
4.4.1.7	Detecta_etiqueta.vi.....	112
4.4.1.8	Guardar_Imagen.vi.....	112
4.4.1.9	Área_de_análisis.vi.....	113
4.4.2	Descripción de código .....	113
4.5	Implementación física .....	124
4.5.1	Sistema de iluminación:.....	125

4.5.2	Cámara de análisis:.....	126
4.6	Control del brazo robótico.....	127
4.6.1	Brazo.vi .....	128
4.6.2	Empuja_Frascos.vi.....	130
4.7	Análisis de costos.....	131
5.	Datos experimentales.....	133
5.1	Pruebas realizadas.....	133
5.1.1	Análisis de frascos.....	134
5.1.2	Análisis de frascos acoplado a la banda transportadora y el brazo robótico.....	135
5.1.2.1	Detección de la botella utilizando la cámara como sensor	136
5.1.2.2	Detección del frasco utilizando un sensor reflector y el empuja frasco.....	137
5.2	Datos obtenidos.....	142
5.3	Análisis de resultados.....	148
5.3.1	Comparación manual .....	149

5.3.2	Comparación obtenida de LabVIEW .....	153
-------	---------------------------------------	-----

CONCLUSIONES Y RECOMENDACIONES

ANEXOS

BIBLIOGRAFÍA

# ABREVIATURAS

<b>BAUDIOS</b>	Bits por segundo
<b>BIT</b>	Unidad de medida de información equivalente a la elección entre dos posibilidades igualmente probables
<b>BMP</b>	Mapa de bit.
<b>CMOS</b>	Semiconductor complementario de oxido metálico
<b>CRT</b>	Tubo de rayos catódicos
<b>DSC</b>	Controlador Digital de Señales.
<b>DSP</b>	Procesador digital de señales
<b>dsPIC</b>	Microcontrolador de Microchip con soporte para procesamiento de señales
<b>EEPROM</b>	Memoria ROM programable, Regrabable

<b>E/S</b>	Entradas y Salidas
<b>FPGA</b>	Arreglos de puertas programables por campos
<b>FPS</b>	Fotos por segundo
<b>HDI</b>	Interfaz Dispositivo-Hombre
<b>IVT</b>	Tabla de vector de interrupción
<b>LavBIEW</b>	Laboratory Virtual Instrument Engineering Wordbench
<b>LCD</b>	Pantalla de cristal liquido
<b>Mbps</b>	Mega bits por segundo
<b>MDF</b>	Material parecido a la madera
<b>MCU</b>	Microcontrolador
<b>NI</b>	National Instrument
<b>NI-IMAQ</b>	Software controlador para dispositivos de adquisición de imágenes de National Instrument
<b>PC</b>	Computadora personal
<b>PWM</b>	Modulación por ancho de pulso

<b>RAM</b>	Memoria de acceso aleatorio
<b>ROI</b>	Región de interés
<b>RS-232</b>	Protocolo de comunicación serial
<b>SFR</b>	Registro de funciones especiales
<b>SubVI</b>	Subrutinas en LabVIEW
<b>USART</b>	Universal Synchronous Asynchronous receiver Transmitter
<b>USB</b>	Bus serial universal
<b>VI</b>	Instrumento virtual
<b>Watt</b>	Unidad de medida de Potencia (Vatio)

# INDICE DE IMÁGENES

Figura 1.1 Proceso parcialmente automatizado.....	6
Figura 1.2 Colocación manual de etiqueta.....	8
Figura 1.3 Sensor infrarrojo de proximidad.....	8
Figura 1.4 Maquina etiquetadora para botellas.....	9
Figura 1.5 Inspección visual de etiquetas en los frascos .....	10
Figura 1.6 Frascos con etiqueta.....	11
Figura 2.1 Representación del espacio de un objeto y del espacio de una imagen [Adaptación de Pearson 1991].....	17
Figura 2.2 Representación matricial de una imagen.....	17
Figura 2.3 Imagen a color y píxel.....	18
Figura 2.4 Esquema general de un sistema de visión por computador .....	24

Figura 2.5 Diferentes tipos de luz .....	25
Figura 2.6 Cámaras analógicas Panasonic .....	26
Figura 2.7 Componentes básicos del procesamiento de imágenes [Green, 1989].....	29
Figura 2.8 Operación resta .....	31
Figura 2.9 Histograma de imagen.....	32
Figura 2.10 Perfil de línea.....	32
Figura 2.11 Rotación de una imagen.....	34
Figura 2.12 Detección de bordes .....	36
Figura 2.13 Método de primera y segunda derivada.....	37
Figura 2.14 Detección de bordes a) Operador Gradiente,b) Operador Laplaciano b)Prewit, c) Sobel d) Canny.....	38
Figura 2.15 Ecografía normal (derecha) y la ecografía adquirida en visión por computadora (izquierda) .....	41
Figura 2.16 Muestra sanguínea .....	42
Figura 2.17 Reconocimiento de iris.....	43
Figura 2.18 Análisis de terreno .....	44



Figura 2.19 Sistema de visión estéreo montado sobre un robot móvil.....	45
Figura 2.20 Control de calidad con visión artificial .....	47
Figura 2.21 Control de un brazo robótico.....	48
Figura 2.22 Matrículas de vehículos .....	49
Figura 2.23 Recoge objetos.....	49
Figura 2.24 Ventana del software I+Solex .....	55
Figura 2.25 Software MvImpact .....	55
Figura 2.26 Funciones del LightPix AE20 .....	59
Figura 2.27 Iris Access 3000.....	60
Figura 3.1 Panel Frontal y Diagrama de Bloques .....	64
Figura 3.2 Estandarización de colores para los diferentes tipos de datos ....	67
Figura 3.3 Paleta de herramientas.....	67
Figura 3.4 Paleta de controles .....	68
Figura 3.5 Paleta de funciones .....	69
Figura 3.6 Pasos para seleccionar NI-IMAQ USB .....	70
Figura 3.7 Funciones de NI-IMAQ USB .....	70

Figura 3.8 Icono de IMAQ USB Init.....	71
Figura 3.9 Icono de IMAQ USB Snap .....	71
Figura 3.10 Icono de IMAQ USB Grab Setup .....	72
Figura 3.11 Icono de IMAQ USB Grab Acquire.....	72
Figura 3.12 Icono de IMAQ USB Close .....	72
Figura 3.13 Identificación de la cámara .....	73
Figura 3.14 Diagrama de bloques de adquisición.....	74
Figura 3.15 Programa para adquirir imágenes .....	75
Figura 3.16 Puerto serie RS-232 de 9 y 25 pines macho. ....	78
Figura 3.17 Símbolo USB .....	79
Figura 3.18 Conectores USB, de izquierda a derecha: USB1.1 y USB2.0 ...	80
Figura 3.19 Periféricos USB conectados a la PC.....	81
Figura 3.20 Rendimiento Vs. Precio entre los DSC y microcontroladores ....	84
Figura 3.21 Memoria de programa y datos .....	85
Figura 3.22 Registro de trabajo w.....	89
Figura 3.23 Sistema de desarrollo PIC school.....	92

Figura 3.24 Imagen del robot móvil.....	94
Figura 4.1 Formato de la etiqueta.....	100
Figura 4.2 Formato de la Botella.....	100
Figura 4.3 Diagrama de bloques de la aplicación.....	108
Figura 4.4 Diagrama de flujo parte 1.....	103
Figura 4.5 Diagrama de flujo parte 2.....	104
Figura 4.6 Panel frontal.....	105
Figura 4.7 Icono de Inicio sesión.....	108
Figura 4.8 Icono de ruta.....	109
Figura 4.9 Icono detecta_botella.....	109
Figura 4.10 Icono de punto_interés.....	110
Figura 4.11 Icono función_bordes.....	110
Figura 4.12 Icono giro_botella_detecta_etiqueta.....	111
Figura 4.13 Icono detecta_etiqueta.....	112
Figura 4.14 Icono guardar_imagen.....	112
Figura 4.15 Icono área_de_análisis.....	113

Figura 4.16 Código de calibrar cámara.....	114
Figura 4.17 Código de adquirir plantilla .....	114
Figura 4.18 Inicialización de variables .....	115
Figura 4.19 Código de configuración de puertos series.....	116
Figura 4.20 Configuración del movimiento de banda.....	116
Figura 4.21 Posicionamiento del empuja frasco .....	117
Figura 4.22 Reporte del sensor de tapa.....	118
Figura 4.23 Tiempo de espera .....	119
Figura 4.24 Detener banda transportadora.....	119
Figura 4.25 Modo de captura continua .....	120
Figura 4.26 Búsqueda de la plantilla.....	120
Figura 4.27 Agrupación de los puntos de interés.....	121
Figura 4.28 Ubicación del brazo robótico.....	122
Figura 4.29 Análisis de etiqueta.....	122
Figura 4.30 Elimina botella del sistema .....	123
Figura 4.31 Ubicación de brazo en la posición inicial .....	123

Figura 4.32 Movimiento de la banda transportadora.....	124
Figura 4.33 Se cierra la configuración de los puertos utilizados .....	124
Figura 4.34 Vista del sistema de iluminación .....	125
Figura 4.35 Vista exterior de la cámara de análisis .....	126
Figura 4.36 Interior de la cámara de análisis .....	127
Figura 4.37 Icono del brazo robótico.....	128
Figura 4.38 posición de los motores servo .....	129
Figura 4.39 Icono del empuja frascos .....	130
Figura 5.1 Mesa de trabajo .....	135
Figura 5.2 Imágenes donde se aprecia el desfase en la detención de los frascos. ....	137
Figura 5.3 Visualización del sensor y de la barrera de detención .....	138
Figura 5.4 Imágenes de los frascos utilizados para el análisis de etiqueta 1,2,3.....	140
Figura 5.5 Imágenes de los frascos utilizados para el análisis de etiqueta 4,5,6.....	140
Figura 5.6 Imagen del frasco utilizado para el análisis de etiqueta 7.....	141

Figura 5.7 Plantilla para seccionar el área de análisis .....	142
Figura 5.8 Plantilla de la etiqueta.....	142
Figura 5.9 Imagen frasco correcto .....	149
Figura 5.10 Etiqueta con otra coloración .....	150
Figura 5.11 Etiqueta Incorrecta.....	150
Figura 5.12 Etiqueta desplazada hacia abajo .....	151
Figura 5.13 Etiqueta rayada.....	152
Figura 5.14 Etiqueta rayada.....	152
Figura 5.15 Frasco sin etiqueta.....	153
Figura 5.16 Frasco con etiqueta correcta .....	154
Figura 5.17 Etiqueta con otra coloración .....	155
Figura 5.18 Etiqueta incorrecta .....	155
Figura 5.19 Etiqueta desplazada .....	156
Figura 5.20 Etiquetas rayadas .....	157
Figura 5.21 Frasco sin etiqueta.....	157

# INDICE DE TABLAS

Tabla 1 Costos de los equipos y materiales .....	131
Tabla 2 Identificación numérica de los frascos analizados .....	141
Tabla 3 Imágenes adquiridas y área de análisis de pruebas realizadas....	146
Tabla 4 Prueba realizada 1 .....	147
Tabla 5 Prueba realizada 2.....	147

# INTRODUCCIÓN

Para el desarrollo de la APLICACIÓN DE VISIÓN CON LABVIEW PARA LA DETECCIÓN DE FRASCOS SIN ETIQUETA se ha escogido cumplir con los siguientes objetivos:

## OBJETIVOS GENERALES

- Desarrollar en LabVIEW un programa que permita la detección de frascos sin etiquetas usando el paquete IMAQ Visión de LabVIEW
- Acoplar el programa a la plataforma empleada tomando en cuenta todas sus características
- Permitir la realización de cambios rápidos o modulares en la programación que se acoplen a nuevas posiciones de trabajo
- Armonizar, documentar y sistematizar el uso del programa desarrollado para permitir su utilización como una opción modular de la plataforma empleada.



## OBJETIVOS ESPECÍFICOS

- Aprender a manipular la herramienta de programación como LabVIEW y su paquete IMAQ Vision.
- Comprender los principios básicos de visión artificial..
- Comprender y manejar las interfaces de control entre: LabVIEW - brazo robótico y LabVIEW - banda transportadora.

El desarrollo de este documento está comprendido en cinco capítulos, acerca de los cuales se va a dar una breve introducción a continuación:

- En el capítulo I se identifica y explica una descripción del problema a tratar. Se presentan los antecedentes, causales y efectos del problema. Por último en este capítulo se detalla el alcance que tendrá la solución planteada.
- En el capítulo II se ofrece una introducción de lo que es visión por computadora. Se explica brevemente el procesamiento de imágenes digitales, también se expone aplicaciones de la visión artificial y tipos de softwares utilizados para la adquisición de la misma y además se trata sobre los hardwares implementados.
- En el capítulo III se explica en forma general la plataforma de desarrollo que se va a utilizar para la elaboración del presente

proyecto. Se describen los componentes y librerías. La forma de capturar imágenes. Los protocolos de comunicación y periféricos. Se especifican además las características generales de los dsPICs, su arquitectura, funcionamiento e interfaces. Al final de este capítulo se relatan las aplicaciones prácticas de dichos dispositivos.

- En el capítulo IV se realiza el diseño e implementación del proyecto de tesis para lo cual se explican sus requerimientos y consideraciones presentando el flujo del código realizado en LabVIEW para la captura de imágenes proporcionando una explicación detallada de su funcionamiento. Además la aplicación se implementará en la plataforma de bandas transportadoras con un brazo robótico. Como último paso se detalla un análisis de los costos relacionados con la ejecución del proyecto.
- En el capítulo V comprende la recopilación de datos de las pruebas realizadas para posteriormente realizar su análisis y verificar que el sistema desarrollado esté funcionando debidamente

# **CAPÍTULO 1**

## **1. ESPECIFICACIÓN DEL TEMA**

El presente capítulo explica el problema que se pretende solucionar a través de la ejecución del proyecto. Para este fin, es fundamental conocer de los inicios de la industria como medio para la producción en masa. Además se trata sobre la industria embotelladoras debido a la utilización de frascos. Así mismo se trata acerca de las justificaciones por las cuales se decidió emprender el proyecto y el posible impacto que éste va a tener en el medio y demás efectos.

### **1.1 Descripción del problema**

El problema planteado es la detección de frascos sin etiqueta dentro de procesos industriales debido a los diferentes pasos de control de calidad existentes que buscan ofrecer al mercado un producto que cumpla con los requerimientos de los clientes o consumidores que cada vez son más exigentes. Actualmente las empresas utilizan

diversos sistemas para la realización de este objetivo; desde el más rústico que consiste en la detección de errores realizados por operadores por medio de simple inspección visual hasta los que utilizan infrarrojo.

Con la presentación de esta tesis se propone un método alternativo utilizando un programa de entorno gráfico como LabVIEW que en los últimos años ha incorporado a su entorno el manejo de cámaras USB.

Esto permite realizar simulaciones de laboratorio para la detección de etiquetas en frascos a muy bajo costo debido a que es un prototipo de diseño para futuras investigaciones en el campo del control de calidad en una industria.

## **1.2 Antecedentes, causales y efectos del tema planteado**

### **1.2.1 Antecedentes**

Antes de comenzar a recabar información acerca del tema, es importante definir el significado de “industria”, se define como el conjunto de actividades y procesos que tienen como principal objetivo convertir la materia prima en productos elaborados para así satisfacer las necesidades del hombre [1].

Con el transcurso de los años las industrias se han incrementado y modernizado dando lugar al desarrollo de la

mayoría de los pueblos, este gran crecimiento se debe a la aparición de la automatización de procesos donde el gran objetivo es crear un sistema en donde se asignan tareas de producción por medio de dispositivos programados que logran trabajar de forma independiente o parcialmente dependiente del humano.



Figura 1.1 Proceso parcialmente automatizado

En el mercado mundial existen una gran variedad de industrias como lo son las industrias textiles, farmacéuticas, alimenticias, metalúrgicas entre otras y cada una de estas industrias

mencionadas ha aportado la necesidad y exigencia de creación de nuevos sistemas de control de procesos de manera autónoma en el campo de ingeniería.

Las industrias embotelladoras han tenido mejoras en sus distintos procesos de elaboración, presentación y empaquetamiento de un producto, los cuales han contribuido al reconocimiento de marcas a nivel nacional e internacional. Sin embargo en lo que respecta al proceso de verificación de etiqueta en muy pocas industrias envasadoras se realiza esta inspección.

Hace unas décadas la colocación y verificación de la etiqueta en el frasco se realizaba de manera manual como se aprecia en la Figura 1.2; donde el operario colocaba el adhesivo, pero esta acción no se realizaba de manera uniforme.



Figura 1.2 Colocación manual de etiqueta

Con el desarrollo de la tecnología, se han creado nuevos métodos para la colocación y verificación de etiqueta, uno de estos es el de utilizar un sensor infrarrojo como el de la Figura 1.3 que detecta la existencia de un objeto a una cierta distancia esto ayudaría a verificar si la etiqueta está presente o no en el frasco.

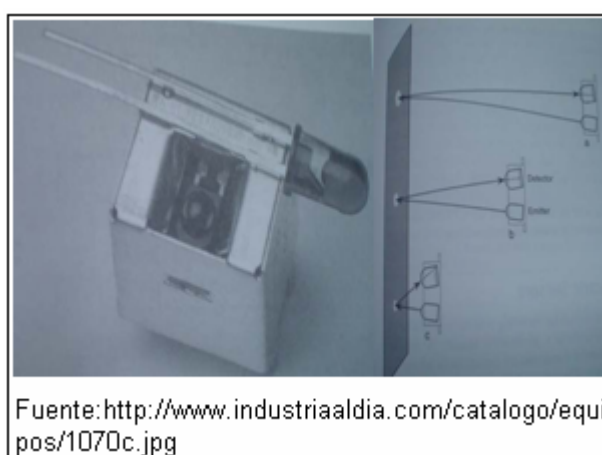


Figura 1.3 Sensor infrarrojo de proximidad

Otra alternativa para la inspección de la etiqueta es adquirir una máquina para la colocación de la misma como se muestra en la Figura 1.4, entonces mediante prueba y error se calibra la máquina etiquetadora para que la etiqueta esté siempre presente.



Figura 1.4 Máquina etiquetadora para botellas

### 1.2.2 Causales

Dado que las Embotelladoras buscan procesos más eficientes, estos deben de ir asociados con unos altísimos niveles de calidad en el acabado del producto, esto conlleva una verificación en el contenido y la presentación antes de salir al mercado, los parámetros medidos son: nivel de líquido adecuado, la tapa debe de estar sellada, la etiqueta debe de tener la información necesaria para su consumo entre otros.



Una de las necesidades que origina el desarrollar este proyecto, es que una vez etiquetado el frasco, se realiza un control de la misma con una simple inspección visual antes de empaquetarlo. Ésta verificación se muestra en la Figura 1.5 , es realizada por el ser humano y la efectúa por muestreo para lo cual toma un frasco de un grupo y lo revisa, pero como seres humanos tendemos a cansarnos con una vida rutinaria y monótona.

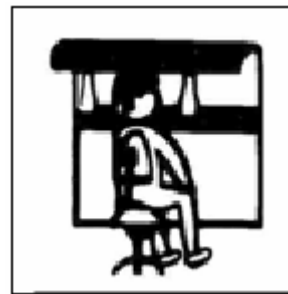


Figura 1.5 Inspección visual de etiquetas en los frascos

Las industrias en busca de mejoras en sus productos se han visto obligados en automatizar sus procesos de producción y por ende se han hecho dependientes de la tecnología.

Por los motivos mencionados anteriormente y observando que en la actualidad nuestro país está en auge del desarrollo de novedosos sistemas de monitoreo y control supervisados por computador, se ha decidido realizar el proyecto “APLICACIÓN DE VISIÓN CON LABVIEW PARA LA DETECCIÓN DE

FRASCOS SIN ETIQUETA” el cual ofrece una alternativa de mejorar en gran parte la adecuada presentación del producto final.

### 1.2.3 Efectos

Las industrias embotelladoras al no tener un sistema autónomo dedicado a la inspección de etiquetas en frascos, están dando mayor probabilidad de que uno de estos salga al mercado sin la adecuada presentación: Estos detalles son importantes debido que influyen en la calidad de presentación del producto final, lo cual es trascendental para el cliente porque es una forma de diferenciar los distintos productos existentes en el mercado. A continuación la Figura 1.6 muestra con una presentación correcta.



Figura 1.6 Frascos con etiqueta

Al optar por esta aplicación, las industrias estarían entrando a un nivel de automatización que garantiza un control continuo sin intervención de operarios responsables de la verificación sino que serían responsables del funcionamiento de la máquina.

Se libera al ser humano de lo tedioso y cansado que puede ser el trabajo rutinario y como resultado la industria se ahorra costos de mano de obra producción.

Con la alternativa propuesta a este problema se puede realizar un control más frecuente de los frascos producidos por la industria sin tener que sacarlos de la banda transportadora.

### **1.3 Alcance del proyecto**

El proyecto está enfocado a la verificación de frascos que no posean etiqueta por medio de una cámara USB, su implementación se realizará sobre una plataforma y un brazo robótico prototipos realizados en el tópico: “Plataforma para el desarrollo de proyectos con DSPICS y visión robótica con LabVIEW”, estos dispositivos tienen unas características de hardware y software definidas las cuales se explicarán en el capítulo 4.

Una alternativa para la aplicación es que pueda ser acoplada a cualquier tipo de industria pero para poder llevar a cabo esto se necesita realizar las variaciones y análisis tanto en hardware como en software.

#### **1.4 Planteamiento de solución**

Para comenzar a realizar la solución, se va a utilizar el principio básico que fue enseñado a lo largo de la mayoría de la materias tomadas en la universidad y el cual sin duda es de mucha utilidad para resolver problemas complejos como es el divide y vencerás [2].

Al descomponer el proyecto de aplicación de visión con LabVIEW para la detección de frascos sin etiqueta, se obtiene las siguientes etapas: Realizar la adquisición de la imagen, se continúa con su procesamiento, después se realiza la comparación con una imagen preestablecida y finalmente cómo enviar señales a la tarjeta que controla el brazo robótico y así este debe generar determinada acción.

A continuación se explican brevemente como solucionar cada una de los pasos mencionados en el párrafo anterior.

Para la adquisición de las imágenes, se va a emplear una cámara USB Standard marca Genius modelo 112 de resolución 352x4288

pixeles que tienen la capacidad de capturar 30 fotos por segundo y su costo es rentable.

Se realiza la aplicación con LabVIEW 8.2 para reconocer la cámara Genius, luego se captura la imagen, una vez capturada la imagen el siguiente paso es desarrollar la aplicación utilizando el paquete IMAQ Vision de LabVIEW que contiene VI's que permiten realizar el procesamiento de la imagen y de igual manera contiene filtros que se necesitan utilizar para eliminar distorsiones de la misma; para la captura de la imagen se utiliza la cámara en modo de video de esta manera realiza la toma de fotos de los frascos para lograr que sea un muestreo constante.

El objetivo final es acoplar la aplicación realizada en la banda transportadora y el brazo robótico, por ende hay que tener en cuenta las características físicas de ambos de las cuales la de mayor importancia es la velocidad de trabajo.

Otro aspecto es la iluminación uniforme que se debe tener en el área de captura de la imagen.

# CAPÍTULO 2

## 2. VISIÓN POR COMPUTADORA

En el presente capítulo se muestran los fundamentos básicos de la visión por computadora. Además se explican las técnicas de procesamiento digital de imágenes, también se expone los campos de aplicación de la visión artificial, por último se describen los programas desarrollados para la adquisición de la misma y hardware implementados que utilizan la visión artificial en el mercado.

### 2.1 Introducción

#### 2.1.1 Imágenes

En términos generales, se entiende por "imagen" la apariencia visible de una forma, entonces una imagen es la representación grabada de una forma o de una entidad determinada [3].

Existen varios tipos de imágenes entre las que tenemos imágenes físicas visibles (fotografía tradicional, digital) y también imágenes físicas invisibles, o imágenes que tienen las mismas características que las visibles pero quedan fuera del rango al que los seres humanos son sensibles, por ejemplo las imágenes infrarrojas o ultravioletas, que pueden ser visualizadas por medio de filtros y dispositivos especiales [4].

También cabe hablar de imágenes abstractas, tal como las imágenes matemáticas que son conceptos y, en consecuencia, invisibles, pero que admiten representaciones de diverso tipo. Se puede puntualizar que las imágenes físicas visibles pueden ser permanentes; por ejemplo un cuadro, un documento impreso, o transitorias tales como las producidas por un monitor CRT.

Matemáticamente la imagen es una función continua  $f$  de dos variables  $x$  i  $y$ , definida en una región rectangular de un plano. Donde  $f(x,y)$  es el valor de luminosidad de la imagen en el punto  $(x,y)$ .

A continuación la Figura 2.1 muestra la representación de un objeto y de una imagen.

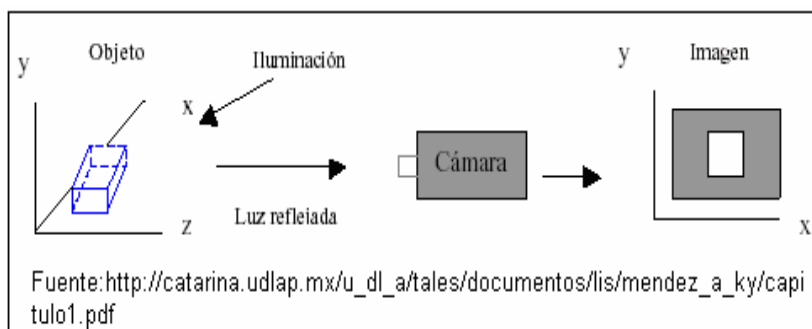


Figura 2.1 Representación del espacio de un objeto y del espacio de una imagen [Adaptación de Pearson 1991]

### 2.1.2 Imágenes digitales

Al transformar la información de una escena, o de cualquier otra toma del mundo real a un medio electrónico, se está generando una imagen digital, entonces la imagen digital es una aproximación discreta  $I(i,j)$  de una imagen  $f(x,y)$  en forma de matriz  $R \times C$ . La figura 2.2. muestra la representación de una imagen digital.

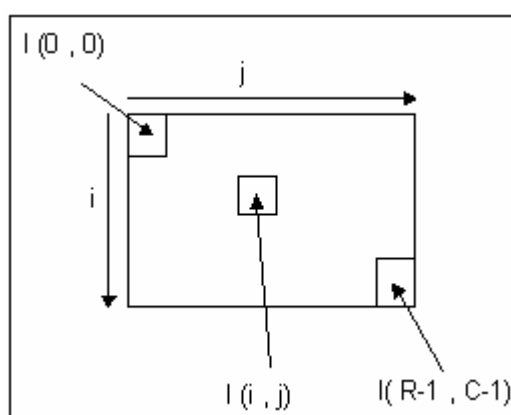


Figura 2.2 Representación matricial de una imagen



**Píxel.-** Es una muestra de  $f(x,y)$  almacenada en  $I(i,j)$ ; usualmente es un entero positivo de 8 bits [0:255].

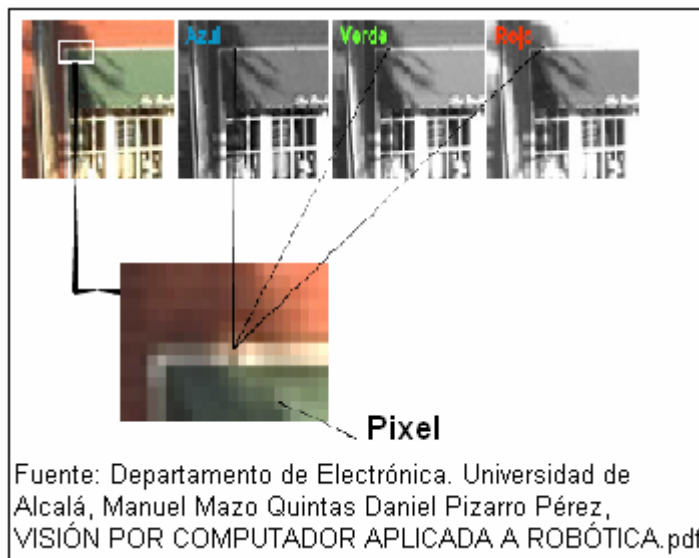


Figura 2.3 Imagen a color y píxel

Cabe recalcar que la imagen, como es función continua, no puede representarse perfectamente en un ordenador digital.

Las imágenes, una vez digitalizadas, pueden ser tratadas como cualquier otro tipo de información en la computadora, es decir, se pueden aplicar diversos procesos sobre ellas. Si se tienen dos imágenes, estas bien pueden ser manipuladas, ya sea con operadores lógicos o aritméticos. Los operadores de imágenes que en este proyecto se usarán son de tipo aritmético. Las operaciones de dos imágenes se hacen tomando píxel a píxel y manipulando su valor o contenido.

Esta información podría no ser un simple número, como es el caso de los píxeles que contienen varios canales de color, por lo que las operaciones podrían tener diferentes definiciones [5].

### **2.1.3 Definición de visión por computadora**

La visión por computador se define como un sistema informatizado que captura la imagen de un objeto para posteriormente realizar el proceso de extracción de información del mundo físico a partir de la misma, utilizando para ello un computador. La visión por computadora es también llamada visión artificial [6].

### **2.1.4 Descripción de la visión por computadora**

La visión artificial es una rama de la inteligencia artificial en el área de percepción, y son todas aquellas técnicas y modelos que nos permitan el procesamiento, análisis y explicación de cualquier tipo de información especial obtenida a través de imágenes digitales [7].

La visión artificial tuvo sus comienzos hace más de 30 años. Desde que se desarrollaron computadores digitales suficientemente potentes para procesar mayor información y

adquirieron una forma más compacta, se pensó en utilizar información obtenida por sensores para dotar de “ojos” al ordenador y este trabajará de manera autónoma.

Desde sus inicios la visión por computadora ha inspirado sus desarrollos en el estudio del sistema visual humano el cual sugiere la existencia de diferentes tipos de tratamiento de la información visual dependiendo de metas u objetivos específicos, es decir, la información visual percibida es procesada en distintas formas con base en las características particulares de la tarea a realizar, por lo que la visión por computadora propone varias técnicas que permiten obtener una representación del mundo a partir del análisis de imágenes obtenidas desde cámaras de video.

