



# **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

## **FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

### **ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES**

#### **"DESARROLLO DE UN SISTEMA AUTOMÁTICO PARA EL CONTROL DE CALIDAD DE PRODUCTOS TERMINADOS, MEDIANTE UN SCÁNER 3D REALIZADO CON TÉCNICAS DE VISIÓN ARTIFICIAL Y TECNOLOGÍA KINECT"**

Trabajo de Titulación presentado para optar al grado académico de:

### **INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES**

**AUTORES:** CRISTIAN ISRAEL LOGROÑO SALAZAR.

KEVIN FERMÍN MISHQUI VILLA

**TUTOR:** ING. ANDRÉS MOROCHO C.

Riobamba-Ecuador

2016

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES  
INDUSTRIALES**

El Tribunal del Trabajo de Titulación certifica que: DESARROLLO DE UN SISTEMA AUTOMÁTICO PARA EL CONTROL DE CALIDAD DE PRODUCTOS TERMINADOS, MEDIANTE UN SCÁNER 3D REALIZADO CON TÉCNICAS DE VISIÓN ARTIFICIAL Y TECNOLOGÍA KINECT, de responsabilidad de los señores: Cristian Israel Logroño Salazar y Kevin Fermín Mishqui Villa, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

Dr. Miguel Tasambay, PhD.

**DECANO**

\_\_\_\_\_

Ing. Jorge Paucar

**DIRECTOR DE ESCUELA  
INGENIERÍA ELECTRÓNICA  
EN CONTROL Y REDES  
INDUSTRIALES**

\_\_\_\_\_

Ing. Andrés Morocho

**DIRECTOR DEL TRABAJO  
DE TITULACIÓN**

\_\_\_\_\_

Ing. Henry Vallejo

**MIEMBRO DEL TRIBUNAL**

\_\_\_\_\_

**DOCUMENTALISTA**

**SISBIB ESPOCH**

\_\_\_\_\_

"Nosotros, **CRISTIAN ISRAEL LOGROÑO SALAZAR** y **KEVIN FERMÍN MISHQUI VILLA**, somos responsables de las ideas, doctrinas y resultados, expuestos en este Trabajo de Titulación; y, el patrimonio intelectual le pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO".

---

Cristian Israel Logroño Salazar

---

Kevin Fermín Mishqui Villa

## **DEDICATORIA**

A mis padres que siempre me apoyaron y motivaron para poder alcanzar mis objetivos profesionales.

**Cristian**

A mis padres que siempre me apoyaron y motivaron para poder alcanzar mis objetivos de mi vida, y a los ingenieros de la FIE por compartir sus enseñanzas y permitirme ser un profesional.

**Kevin**

## **AGRADECIMIENTO**

A la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO y en gran manera al personal docente de la Escuela de Ingeniería en Electrónica, Control y Redes Industriales por haberme sabido formar de excelente forma tanto en la parte profesional como humana, inculcándome valores que me permitirán desenvolverme de manera eficaz dentro del mundo laboral.

**Cristian**

Un especial agradecimiento a mis padres por guiarme en el camino del éxito, gratitud a sus consejos y admiración a sus sacrificios, su esfuerzo desplegado en mi es palpable, a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO y la ESCUELA DE ELECTRÓNICA EN CONTROL por su formación Académica.

**Kevin**

## TABLA DE CONTENIDO

	Páginas
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
TABLA DE CONTENIDO.....	vi
ÍNDICE DE ABREVIATURAS.....	xii
ÍNDICE DE TABLAS.....	xiii
ÍNDICE DE ILUSTRACIONES.....	xiv
ÍNDICE DE ANEXOS.....	xviii
RESUMEN.....	xix
SUMMARY.....	xx
INTRODUCCIÓN.....	1
ANTECEDENTES.....	2
JUSTIFICACIÓN TEÓRICA.....	3
JUSTIFICACIÓN APLICATIVA.....	4
METODOLOGÍA.....	5
OBJETIVO GENERAL.....	6
OBJETIVOS ESPECÍFICOS.....	6
CAPÍTULO I	
<b>1 MARCO TEÓRICO.....</b>	<b>7</b>
<b>1.1 Estado del Arte.....</b>	<b>7</b>
<b>1.2 Visión Artificial.....</b>	<b>9</b>
<b>1.2.1 <i>Imagen Digital</i>.....</b>	<b>11</b>