Debido a que la información visual es una de las principales fuentes de datos del mundo real, resulta útil el proveer a una computadora digital del sentido de la vista (a partir de imágenes tomadas con cámaras digitales o analógicas), que junto con otros mecanismos como el aprendizaje hagan de ésta una herramienta capaz de detectar y ubicar objetos en el mundo real, objetivo principal de la visión por computadora.

## **2.1.5 Comparación entre el sistema de visión humano y el sistema de visión artificial**

Para poder realizar una comparación con el sistema humano debemos comprender como trabaja éste.

### **2.1.5.1 Sistema de visión humano**

La visión se realiza a través del ojo, órgano par colocado en la parte anterior de la cavidad orbitaria de la cara [8].

En los seres humanos casi el 80% de la información que recibimos del mundo exterior es a través del sistema visual.

El funcionamiento del sistema de visión humano de manera general comienza con la recepción de estímulos de los rayos de luz procedentes del entorno y lo transforma en impulsos nerviosos. Estos impulsos llegan hasta el centro cerebral de la visión, donde se decodifican y se convierten en imágenes.

Al igual que la cámara, un ojo normal enfoca la luz y produce una imagen nítida. Sin embargo, los mecanismos mediante los cuales el ojo controla la

cantidad de luz admitida y los ajustes para producir imágenes enfocadas correctamente son mucho más complejos, intrincados y efectivos que los correspondientes incluso a una cámara más avanzada. A continuación se indican las ventajas de ambos sistemas en procesos de control de calidad utilizando visión.

#### **2.1.5.2 Ventajas de la visión humana**

- Uniendo los 5 sentidos se obtiene una percepción total de la realidad y en caso de controlar algún dispositivo las operaciones que puede realizar no están limitadas a lo que tenga un procesador.
- La iluminación es una parte crítica dentro de un sistema de visión. Las cámaras, de momento, son mucho menos sensibles y versátiles que la visión humana y las condiciones de iluminación deben optimizarse al máximo para que una cámara pueda capturar una imagen que el ojo humano podría distinguir sin necesidad de una iluminación tan especializada, por lo

mencionado anteriormente el ojo es considerado una maravilla fisiológica.

### **2.1.5.3 Ventajas de la visión artificial**

- En muchos procesos de fabricación, los límites de detección de defectos han superados la percepción del ojo humano.
- La visión por computador se está convirtiendo rápidamente en un factor clave en el desarrollo de la calidad total dentro de los diferentes procesos de automatización industrial.
- La visión por computador permite inspeccionar un proceso de producción sin fatigas ni distracciones y de esta manera se consigue un control más continuo.
- Su implementación en una empresa genera un aumento en el nivel de producción y una reducción en los costos de fabricación.

## 2.1.6 Esquema general de un sistema de visión por computadora

Los sistemas de visión por computadora poseen varios elementos en su esquema general, a continuación la Figura 2.4 muestra el esquema:

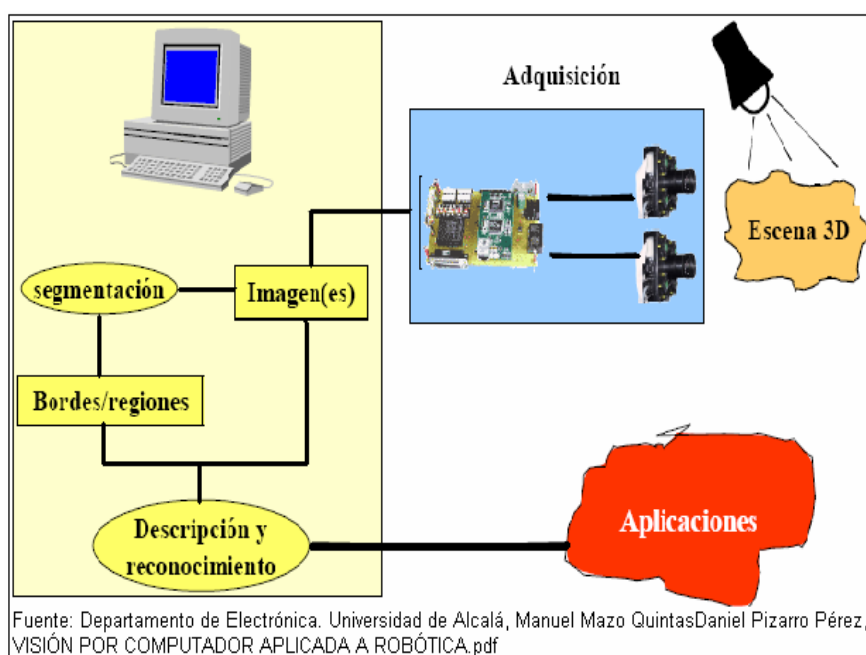


Figura 2.4 Esquema general de un sistema de visión por computador

A continuación se describen cada una de los elementos básicos que posee un sistema de visión basado en PC :

### 2.1.6.1 Escena 3D

Debe existir un objetivo a capturar el cual puede estar estático o en movimiento.

### 2.1.6.2 Iluminación

Es muy importante poseer una buena fuente de luz y su distribución tiene que ser uniforme en el área de captura de la imagen. Existen distintas técnicas de iluminación:

- Iluminación posterior.
- Iluminación Frontal Oblicua y Direccional.
- Iluminación Frontal Axial (difusa).
- Iluminación de Campo Oscuro
- Arreglo de luces.



Figura 2.5 Diferentes tipos de luz



### 2.1.6.3 Adquisición de imagen

Se realiza por medio del sensor, la función básica del sensor es convertir la luz a energía eléctrica. En la Figura 2.6 se aprecia el sensor más utilizado que es la cámara.



Figura 2.6 Cámaras analógicas Panasonic

Existen dos tipos de cámaras: la analógica y digital, a su vez esta última se puede subdividir en IEEE1394, Camera link y Digital Parallel.

### 2.1.6.4 Digitalizador

Convierte la energía continua en discreta. Para procesar una imagen se debe convertir primero en una forma numérica; esta conversión numérica es llamada digitalización [Castleman, 1979] [9].

### **2.1.6.5 Preprocesamiento**

El objetivo es modificar la imagen que se acaba de adquirir con el fin de enriquecerla en detalles de acuerdo a los parámetros a analizar, las modificaciones pueden ser:

- En caso de existencia de ruido se procederá a su eliminación.
- Acentuar o perfilar las características de una imagen tales como bordes y límites.
- Contrastar la imagen para que sea más útil la visualización gráfica y el análisis de la misma.

### **2.1.6.6 Segmentación**

Su objetivo es dividir la imagen en las partes que la constituyen y de esta manera podemos obtener nuestra región de interés. Un ejemplo de este proceso puede ser la diferencia entre el objeto y el fondo.

### **2.1.6.7 Descripción**

En esta etapa se pretende etiquetar los objetos teniendo en cuenta información suministrada por la inspección que puede ser:

- Cuantitativa: Realización de mediciones (áreas, longitudes, perímetros etc.) y ángulos de orientación.
- Cualitativa: Verificación de la correcta realización del trabajo como el ensamblado, el embotellado, el etiquetado, el empaquetado etc.

### **2.1.6.8 Reconocimiento e interpretación.**

Con los datos analizados se identifica los objetos de una escena y les asigna una interpretación para después realizar la toma de decisiones. Frecuentemente los sistemas de visión artificial controlan aparatos mecánicos que efectúan la manipulación de los productos.

## 2.2 Procesamiento digital de imágenes

Continuando con los componentes básicos del procesamiento de imágenes mostrados en la Figura 2.7 se debe realizar el análisis y procesado de la imagen para después de esta etapa obtener la información útil requerida.



Figura 2.7 Componentes básicos del procesamiento de imágenes [Green, 1989]

Con el procesamiento digital de la imagen se obtiene una modificación rápida de la imagen prácticamente libre de ruido en forma de una matriz de enteros (píxeles) en vez de las clásicas manipulaciones en el cuarto oscuro o filtración de voltajes dependientes del tiempo necesario para las imágenes analógicas y señales de video. Aunque muchos algoritmos de procesamiento de imágenes son extremadamente robustos, a menudo el usuario aplica operaciones a imágenes digitales sin tener en cuenta los principios subyacentes tras dichas manipulaciones.

Algunas de las operaciones principales en el procesamiento de imágenes son la transformación, codificación, extracción de

características (detección de bordes), reconocimiento de patrones, entre otras [Chellapa, et.al., 1985] [10].

### **2.2.1 Transformaciones de imágenes.**

Se refiere al tratamiento o edición de imágenes digitales y se manifiesta una tendencia clara al acercamiento entre el sistema vectorial (objetos) y los píxeles. Algunas de las operaciones que se pueden realizar son:

- Operaciones Puntuales
- Operaciones Locales
- Operaciones Globales
- Operaciones Geométricas

### **2.2.2 Operaciones Puntuales**

Son aquellas operaciones que se realizan sobre una o varias imágenes y se las realiza de manera dependiente o independiente de las características de la misma, su objetivo es mejorar la calidad de la imagen y facilitar la segmentación de objetos.

Las operaciones más comunes considerando la independencia de las características globales son: suma, resta, multiplicación, división, inversa, AND, OR entre otras.

A continuación se muestra la operación resta entre dos imágenes idénticas.

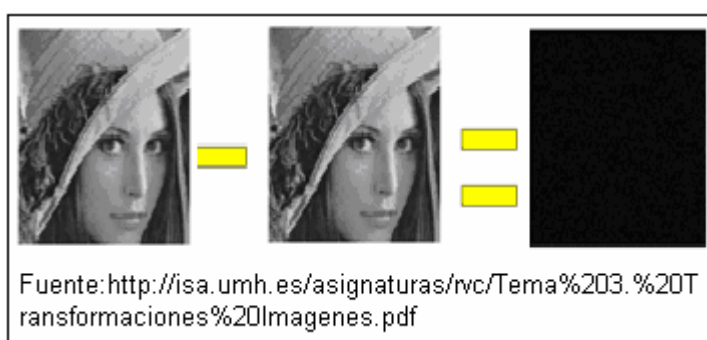


Figura 2.8 Operación resta

Las operaciones más comunes considerando la dependencia de las características globales son: Utilización de histogramas, medidas de intensidad, perfiles de líneas entre otras.

La información que proporcionan los histogramas son el número de ocurrencias de los niveles de grises presentes en una imagen, donde el eje de las x se encuentra la frecuencia de cada nivel de la imagen y en el eje de las y se encuentra el nivel de gris[11].

A continuación se presenta una imagen y su histograma respectivo.

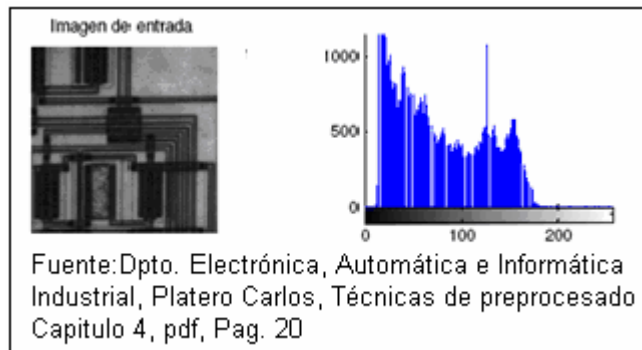


Figura 2.9 Histograma de imagen

Los perfiles de líneas nos proporcionan información de variaciones de intensidad a lo largo de una línea, se utiliza para delimitar fronteras entre objetos, también se utiliza para detectar la presencia de patrones repetitivos.

La Figura 2.10 es un ejemplo de perfil de línea.

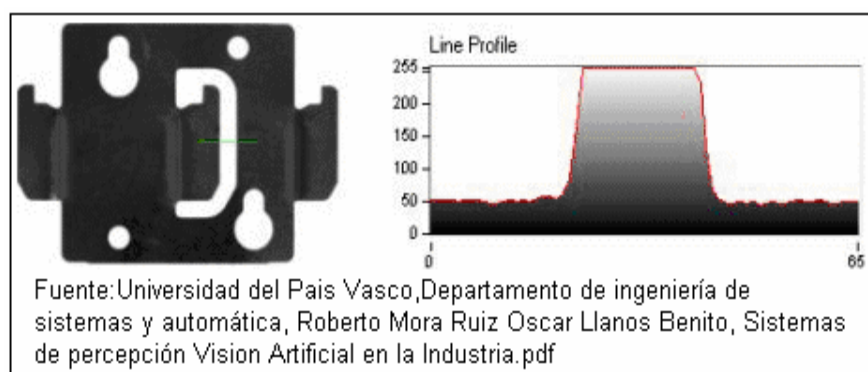


Figura 2.10 Perfil de línea

### **2.2.2.1 Operaciones locales**

Con este tipo de operaciones se pretende suavizar determinadas irregularidades que pueden aparecer en la imagen ya que afectan directamente a la nitidez de esta y no se puede apreciar los pequeños componentes.

El método más utilizado para el filtrado de imágenes es la convolución que es una operación de matrices que hace una especie de media entre las dos mientras, sucesivamente, una de las matrices se arrastra por encima de la otra. Y al final se obtiene un promedio del nivel de gris del píxel.

### **2.2.2.2 Operaciones globales**

Estas operaciones se realizan sobre toda la imagen directamente no se trabaja a nivel de píxel y entre las más empleadas se destacan:

Transformada de Fourier que es el cambio del dominio del tiempo al dominio de la frecuencia.

Cambio entre modelos de color que usualmente se realiza entre el modelo RGB que es el básico y el



modelo HSV que es muy útil por la pureza de sus colores en el campo de visión por computadora [12].

Transformada de Hough permite detectar curvas o fronteras de un objeto en una imagen.

### 2.2.2.3 Operaciones geométricas

Su objetivo es la corrección de perspectiva de la escena, estas operaciones están presentes en la mayoría de programas de diseño gráfico las más comunes son: Variación de tamaño, traslación, rotación, recorte entre otros.

La Figura 2.11 muestra operaciones geométricas de un imagen.



Figura 2.11 Rotación de una imagen

### 2.2.3 Detección de bordes

La definición de bordes o contornos son los extremos u orillas de alguna cosa. El sistema visual humano utiliza información del color de los objetos, de sus texturas, de las sombras, pero sobre todo de los contornos o bordes de los objetos para la percepción de escenas del mundo real, de igual manera la visión artificial emula la detección automática de bordes, la cual es una tarea esencial en muchas aplicaciones de interpretación de las imágenes.

En visión artificial se entiende como borde aquella región donde aparece una fuerte variación del nivel de intensidad en los píxeles adyacentes. Su causa principal es originada por la intersección de varios objetos, con diferentes niveles de reflexión, que al ser proyectados sobre la cámara generan discontinuidades de intensidad en los píxeles asignados.

Sin embargo, estas discontinuidades también aparecen de forma no deseada por la presencia de ruido, por el efecto de sombras sobre los propios objetos o por una iluminación no uniforme dentro de la escena [13]

Al realizar la detección de bordes se puede transformar la imagen en un dibujo a base de líneas, los contornos están

relacionados con propiedades relevantes de los objetos de la escena por lo que suele ser una etapa muy empleada para las tareas de segmentación o para la búsqueda de objetos geométricos más complejos como líneas, elipses, esquinas, etc.



Figura 2.12 Detección de bordes

La mayor parte de las aplicaciones de visión artificial detectan la presencia de partes y medida de dimensiones para ver si cumplen especificaciones, por eso es importante la detección de bordes.

El razonamiento utilizado para la detección de los bordes está en la aplicación del operador: primera derivada y la segunda derivada en un entorno de vecindad mostrados en la Figura 2.13.

Primera derivada.- Realizando esta operación se produce un resalte en las zonas donde la intensidad no es homogénea porque se observa que esta toma un valor de máximo o mínimo.

Segunda derivada.- Cuando se realiza esta operación proporciona la ubicación del borde ya que se genera un cambio de signo (paso por cero) en el punto de inflexión de la primera derivada[14].

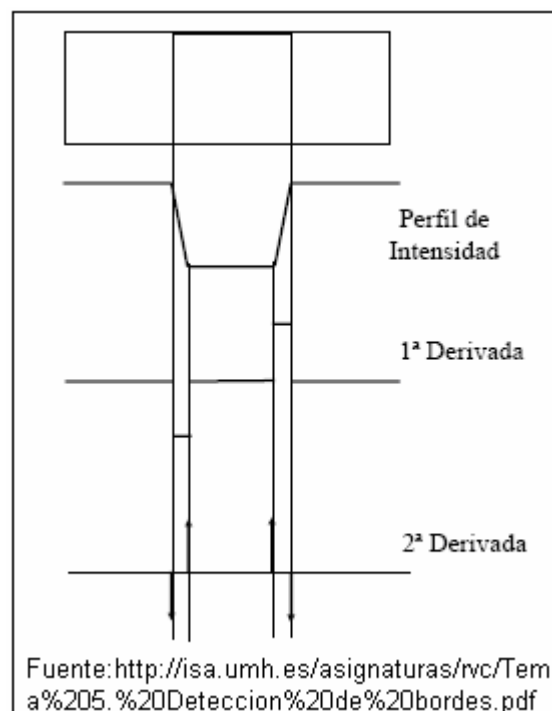


Figura 2.13 Método de primera y segunda derivada

En la práctica para la detección de bordes existen diferentes técnicas que se basan en el fundamento básico, a

continuación se mencionan y se muestran las técnicas más utilizadas:

- Técnica basada en el operador gradiente
- Operadores basados en la Laplaciana
- Operador Sobel
- Operador de Canny

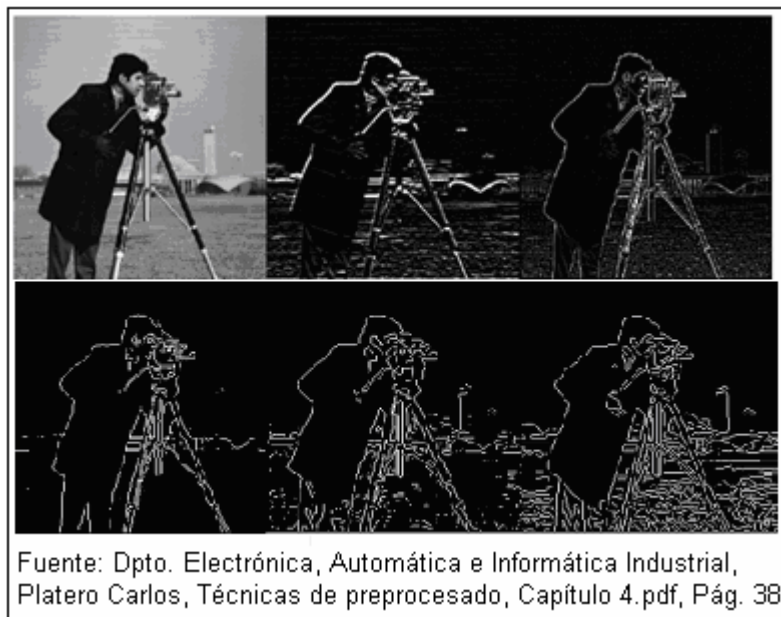


Figura 2.14 Detección de bordes a) Operador Gradiente, b) Operador Laplaciano b)Prewit, c) Sobel d) Canny

#### 2.2.4 Segmentación

Es la base para el análisis de imágenes, se realiza según la información que se busque dentro de la imagen, entre algunas

de las opciones tenemos segmentación por color, por umbral y por la región delimitada por los bordes, entre otros.

A continuación se explica la segmentación por umbralización debido a que es una técnica muy utilizada en la actualidad.

#### **2.2.4.1 Umbralización**

Con este método se puede segmentar una imagen en función de los niveles de intensidad de los píxeles. Existen tres tipos de umbrales: el global, local y dinámico-zonal.

Umbral Global.- Se define un umbral a partir de la selección del histograma y es único para toda la imagen y la desventaja es que con el ruido no se definen los puntos de inflexión.

Umbral local.- Se realiza una subdivisión de la imagen capturada, luego se define un umbral único para cada subimagen.

Umbral dinámico-zonal.- Es también llamado multiumbral, si se poseen  $N$  objetos en la imagen se puede tener un rango de  $N-1$  umbrales para el proceso de segmentación.

La segmentación de imagen es una temática actual de investigación y se han definido numerosos métodos diseñados para aplicaciones particulares. No obstante, no hay métodos que sean útiles en una amplia variedad de situaciones, por lo que cada problema necesita la adaptación o desarrollo de nuevas técnicas.

### **2.3 Aplicaciones de visión artificial**

Debido al auge de la tecnología, los sistemas de adquisición de imágenes como cámaras, sensores, computadoras, entre otros que realizan el procesamiento de imágenes han mejorado cada vez más su nivel de funcionamiento y se encuentran en el mercado a precios accesibles. La mezcla de estos factores ha contribuido a la expansión de los campos de aplicación de la visión artificial en muchas áreas.

Como ya se mencionó anteriormente la visión artificial consiste en la captación de imágenes en vivo mediante cámaras CCD y para su posterior tratamiento a través de las técnicas de procesamiento avanzadas, permitiendo así poder intervenir sobre un proceso (modificación de variables del mismo) o producto (detección de unidades defectuosas), para el control de calidad y seguridad de toda la producción.

El campo de aplicación de las capacidades de visión artificial es muy variado pues abarca desde las telecomunicaciones, industria, medicina, seguridad, entre los más destacados tenemos:

### 2.3.1 Aplicaciones en Medicina

En la medicina se tiene algunas aplicaciones en las que aparece el procesamiento de imágenes, la mayoría de ellas orientadas hacia el diagnóstico de dolencias o enfermedades, entre las que se incluyen radiografías, resonancias magnéticas, tomografías y ecografías como la mostrada a continuación <sup>1</sup>.



Figura 2.15 Ecografía normal (derecha) y la ecografía adquirida en visión por computadora (izquierda)

---

<sup>1</sup> Reyes K, Delgado L. "Diseño de un software basado en formato DICOM para, almacenamiento y transmisión de imágenes en Telemedicina" Tesis, Escuela Superior Politécnica del Litoral, Facultad de Ingeniería en Electricidad y Computación, 2007.



La Figura 2.16 muestra que la visión por computadora también se utiliza para realizar el conteo del número de glóbulos rojos, blancos y plaquetas de una muestra sanguínea, el conteo se lo realiza de una manera estadística<sup>2</sup>.



Figura 2.16 Muestra sanguínea

### 2.3.2 Aplicaciones a nivel de Identificación

En este campo se han desarrollado muy importantes y distintos sistemas para varios requerimientos, uno de ellos es el reconocimiento de la iris para el acceso a determinada área restringida, una variante de este diseño es la lectura de huellas

---

<sup>2</sup> Proyectos de telemedicina: <http://www.itc.ucm.es/>

dactilares. A continuación se muestra el sistema de reconocimiento de iris:



Figura 2.17 Reconocimiento de iris

### **2.3.3 Aplicaciones en el área militar.**

Las aplicaciones no son muy variadas pero si son muy complejas entre las principales se puede encontrar: detección de seguimiento de objetivos, análisis del terreno y armas inteligentes.

Se ha empleado para la localización de infraestructuras y objetos desde satélites; un ejemplo de esto se puede encontrar en Kass y col. (1987) donde se describe la detección de carreteras mediante el procedimiento de contornos deformables. También en Fischler y col. (1987) se presenta un método de detección de carreteras el resultado de este método es la Figura 2.18.

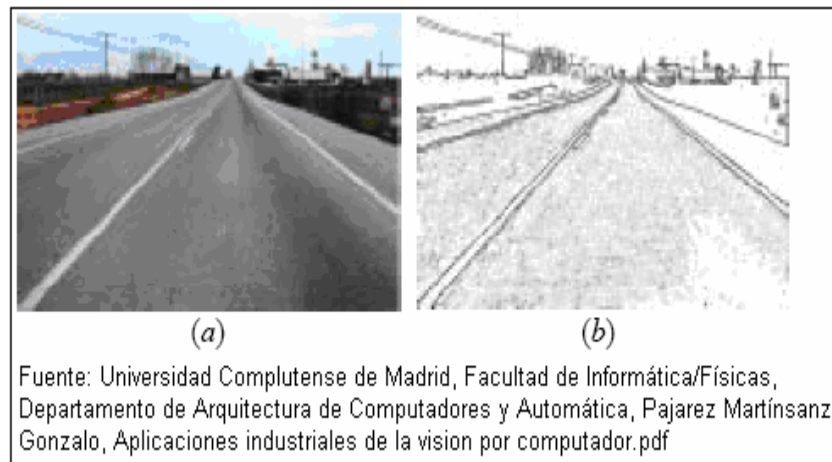


Figura 2.18 Análisis de terreno

#### 2.3.4 Aplicaciones en la robótica

En la robótica las aplicaciones que más se subrayan son: la navegación de los robots de manera autónoma a partir de un método de localización y mapeo simultáneos (SLAM). La información procedente de la visión es validada, comparada y finalmente integrada con el resto de la información proporcionada por otro tipo de sensores; de esta manera los robots pueden tener una apreciación del mundo real [15].

La Figura 2.19 se muestra el robot desarrollado para la navegación autónoma.

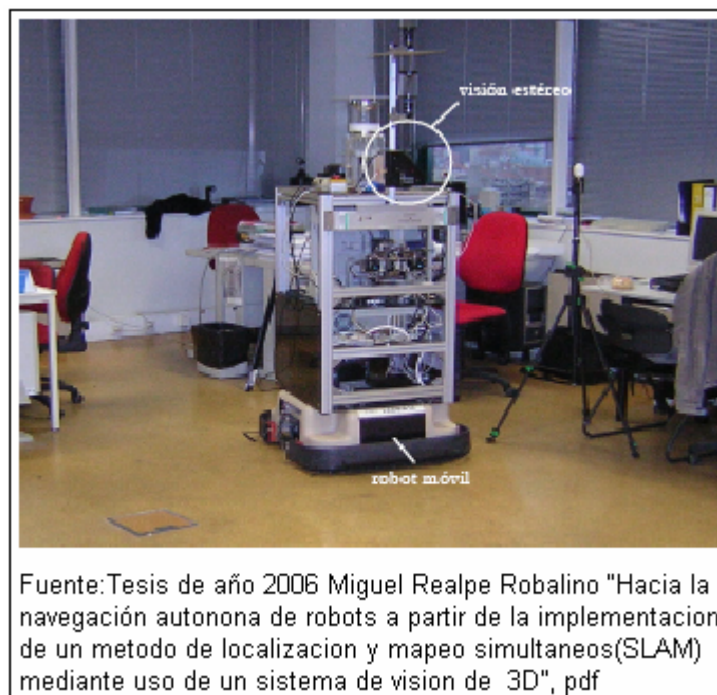


Figura 2.19 Sistema de visión estéreo montado sobre un robot móvil

Otra aplicación, la cual ha tomado una gran importancia desde el 2002 e incluso se realizan campeonatos entre universidades, es el denominado fútbol robótico, también tenemos en este campo los robots teleoperados que utilizan lo que se denomina bits tangibles [16]

### 2.3.5 Aplicaciones en el control de calidad en las industrias

En el control de calidad que poseen una variedad de aplicaciones que dependen de los parámetros a inspeccionar, aquí se pueden realizar el control de:

- Verificación de etiquetas
- Inspección de soldaduras
- Inspección de motores
- Inspección de cristales
- Control de calidad de comida
- Inspección de circuitos impresos
- Inspeccionar impurezas en líquidos y el nivel de llenado de líquido

A continuación se muestra un diseño de planta donde se realiza el control de calidad del nivel de líquido del envase.

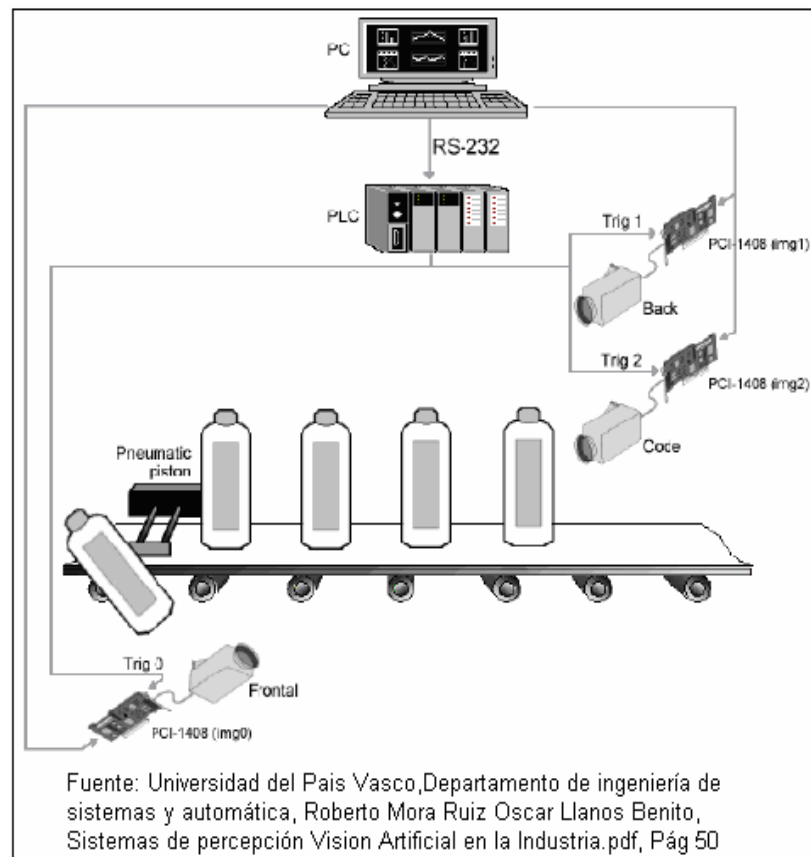


Figura 2.20 Control de calidad con visión artificial

### 2.3.6 Aplicación en industrias

En el Ecuador las empresas de integración de soluciones industriales han desarrollado varios sistemas de visión artificial, autónomos e integrados, en el campo industrial con el fin de automatizar procesos, entre los cuales tenemos:

Control numérico de un brazo robótico, como respuesta a algoritmos de procesamiento de imágenes y sistemas expertos mostrado en la Figura 2.21.

Sistema de monitoreo de fugas de caudal de líquido, utilizando procesamiento digital de imágenes para la unidad civil geominera.

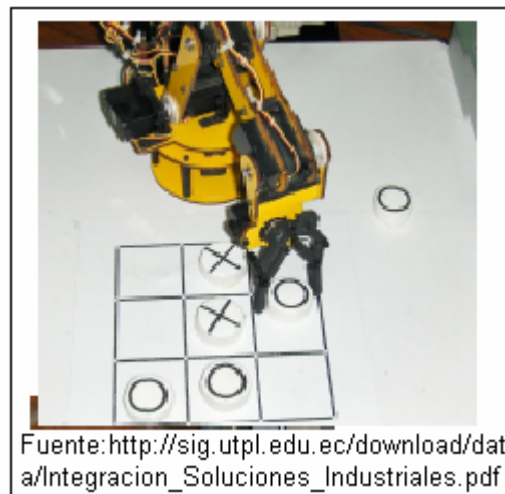


Figura 2.21 Control de un brazo robótico

### 2.3.7 Aplicación en otros campos

En el área de las telecomunicaciones la visión se enfoca en la telefonía móvil con la implementación de las videos conferencias.

Otro campo de aplicación es el control de velocidad vehicular para lo cual se realiza lectura de matrículas de carros en determinadas calles de la ciudad, esta aplicación permite la identificación y procesado de matrículas en movimiento.



Figura 2.22 Matrículas de vehículos

En el campo de diversión (ocio) muchas personas diseñan e implementan dispositivos como pasatiempos de las cuales se puede encontrar información en el Internet; un ejemplo de esto es el “recoge objetos” el cual fue realizado por la librería Saphira, controlando el movimiento de un robot guiándose por las características de forma y cromaticidad del objeto a recoger. Esta aplicación puede tener un uso mas apropiado con los respectivos cambios.



Figura 2.23 Recoge objetos



## **2.4 Software utilizado para la adquisición de imágenes**

El objetivo de toda aplicación de visión por computador es tomar decisiones basándose en los objetos de una escena captada de una imagen. Para poder tomar estas decisiones es importante realizar un procesamiento de la imagen y obtener información de interés.

La adquisición y procesamiento de imágenes se logra a través de software de visión artificial.

### **2.4.1 Requerimientos del software**

Para poder realizar la adquisición de imágenes, necesitamos de un software que cumpla con los siguientes requisitos:

#### **2.4.1.1 Selección de la cámara**

Al momento de escoger un software de visión, lo más importante que se debe considerar es que trabaje con la cámara que mejor se ajusta a la aplicación y a las condiciones económicas de la empresa.

Con el desarrollo económico es demasiado fácil hallar cámaras analógicas de bajo costo pero, a menudo, una aplicación requiere más resolución VGA y una

mayor calidad de imagen general que lo que posee una cámara analógica común.

#### **2.4.1.2 Flexibilidad del hardware**

El software debe poder acoplarse a cualquier tipo de cámaras ya sean analógicas o digitales debido a que la tecnología de cámaras está avanzando rápidamente, estos avances brindan la posibilidad de tener una mejor calidad de imagen para aplicación.

#### **2.4.1.3 Precisión y rendimiento del algoritmo**

Lo más importante después de haber adquirido las imágenes, es la etapa de procesado para lo cual se necesita la manipulación de herramientas de procesado, hoy en día existe una diversidad de algoritmos para procesar imágenes. Al elegir el software, este debe aprovechar al máximo los algoritmos para que sea preciso y confiable obteniendo de esta manera un mínimo error en la aplicación.

Otra consideración es la velocidad de ejecución de los algoritmos. No importa la gran cantidad de algoritmos

que se tengan para elegir o cuán rápido se puede construir una aplicación con ellos ya que, si las herramientas de inspección son ineficientes y toman mucho tiempo para ejecutarse, entonces la mayor parte del trabajo se desperdicia.

#### **2.4.1.4 Facilidad al utilizar**

Un software debe ser amigable al momento de utilizarlo, debe prestar todas las herramientas posibles para el procesado de imágenes.

#### **2.4.1.5 Nivel de integración con otros dispositivos**

Después de haber realizado la aplicación de visión se tiene claro que se debe interconectar a un sistema de control mucho mayor.

En automatización industrial, la aplicación de visión puede requerir el control de actuadores para clasificar productos, comunicar los resultados de la inspección a un controlador robot, PLC o controlador de automatización programable (PAC entre otros).

#### **2.4.1.6 Costo del software**

Los precios deben ser accesibles para los clientes y deben depender del grado de exactitud y precisión de los algoritmos implementados, también se debe incluir un soporte técnico directo en caso de que el consumidor no pueda obtener la función deseada por el software.

#### **2.4.2 Tipos de software**

Una vez considerado los requerimientos, se procede a mencionar y describir de manera general algunos de los programas que existen en el mercado:

##### **2.4.2.1 Pyvison<sup>3</sup>**

Es un software que tiene como objetivo brindar soluciones de visión artificial para el sector industrial, se caracteriza por su simplicidad, flexibilidad, su

---

<sup>3</sup> Descarga de manuales: [www.pyvision.asp.htm](http://www.pyvision.asp.htm)

eficacia, su software es configurable para cualquier aplicación

#### **2.4.2.2 I+Solex<sup>4</sup>**

Es un software de procesamiento y análisis de imágenes desarrollado en Java. Permite capturar, mostrar, editar, analizar imágenes; crear y analizar vídeo. Está diseñado especialmente para aplicaciones científicas.

Una característica de I+Solex permite el manejo de diversos tipos de cámaras, tanto digitales, vía firewire (IEEE 1394), USB, Camera Link, como analógicas, vía tarjeta digitalizadora.

---

<sup>4</sup> Para obtener mayor información ir al enlace [www.solexvision.com](http://www.solexvision.com)

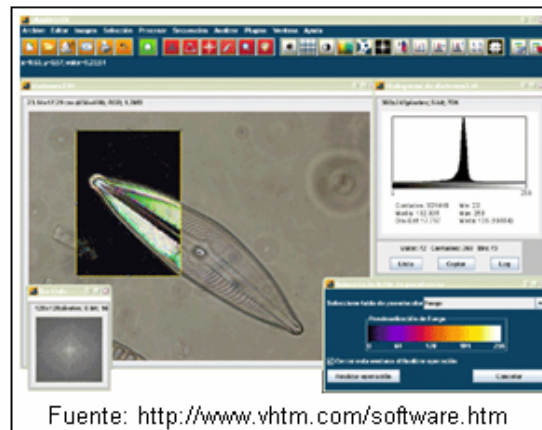


Figura 2.24 Ventana del software I+Solex

### 2.4.2.3 MvImpact

Es un software modular para el procesamiento de imágenes de última generación, para integradores de visión y fabricantes de equipos. Compatible con las cámaras y tarjetas matrix vision.

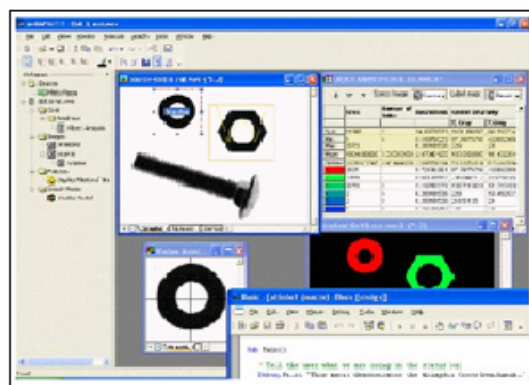


Figura 2.25 Software MvImpact

#### **2.4.2.4 OpenCV<sup>5</sup>**

OpenCV es la abreviación de The Open Computer Vision Library, esta es una librería de visión artificial desarrollada por Intel en el 2003, esta librería, que es código abierto desarrollado en C, presenta las siguientes características: estabilidad, escalabilidad, rápido crecimiento y desarrollo lo que garantiza su continuidad y la hacen especialmente útil en el ámbito universitario.

#### **2.4.2.5 LabVIEW**

Este software fue creado por Nacional Instruments, su programación se realiza en lenguaje gráfico, su completo stock de herramientas permite realizar una variedad de aplicaciones para las distintas áreas laborales del ser humano. En lo que respecta al sistema de visión artificial hace más de una década se ha convertido en el líder resolviendo problemas

---

<sup>5</sup> [www.intel.com/software/products/opensource/libraries/cvfl.htm](http://www.intel.com/software/products/opensource/libraries/cvfl.htm), 2003

para el control de calidad en la industria, en obtención imágenes médicas de mejor calidad, etc.

Nacional Instruments ofrece dos herramientas muy potentes para el desarrollo de aplicaciones de Visión como lo son: la NI Vision Assistant y NI Vision Builder, también contiene la librería IMAQ Vision con la cual se puede realizar la captura y procesamiento imágenes.

En el desarrollo de este proyecto se utilizará software LabVIEW debido a que la universidad posee la licencia para su utilización. Otra característica para su elección es que es un lenguaje muy amigable, además cumple con todos los requerimientos mencionados anteriormente.

En el capítulo 3 se provee mayor información sobre el manejo de los componentes y librerías.

## **2.5 Hardware implementados**

En la actualidad existen una gran variedad de dispositivos implementados que utilizan visión artificial, de esta manera se ha



pretendido dar solución a las exigencias requeridas en el mundo real que utilizan visión artificial.

El área donde el avance ha sido notable es en el control de calidad en las industrias para lo cual se han diseñado sistemas como los que se menciona a continuación:

### **2.5.1 LightPix AE20**

Es un dispositivo desarrollado por Panasonic que utiliza visión artificial a través de una fotocélula, la cual es muy fiable y económica en procesos industriales e inclusive posee una adaptabilidad para controlar distintos procesos, los cuales se configuran a través de una PC que posea Windows y su control se realiza mediante una pantalla táctil. La empresa Panasonic posee una gama de dispositivos que utilizan visión artificial desde los más sencillos como los realizados con fotoeléctricos, hasta los más completos los cuales utilizan cámaras inteligentes<sup>6</sup>.

---

<sup>6</sup>[http://www.panasonic-electric-works.es/pewes/es/downloads/sf\\_x633\\_vision\\_artificial\\_es\\_0707.pdf](http://www.panasonic-electric-works.es/pewes/es/downloads/sf_x633_vision_artificial_es_0707.pdf)

La Figura 2.26 muestra algunas de las utilidades de LightPix AE20.

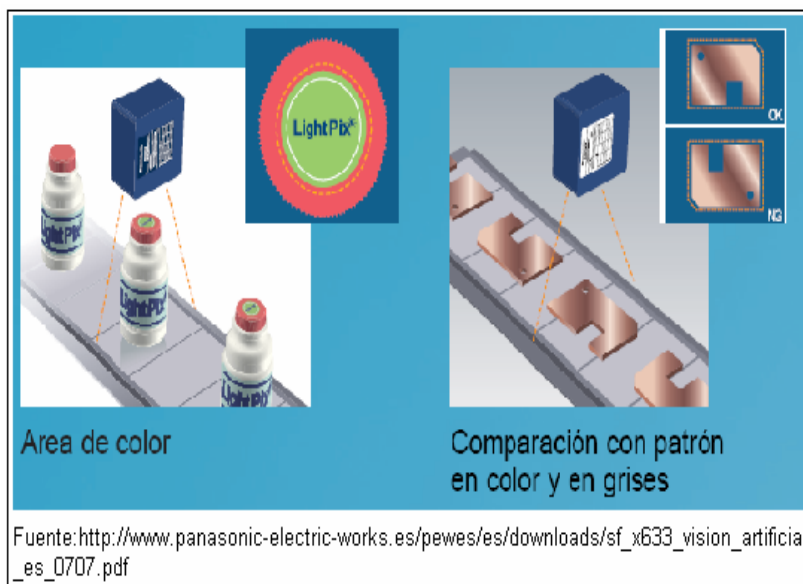


Figura 2.26 Funciones del LightPix AE20

### 2.5.2 Iris access 3000

Fue desarrollado por LG Electronics, su implementación es realizada con los estándares de la tecnología iclass de HID para el reconocimiento de iris.

Su implementación provee alto grado de seguridad donde el enfoque está en la protección de bienes o de recursos humanos.

A continuación se presenta la apariencia física del equipo desarrollado por la marca LG.



Figura 2.27 Iris Access 3000

### **2.5.3 SIMATIC VS100**

Este dispositivo es la mejor opción para el control de forma, tipo o posición de piezas menudas, su instalación es muy sencilla y el manejo se lo realiza de una manera amigable ya que no necesita ser un experto para comprender su funcionamiento. Los sensores de visión se integran con flexibilidad en el sistema de automatización gracias a interfaces estandarizadas.

El SIMATIC VS 100 es desarrollado por Siemens quien posee una gran variedad de dispositivos para visión artificial.

# CAPÍTULO 3

## 3. PLATAFORMA DE DESARROLLO CON DSPICS

En este capítulo se da una introducción a la programación con LabVIEW detallando sus elementos, así también el manejo de los componentes de IMAQ VISION USB haciendo capturas de imágenes con un ejercicio práctico, debido a su utilización en el proyecto de tesis, y al manejo de las cámaras USB. Además se da una explicación de los protocolos de comunicación con los periféricos y la interfaz que se utiliza para el manejo y control. También se describe las características de los dsPICS, su arquitectura, funcionamiento e interfaces dando algunas aplicaciones prácticas.

### 3.1 Programación en LabVIEW

#### 3.1.1 Introducción

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) fue creado por National Instruments en 1976, Es

una herramienta de programación gráfica destinado al desarrollo de aplicaciones que involucren sistemas de adquisición de datos, instrumentación, control, análisis y presentación de datos. LabVIEW da la facilidad de crear una interfaz de usuario que permite estar interactuando con el sistema, válido para cualquier clase de programador incluyendo a personas con pocos conocimientos en programación debido a su fácil programación gráfica ó lenguaje G para crear programas basados en diagramas de bloques llamados instrumentos virtuales o VI [17].

La instrumentación virtual se puede describir como un sistema de medición, precisión, análisis y control de señales físicas, con la ayuda de instrumentos virtuales desarrollados desde un computador.

Entre sus principales ventajas se puede citar:

- Es un lenguaje gráfico intuitivo por su facilidad de programación.
- Puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab/Simulink, .Net, DLL, AutoCAD.

- Facilidades en el manejo de interfaces de comunicación, como por ejemplo: Puerto Serie, Puerto Paralelo, USB, entre otros.
- Fácil integración con el hardware, por ejemplo: tarjetas de adquisición, medición y procesamiento de datos, incluyendo adquisición de imágenes.

### **3.1.2 Área de aplicación**

Su mayor aplicación radica en sistemas de medición, monitoreo de procesos y aplicaciones de control. Por ejemplo:

- Automatización de máquinas
- Sistemas de monitoreo de condiciones de máquina
- Visión artificial
- Adquisición y tratamientos de imágenes y datos
- Laboratorios de aprendizaje en universidades
- Procesos de control industrial
- Diseño y análisis de sistemas empujados como las FPGAs y Microprocesadores

- LabVIEW también es utilizado en el procesamiento digital de señales, procesamiento en tiempo real de aplicaciones biomédicas, automatización, manipulación de imágenes y audio, diseño de filtros digitales, generación de señales, etc.

### 3.1.3 Entorno de LabVIEW

Todos los VI tienen un panel frontal y un diagrama de bloques como se aprecia en la Figura 3.1.

Las paletas contienen las opciones que se emplean para crear y modificar los VI. A continuación se realizará una descripción de estos conceptos.

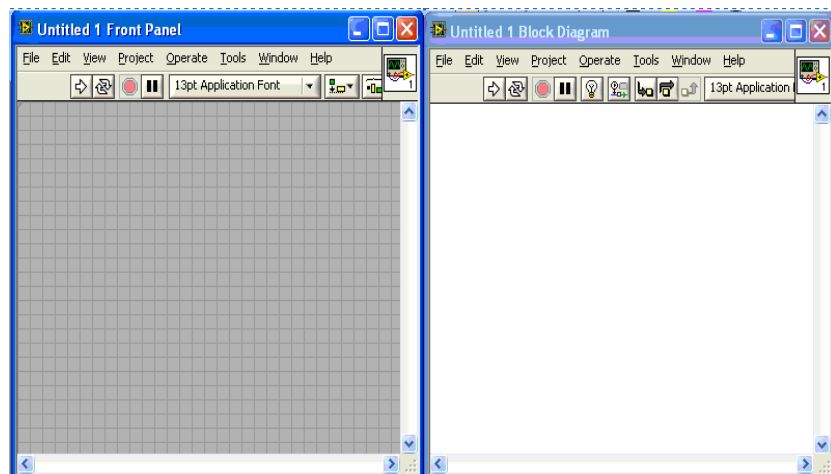


Figura 3.1 Panel Frontal y Diagrama de Bloques

### **3.1.3.1 Panel Frontal**

El panel frontal es la interfaz del usuario con el VI, donde el usuario interactúa con el código, aquí se definen los controles e indicadores. Los controles representan los dispositivos de entrada para ingresar datos desde el Mouse o el teclado como son: pulsadores, marcadores, potenciómetros y otros componentes de entradas. Los indicadores representan los dispositivos de salidas a mostrar, los más utilizados son: gráficos y luces .

### **3.1.3.2 Diagrama de Bloques**

El diagrama de bloques constituye el código fuente para el VI. Donde se define la funcionalidad del sistema realizando la implementación del programa del VI para controlar o realizar cualquier proceso creado en el panel frontal.

El diagrama de bloques contiene funciones y estructuras integradas en las librerías que incorpora LabVIEW. También está compuesto de nodos y conectores que son iconos ejecutables llevando datos entre nodos. Estos nodos son elementos de ejecución



los cuales consisten de funciones, estructuras y subVIs, realizando todas las funciones específicas necesarias para el VI, análogamente son como los operadores o librerías de funciones de los lenguajes convencionales.

Cuando se ingresa en el panel frontal un indicador o control, se crea un terminal en el diagrama de bloques correspondiente a ese indicador o control, los enlaces o conectores son los caminos para la comunicación entre los terminales.

LabVIEW utiliza una estandarización gráfica de colores para describir el tipo de dato con el que se está trabajando ya sean controles, indicadores o simplemente nodos, como se puede apreciar en la Figura 3.2 los datos de tipo numéricos son representados de color naranja, los enteros de color azul, los boléanos de color verde.

<b>TF</b>	Boolean	<b>U32</b>	Integer 32 bit (positivo)
<b>EXT</b>	Extended	<b>I8</b>	Integer 8 bit
<b>DBL</b>	Double	<b>EXT</b>	Numeric 7
<b>SGI</b>	Single	<b>CD8</b>	Numeric 8
<b>I32</b>	Integer 32 bit	<b>CSG</b>	Numeric 9
<b>I16</b>	Integer 16 bit		

Fuente: "AUTOMATIZACIÓN DEL PROCESO DEL LÍQUIDO EN LA ELABORACIÓN DE BEBIDAS GASEOSAS" Tesis de grado ESPOL, 2007

Figura 3.2 Estandarización de colores para los diferentes tipos de datos

### 3.1.3.3 Paletas

Proporcionan las herramientas necesarias para crear y modificar el panel frontal, el diagrama de bloques estableciendo propiedades como borrar, pegar, copiar, seleccionar a los objetos. Existen las siguientes paletas:

- Paleta de Herramienta

Contiene las herramientas necesarias para editar y depurar los objetos en el panel frontal como en el diagrama de bloques.

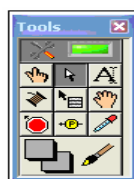


Figura 3.3 Paleta de herramientas

- Paleta de Controles

Contiene los controles e indicadores para crear la interfaz del VI con el usuario. Sólo se utiliza en el panel frontal.

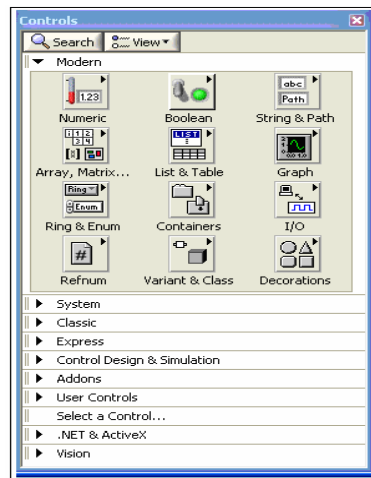


Figura 3.4 Paleta de controles

- Paleta de funciones

Contiene todos los elementos necesarios para la implementación del programa del VI como las funciones aritméticas, estructuras, arreglos y más; sólo se encuentra en el diagrama de bloques.

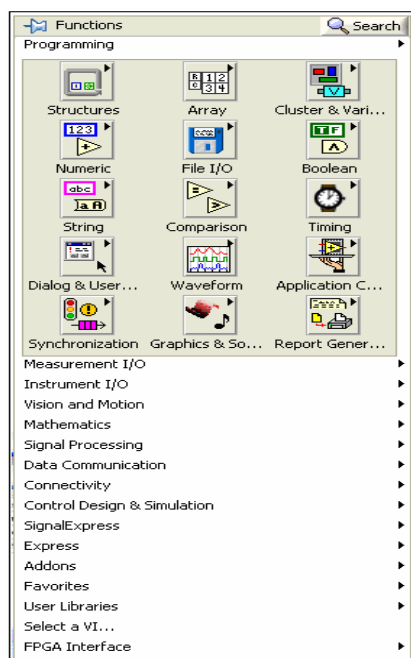


Figura 3.5 Paleta de funciones

### 3.2 Manejo de los componentes de la librería IMAQ VISION

NI-IMAQ es una extensa librería de funciones que se encuentra en la paleta de funciones dentro del diagrama de bloques. Permite adquirir imágenes continuamente en memoria, puede iniciar la adquisición de imágenes por evento, seleccionar una región de interés (ROI) para poder reducir la imagen que se está transfiriendo, entre otros. [18].

A continuación se muestra la forma en la que se accede a la librería de visión, que se encuentra en el diagrama de bloques.

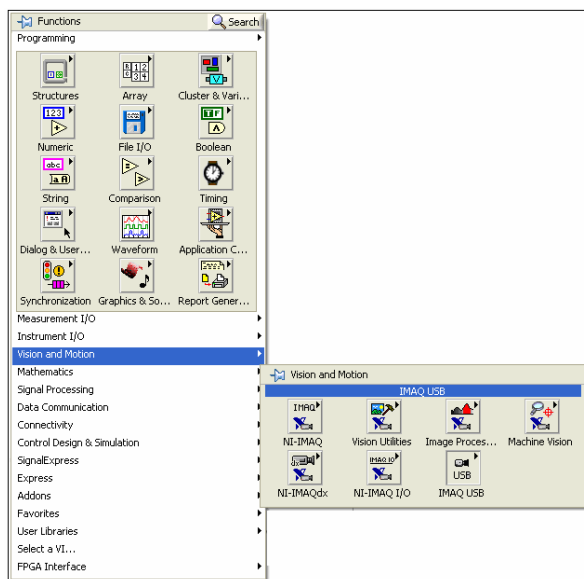


Figura 3.6 Pasos para seleccionar NI-IMAQ USB

Una de las herramientas que posee la librería IMAQ VISION para la utilización de cámaras digitales y adquirir imágenes es NI-IMAQ USB, se explica esta librería porque se utilizó para el desarrollo del proyecto.

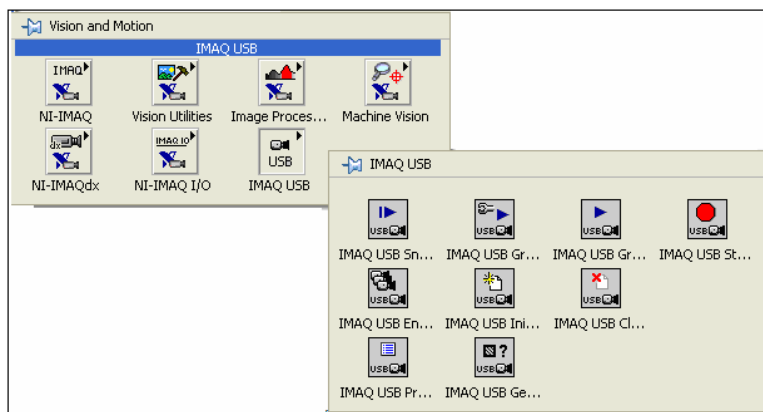


Figura 3.7 Funciones de NI-IMAQ USB

A continuación se dará una breve descripción de cada función para mejor entendimiento.

### 3.2.1 IMAQ USB Init:

Configura el dispositivo de visión, esto significa que abre una sesión para la cámara que se está utilizando para poder capturar imágenes.

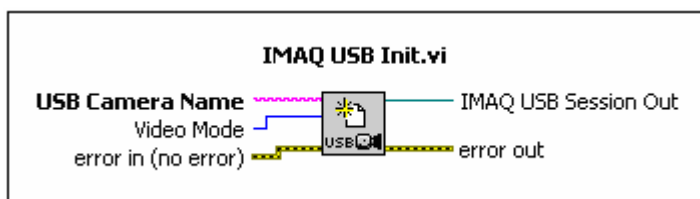


Figura 3.8 Icono de IMAQ USB Init

### 3.2.2 IMAQ USB Snap

Captura una única imagen cada vez que se ejecuta, identificando el dispositivo conectado que viene del bloque anterior; es bueno para aplicaciones a baja velocidad.

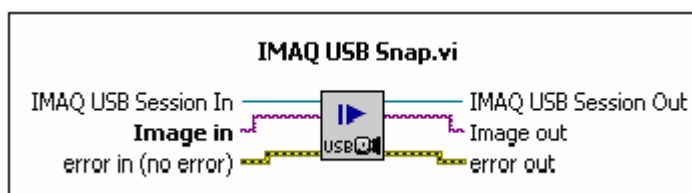


Figura 3.9 Icono de IMAQ USB Snap

### 3.2.3 IMAQ USB Grab Setup

Configura e inicia una grabación en modo continuo y las almacena en un buffer (toma de imágenes de alta velocidad). Sólo se lo puede llamar una vez.

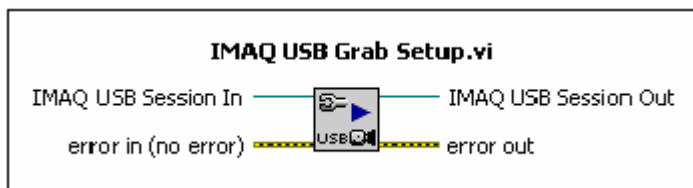


Figura 3.10 Icono de IMAQ USB Grab Setup

### 3.2.4 IMAQ USB Grab Acquire

Adquiere una imagen durante el modo continuo y la obtiene del buffer anterior en múltiples ocasiones, pero sólo de una cámara a la vez.

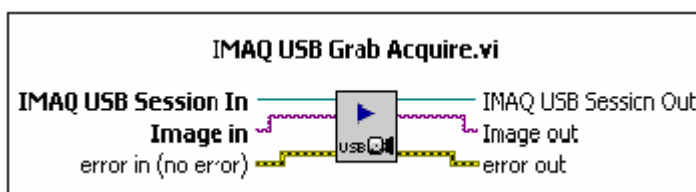


Figura 3.11 Icono de IMAQ USB Grab Acquire

### 3.2.5 IMAQ USB Close

Termina la adquisición de imágenes y cierra sesión a todos los recursos utilizados tanto de hardware como de software.

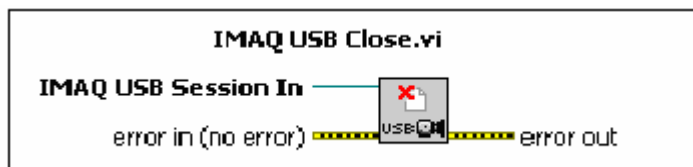



Figura 3.12 Icono de IMAQ USB Close

### 3.3 Captura de imágenes mediante LabVIEW

Antes de comenzar a realizar capturas, se debe de tener instalado el software de la cámara adquirida, debido a que LabVIEW debe trabajar con el nombre que reconoce Windows, en este ejemplo se utilizará la librería IMAQ VISION USB y una cámara digital marca Genius.

Para realizar la identificación de la cámara con la que se va a trabajar. Primero se crea un nuevo VI, a continuación utilizar la función IMAQ USB enumerate cameras.vi de la librería IMAQ VISION y crear un indicador a la salida de esta función dando clic derecho a USB cameras Name. Finalmente hacer clic en  y el programa le dará el nombre de su cámara, en este caso el nombre es i-look 111.

La Figura 3.13 muestra el VI y el resultado en el panel frontal.

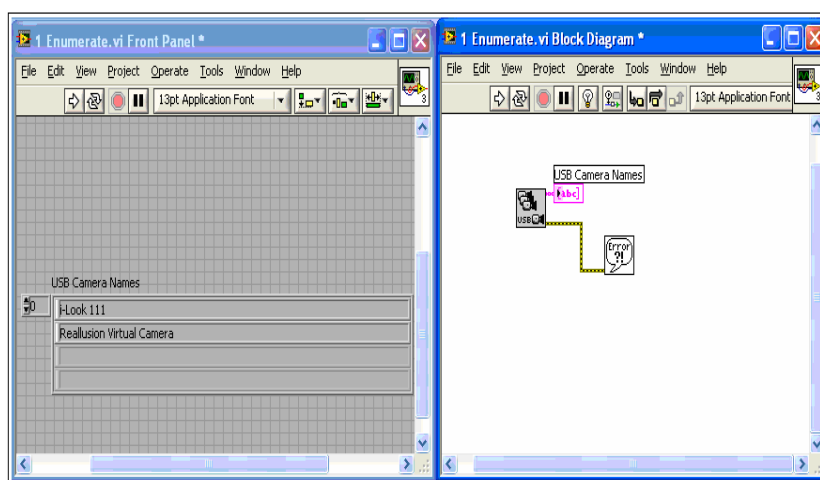


Figura 3.13 Identificación de la cámara



Una vez identificado el nombre de la cámara se procederá a adquirir la imagen, para lo cual se creará un nuevo VI, luego en la librería IMAQ VISION USB se arrastra la función IMAQ USB init que permitirá abrir la sesión con el nombre de la cámara descrito en el ejemplo anterior, se creará un control en USB camera name de esta función porque en este campo se pondrá el nombre de la cámara. Luego se necesita grabar continuamente la imagen por ello se llama a la función IMAQ USB grab setup y se conecta la sesión de salida de IMAQ USB init a la sesión de entrada de IMAQ USB grab setup, lo mismo se hace con el error out y con error in como se muestra en Figura 3.14

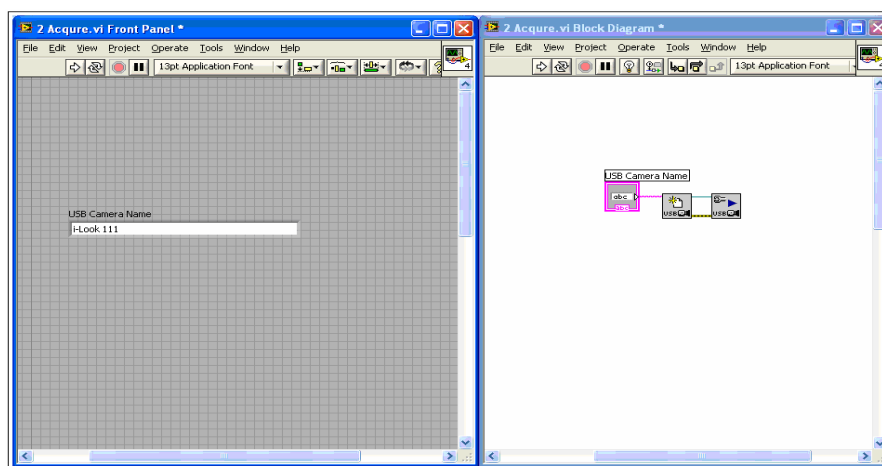


Figura 3.14 Diagrama de bloques de adquisición

Después, es necesario adquirir imágenes en modo continuo, para lo cual se necesita reservar temporalmente espacio de memoria; por ello se llama a la función IMAQ USB Grab adquiere y de IMAQ create

que se encuentra en la misma librería; Grab acquire se conecta luego de USB grab setup y se crea un indicador en image out para observar la imagen que se está adquiriendo, pero es necesario agregarle un lazo while que encuentra en la librería structures de este modo siempre se está llamando a la función para adquisición continua.

El lazo utilizado debe incluir una botonera de stop para el término del ciclo o de la toma de imágenes. New image de IMAQ create se conecta con image in de IMAQ grab setup y se crea una constante con un nombre en image name de esta función para reservar memoria. Todo lo mencionado anteriormente se muestra en el siguiente gráfico.

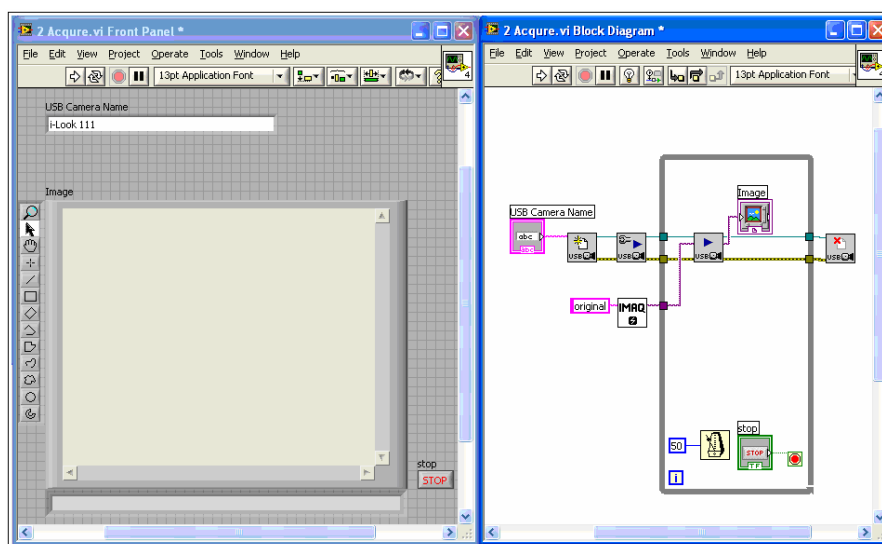


Figura 3.15 Programa para adquirir imágenes

Se puede notar que dentro del ciclo continuo se ha agregado un retardo para el refrescamiento de la cámara y de las tomas de

imágenes, al final del ciclo se añadió la función IMAQ USB close que cierra la sesión de la cámara.

### **3.4 Protocolos de comunicación con periféricos**

Debido al uso de los diferentes periféricos utilizados en el proyecto de tesis, se debe de conocer el conjunto de reglas que se especifican para el intercambio de datos durante la comunicación entre si y con labVIEW. A esto se la llama protocolo de comunicación.

Hay diferentes tipos de protocolos pero en este tema se va a explicar la comunicación serial por el puerto RS-232 y el puerto USB debido a que se utilizan para el desarrollo de este proyecto.

#### **3.4.1 Comunicación serial**

La comunicación serie consiste en la transmisión y recepción de pulsos digitales que representan un dato binario a una misma velocidad.

La interfaz física para esta comunicación es el puerto serie donde la transmisión es de bit a bit de forma ordenada. Se ha usado para conectar las computadoras a dispositivos externos como: Mouse, teclado y otros periféricos.

Los bits se representan por códigos que se transmiten en un período de tiempo fijo. Este tiempo se conoce como período baudios.

Si se tienen que comunicar dos dispositivos a una velocidad de 2400 bits por segundo (Baudios), el receptor capturará las señales de voltaje que le está enviando el transmisor, y por cada  $1/2400$  de un segundo, interpretará dicha señal como un nuevo bit de datos. Si el voltaje es alto (comúnmente 5V), interpretará el dato como 1, y si el voltaje es bajo (0 voltios), interpretará el dato como 0. Así el receptor puede obtener un mensaje interpretando una secuencia de datos.

#### **3.4.1.1 RS-232**

Generalmente se utiliza el protocolo RS-232 para estas comunicaciones, es una interfaz que define un estándar para el intercambio serie de estos datos utilizando un cableado simple desde 3 hilos hasta 25 hilos entre un transmisor y un receptor a una distancia máxima de 15 metros.

A inicios, el RS-232 original tenía un conector tipo D de 25 pines de los cuales muchos no se utilizaban por lo que se incorporó un conector mas pequeño de 9

pinos que actualmente se utiliza y se lo conoce como DB9<sup>7</sup>.

A continuación se muestra de manera general el conector DB9 y el DB25.

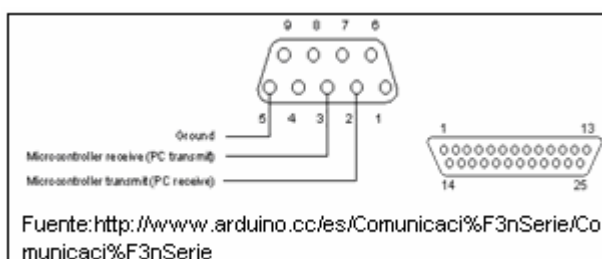


Figura 3.16 Puerto serie RS-232 de 9 y 25 pines macho.

### 3.4.1.2 USB

Universal Serial Bus (Bus universal en serie) es un puerto que sirve para conectar la computadora a diferentes periféricos a una importante velocidad de transferencia [19].

La Figura 3.17 muestra la representación simbólica de los dispositivos USB.

---

<sup>7</sup> Anexo A

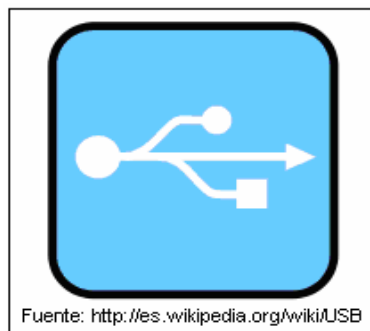


Figura 3.17 Símbolo USB

La ventaja del USB ante el puerto serie y paralelo es la velocidad debido a que el USB trabaja a 12Mbps para el USB 1.1 y 480Mbps para el USB 2.0 esto es más o menos de 3 a 5 veces más rápido que un dispositivo del puerto paralelo y de 20 a 40 veces más rápido que un dispositivo del puerto serie porque trabajan a una velocidad de transferencia inferior a 1Mbps. También por su capacidad de que con sólo conectar el dispositivo al CPU, éste sea capaz de interpretar la información almacenada y reproducirla inmediatamente porque el servidor le agrega el software necesario. Es decir, que el computador y el dispositivo hablan el mismo idioma y se entienden entre sí. Sólo es necesario que el sistema operativo lleve incluido el correspondiente controlador o driver, esto ya es posible para la mayoría de ellos, sobre todo

si se dispone de un sistema operativo como por ejemplo Windows XP, de lo contrario el driver le será solicitado al usuario [20].

Otra característica de este sistema es que permite conectar y desconectar los diferentes dispositivos sin necesidad de reiniciar el equipo.

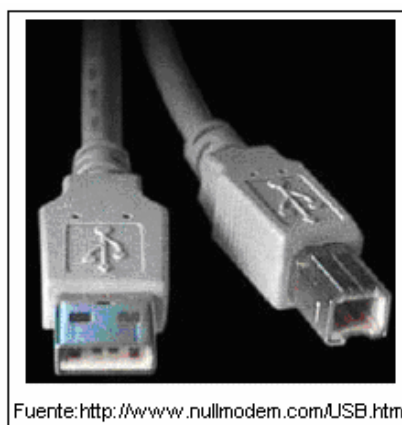


Figura 3.18 Conectores USB, de izquierda a derecha: USB1.1 y USB2.0

### 3.4.1.3 Periféricos

Los USB soportan periféricos de baja y alta velocidad siendo esta para la transmisión de datos de 1.5Mbps y 12Mbps; consiguiendo una utilización más eficiente de sus recursos. Los periféricos de baja velocidad son: Mouse, teclado y otros periféricos. Los de alta velocidad que requieren más recursos del sistema son

los monitores, impresoras, equipos de audio, entre otros que necesitan velocidades mayores para la transmisión de datos.

El siguiente grafico muestra los periféricos mas comunes que utilizan conexión USB.

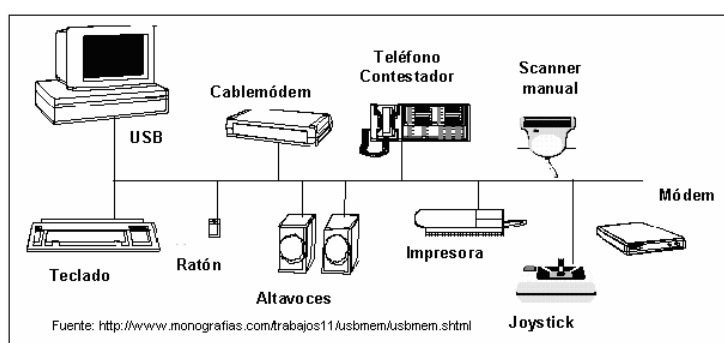


Figura 3.19 Periféricos USB conectados a la PC.

### 3.5 Interfaces de manejo y control

En la implementación del proyecto que cuenta con dos interfaces, uno es entre LabVIEW y la banda transportadora; el otro es entre LabVIEW y el brazo robótico, a continuación se va abordar el tema de interfaces.

Para trabajar con una interfaz entre LabVIEW y el control de la banda transportadora, se utilizó una placa desarrollada por compañeros del tópicos cuya tesis es “Diseño de dos bandas transportadoras de 100x10cm y dos de 30x10cm con motores controlados con DSPICS”



para la adquisición, procesamiento y envío de datos; que cuenta con un dsPIC30F4011 de marca microchip y que su código fue realizado en MikroBasic para dsPIC que es un compilador para la programación de microcontroladores.

El propósito de esta interfaz es de controlar la velocidad de los cuatro motores por medio de los PWM que cuenta el dsPIC, el cual sensa la velocidad por medio de los 4 sensores reflectores que entregan la velocidad de la banda; y un sensor reflector para la detección de las botellas; además administra el panel de control de la banda transportadora para su funcionamiento manual. Todo esto lo hace por comunicación serial debido a su facilidad de transmisión. Por último, La tarjeta posee resistencias lineales para calibrar los sensores para obtener una mejor lectura de la velocidad.

Por otra parte, la segunda tarjeta se diseñó con el propósito de controlar el brazo robótico y del empuja frascos por medio de LabVIEW, enviando las posiciones deseadas a diferentes velocidades, con esto se logra el control de los 5 servo motores del brazo y del motor del empuja frascos con el fin de tener una mayor precisión.

El tipo de comunicación que utiliza esta tarjeta es el serial y posee un dsPIC30F4012 que es el cerebro de la misma que se programó en

MiKroBasic para dsPIC, el programa realizado en LabVIEW es muy fácil de manejar debido a que sólo se envían las posiciones de cada motor. Si desean tener mayor información sobre la tarjeta, pueden consultar la tesis de “Diseño de brazo robótico y diseño de brazo empuja frascos controlados por Labview y DSPICS”.

### **3.6 Características generales de los dsPICS**

En este proyecto se hace referencia a los dsPIC30F que es la primera familia de microchip que ha salido al mercado y sobre estos se ha sacado información relevante para su explicación.

El dsPIC es un Controlador Digital de Señales (DSC) basado en características más potentes que los microcontroladores de 16 bits al que se les han incorporado las principales prestaciones de los DSP para resolver óptimamente algoritmos en el tratamiento de imágenes y el sonido, ecuaciones matemáticas, etc; permitiendo su aplicación en el campo del procesamiento de las señales analógicas y digitales.

Un DSC posee lo mismo que un microcontrolador: potencia, velocidad, manejo flexible de interrupciones, amplio campo de funciones, opciones de reloj, protección o reset programable, perro guardián, seguridad del código, simulación en tiempo real, y su precio es similar al de los microcontroladores, en la siguiente figura se

puede ver el rendimiento vs. Precio entre los DSC y microcontroladores.

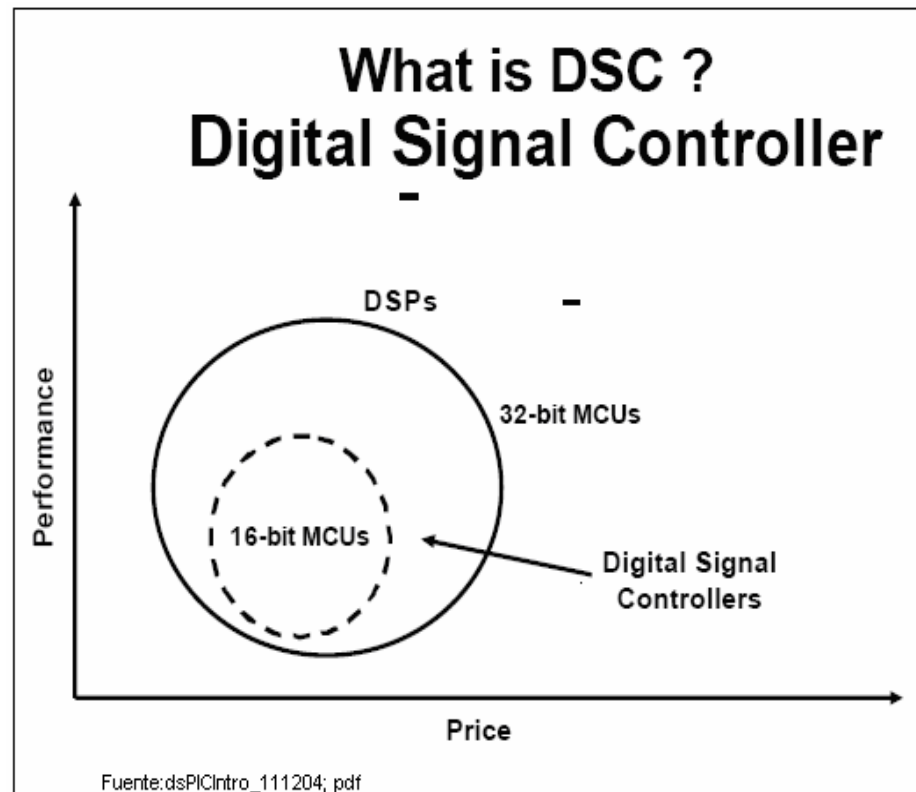


Figura 3.20 Rendimiento Vs. Precio entre los DSC y microcontroladores

Algunas características de los DSP que se han incorporado en la arquitectura básica de los dsPIC30F:

- Multiplicador de 17 x 17 bits
- Registro de desplazamiento de 40 bits
- 2 acumuladores de 40 bits
- Acceso simultáneo de dos operandos
- Bucle de registro de trabajo

- Juego flexible de interrupciones
- Perro guardián
- Optimización para programación en C
- Poseen hasta 6 PWM.

### 3.7 Arquitectura, funcionamiento e interfaces

Presentando las características de los dsPIC, se pasará a la descripción de la arquitectura de estos que es muy similar a la de un microcontrolador.

#### 3.7.1 Arquitectura Harvard

Con dos memorias independientes, modificada (núcleos de CPU de 16/24bits una para datos que es llamada memoria de datos y otra para instrucciones que es llamada memoria de programa). Un esquema general se presenta en la figura 3.21.

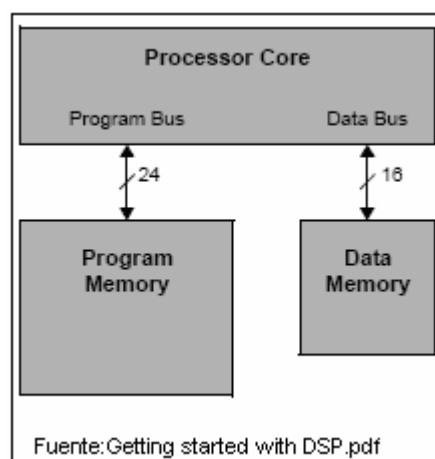


Figura 3.21 Memoria de programa y datos

Este diseño mejora la eficiencia del conjunto de instrucciones permitiendo mayor rapidez de procesamiento debido a que utiliza dos buses diferentes para cada memoria, y las ventajas de esta arquitectura es que puede ir a la siguiente instrucción de memoria del programa mientras se ejecuta la instrucción actual permitiendo acceder a los datos de la memoria RAM, y el tiempo de acceso a las instrucciones puede superponerse con el de los datos logrando una mayor velocidad de operación.

### **3.7.2 Motor DSP**

Es un conjunto de recursos matemáticos que le permite al procesador realizar operaciones a una gran velocidad, sus características son: multiplicador rápido de 17x17 bits, registro de desplazamiento de 40 bits de longitud, sumador/restador de 40 bits, módulo de redondeo y saturación lógica, todo esto es en un ciclo<sup>8</sup>.

---

<sup>8</sup> Véase en el anexo B

Las operaciones de multiplicación y desplazamiento utilizan el motor DSP para la obtención de sus resultados, utilizando un bus de datos X para la lectura y escritura en dichas operaciones.

Pueden realizar una división fraccionaria con signo de 16/16 bits, así como división de enteros de 32/16 bits y de 16/16 bits con signo y sin signo.

### **3.7.3 Memoria y contador del programa**

La memoria de programa posee un espacio de 4M con posiciones de 24 bits cada una, el contador del programa es de 24 bits e incrementa por cada 2 instrucciones. El programa de memoria contiene la ubicación del vector de Reset, la tabla del vector de interrupciones, la memoria del programa de usuario con una capacidad máxima de 48Kbytes de 24 posiciones cada una, datos EEPROM de 4KBytes y la configuración de memoria<sup>9</sup>.

---

<sup>9</sup> Véase en el anexo C

### **3.7.4 Memoria de datos**

Esta memoria es de 64Kbytes y es tratado como un espacio de memoria lineal para la mayoría de las instrucciones que consta de posiciones de memoria de 16bits en las que se agrupan 2 bytes, cuando determinadas instrucciones son usadas como las DSP, la memoria se divide en dos bloques o espacios llamados memoria de datos X i Y; los datos pueden ser leídos simultáneamente de los bloques X i Y; accediendo independientemente a las instrucciones DSP.

El primer bloque de datos es de 2-KByte que se les asigna el registro de funciones especiales. El SFR son los registros de control y estado para el núcleo y periféricos en funciones de los dispositivos dsPIC.

Después de los SFR, viene la implementación de la RAM de datos que es de 8Kbytes. La memoria RAM es una memoria de propósito general que puede ser usada para el almacenamiento de datos.

### **3.7.5 Registro de trabajo**

Estos procesadores disponen de un banco de 16 registros de trabajo que van desde W0 a W15 y cada uno es de 16 bits.



Figura 3.22 Registro de trabajo w

### 3.7.6 Grupo de instrucciones

El dsPIC tiene 2 clases de instrucciones: MCU y DSP. Estas dos clases de instrucciones se integran perfectamente en la arquitectura y se llevan a cabo desde una única unidad de ejecución. El conjunto de instrucciones incluye muchos modos de abordar y fue diseñado para una óptima eficiencia.

También tiene la capacidad de ejecutar instrucciones de 3 operaciones en un ciclo, por ejemplo: la suma  $A+B=C$ . El resultado puede almacenarse en un banco de registros W o en una posición de memoria de datos.



### 3.7.7 Interrupciones

El Dspic30F tiene una tabla de vector de interrupciones<sup>10</sup> (IVT). Cada fuente de interrupción tiene su propio vector y puede ser asignada dinámicamente en uno de los siete niveles de prioridad. La interrupción de entrada y retorno de latencias son fijos, proporcionando un tiempo determinístico para aplicaciones de tiempo real. El IVT contiene 62 vectores formando un grupo de 8 interrupciones no enmascarables (siempre habilitado) y 54 vectores de interrupciones. Cada interrupción salta una posición.

### 3.7.8 Periféricos

Los dsPIC están disponibles con una amplia gama de periféricos para adaptarse a una diversa variedad de aplicaciones. Los principales periféricos incluyen:

- Puertos de E/S
- Temporizadores

---

<sup>10</sup> Ver anexo D

- Input Capture
- Salida comparadora/PWM.
- Control de motor PWM.
- Convertidor A/D de 10-bit o 12 bit.
- UART.
- SPI.
- I2C.
- Controller Area Network.

### **3.8 Aplicaciones prácticas de los dsPICs**

Detallada la arquitectura de los dsPICs, se proporcionará dos ejemplos para conseguir una familiarización en este campo.

Se va a explicar el sistema de desarrollo PIC School de ingeniería de microsistemas programados, que se muestra en la figura 3.23.

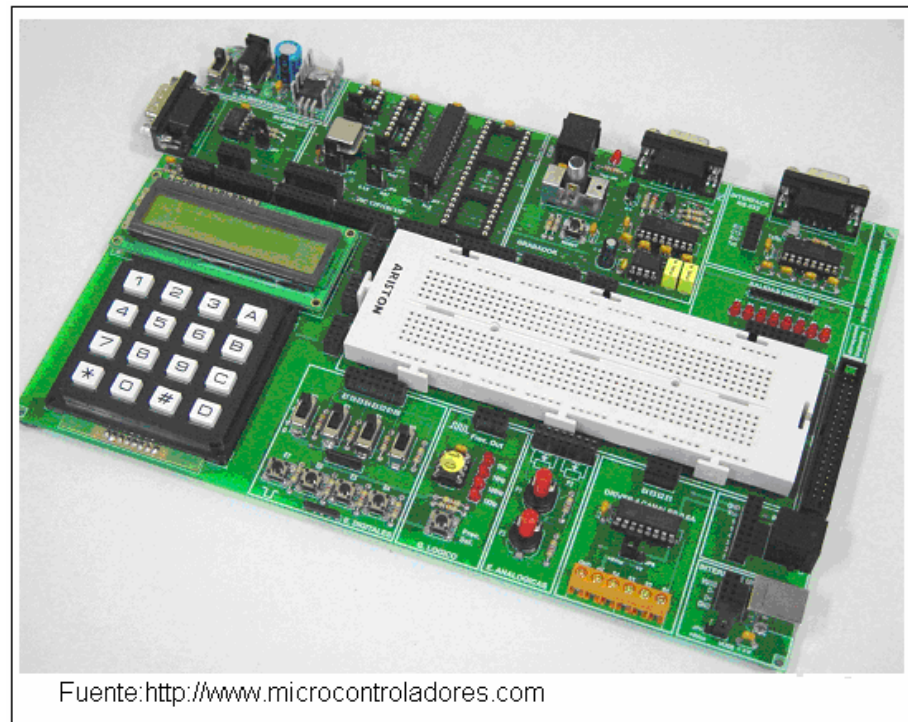


Figura 3.23 Sistema de desarrollo PIC school

El sistema de desarrollo PIC School está diseñado para soportar la grabación e implementación de sistemas basados en toda la gama de familias de microcontroladores de Microchip. Dada la incorporación de comunicaciones, recursos y sistemas de grabación, para llevar a cabo esta explicación se utiliza un dsPIC30F4013, un adaptador para dicho módulo, un motor de corriente continua y un controlador de LCD [23].

Se simulará un bólido de Formula 1, el cual se basa en la rapidez y en la precisión. Por eso en este ejemplo experimental se ha tratado de aprovechar determinadas características de los dsPIC para simular el control del bólido para este tipo de competencia.

El sistema tiene una pantalla LCD que simula la pantalla del volante del bólido, sirve para visualizar la temperatura del motor (potenciómetro) y la velocidad del mismo, para ello se utiliza la comunicación UART entre el dsPIC y la pantalla, los cambios de velocidad del bólido se realizan por medio de interrupciones, se simula el motor con uno de corriente continua utilizando el módulo comparador del dsPIC haciendo que varíe la velocidad acorde a las variables que intervienen.

Para el control de las variables de este proyecto, se aprovechó la velocidad de cálculo del dsPIC, su potente conversor analógico digital y sus periféricos de comunicaciones, que se ejecuta con la rapidez necesaria en aplicaciones de esta índole.

El siguiente ejemplo es de un diseño y construcción de un robot móvil que pueda desplazarse dentro de un laberinto en base a un algoritmo de control de posición, realizado por los alumnos de la Universidad Politécnica Nacional.

Este robot es capaz de explorar áreas desconocidas de tal manera que podrá ir creando internamente mapas de entorno del camino que ha recorrido y en base a esa información resolver un laberinto.

La implementación se basa en el control de motores, acondicionamiento y manejo de sensores que recogen información

del medio en que se desenvuelve el robot, utilizando un dsPIC30F3010 para el desplazamiento del robot.

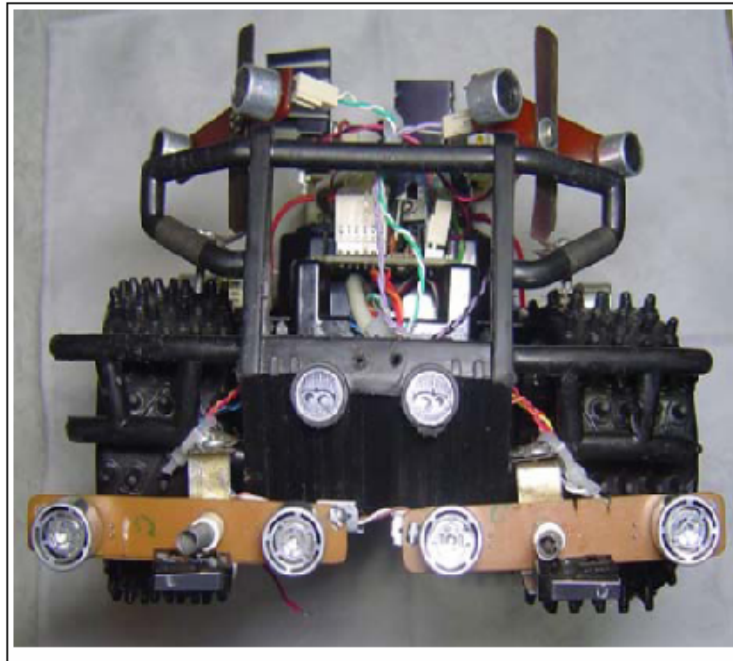


Figura 3.24 Imagen del robot móvil

La configuración del robot es de dos ruedas con tracción trasera comandado por un motor de corriente continua y dos ruedas con dirección delantera comandado por un servomotor, además de la utilización de sensores ultrasónicos constituidos por un par transmisor-receptor para el reconocimiento de obstáculos como son las paredes frontales y laterales. El programa fue realizado en lenguaje ensamblador, utilizando el software C30 que incluye el compilador MPLAB IDE v7.2.

# CAPÍTULO 4

## 4. DISEÑO E IMPLEMENTACIÓN

En este capítulo se da los requerimientos de implementación detallando el procesador, el software, entre otros; además se explica las consideraciones de diseño del programa realizado debido a las restricciones que se dan en las etiquetas, por ejemplo: la forma, así también se expone el diagrama de flujo del control en LabVIEW y el código realizado en el mismo para la captura de imágenes con el diagrama de bloques. Además se explica la implementación física como es la selección de la luz y de una cabina para que no permita el paso de la luz externa. También se describen los costos de implementación del proyecto.

## **4.1 Requerimiento de implementación**

Para que la aplicación desarrollada en LabVIEW funcione correctamente, se debe de seguir los siguientes requerimientos necesarios:

### **4.1.1 Requerimientos del computador**

La computadora es la unidad principal del sistema, puesto que ejecuta el programa que controla todo el proceso. El programa que se desarrolló funciona bajo Windows XP. El cuál tiene los siguientes requisitos:

- Pentium 4 2.46 GHz, Service Pack 2
- Mínimo 256MB de RAM
- Monitor para visualizar el proceso
- La computadora debe de estar provista de puertos USB para conectar la cámara Web.

### **4.1.2 Requerimientos del Software**

El software utilizado es LabVIEW 8.2 debido a que permite diseñar las pantallas que observará el usuario y a través de las cuales interaccionará con el proceso a controlar, además de

diseñar diferentes subrutinas de control de manera ordenada y simplificada que pueden ser llamadas del programa principal, así también, que la ESPOL tiene licencia para su uso y por la familiarización en el Tópico.

#### **4.1.3 Requerimientos de la cámara**

Se parte del hecho de que una cámara es un transductor óptico que captura las imágenes y las convierte en una señal eléctrica, para luego con los demás elementos del sistema de adquisición y un PC se procesa y analiza. La cámara utilizada es la e-Messenger 112 de la marca Genius, sus características son:

- Resolución 352x288
- Capacidad de transmisión de imágenes: 30fps
- Ángulo de visión de 52°
- Interfaz: USB
- Sensor de imágenes CMOS

Cabe indicar que las CMOS son pocos sensibles a la luz, de menor calidad en resolución pero más baratos que otros tipos. Para la implementación se eligió ésta cámara principalmente



por su costo, debido a que una cámara industrial cuesta mínimo \$ 500, además de esto, es un proyecto de tipo académico debido a que no se están considerando las variables reales de un proceso en una industria como es el caso de la velocidad de procesamiento, el desgaste de los motores y entre otros.

#### **4.1.4 Requerimientos de iluminación**

Se ha considerado un agente externo como es la fuente luz, que es un factor importante en el acondicionamiento de la imagen. De esta manera se optimizará y mejorará el rendimiento en su procesamiento.

Las cámaras capturan la luz reflejada de los objetos. El propósito de la iluminación es controlar la forma en que la cámara aprecia el objeto. Una iluminación adecuada permitirá emplear menos filtros de software en la imagen y por tanto aumentar la velocidad de proceso en esa aplicación.

En este caso se utiliza la iluminación por fluorescente ubicado detrás de la cámara, reduciendo las sombras y de las rayas que se generan, es indicado para este tipo de aplicación así detecta los diferentes colores, caracteres y de cualquier objeto con un cambio de color.

## 4.2 Consideraciones de diseño

Entendido los requerimientos de la implementación. Se trata de montar un sistema de visión artificial para la detección de frascos sin etiqueta, en el cual los frascos viajarán por una banda transportadora a una velocidad determinada por consideración de las características de sus motores y de la cámara que va a captar las imágenes.

El sistema debe de ser capaz de detectar la botella por medio de un sensor reflector que va a estar montado sobre la banda a una distancia cercana al área de análisis, el cual sensa la tapa de la botella y detendrá la banda. Además se utiliza el empuja frascos para que al momento de detener la banda, la botella no se pase del área de interés y así poder analizarla. El trabajo del empuja frascos además de parar la botella, es de botarla si es que el resultado del análisis lo requiere..

Otro elemento que se utiliza es el brazo robótico, que girará la botella 360° en pasos de 18°, con esto se logra analizar toda la botella y obtener mejores resultados.

El formato de la etiqueta es propuesto por el director de tesis, ésta tiene los bordes de color azul debido a que el sistema en primera detecta un cambio de color en una región de interés, en su interior tiene el logo de 50 años ESPOL.



Figura 4.1 Formato de la etiqueta

A continuación, se muestra una imagen del frasco con la etiqueta en blanco y negro, que va ser el modelo para la comparación y verificación de las demás botellas:



Figura 4.2 Formato de la Botella

Cabe resaltar que para la detección, debe de ser el mismo formato de la etiqueta puesto que se ha trabajado con estos parámetros para el logro del análisis, y si se requiere diseñar otro formato, se debe considerar el cambio en el parámetro del software.

### **4.3 Diagrama de flujo**

Para poder realizar el diagrama de flujo, se comenzó con etapas muy generalizadas como lo son las que se incluyen en el diagrama de bloques en el cual se plantea la forma cómo funcionará la aplicación de detección de frascos sin etiqueta.

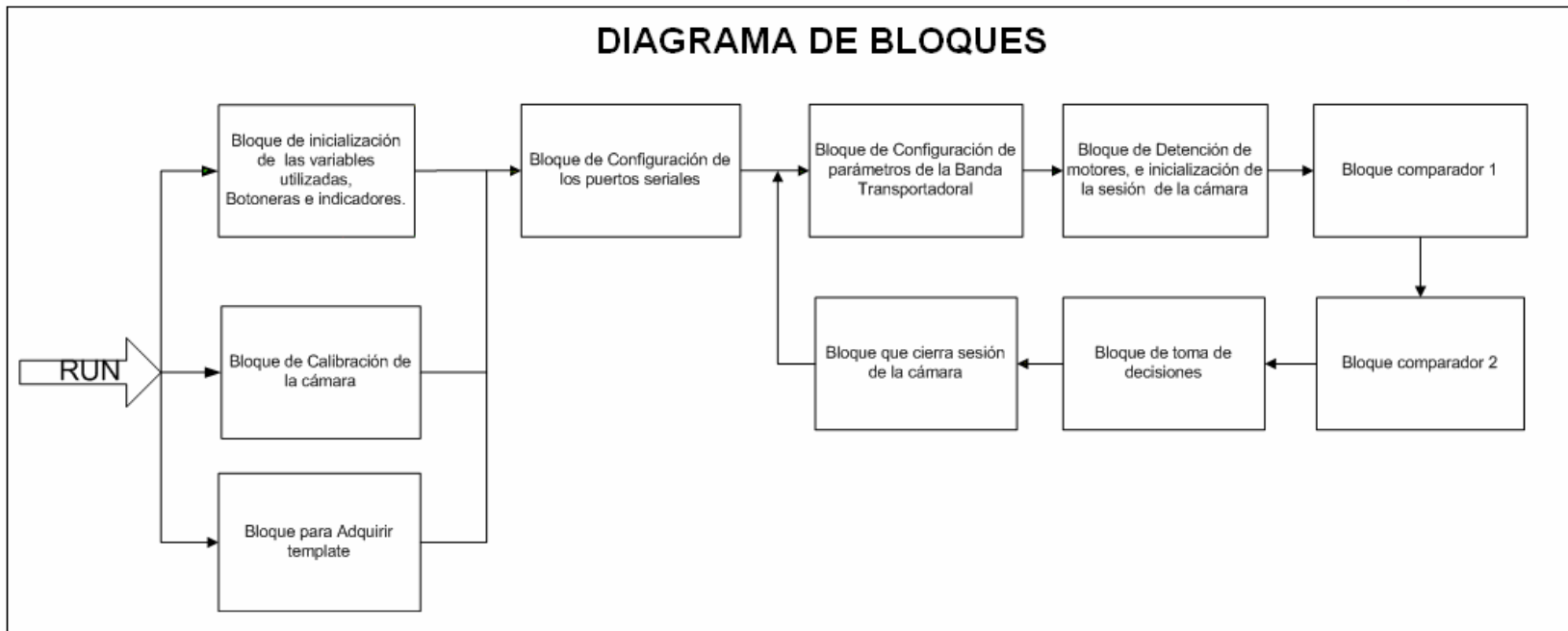


Figura 4.3 Diagrama de bloques de la aplicación

Ahora se presenta el diagrama de flujo acoplado con el funcionamiento de la banda transportadora y el brazo robótico, posterior a esto se da una explicación para su mayor comprensión.

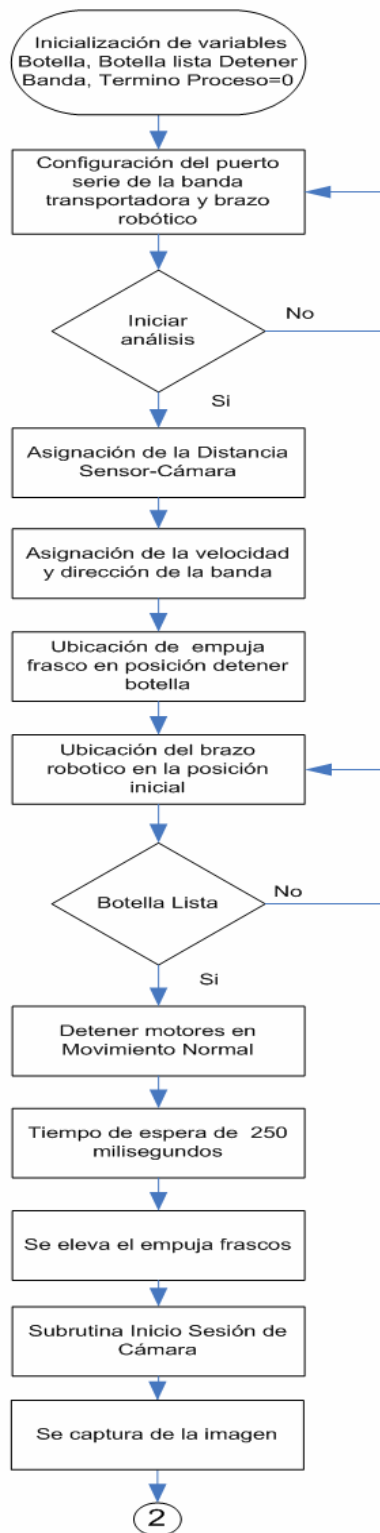


Figura 4.4 Diagrama de flujo parte 1

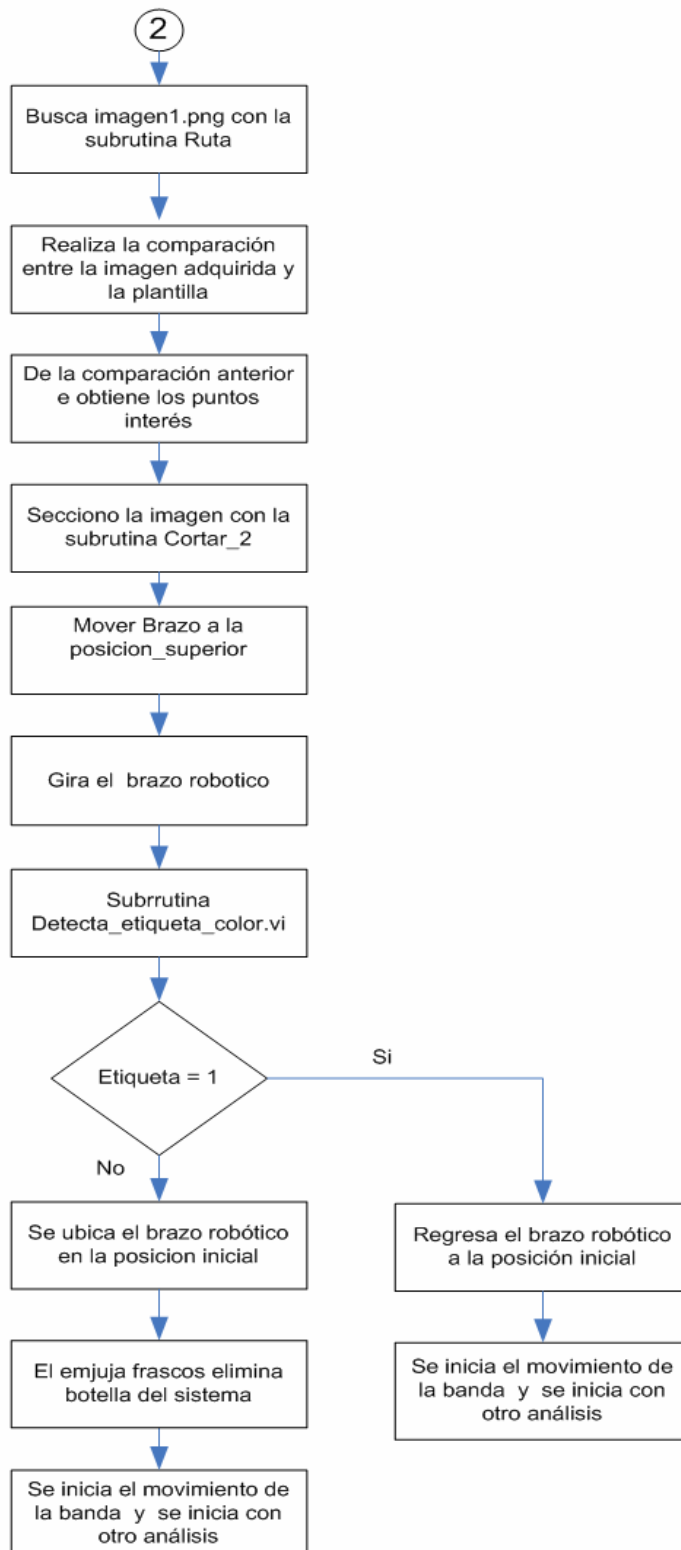


Figura 4.5 Diagrama de flujo parte 2

La aplicación contiene un panel frontal donde se puede seleccionar la acción a realizar, posee una pantalla para observar la imagen que se está capturando y algunas botoneras e indicadores para los diferentes procesos.

La figura 4.6 muestra el panel frontal de la aplicación desarrollada.

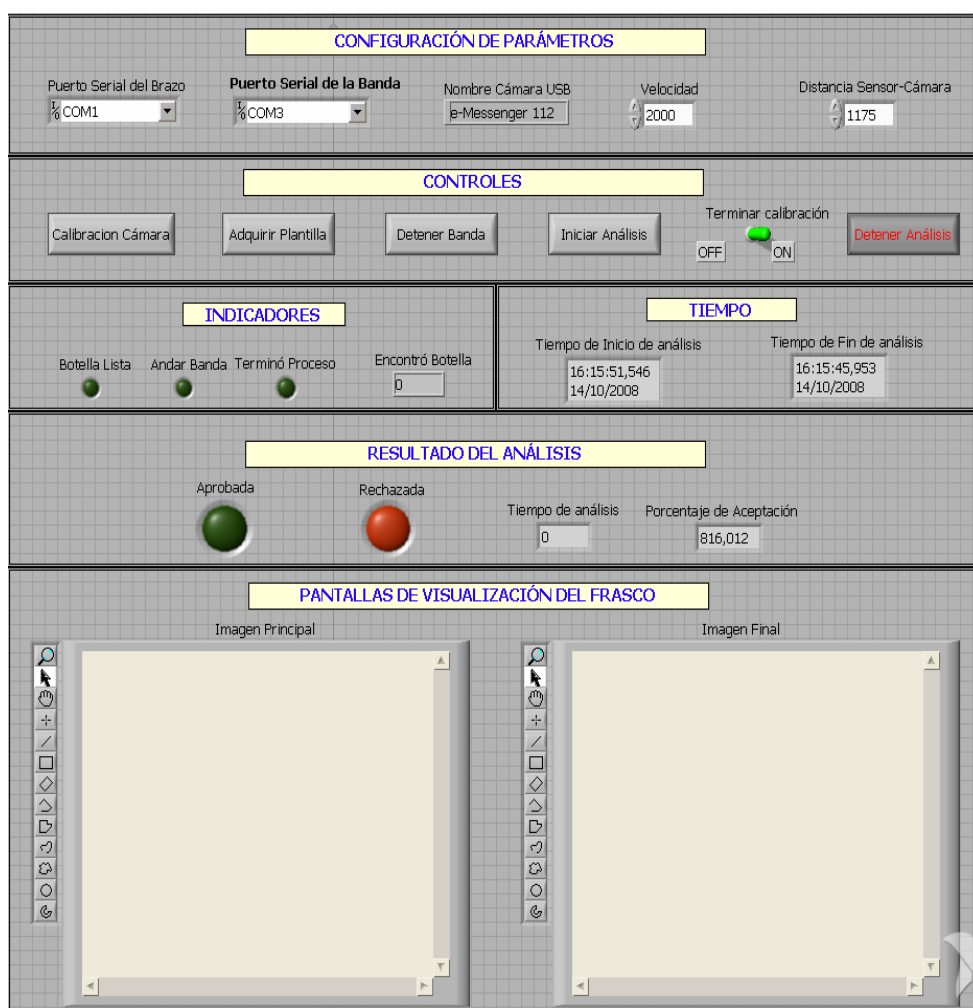


Figura 4.6 Panel frontal



El primer paso que se debe realizar es calibrar la cámara para poder enfocar el área de interés y ajustar el lente de la misma. Además de esto, se deben adquirir las plantillas tanto de la botella como de la etiqueta porque para realizar los análisis se debe tener alguna imagen para poder comparar, y de estas imágenes depende la calidad de reconocimiento y así tener un mayor porcentaje de aceptación.

Luego de los pasos anteriores, se presiona la botonera Iniciar análisis para que comience a correr el programa inicializando todas las variables que el proceso utiliza, se deshabilitan las botoneras: Calibración cámara, Adquirir plantilla e iniciar análisis para evitar interrupciones no deseadas en el proceso de análisis.

Además, se configuran los puertos de comunicación tanto de la banda transportadora como del brazo robótico y empuja frascos, se debe tener en cuenta que el brazo robótico y el Empuja frasco trabajan con el mismo puerto de comunicación (COM1). Esto es debido a que los dos son controlados con la misma tarjeta que fue desarrollada en el tópico **“Plataforma para el desarrollo de proyectos con dsPICs y visión Robótica con LabVIEW”**.

Un parámetro que se debe ingresar, es la distancia entre el sensor de botellas y la cámara porque de este depende la precisión al

momento de detener la botella en el área de interés, también se configura la velocidad de la banda transportadora a la que se desea desplazar las botellas. Se debe indicar que la banda transportadora tiene su propia tarjeta que la controla, ésta fue desarrollada por compañeros con el tema: "Diseño de dos Bandas Transportadoras de 100x10cm y dos de 30x10cm con Motores Controlados con DSPICS". Una vez terminada las configuraciones, se da paso al movimiento de la banda transportadora; el brazo robótico y el empuja frascos se ubican en la posición de reposo.

A medida que se desplacen las botellas, el sensor estará funcionando para detectar la tapa de las botellas, si el sensor detectó alguna de ellas, éste detendrá la banda con la botella en el área de interés, levantará el empuja frascos y dará inicio a la sesión de la cámara para adquirir la imagen de la botella detectada.

Con esta imagen, se compara con la plantilla adquirida al inicio del programa, si son parecidas, un indicador se encenderá mostrando que detectó la botella, luego el brazo robótico hará girar la botella 360° a pasos de 18° para detectar si tiene la etiqueta correcta; si la detecta, el programa envía un pulso de aprobada, caso contrario el empuja frasco se activa botando la botella analizada. Luego de ser analizada la botella, el programa vuelve a mover la banda

transportadora para efectuar el siguiente análisis volviendo a repetir los pasos descrito anteriormente.

#### 4.4 Código realizado en LabVIEW para captura de imágenes

Después de tener clara la idea del proceso, se procedió a crear subrutinas que ayuden a modular la aplicación creada, a continuación se muestra las subrutinas creadas y el código realizado en labVIEW.

##### 4.4.1 SubVIs creados

Con lo descrito anteriormente, se procederá a detallar los sub-programas utilizados en LabVIEW con el propósito de realizar un programa modular.

##### 4.4.1.1 Inicio\_sesion.vi

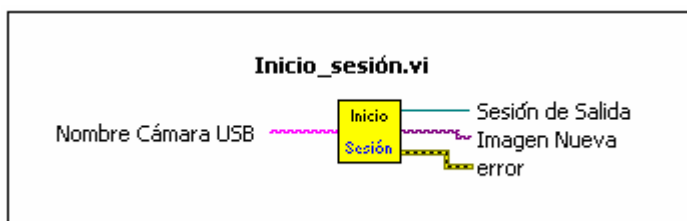


Figura 4.7 Icono de Inicio sesión

Inicia la sesión de la cámara, configurando los parámetros del mismo. Lo único que se le envía es el nombre con el que se reconoce en Windows.

#### 4.4.1.2 Ruta.vi

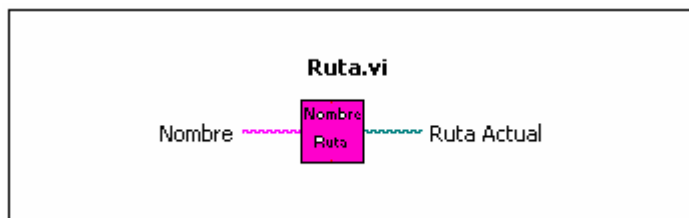


Figura 4.8 Icono de ruta

Busca el archivo con el nombre que se le envía en *nombre*, retornando la ruta de acceso de éste.

#### 4.4.1.3 Detecta\_Botella.vi

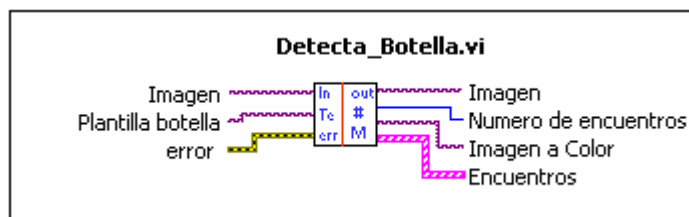


Figura 4.9 Icono detecta\_botella

Busca una similitud del frasco definida con *plantilla botella* con la imagen adquirida; esta subrutina retorna el área donde se encuentra el patrón por medio de *Imagen*, *número de encuentros* devuelve el número de imágenes encontradas, así también retorna las coordenadas de la posición en *Encuentros*. Por medio de esta subrutina se detecta toda la botella para luego efectuar cortes y análisis posteriores.

#### 4.4.1.4 Punto\_interés.vi

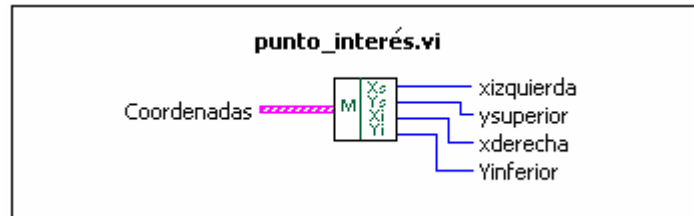


Figura 4.10 Icono de punto\_interés

Recibe la salida *Encuentros* de *Detecta\_Botella.vi* para ordenar las coordenadas, enviando ordenadamente los dos puntos de interés. En este caso es (xizquierda, ysuperior) y (xderecha 2, yInferior).

#### 4.4.1.5 Función\_bordes.vi

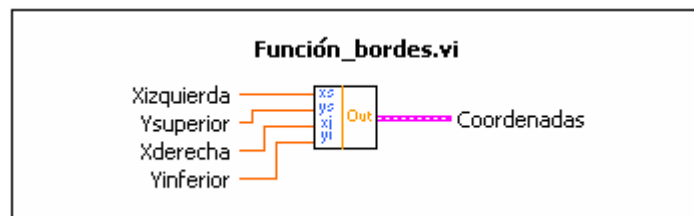


Figura 4.11 Icono función\_bordes

Recibe las dos coordenadas ordenadas de *Punto\_interes.vi* para agruparlos en un arreglo de arreglos, debido a que las siguientes funciones reciben este tipo de datos.

#### 4.4.1.6 Girar\_Botella\_detecta\_etiqueta.vi

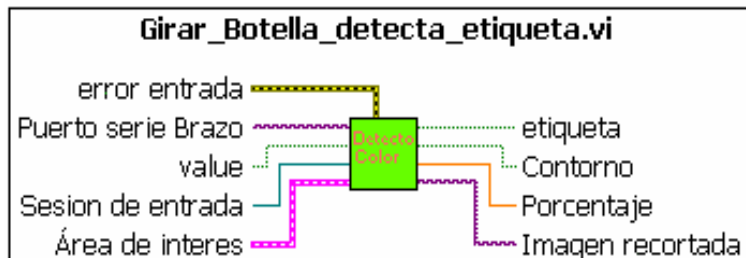


Figura 4.12 Icono giro\_botella\_detecta\_etiqueta

Detecta la etiqueta en la imagen de la botella cortada anteriormente, efectuando giros por medio del brazo robótico para analizar todo el contorno de la botella. Ésta subrutina posee las siguientes entradas : Arreglo de coordenadas de la región de interés, puerto de comunicación con el Brazo Robotico. Las salidas que posee son:

- Contorno: que significa que la botella giró los 360° y por ende no fue aceptada.
- Etiqueta: El patrón de la etiqueta encontrada.
- Porcentaje: Es el porcentaje de aceptación por la etiqueta encontrada.

La detección se la obtiene de otro SubVI que se encuentra dentro de éste, y se lo detalla en el siguiente párrafo.

#### 4.4.1.7 Detecta\_etiqueta.vi

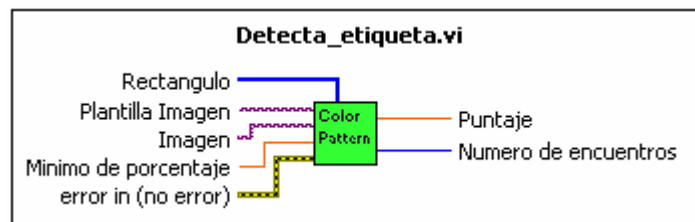


Figura 4.13 Icono detecta\_etiqueta

Se encarga de la detección de la etiqueta por medio del patrón que se le envía en *plantilla imagen*, además se le da un valor de referencia que permite filtrar el tipo de imagen y así obtener un mayor índice de aprobación. Su salida es el porcentaje real de aceptación con el que detecto la etiqueta.

#### 4.4.1.8 Guardar\_Imagen.vi

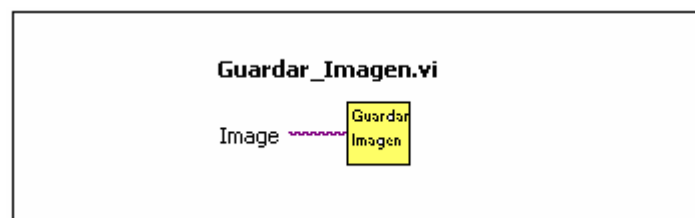


Figura 4.14 Icono guardar\_imagen

Se encarga de guardar la imagen aceptada por la aplicación, su propósito es para que en algún momento se pueda observar los detalles y poder corregirlo.

#### 4.4.1.9 Área\_de\_análisis.vi

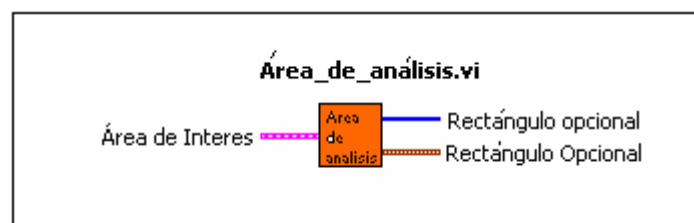


Figura 4.15 Icono área\_de\_análisis.

Se encarga de crear un área para sólo analizar la etiqueta, esto es, del área total de la botella, se analiza una región pequeña de la etiqueta para mayor eficiencia .

#### 4.4.2 Descripción de código

Una vez detenida la banda transportadora, la cámara comienza a adquirir la imagen.

Antes de hacer funcionar el programa, se tiene las opciones de adquirir el plantilla y de calibrar la cámara:



En la figura que se muestra a continuación, se puede observar la parte de Labview que se encarga de calibrar la cámara, que permite ajustarla para que las imágenes captadas estén correctamente enfocadas:

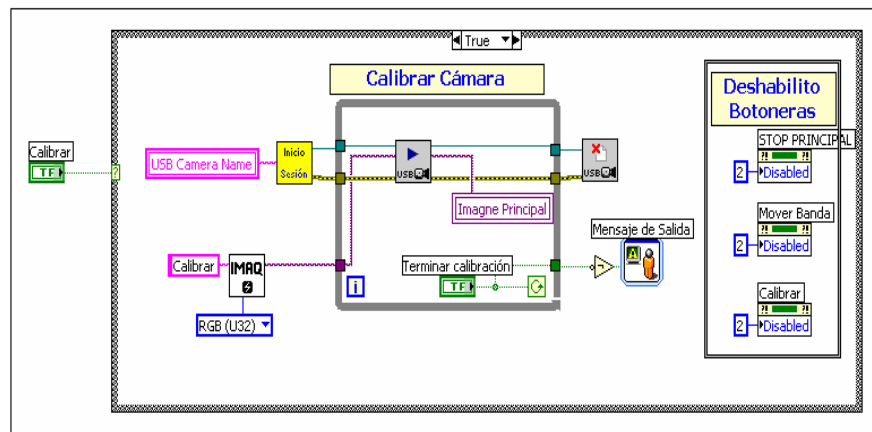


Figura 4.16 Código de calibrar cámara

También se muestra a continuación la parte de LabVIEW que se encarga de adquirir la plantilla:

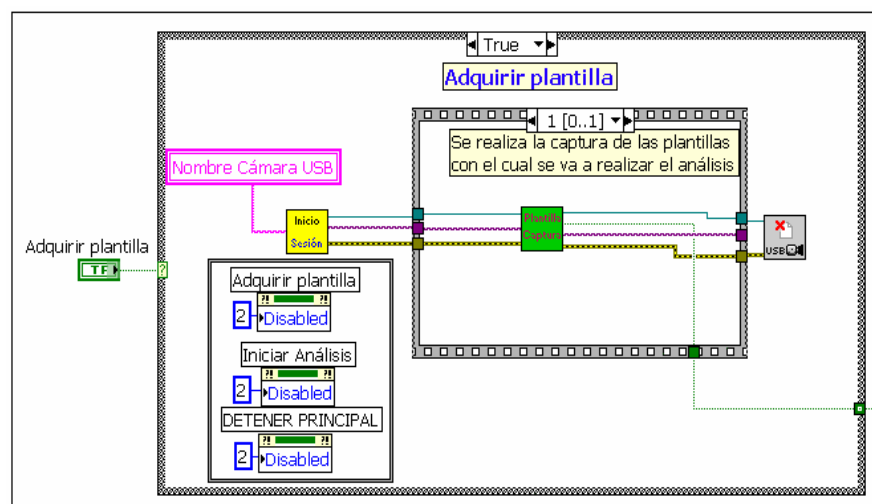


Figura 4.17 Código de adquirir plantilla

Después de estas etapas que son necesarias para poder trabajar, se procede a inicializar las variables y el puerto serie para el control de la banda transportadora y el brazo robótico al inicio del programa, igualmente se configura la distancia entre el sensor y la cámara, en este caso es de 1000 mm, a continuación se muestran las figuras de inicialización de variables y configuración de los puertos.

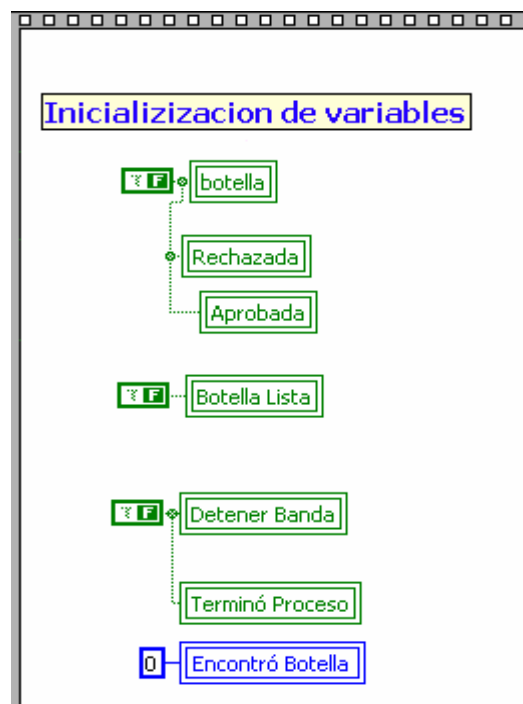


Figura 4.18 Inicialización de variables

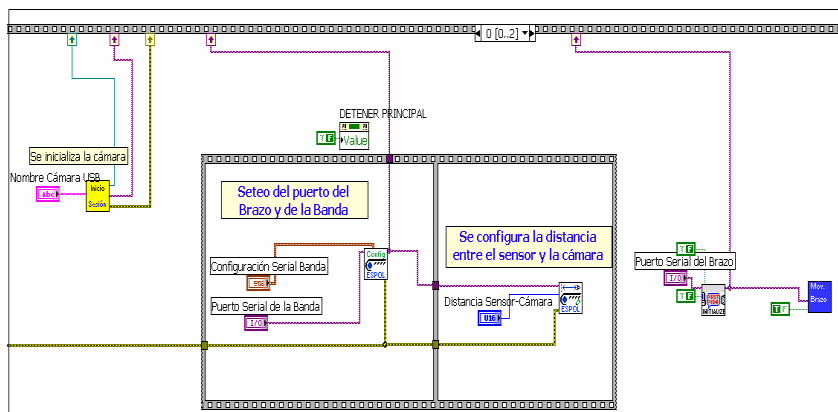


Figura 4.19 Código de configuración de puertos series

A continuación, empieza a moverse la banda por medio de dos subVIs creados; el primer VI sirve para establecer la velocidad con la que se va a trabajar, y el segundo para ordenar a la tarjeta para hacer girar a los motores.

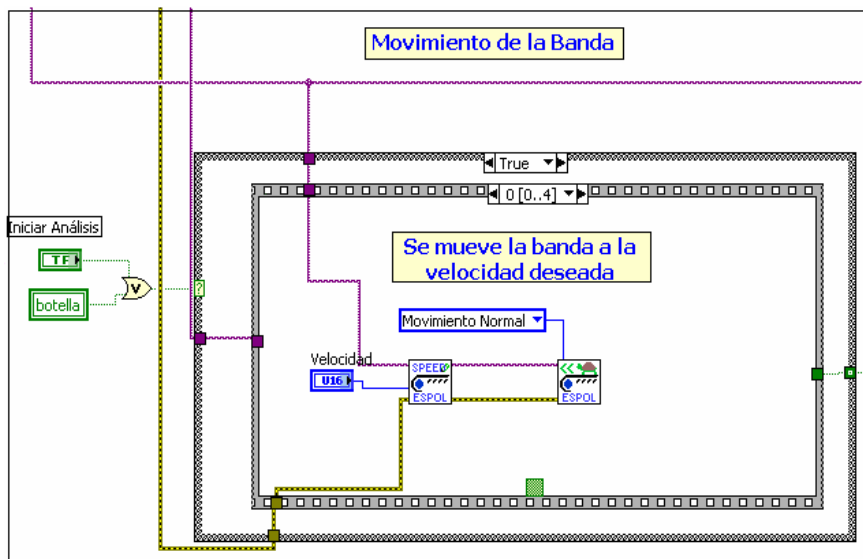


Figura 4.20 Configuración del movimiento de banda

En la siguiente etapa se ordena que el servo del empuja frascos se mueva a la posición de *parar el frasco*, además de limpiar las variables: Botellas, rechazada y aprobada.

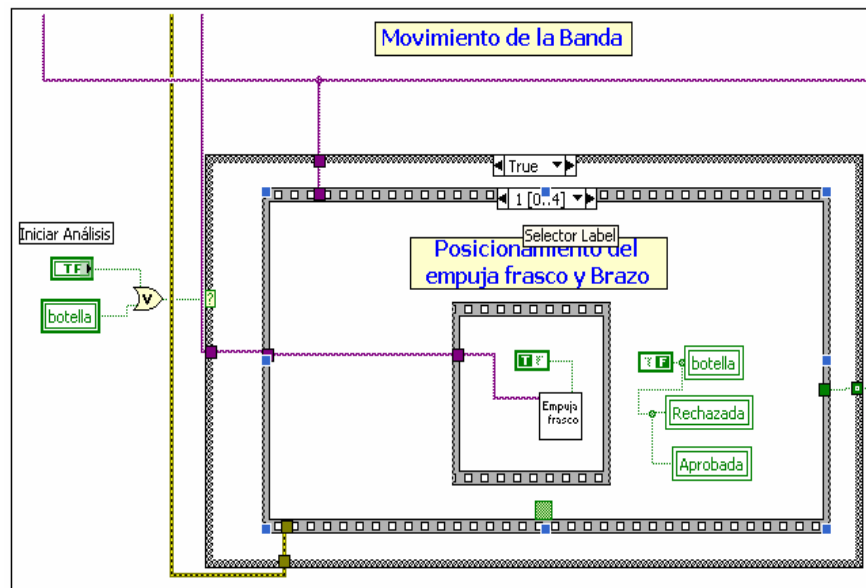


Figura 4.21 Posicionamiento del empuja frasco

En el paso posterior, se utiliza el subVI Espol Lab Conveyor Report.vi que permite obtener los datos de los sensores de la tarjeta que controla la banda y está configurado en reporte manual, este subVI se encuentra dentro de un lazo *while* debido a que constantemente el subVI pide información de los sensores y si no se le añade el lazo, se cae la comunicación perdiendo así el control de la banda.

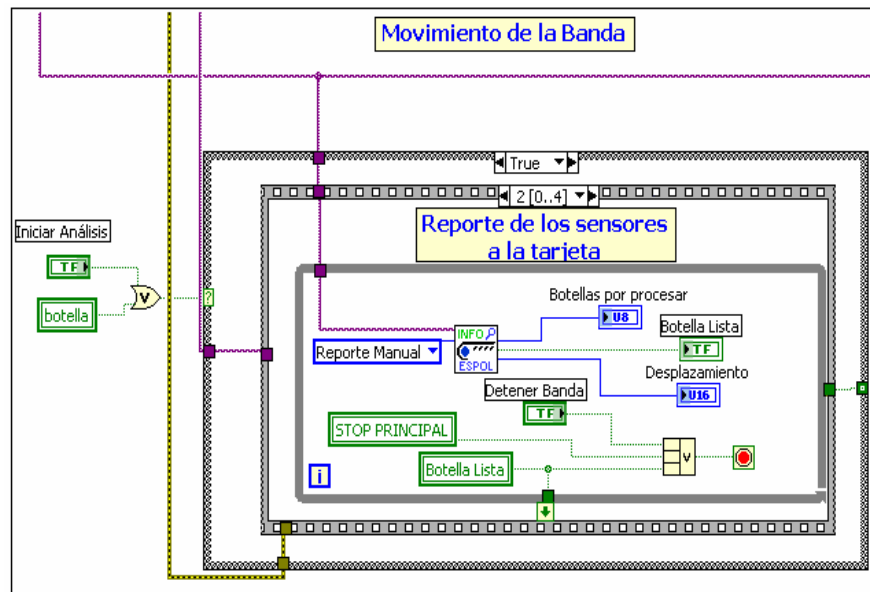


Figura 4.22 Reporte del sensor de tapa

Si el sensor de botellas detecta la tapa de ésta, este subVI detiene la banda con la señal de *botella lista* enviándolo al subVI Espol Lab Conveyor Stop y así saliendo del lazo para esperar 250ms, luego del cual, se levanta el empuja frasco en la posición *retorno*.

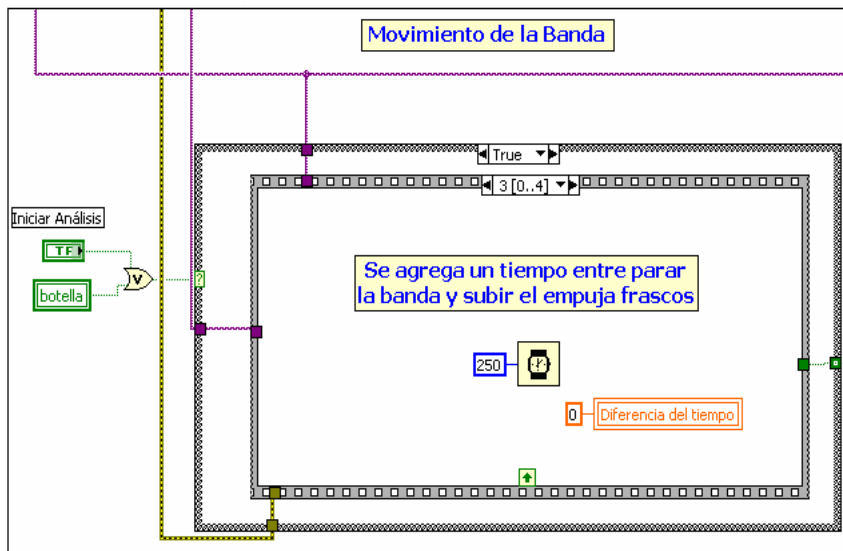


Figura 4.23 Tiempo de espera

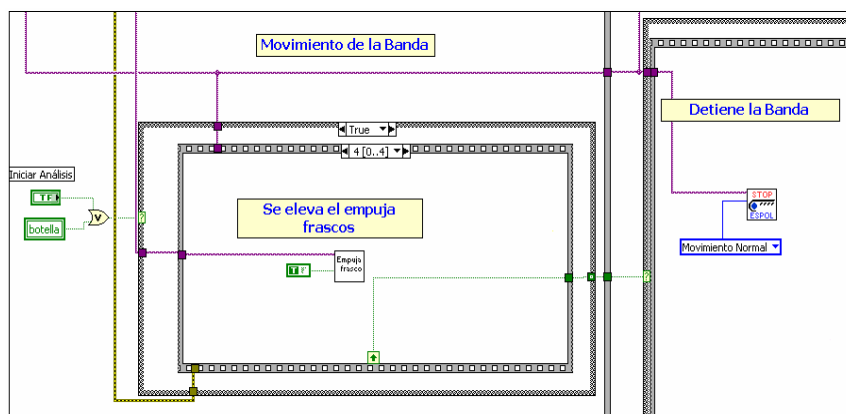


Figura 4.24 Detener banda transportadora

Luego de retornar el empuja frascos, se inicia la sesión de la cámara con el subVI inicio\_sesion.vi asignando un tiempo necesario para su correcto funcionamiento. El tiempo se logra ingresando a modo de captura continua.

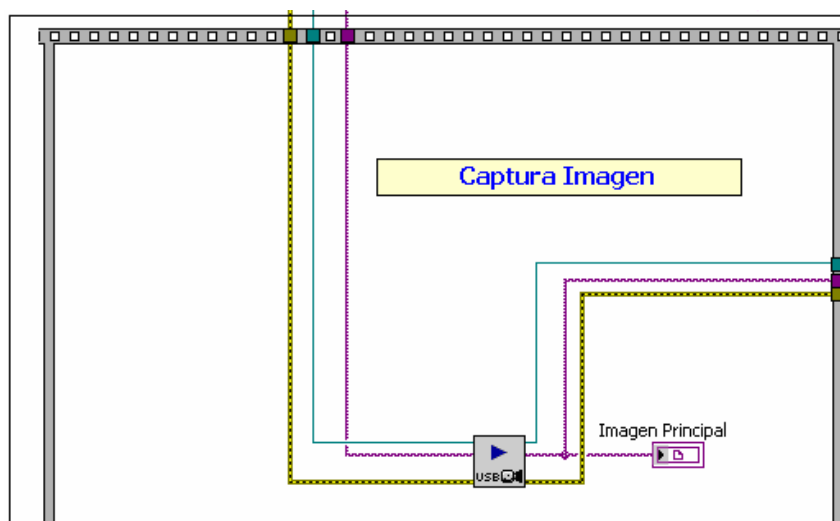


Figura 4.25 Modo de captura continua

Capturada la imagen, se procede a reservar un espacio de memoria para la imagen adquirida, y luego cortar dicha imagen comparando con la plantilla de imagen1.png que se obtuvo al inicio del programa.

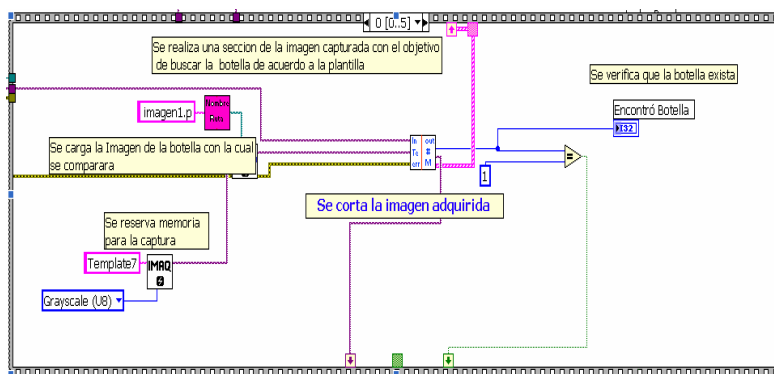


Figura 4.26 Búsqueda de la plantilla

Si encontró la botella entonces el subVI Detecta\_Botella.vi enviará una señal *encontró botella* que será 1, y entregará las coordenadas del frasco encontrado para seccionar la imagen

a un tamaño menor evitando errores con las botellas cercanas y así se asegura que en la región de análisis encontrará una sola botella y poder compararla para detectar si tiene etiqueta o no.

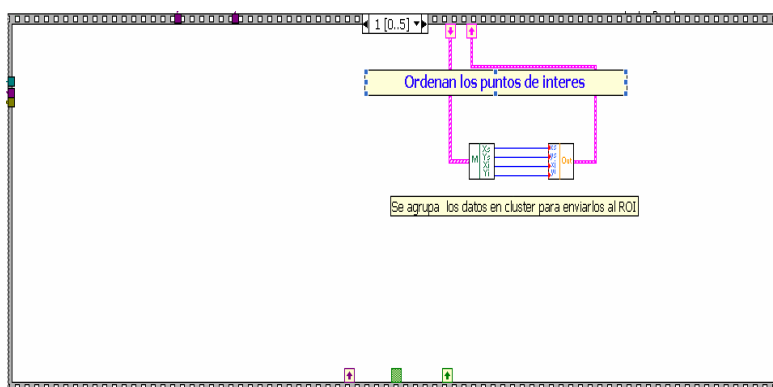


Figura 4.27 Agrupación de los puntos de interés

En el siguiente procedimiento, se envía la orden para que el brazo se posicione en la parte superior de la botella. Esto se logra con el subVI Brazo.vi que fue desarrollado por los compañeros de la tesis “Diseño de brazo robótico y diseño de brazos empuja frascos controlados por LabVIEW y DSPICS”, sólo se le envía el puerto con el que se comunica que en este caso es el COM1, y la orden que al inicio es el *reset*.



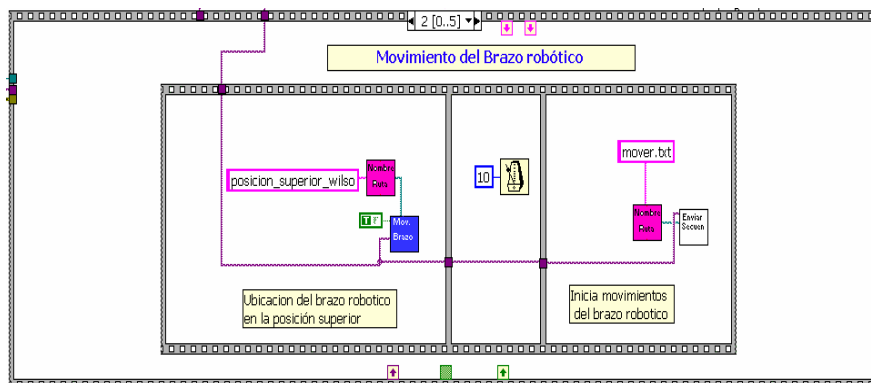


Figura 4.28 Ubicación del brazo robótico

Luego de que se posicionó el brazo encima de la botella, el subVI Girar\_Botella\_detecta\_etiqueta.vi, adquirirá la imagen de la botella y buscará la etiqueta, si no la encuentra, envía la orden al brazo para que haga girar la botella en pasos de  $18^\circ$  y en cada paso buscará la etiqueta. El resultado es un valor de falso o verdadero.

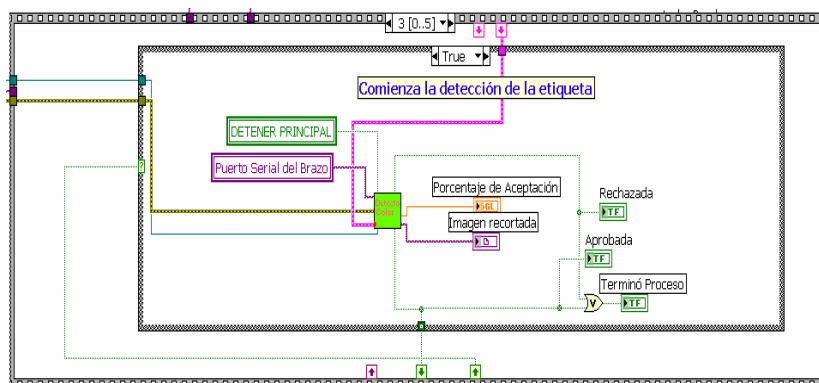


Figura 4.29 Análisis de etiqueta

El resultado del análisis del subVI anterior, va ser útil debido a que si no encontró la botella, el resultado será FALSO y ordenará al brazo a regresar a la posición inicial, luego activará

al empuja frascos para que bote la botella, como se puede apreciar en el código:

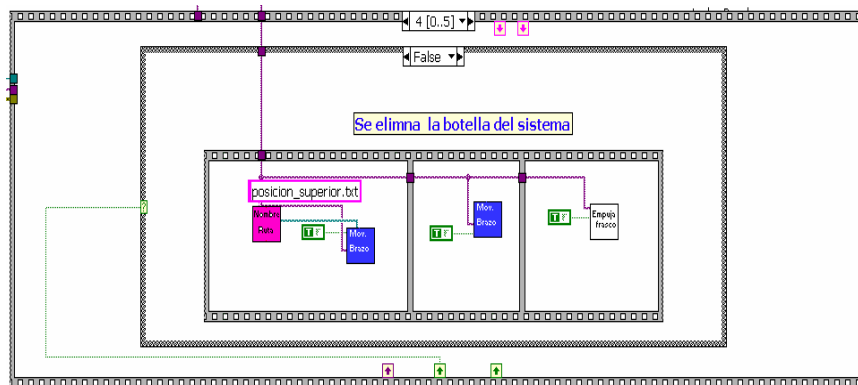


Figura 4.30 Elimina botella del sistema

De lo contrario, si el resultado es VERDADERO, entonces el brazo se ubicará en la posición inicial .

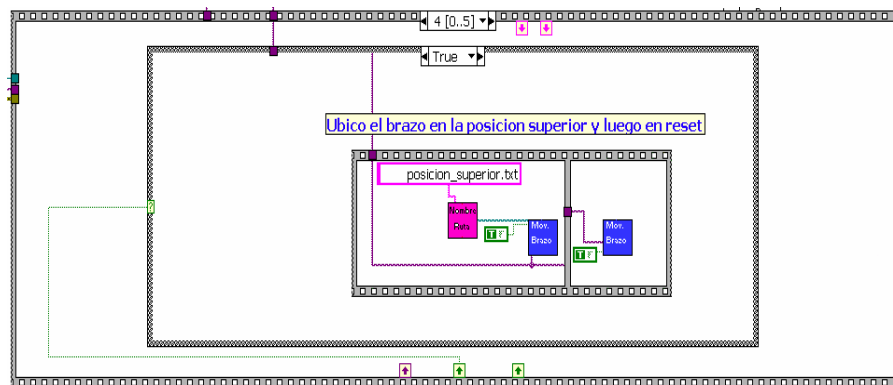


Figura 4.31 Ubicación de brazo en la posición inicial

Después de las acciones mostradas anteriormente, se cierra la sesión de la cámara y moverá a la Banda para analizar a la siguiente botella, y así se repetirá el proceso cada vez que encuentre botellas.

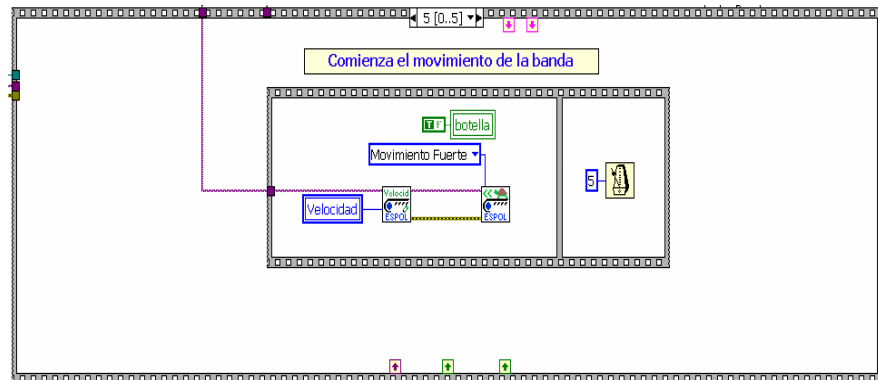


Figura 4.32 Movimiento de la banda transportadora

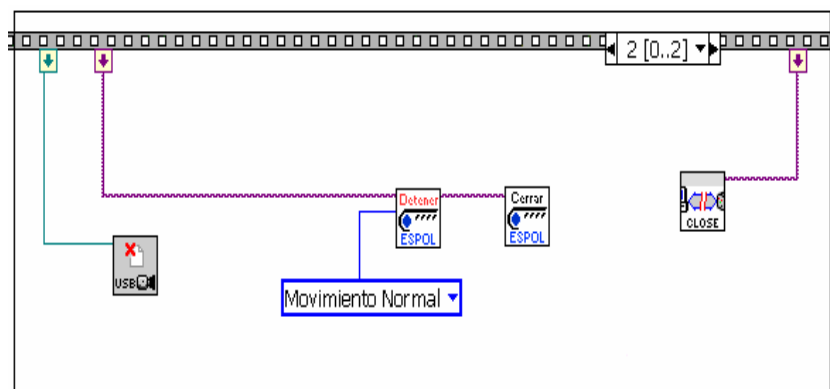


Figura 4.33 Se cierra la configuración de los puertos utilizados

## 4.5 Implementación física

De acuerdo con el requerimiento del proyecto, se han diseñado dos cabinas: una para la iluminación trasera y una para el análisis, para ello a continuación se detallarán.

#### 4.5.1 Sistema de iluminación:

Es un factor importante en la detección, debido a que si existe algún cambio de luminosidad puede afectar el análisis, por tal motivo se ha diseñado una cabina de luz cuyas dimensiones son 20x17x10cm y cuyo interior tiene 2 fluorescentes ahorradoras de energía de 15Watt y 20Watt, que se ubicará detrás de la cámara Genius para enfocar la botella a analizar y así eliminar las sombras y rayas como se mencionó anteriormente.

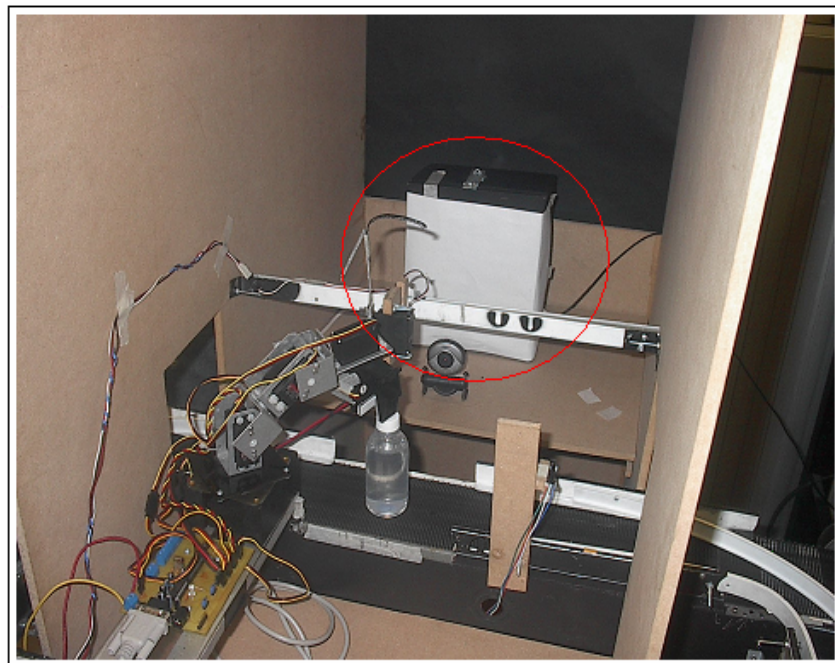


Figura 4.34 Vista del sistema de iluminación

#### 4.5.2 Cámara de análisis:

Además, para aislar el sistema de las condiciones ambientales, se ha construido una cámara de análisis de tal manera que pueda conseguir niveles de luz constantes que favorezcan al proceso de inspección. El material utilizado es MDF tal como se muestra en el siguiente grafico:

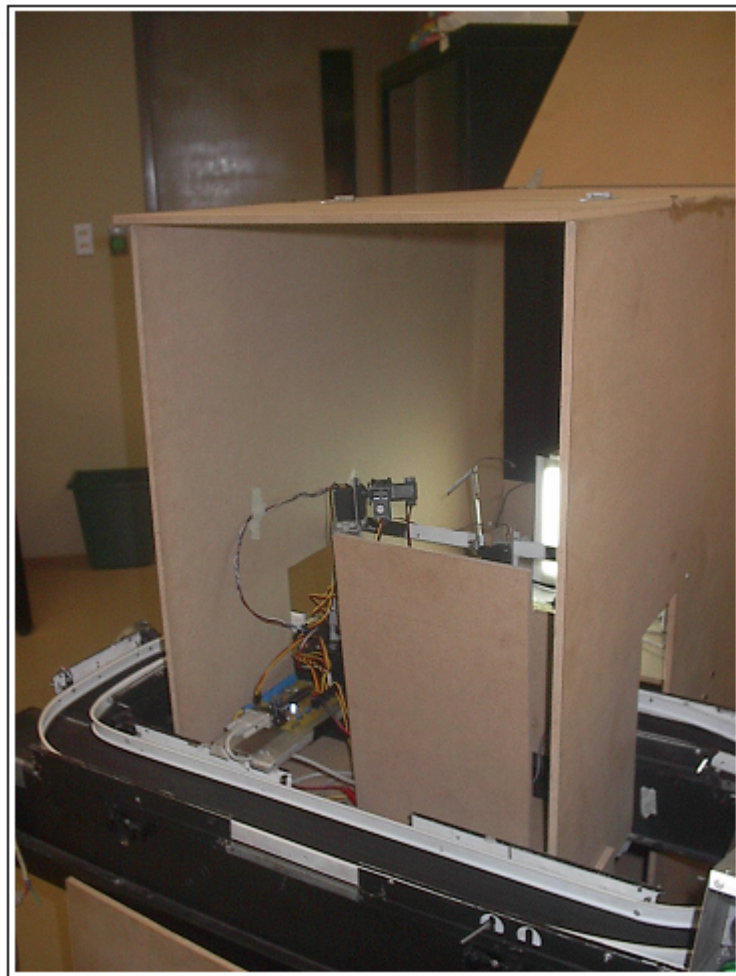


Figura 4.35 Vista exterior de la cámara de análisis

En el interior del armario, se puede apreciar la cámara, el sistema de iluminación, la banda transportadora con la botella a analizar como se muestra en la siguiente figura:

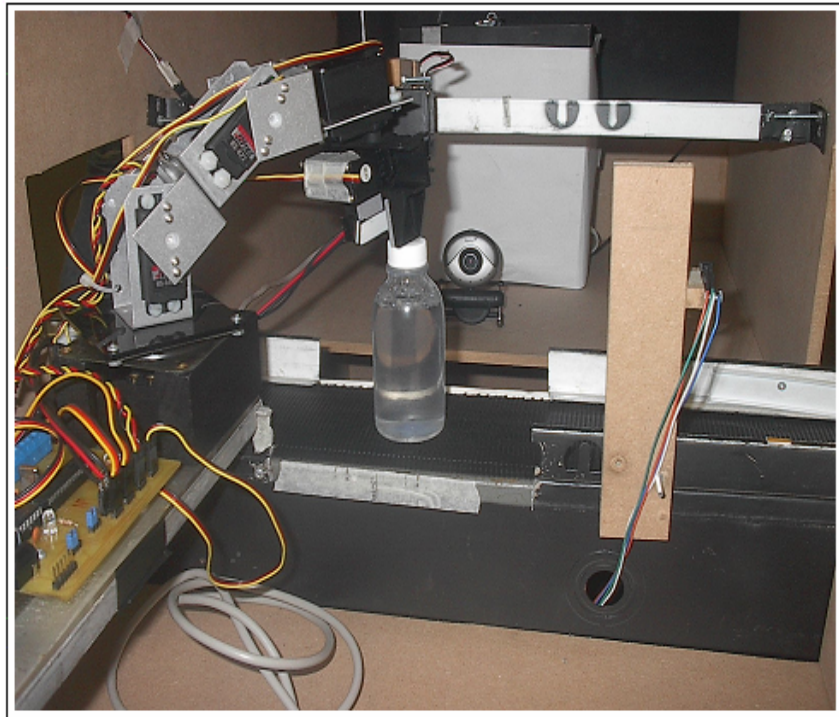


Figura 4.36 Interior de la cámara de análisis

#### 4.6 Control del brazo robótico

Debido a que la etiqueta no cubre toda la superficie lateral de la botella, y puede darse el caso de que la etiqueta no coincida con el enfoque de la cámara, se logró utilizar y controlar el brazo robótico para cuando la botella llegue al área de análisis, el brazo haga girar la botella los 360°.

El control se lo efectúa mediante un programa realizado en LabVIEW, en el cual se asignan los valores de posición a los cinco servos motores mencionados anteriormente. Esto se logra mediante la comunicación serial entre LabVIEW y la tarjeta que controla el Brazo robótico y el empuja frasco.

De manera general, el funcionamiento se basa en enviar el periodo de trabajo de los servomotores, que para este caso es: 900 corresponde a 0° y 6000 es 180°, pudiéndose obtener valores intermedios para tener mayor precisión en la ubicación del brazo.

De acuerdo con lo mencionado, se va a describir los bloques que controlan a esta tarjeta.

#### 4.6.1 Brazo.vi

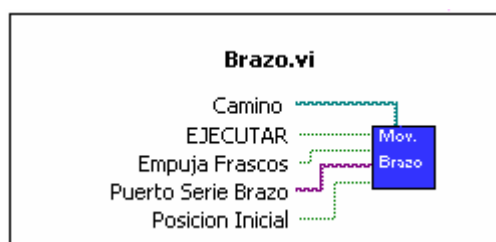
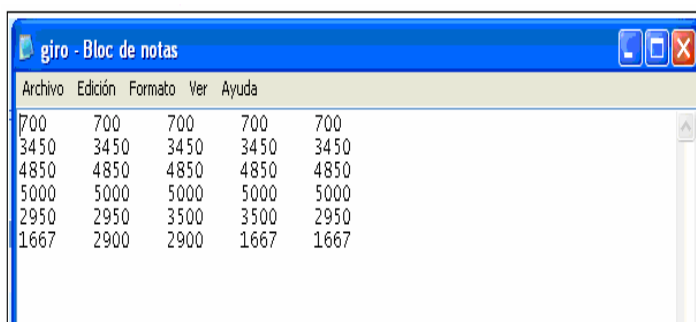


Figura 4.37 Icono del brazo robótico

Este bloque envía y recepta los datos por el puerto de comunicación, que en este caso se conectará por *Puerto Serie Brazo*, *EJECUTAR* debe de estar activado para que el brazo robótico pueda moverse, caso contrario no realizará alguna

acción. Si el objetivo es mover el brazo a una posición deseada, se debe de crear un archivo con extensiones xls ó txt que son archivos de Excel y bloc de notas respectivamente, escribiendo en este orden: la velocidad de movimiento del brazo, la posición del motor 1, la posición del motor 2 hasta el motor 5..

En el siguiente gráfico se muestra las posiciones de los motores en un archivo de bloc de notas:



Archivo	Edición	Formato	Ver	Ayuda
700	700	700	700	700
3450	3450	3450	3450	3450
4850	4850	4850	4850	4850
5000	5000	5000	5000	5000
2950	2950	3500	3500	2950
1667	2900	2900	1667	1667

Figura 4.38 posición de los motores servo

Hay 5 columnas que corresponde a 5 datos enviados, cada una de 6 filas que corresponde a las diferentes posiciones para los servos excepto la primera que es la velocidad del movimiento del brazo. En el gráfico de arriba se muestra el archivo para el giro de la botella que lo hace para buscar si tiene etiqueta.



#### 4.6.2 Empuja\_Frascos.vi

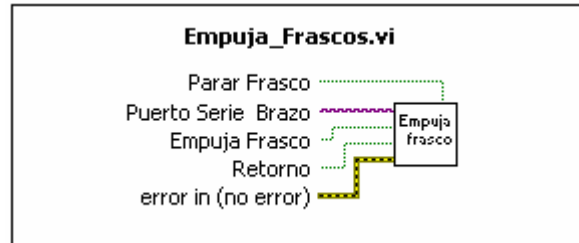


Figura 4.39 Icono del empuja frascos

Este bloque es más fácil de entender debido a que sólo se le envía el puerto del empuja frasco que es el mismo que del brazo en *Puerto Serie Brazo*, y la posición que desea ubicar al empuja frasco que para este caso sólo tiene 3 posiciones: Parar frasco, Empuja frasco y retorno.

Parar frasco: Es para que se ubique de frente a las botellas y no los permita pasar.

Empuja frasco: Como su nombre lo dice, es para botar la botella en caso de no encontrarse la etiqueta.

Retorno: Para que se ubique en una posición que no obstruya a las botellas

#### 4.7 Análisis de costos

Los costos de los equipos y materiales que se utilizaron en la implementación del proyecto, se describen en la siguiente tabla, analizando cada uno de ellos a continuación.

Ítem	Descripción	Precio Unitario	Cantidad	Precio Total
1	Cámara digital Genius	\$ 15,00	1	\$ 15,00
2	Cabina de luz de: 20x17x10cm	\$ 5,00	1	\$ 5,00
3	Cabina de Análisis MDF	\$ 70,00	1	\$ 70,00
4	Frascos plásticos	\$ 0,10	10	\$ 1,00
5	Horas de trabajo	\$ 2,10	480	\$ 700
	<b>TOTAL</b>			<b>\$ 791</b>

Tabla 1 Costos de los equipos y materiales

Para establecer los costos de los equipos y materiales, se consideraron:

La cámara de video es la que se utiliza para realizar conferencias en red, se puede comprar en cualquier centro de cómputo a un precio bajo, lo que se tiene que considerar es la resolución, que en este caso es de 352x288, además se conecta vía USB.

El material de construcción de la cabina de luz es madera, su elaboración fue en un taller de ebanistería, su costo depende del tamaño que se desee, y del lugar en donde se lo manda hacer.

La cabina de análisis es de MDF un material liviano pero resistente, se debe realizar la compra de un cuadro grande de 2.5x4m en el HyperMarket, la mano de obra está incluida en el valor de compra.

El tipo y tamaño del frasco fue escogido por el director de tesis con lo que se pudo realizar pruebas, los frascos fueron comprados a ese precio por su tamaño

Otro costo que se debe de tener en cuenta son las horas empleadas en el desarrollo del proyecto, el número de horas resultado del interés de culminar en un corto plazo sin dejar de lado los requerimientos exigidos para tener un programa eficiente.

# CAPÍTULO 5

## 5. DATOS EXPERIMENTALES

En este capítulo se detallan las pruebas realizadas del código desarrollado en LabVIEW para la detección de frascos sin etiqueta en conjunto con la banda transportadora y el brazo robótico, además se presentan los datos obtenidos de las pruebas, así también se expone un análisis de los resultados de las pruebas realizadas.

Para la culminación de este proyecto se realizaron algunas pruebas donde las variables son: movimiento de la banda, iluminación, precisión del brazo robótico y la posición de la etiqueta, de las cuales al final se llegó a un consenso donde se trató de eliminar la mayor cantidad de variables posibles y ajustarse a los elementos disponibles.

### 5.1 Pruebas realizadas

Para la elaboración del algoritmo de la detección de frascos sin etiquetas desarrollado en LabVIEW se realizó 2 tipos de pruebas:

- Análisis de frascos estáticamente.
- Análisis de frascos acoplado a la banda transportadora y el brazo robótico.

### **5.1.1 Análisis de frascos**

Esta parte del proyecto también se denominó estudio estático debido a que sólo se realizaba pruebas con los frascos ubicados de manera manual sobre una mesa de análisis donde se fijaba el área de interés, la posición donde sólo se realizaría el análisis y el tipo de iluminación.

En el estudio estático se elaboró la gran mayoría de bloques de instrumentos virtuales necesarios para el análisis de frascos sin etiqueta como: inicialización de la cámara, capturar plantilla, recortar imágenes entre otros, estos ya se mencionaron en el capítulo 4.

La Figura 5.1 muestra la mesa de trabajo donde se realizaron las pruebas estáticas.

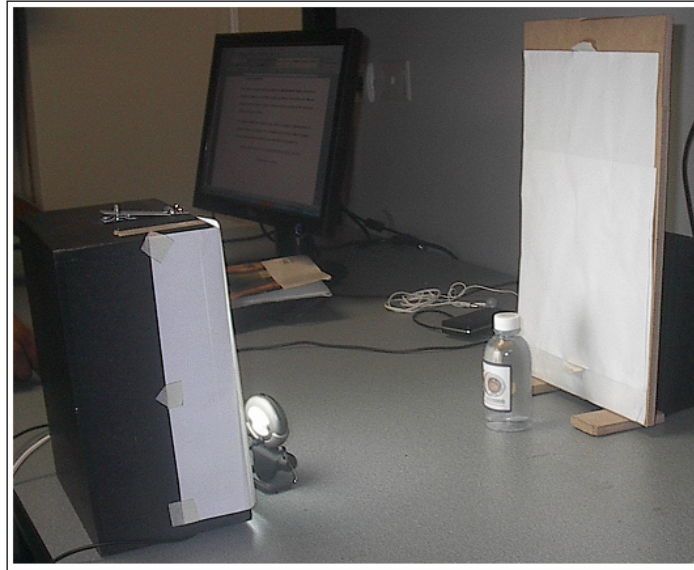


Figura 5.1 Mesa de trabajo

### **5.1.2 Análisis de frascos acoplado a la banda transportadora y el brazo robótico**

Esta etapa consistía en ejecutar todo el acoplamiento de las rutinas realizadas por los compañeros que desarrollaron la banda transportadora y el brazo robótico con el bloque de detección de frascos sin etiqueta.

Para la elaboración del acoplamiento se siguió 2 escenarios de los cuales el último fue escogido debido a que presentaba una mejor precisión obteniendo excelentes resultados al momento de realizar la detención de la banda transportadora, en los párrafos siguientes se presenta todos los escenarios utilizados.

### **5.1.2.1 Detección de la botella utilizando la cámara como sensor**

En esta etapa se pretendía detener la banda transportadora una vez que la botella era censada por la cámara, su objetivo se logró pero no con gran exactitud porque su posición final era dependiente de la velocidad con la que la banda se desplazaba, entonces al momento de utilizar el brazo robótico para realizar los giros de la botella, el frasco no quedaba en el rango de cobertura, este problema se debe principalmente porque la cámara es demasiado lenta al momento de capturar la imagen y después cuando enviaba la información de detener la banda; el frasco en la realidad se había desplazado según su referencia de captura.

En este escenario sólo se revisó la posición de la botella para después continuar con el respectivo análisis, el cual no se desarrolló por los problemas presentados en ubicar el frasco en la zona de análisis.



Figura 5.2 Imágenes donde se aprecia el desfase en la detención de los frascos.

#### 5.1.2.2 Detección del frasco utilizando un sensor reflector y el empuja frasco.

Otro cambio que se realizó para el desarrollo del proyecto fue el de utilizar un sensor distinto al anterior (cámara), como es el caso del sensor de reflexión ubicado antes de que la botella llegue al área de procesamiento, la utilización de este sensor dio una mejor respuesta al momento de detener la banda transportadora pero aún no era muy preciso por lo que se decidió utilizar una barrera de detención para evitar que éste se aleje de la zona de estudio. Con esta



barrera se logró una mejor precisión al momento de realizar los análisis

En la siguiente figura se aprecia el sensor y barrera implementadas en el proyecto.

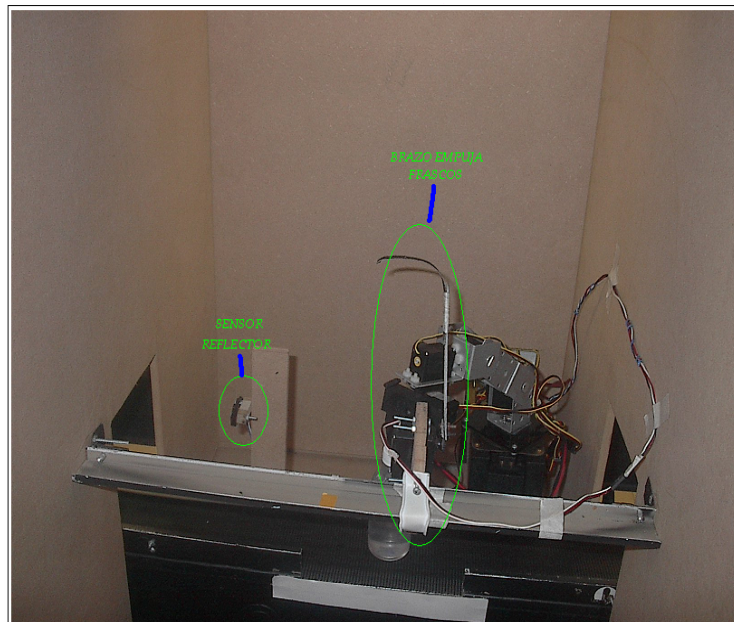


Figura 5.3 Visualización del sensor y de la barrera de detención

Una vez establecido el escenario el cual provee una mejor precisión, se comenzó a realizar pruebas de los VI's creados en el estudio estático para lo cual se dividieron en dos etapas:

La primera consiste en realizar la detección de la botella, esto se logra una vez detenida la banda, se realiza una captura fotográfica con la cámara, después

se busca el patrón con el objetivo de seccionar el área de análisis debido a que puede existir otra botella cercana.

En la segunda etapa se realiza la inspección de la etiqueta con la imagen anteriormente capturada y seccionada para lo cual se utiliza el brazo robótico, permitiendo de esta manera examinar todo el contorno de la botella debido a que la etiqueta puede venir ubicada en distintas áreas, también se realiza una reducción del área donde debe venir colocada la etiqueta.

Las botellas que se utilizaron para llevar a cabo las pruebas de la detección de frascos sin etiquetas fueron las que se muestran en las siguientes figuras :



Figura 5.4 Imágenes de los frascos utilizados para el análisis de etiqueta 1,2,3



Figura 5.5 Imágenes de los frascos utilizados para el análisis de etiqueta 4,5,6



Figura 5.6 Imagen del frasco utilizado para el análisis de etiqueta 7

1	Frasco con etiqueta correcta
2	Frasco con etiqueta de otro color
3	Frasco con etiqueta incorrecta
4	Frasco con etiqueta desplazado hacia abajo
5	Frasco con etiqueta rayada W
6	Frasco con etiqueta rayada
7	Frasco sin etiqueta

Tabla 2 Identificación numérica de los frascos analizados

De las imágenes de los frascos mostrados anteriormente, se tomaron dos imágenes patrón para la identificación de las demás, y son las siguientes:



Figura 5.7 Plantilla para seccionar el área de análisis



Figura 5.8 Plantilla de la etiqueta

## 5.2 Datos obtenidos

Los datos obtenidos del procesamiento por etapas realizado en LabVIEW de cada uno de los frascos presentados anteriormente fue el siguiente:

Se realiza la etapa de selección de la región de interés mostrado en las figuras siguientes con la ayuda de la plantilla de una botella.



Imagen completa	Área de análisis
 A photograph of a clear plastic bottle with a white cap and a label, sitting on a dark surface. The label features a circular logo with a stylized 'M' and the number '19582001' below it. The bottle is centered in the frame.	 A zoomed-in photograph of the same bottle, focusing on the label. The label is clearly visible, showing the circular logo and the number '19582001'.











Tabla 3 Imágenes adquiridas y área de análisis de pruebas realizadas.

En las siguientes tablas se exponen los datos obtenidos del análisis de 7 muestras de frascos, realizando 2 repeticiones, donde se indica si el frasco fue aceptado o rechazado y su porcentaje de aceptación obtenido en cada uno de los análisis.

Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	832.931	*	
2 Etiqueta con otra coloración	615.1		*
3 Etiqueta incorrecta	596.515		*
4 Etiqueta desplazada hacia abajo	660.545		*
5 Etiqueta rayada	612		*
6 Etiqueta rayada	618.894		*
7 Sin etiqueta	500.196		*

Tabla 4 Prueba realizada 1

Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	827.963	*	
2 Etiqueta con otra coloración	646.933		*
3 Etiqueta incorrecta	580.502		*
4 Etiqueta desplazada hacia abajo	609.148		*
5 Etiqueta rayada	686.109		*
6 Etiqueta rayada	605.301		*
7 Sin etiqueta	482.062		*

Tabla 5 Prueba realizada 2

### 5.3 Análisis de resultados

Con los datos obtenidos desde LabVIEW se presenta que la mayoría de las botellas aprobadas deben poseer un nivel de aceptación mayor al 80 %, de esta manera se evita que los frascos incorrectos aprueben el análisis del sistema.

Para el análisis de la etiqueta, se utiliza el VI IMAQ Match color pattern, el cual provee la capacidad de evaluar el color y forma del contenido del objeto seleccionado, pero estos parámetros de análisis dependen de la distancia a la que fue tomada la plantilla y la iluminación que existe en ese momento, por eso es recomendable tener una iluminación uniforme.

En el proyecto cuando se realiza el giro con el brazo robótico, éste afecta la uniformidad de la iluminación debido a que realiza sombra con su movimiento, esto se observa muy claramente cuando se realiza la segunda ronda de análisis de las mismas botellas porque el valor de aceptación generado varía.

Otro análisis que se realiza es una comparación manual y una comparación de los valores obtenidos desde LabVIEW de cada uno de los frascos analizados.

### 5.3.1 Comparación manual

Cuando se refiere a realizar una comparación manual, es el análisis que haría un inspector humano por muestreo, que según su criterio y los casos presentados tomarían una decisión, a continuación se muestra los casos analizados en el presente proyecto

En la Figura 5.9 se observa que el frasco analizado posee la etiqueta ubicada de manera correcta y como es la plantilla, debe ser aprobada sin ningún problema.



Figura 5.9 Imagen frasco correcto

En la Figura 5.10 se observa que el frasco analizado posee la etiqueta correcta pero con otro color por lo tanto debe ser eliminada del sistema.



Figura 5.10 Etiqueta con otra coloración

En la Figura 5.11 el frasco analizado posee una etiqueta incorrecta entonces el frasco debe ser rechazado.



Figura 5.11 Etiqueta Incorrecta

En la Figura 5.12 se observa que el frasco analizado posee la etiqueta desplazada hacia la parte inferior de la misma, por lo tanto debe ser rechazado.



Figura 5.12 Etiqueta desplazada hacia abajo

La Figura 5.13 muestra que el frasco posee unas rayas con marcador en el centro de la etiqueta, sin lugar a duda éste debe ser rechazado por el sistema.



Figura 5.13 Etiqueta rayada

En la Figura 5.14 se observa que la botella analizada posee la etiqueta con rayas en el interior de la etiqueta, entonces el sistema debe rechazar el frasco.



Figura 5.14 Etiqueta rayada

La Figura 5.15 Se muestra el peor de los casos, el cual es que la botella no posea etiqueta.



Figura 5.15 Frasco sin etiqueta

### 5.3.2 Comparación obtenida de LabVIEW

Según el valor ingresado en el IMAQ Match color pattern en la opción de minimum score, éste dará una referencia de los valores de aceptación para establecer que una etiqueta sea considerada correcta y según las pruebas y análisis realizados, el valor debe ser superior al 80%, para los mismos frascos analizados de manera manual se realizó su inspección automática y se muestra el respectivo análisis.

En la Figura 5.16 el frasco analizado es el que posee la etiqueta ubicada de manera correcta, como se realizó varios



análisis, sus valores de aceptación varían porque depende de la ubicación en que la banda transportadora la sitúe después de hacer el recorrido total, otro factor que afecta es que el brazo robótico al momento de realizar los giros, la desplaza muy levemente, aun así con estos factores en contra, la botella es aprobada y se continua con el siguiente frasco.



Figura 5.16 Frasco con etiqueta correcta

En la Figura 5.17 Posee un frasco con la etiqueta, con una coloración distinta y debido a que se a configurado el puntaje mínimo a 800, lo hace bastante estricto al sistema en lo que tiene que ver con color y forma.



Figura 5.17 Etiqueta con otra coloración

La Figura 5.198 Es un frasco con una etiqueta incorrecta, para lo cual la aplicación le asigna un valor de aceptación bajo debido a que no posee coincidencias con la plantilla original, entonces el frasco es descartado del sistema.



Figura 5.18 Etiqueta incorrecta

El frasco de la Figura 5.19 posee la etiqueta correcta pero como el VI de IMAQ Match Color pattern proporciona la opción de escoger el área de interés y la etiqueta en estudio está desplazada hacia la parte inferior del mismo, su valor de aceptación es 0, por ende el frasco es rechazado con el empuja frascos.



Figura 5.19 Etiqueta desplazada

La Figura 5.20 Muestra frascos con etiquetas en la posición correcta pero poseen unas rayas con marcador en el centro de la mismas, por lo tanto su valor de aceptación es muy inferior al preestablecido y son rechazadas.



Figura 5.20 Etiquetas rayadas

En la Figura 5.21 se observa el peor de los casos, donde el frasco no posee etiqueta, el sistema después de realizar su análisis, lo rechazará porque no encontrará nada y su valor de aceptación es mayor a 100 y menor a 200.



Figura 5.21 Frasco sin etiqueta

Otro aspecto que es un muy importante de señalar, es el tiempo que toma la aplicación realizada en LabVIEW para detectar si un frasco posee la etiqueta o no, en términos generales, el análisis de un frasco es de 25 a 75 segundos, es decir, el tiempo de análisis depende de la posición en que la etiqueta llegue al área de análisis, y por las pruebas realizadas, el brazo se demora de 30 a 35 segundos en dar un giro de 360 grados.

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

Al finalizar el proyecto utilizando la herramienta LabVIEW, se puede concluir que resulta muy eficiente y fácil de usar, debido a su programación gráfica.

El acoplamiento de la aplicación desarrollada con la banda transportadora y el brazo robótico, funciona con restricciones debido a la características físicas de cada uno de estos elementos.

Con las 49 pruebas realizadas, el sistema funcionó correctamente descartando los frascos que eran necesarios y aprobando los correctos.

Los datos obtenidos, al ser analizados con la herramienta búsqueda de patrones de LabVIEW, arrojan un porcentaje de coincidencia superior al 80%, dicho algoritmo hace comparaciones de la imagen capturada en forma y color

con una plantilla que ha sido previamente cargada en el programa. Este porcentaje de coincidencia es aceptable

Una desventaja de este proceso, es el tiempo de análisis, el cual toma como máximo valor 75 segundos, debido a que se analiza todo el contorno de la botella utilizando el brazo robótico (35 segundos), el cual posee una velocidad limitada, otro factor que afecta, es el tipo de cámara, la cual necesita un tiempo de refrescamiento para realizar una captura aceptable.

Esta aplicación prototipo con una inversión relativamente baja logra ofrecer una alternativa a los controles de calidad existentes para la inspección de etiquetas en frascos, de esta manera se pretende incentivar a futuros investigadores con una mayor inversión a realizar mejoras al proceso.

Se automatiza el control de inspección visual, por lo tanto se libera al operario de una labor tediosa y monótona la cual lo expone a cometer errores involuntariamente o sufrir de enfermedades.

Para la culminación del proyecto se han aplicado conocimientos adquiridos en la vida universitaria y en el presente tópico, como son: análisis de ingeniería y utilizar las herramientas que se posee en el mercado nacional.

## **Recomendaciones**

Se sugiere la utilización de un computador con una memoria RAM elevada debido a que LabVIEW consume bastantes recursos, si esta no es, la aplicación podría ocasionar que el proceso se torne demasiado lento o arroje valores incorrectos.

Para el desarrollo de aplicaciones de visión artificial es importante la utilización de técnicas de iluminación, en este proyecto se utilizó iluminación frontal que sirve para detectar contornos.

En procesos industriales reales se aconseja el uso de cámaras de mayor velocidad y mejor resolución, lo cual contribuirá a elevar la velocidad del proceso y la calidad de aceptación del mismo.



# ANEXO A

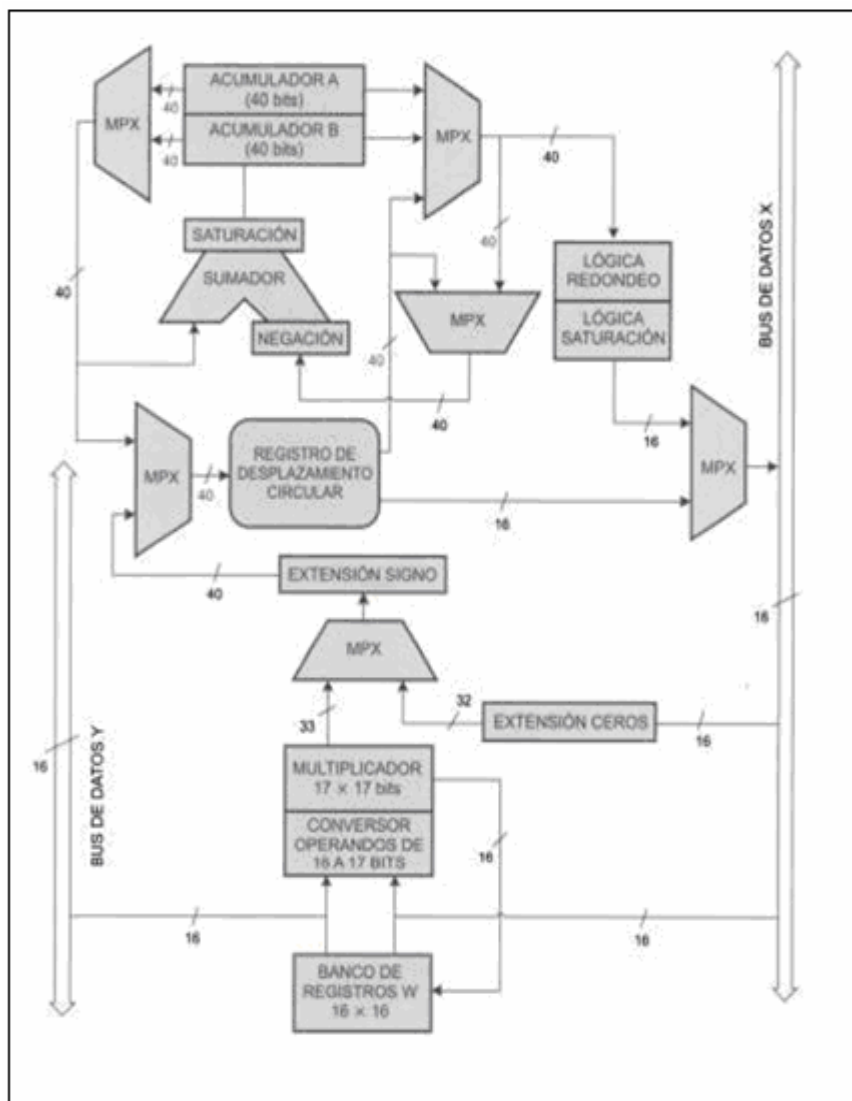
La siguiente tabla muestra la asignación de pines para los dos tipos de conectores.

Señal		<u>DB-25</u>	<u>DB-9</u>
Tierra común	G	7	5
Dato transmitido	TD	2	3
Dato recibido	RD	3	2
Terminal de datos listo	DTR	20	4
Configuración de dato listo	DSR	6	6
Pedido de envió	RTS	4	7
Limpiar para enviar	CTS	5	8
Portador de detectar	DCD	8	1
Sonido indicador	RI	22	9

Tabla 1 Asignación de pines

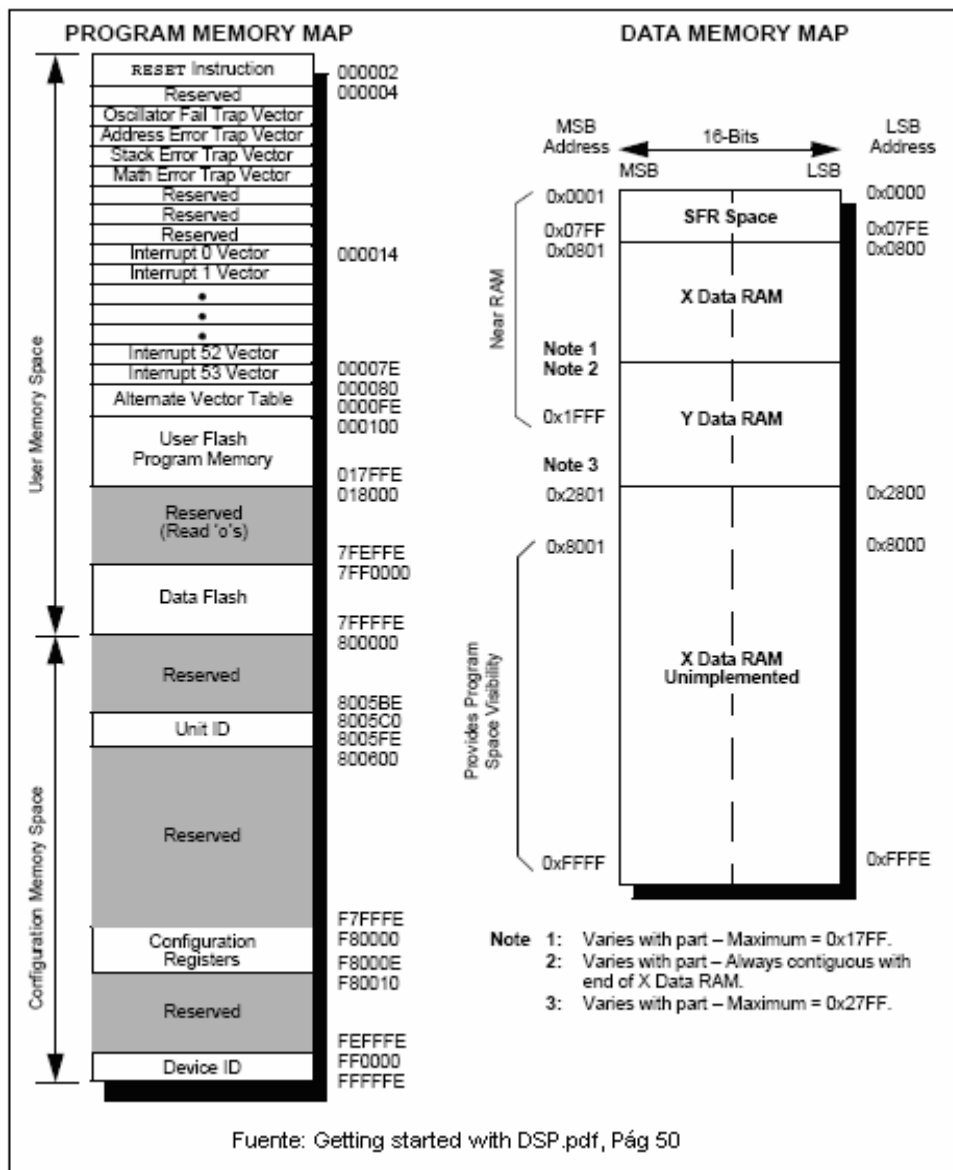
# ANEXO B

Arquitectura interna del motor DSP



# ANEXO C

Mapa de memoria de programa y datos del dsPIC30F



# ANEXO D

Figura de la tabla de vector de interrupciones

↓ Decreasing Priority  ↑  ↓  ↑  ↓	↑ $\sqrt{T}$	Reset – GOTO Instruction	0x000000
		Reset – GOTO Address	0x000002
		Reserved	0x000004
		Oscillator Fail Trap Vector	
		Address Error Trap Vector	
		Stack Error Trap Vector	
		Math Error Trap Vector	
		Reserved Vector	
		Reserved Vector	
		Reserved Vector	
		Interrupt 0 Vector	0x000014
		Interrupt 1 Vector	
		—	
		—	
		—	
		Interrupt 52 Vector	
		Interrupt 53 Vector	0x00007E
		Reserved	0x000080
		Reserved	0x000082
		Reserved	0x000084
		Oscillator Fail Trap Vector	
		Stack Error Trap Vector	
		Address Error Trap Vector	
		Math Error Trap Vector	
		Reserved Vector	
		Reserved Vector	
		Reserved Vector	
		Interrupt 0 Vector	0x000094
	Interrupt 1 Vector		
	—		
	—		
	—		
	Interrupt 52 Vector		
	Interrupt 53 Vector	0x0000FE	
	$\sqrt{T}$  AIVT		

Fuente: Getting started with DSP.pdf, Pág 53

# ANEXO E

Tablas de las pruebas realizadas:

Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	832.193	*	
2 Etiqueta con otra coloración	604.489		*
3 Etiqueta incorrecta	576.654		*
4 Etiqueta desplazada hacia abajo	726.175		*
5 Etiqueta rayada	752.259		*
6 Etiqueta rayada	623.307		*
7 Sin etiqueta	491.928		*

Tabla 2 Prueba realizada 3

Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	831.782	*	
2 Etiqueta con otra coloración	612.651		*
3 Etiqueta incorrecta	553.739		*
4 Etiqueta desplazada hacia abajo	733.265		*
5 Etiqueta rayada	780.011		*
6 Etiqueta rayada	657.262		*
7 Sin etiqueta	584.163		*

Tabla 3 Prueba realizada 4

Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	823.545	*	
2 Etiqueta con otra coloración	604.801		*
3 Etiqueta incorrecta	542.325		*
4 Etiqueta desplazada hacia abajo	726.151		*
5 Etiqueta rayada	614.437		*
6 Etiqueta rayada	605.372		*
7 Sin etiqueta	412.017		*

Tabla 4 Prueba realizada 5

Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	806.96	*	
2 Etiqueta con otra coloración	606.568		*
3 Etiqueta incorrecta	503.07		*
4 Etiqueta desplazada hacia abajo	695.177		*
5 Etiqueta rayada	670.234		*
6 Etiqueta rayada	644.894		*
7 Sin etiqueta	470.161		*

Tabla 5 Prueba realizada 6



Frasco con:	Valor de aceptación	Aceptada	Rechazada
1 Etiqueta correcta	859.663	*	
2 Etiqueta con otra coloración	680.79		*
3 Etiqueta incorrecta	506.317		*
4 Etiqueta desplazada hacia abajo	700.169		*
5 Etiqueta rayada	651.707		*
6 Etiqueta rayada	623.195		*
7 Sin etiqueta	455.782		*

Tabla 6 Prueba realizada 7

# **ANEXO F**

## **MANUAL DE USUARIO**

**APLICACIÓN DE VISIÓN CON LABVIEW**

**PARA LA DETECCIÓN DE FRASCOS**

**SIN ETIQUETA**

# CONTENIDO

<b>CONTENIDO.....</b>	<b>1</b>
<b>Para empezar.....</b>	<b>3</b>
<b>Pantalla principal .....</b>	<b>7</b>
<b>Parámetros de configuración .....</b>	<b>7</b>
Puertos seriales.....	8
Nombre de la cámara .....	8
Distancia Sensor-Cámara .....	8
<b>Controles .....</b>	<b>9</b>
Calibración Cámara.....	9
Adquirir plantilla.....	9
Iniciar análisis.....	10
Detener análisis.....	10
Detener banda.....	10
<b>Indicadores.....</b>	<b>11</b>

Indicadores booleanos .....	11
Indicadores numéricos .....	11
Indicadores gráficos .....	12
<b>Funciones implementadas .....</b>	<b>12</b>
Adquirir plantilla .....	12
Calibración de la cámara.....	14
Inicio de análisis .....	15
<b>Resolviendo problemas .....</b>	<b>16</b>

# INTRODUCCIÓN

Este documento explica el funcionamiento de la aplicación de detección de frascos sin etiqueta desarrollada en el software LabVIEW, se muestran sus principales componentes como: Pantalla principal con sus indicadores y controles.

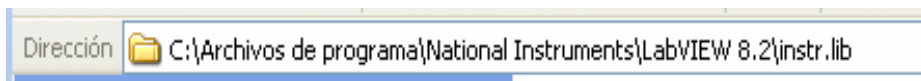
## PARA EMPEZAR

Antes de comenzar a ejecutar el programa se debe asegurar de cumplir con los requerimientos de instalación.

- Mínimo Pentium 4 2.46 GHz, Service Pack 2
- Mínimo 256MB de RAM
- Monitor para visualizar el proceso
- Software LabVIEW 8.2 con el paquete IMAQ Vision desarrollado por National Instruments.
- La computadora debe estar provista de puertos USB y puertos seriales para conectar la cámara Web, la banda transportadora y el brazo robótico respectivamente.

- Cámara de Web Marca Genius modelos 110, 111, 112 (Probados) o cualquier otra cámara compatible.
- Fuente de iluminación adecuada.
- Banda transportadora.
- Brazo robótico.

Una vez instalado LabVIEW se debe, copiar los archivos de configuración de las rutinas del brazo robótico y la banda transportadora que se encuentran en el disco de instalación a la siguiente dirección.



Los archivos se encuentran en las carpetas mostradas a continuación.

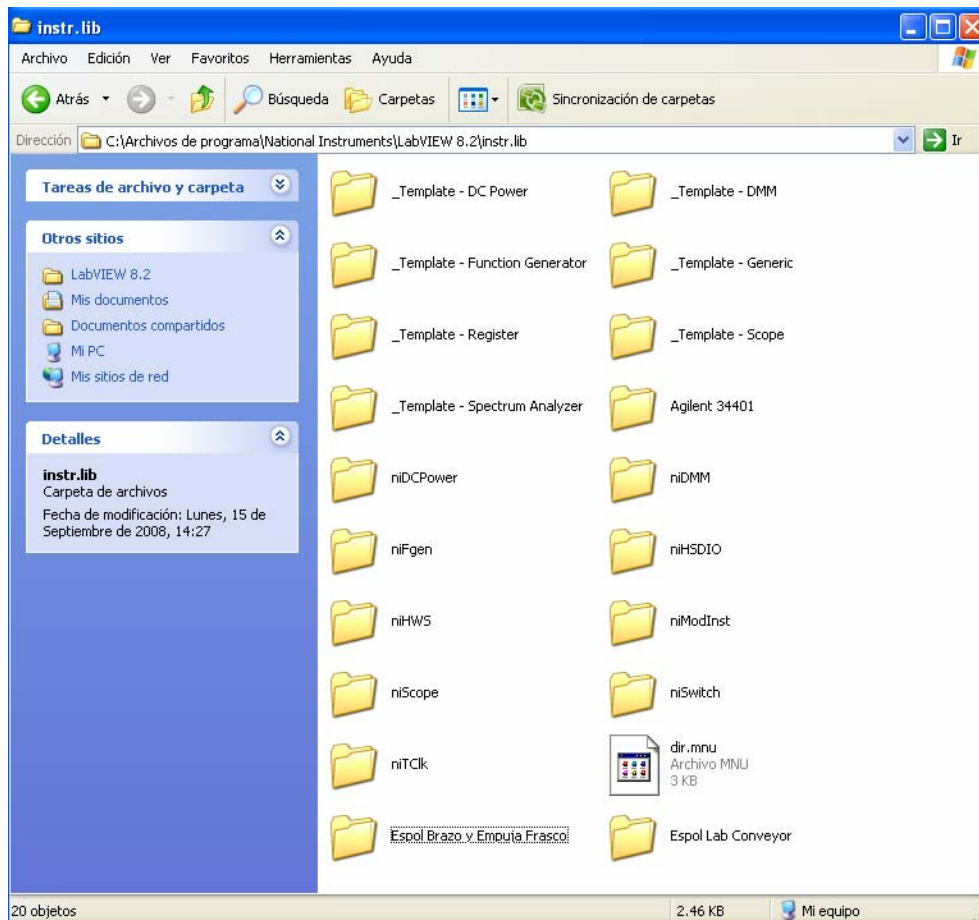


Espol Brazo y Empuja Frasco



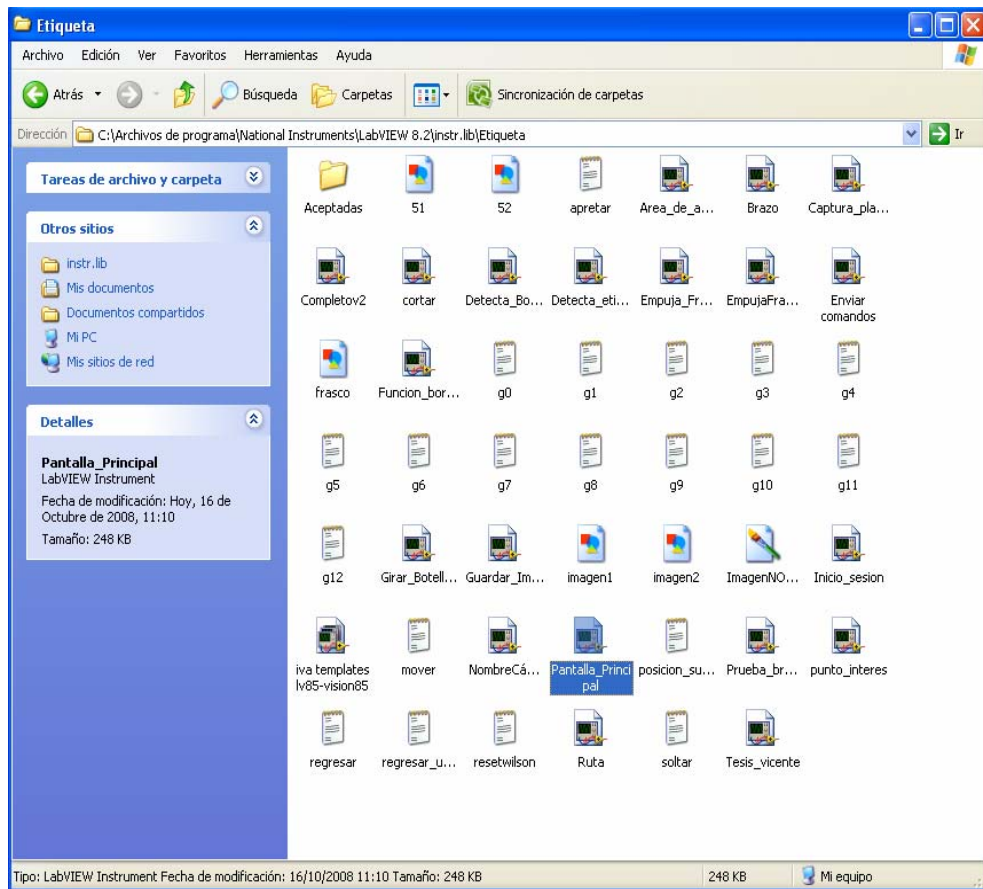
Espol Lab Conveyor

Al final deben aparecer las carpetas de esta manera



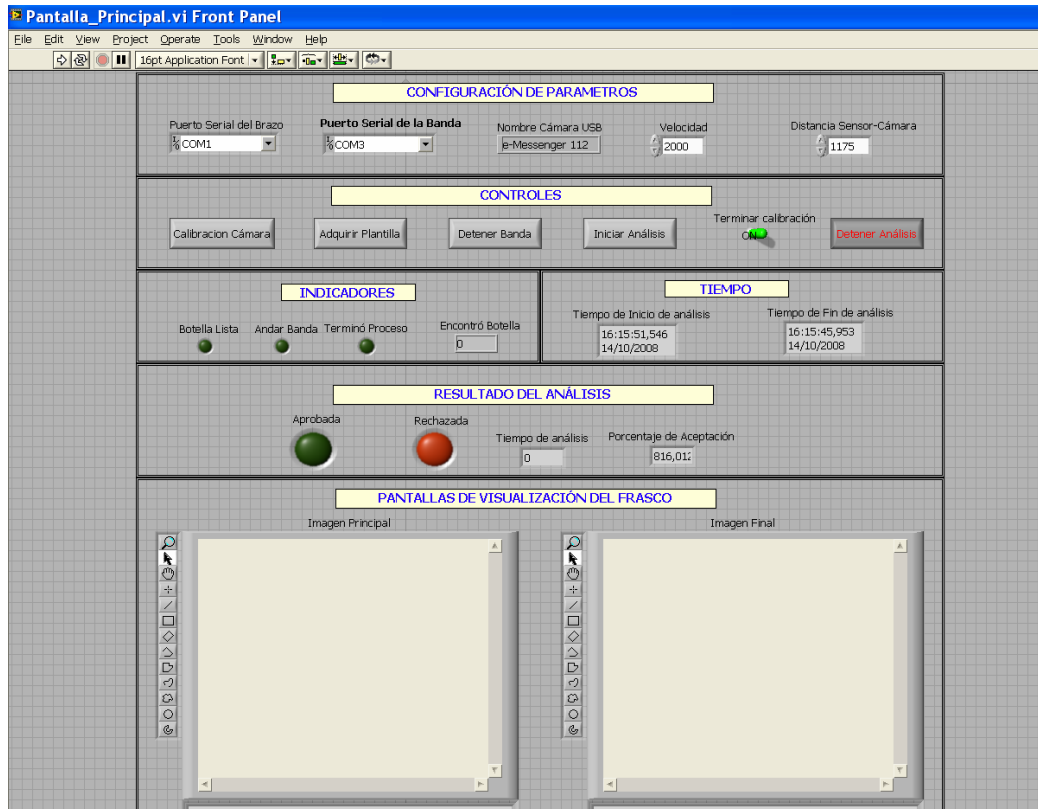
Después buscar la carpeta “Etiqueta” en el disco de instalación, se la copia en la misma dirección descrita anteriormente, esta carpeta contiene un archivo con el nombre de “Pantalla\_Principal”, se da doble clic para abrir.

Nota: No olvidarse de la dirección donde se encuentra la carpeta *etiqueta* porque en esta carpeta se deben guardar las plantillas adquiridas más adelante.





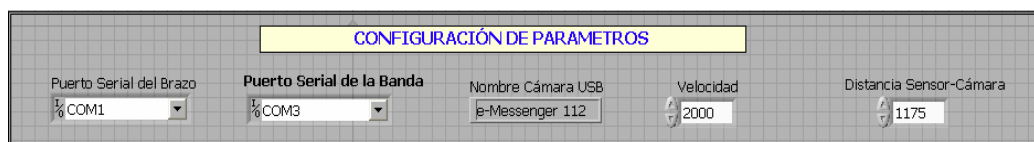
## PANTALLA PRINCIPAL



En la figura anterior se presenta el panel principal de la aplicación el que posee: Variables, indicadores, controles y pantallas de visualización. Estos elementos permiten manipular y observar el proceso de análisis de los frascos.

## PARÁMETROS DE CONFIGURACIÓN

Los parámetros de configuración son muy importantes debido a que definen el variables del proceso.

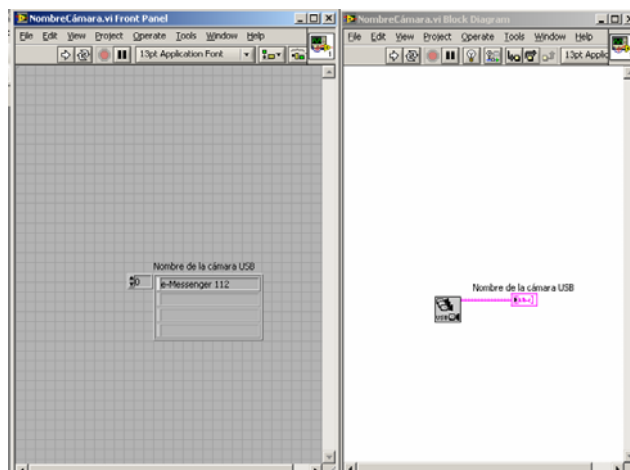


## Puertos seriales

Se deben configurar los puertos seriales de comunicación de la banda transportadora y el brazo robótico de manera correcta.

## Nombre de la cámara

Otra variable a definir es el nombre de la cámara, en caso de no saber el nombre ejecute la siguiente subrutina “Nombre\_Camara.vi”.



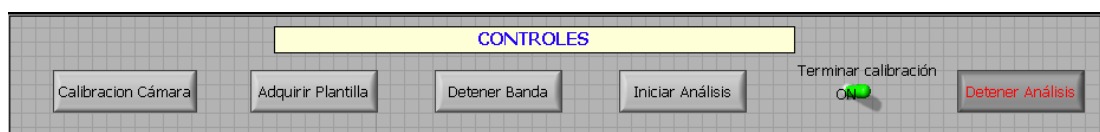
La figura muestra el nombre con el que LabVIEW identifica la cámara usb, en este ejemplo se trabaja con la e-messenger 112.

## Distancia Sensor-Cámara

Definir la distancia entre el sensor y la cámara es muy importante, porque este parámetro depende de la posición

del área de interés, ubicada aproximadamente a una distancia de 20 cm del sensor.

## CONTROLES



Estos proveen la capacidad de ejecutar determinadas acciones como lo son: Calibración de la cámara, adquirir plantilla para el análisis, iniciar análisis.

### Calibración Cámara

La opción de calibración de la cámara se utiliza para enfocar el lente de la cámara y así obtener una imagen clara y definida. Esta operación también permite enfocar el área de interés. Para terminar la calibración se desactiva con un switch de dos posiciones que se encuentra en el panel de controles.

### Adquirir plantilla

Otra opción es la *adquirir plantilla*, con esta opción se puede adquirir nuevos patrones y así acoplarnos a nuevas etiquetas sin tener que realizar grandes cambios en el

programa. En esta parte del programa la primera imagen es la botella completa y la segunda imagen es solo la etiqueta.

Estas imágenes se deben guardar en la carpeta donde se encuentra toda la aplicación.

### **Iniciar análisis**

El control más importante es *iniciar análisis* al presionar esta botonera comienza todo el proceso, mueve la banda transportadora, ubica el brazo robótico ubica el brazo robótico en la posición asignada y el sensor detecta la botella, para posteriormente verificar si posee la etiqueta.

### **Detener análisis**

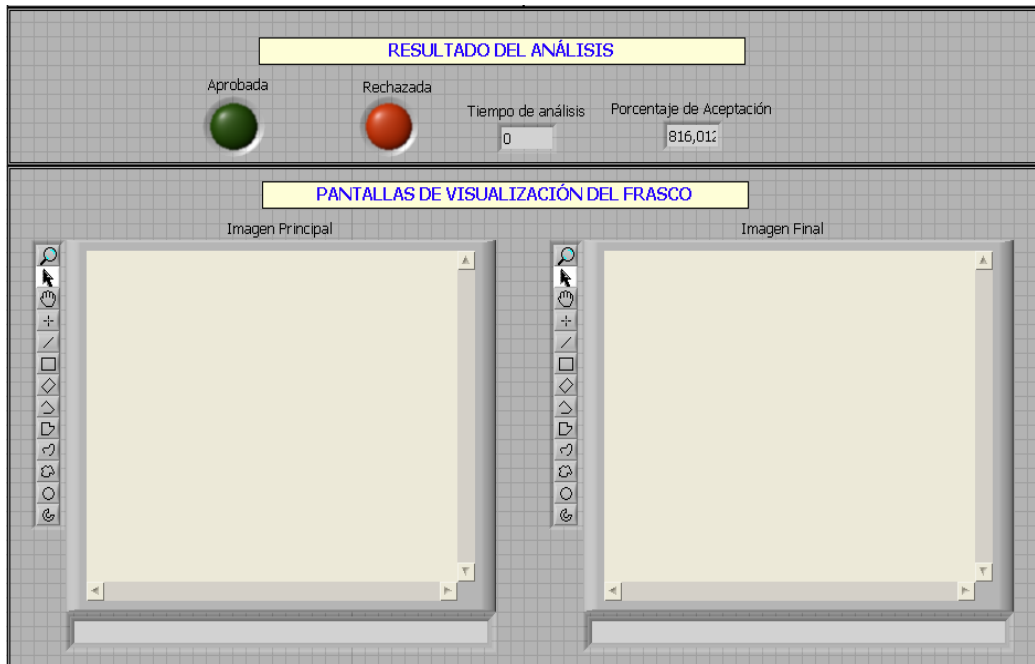
Con esta opción se provee la alternativa de detener el proceso en caso de emergencia y de que se necesite realizar modificaciones en algunos parámetros.

### **Detener banda**

Permite detener la banda transportadora antes que el sensor detecte algún frasco. Usualmente se utiliza cuando no se ha calibrado la cámara.

## INDICADORES

Sirve para visualizar el resultado del análisis del proceso. Se posee 3 tipos de indicadores que se representan según los datos.



### Indicadores booleanos

Presentan mediante dos LED's de manera general si el frasco analizado fue rechazado o aceptado.

### Indicadores numéricos


Sirven para visualizar el porcentaje de aceptación que obtuvo una etiqueta en caso de ser correcta.

## **Indicadores gráficos**

En la pantalla izquierda, se observa la imagen de inicio en donde se identifica el frasco.

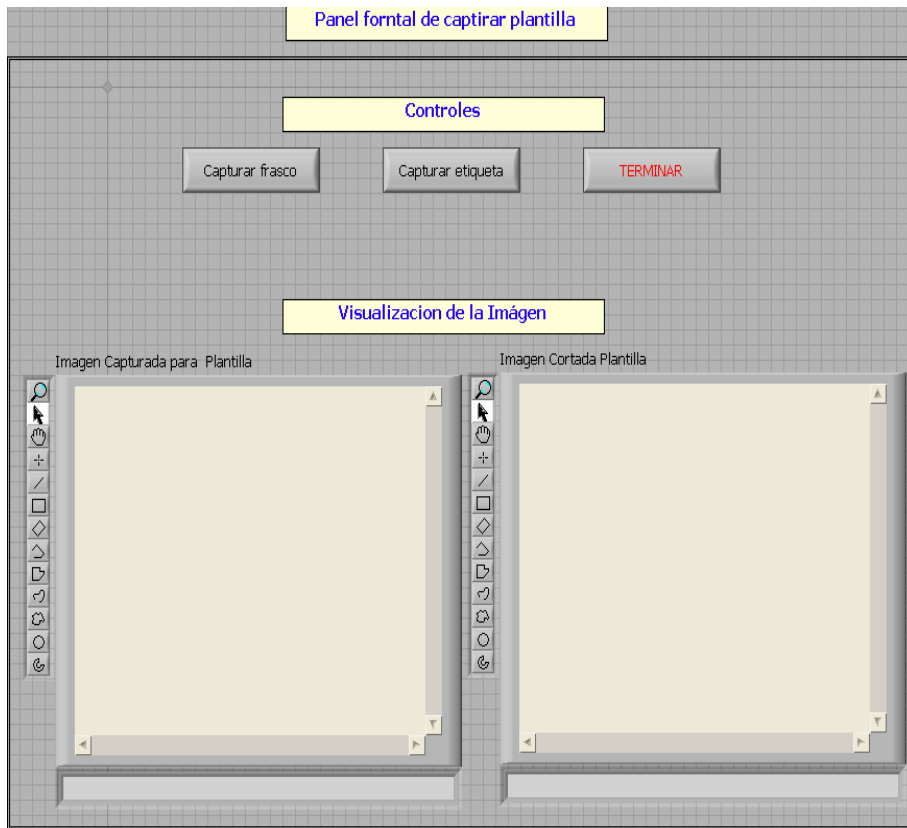
La pantalla de la derecha, muestra la imagen después de todo el proceso de análisis.

## **FUNCIONES IMPLEMENTADAS**

Después de definir parámetros de configuración y conocer el manejo de controles se procede a ejecutar la aplicación, para lograr se presiona la flecha de ejecutar  ubicado en la parte superior derecha.

### **Adquirir plantilla**

Se presiona la botonera respectiva para ejecutar esta acción, a continuación se presenta el panel frontal para capturar la plantilla.



La pantalla de la izquierda sirve de guía para capturar la plantilla de la etiqueta, la cámara debe ver a la etiqueta de manera frontal, la pantalla de la derecha muestra la imagen recortada, esta imagen se utiliza para realizar los respectivos análisis.

Se posee tres opciones: Capturar etiqueta, capturar frasco y terminar captura, estas botoneras nos permiten controlar el orden de captura.

Captura etiqueta se utiliza para realizar la plantilla solo de la etiqueta, la imagen debe ser la siguiente:



Captura frascos se utiliza para obtener la plantilla del frasco en general sin etiqueta, esta plantilla sirve para seccionar la imagen adquirida por la cámara.



Terminar captura permite abandonar el proceso de obtener plantilla, y regresar al menú principal de la aplicación.

### **Calibración de la cámara**

Al ejecutar esta acción se presenta una imagen en el visualizador inferior izquierdo panel principal que está en modo video, esto modo permite realizar la calibración de la posición de la cámara y el enfoque del lente de la misma en tiempo real.



Una vez calibrado el lente y la posición de la cámara se activa el switch *Terminar calibración* para regresar al menú principal.

### Inicio de análisis

Esta es la función principal de toda la aplicación, ya que aquí se realiza el análisis de cada frasco que este en la banda transportadora, para lo cual se utiliza el brazo robótico.

Cuando se está analizando el contorno de todo el frasco se abre una rutina que muestra la siguiente pantalla.



De manera general este panel posee indicadores con los siguientes significados:

**Etiqueta.-** Indica que la etiqueta es la correcta e inmediatamente activa el indicador terminar proceso.

**Contorno.-** Indica que el frasco fue analizado en su totalidad y que no existió ninguna coincidencia con la etiqueta original, también activa el indicador terminar proceso.

**Porcentaje de aceptación y 50.-** Indican que la etiqueta es correcta y cual fue el porcentaje de aceptación con respecto a la plantilla original.

**Pantalla de visualización.-** Muestra una imagen por cada rotación del frasco hasta completar los 360 grados, esto se aprecia como un análisis en movimiento.

## **RESOLVIENDO PROBLEMAS**

Pueden existir errores de ruta al momento de cargar un archivo, por lo que se debe asegurar de que el nombre está bien escrito y que la dirección del archivo es la correcta.

Al realizar la comparación de una imagen a color contra una de blanco y negro existirá un error, debido a que, no existe el mismo nivel de cuantificación en ambas imágenes.

Cuando existe un problema de comunicación de la banda transportadora o el brazo robótico con el software, se debe cerciorar de que ambos elementos estén alimentados correctamente. Si el problema persiste se debe cerrar la aplicación y reiniciar todo los equipos.

Otro problema puede ser la calibración física, donde el brazo robótico no puede capturar la botella para realizar los respectivos giros, para solucionar este problema se debe entrar al modo prueba, donde sólo se analiza la posición en la que se ubica el frasco para el análisis y después de realizar varias pruebas se debe reasignar la distancia sensor – cámara. Este problema se origina generalmente, cuando los compañeros utilizan la banda transportadora y cada uno analiza un parámetro en particular.

# BIBLIOGRAFÍA

[1] LCDO. VINUEZA MAZÓN JOSÉ, "Industria en el Ecuador", imprenta litográfica ORVI,(2006) Biblioteca Municipal

[2] [http://es.wikipedia.org/wiki/Algoritmo\\_divide\\_y\\_vencer%C3%A1s](http://es.wikipedia.org/wiki/Algoritmo_divide_y_vencer%C3%A1s)

[3] Diccionario Océano color, editorial océano, edición 1994, dirección Carlos Gispert.

[4] <http://www.lcc.uma.es/LCCTemario/Asignatura.jsp?idasignatura=24>

[5] "The future of film photography." Digital Photography Centre. HP Digital. 28 Jan. 2006. Disponible en [http://www.hp.ca/portal/hho/dpc/learn/future\\_film\\_photography.php](http://www.hp.ca/portal/hho/dpc/learn/future_film_photography.php).

[6] <http://decsai.ugr.es/mia/teoria/T1/T1bn.pdf>

[7] <http://www.lania.mx/spanish/actividades/newsletters/1999-primavera-verano/aplicaciones.html>

[8] Enciclopedia temática Ilustrada Circulo editorial NORMA SA

[9] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/mendez\\_a\\_ky/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/mendez_a_ky/capitulo1.pdf)

[10] <http://cursos.rvg.ua.es/file.php/19/traspas/segm-reco.pdf>

[11] Dpto. Electrónica, Automática e Informática Industrial, Platero Carlos, Técnicas de preprocesado. Capitulo 4, pdf, Pag. 20

[12] <http://ccc.inaoep.mx/~esucar/Vision/vis05v05-color.ppt>

[13] <http://www.elai.upm.es/spain/Asignaturas/Robotica/ApuntesVA/cap4Procesadov1.pdf>

[14] <http://isa.umh.es/asignaturas/rvc/Tema%205.%20Deteccion%20de%20bordes.pdf>

- [15] Realpe Robalino Miguel, “Hacia la navegación autónoma de robots a partir de la implementación de un método de localización y mapeo simultáneos (SLAM) mediante uso de un sistema de visión de 3D”, 2006, pdf.
- [16] Pallaicho, D y Salamea, V., “Diseño e implementación de un Equipo de Robots Autónomos con Decisiones en Tiempo Real: Fútbol Robótico – Componente Electromecánico”, Tesis, Escuela Superior Politécnica del Litoral, Facultad de Ingeniería en Electricidad y Computación, 2003 pp. 203.
- [17] <http://www.addlink.es/productos.asp?pid=571>  
<http://www.conocimientosweb.net/dcmt/ficha304.htm>
- [18] Martínez Barambio Daniel, Sistema de Visión Artificial, 203, pdf.
- [19] [www.usb.org](http://www.usb.org)  
<http://es.wikipedia.org/wiki/USB>  
<http://www.nullmodem.com/USB.htm>  
[http://multingles.net/docs/alezito/alezito\\_usb.htm](http://multingles.net/docs/alezito/alezito_usb.htm)
- [20] <http://es.wikipedia.org/wiki/Plug-and-play>

[21] <http://es.wikipedia.org/wiki/Microcontrolador>

<http://www.monografias.com/trabajos12/microco/microco.shtml>

[22] [http://es.wikipedia.org/wiki/Procesador\\_digital\\_de\\_se%C3%B1al](http://es.wikipedia.org/wiki/Procesador_digital_de_se%C3%B1al)

[23] Ingeniería de Microsistemas Programados S.L. *Tutorial de PIC School.*, <http://www.microcontroladores.com>, 2006.

[24] Manual de usuario de LabVIEW.