1.2.1.1	<i>Fundamento de color</i> .....	12
<b>1.2.2</b>	<b><i>Adquisición de Imágenes</i></b> .....	14
<b>1.2.3</b>	<b><i>Procesamiento de Imágenes</i></b> .....	15
1.2.3.1	<i>Transformaciones morfológicas</i> .....	15
1.2.3.1.1	<i>Dilatación Binaria</i> .....	16
1.2.3.1.2	<i>Erosión Binaria</i> .....	16
1.2.3.1.3	<i>Apertura</i> .....	17
1.2.3.1.4	<i>Cierre</i> .....	18
<b>1.2.4</b>	<b><i>Histograma de una imagen</i></b> .....	19
<b>1.2.5</b>	<b><i>Segmentación</i></b> .....	19
1.2.5.1	<i>Umbralización</i> .....	20
1.2.5.2	<i>Segmentación orientada a regiones</i> .....	21
1.2.5.3	<i>Basada en Bordes</i> .....	21
1.2.5.4	<i>Por colores</i> .....	21
<b>1.2.6</b>	<b><i>Iluminación de una escena</i></b> .....	22
1.2.6.1	<i>Iluminación frontal</i> .....	22
1.2.6.2	<i>Iluminación por Retroalimentación</i> .....	22
<b>1.3</b>	<b>Software</b> .....	23
<b>1.3.1</b>	<b><i>Linux Ubuntu 14.04 LTS</i></b> .....	23
<b>1.3.2</b>	<b><i>Lenguaje C++</i></b> .....	23
<b>1.3.3</b>	<b><i>OpenCV</i></b> .....	24
<b>1.3.4</b>	<b><i>CMake</i></b> .....	25
<b>1.4</b>	<b>Kinect</b> .....	25
<b>1.4.1</b>	<b><i>Menú de componentes</i></b> .....	26

<b>1.4.2</b>	<b><i>Visión del Kinect</i></b> .....	27
1.4.2.1	<i>Visión RGB</i> .....	27
1.4.2.2	<i>Imagen DEPTH</i> .....	28
1.4.2.2.1	<i>Luz Estructurada</i> .....	29
<b>1.4.3</b>	<b><i>LibFreenect</i></b> .....	30
<b>1.5</b>	<b>Comunicación Serial</b> .....	31
<b>1.5.1</b>	<b><i>Puerto Serie</i></b> .....	31
<b>1.6</b>	<b>Componentes Eléctricos</b> .....	32
<b>1.6.1</b>	<b><i>Motores Eléctricos</i></b> .....	32
1.6.1.1	<i>Funcionamiento</i> .....	32
1.6.1.2	<i>Clasificación de los motores</i> .....	33
1.6.1.2.1	<i>Motores corriente continua</i> .....	33
1.6.1.2.2	<i>Motores corriente alterna</i> .....	34
<b>1.6.2</b>	<b><i>Luz piloto</i></b> .....	34
<b>1.6.3</b>	<b><i>Selector</i></b> .....	35
<b>1.6.4</b>	<b><i>Relé</i></b> .....	35
1.6.4.1	<i>Funcionamiento</i> .....	35
1.6.4.2	<i>Tipos de relés</i> .....	36
<b>1.6.5</b>	<b><i>Sensores</i></b> .....	36
1.6.5.1	<i>Funcionamiento de los sensores</i> .....	37
1.6.5.2	<i>Tipos de sensores</i> .....	39
1.6.5.2.1	<i>Sensores inductivos</i> .....	39
1.6.5.2.2	<i>Sensores de contacto</i> .....	40
1.6.5.2.3	<i>Sensores ultrasónicos</i> .....	40

<b>1.7</b>	<b>Bandas Transportadoras</b> .....	42
1.7.1	<i>Aplicación</i> .....	43
1.7.2	<i>Sistemas de transportadores de bandas</i> .....	43
1.7.3	<i>Ventajas</i> .....	44
<b>1.8</b>	<b>Arduino</b> .....	44
1.8.1	<i>Definición</i> .....	44
1.8.2	<i>Ventajas de usar Arduino</i> .....	44
1.8.3	<i>Hardware</i> .....	45
<b>1.9</b>	<b>Control de Calidad</b> .....	46
1.9.1	<i>Etapas para el control de calidad</i> .....	46
1.9.2	<i>Inspección</i> .....	47
1.9.3	<i>Esquema de un proceso de inspección</i> .....	48
1.9.4	<i>Modalidades de inspección</i> .....	48
<b>1.10</b>	<b>Indicadores de calidad</b> .....	49
1.10.1	<i>Sistema de medición</i> .....	49
1.10.2	<i>Criterio de calidad del producto</i> .....	49
1.10.3	<i>Indicadores de calidad</i> .....	50
 CAPÍTULO II		
<b>2</b>	<b>MARCO METODOLÓGICO</b> .....	51
2.1	<b>Obtención y procesamiento de escenas Tridimensionales</b> .....	51
2.1.1	<i>Método RGB</i> .....	51
2.1.2	<i>Método Estereográfico</i> .....	52
2.1.3	<i>Método Time-of-flight</i> .....	53
2.1.4	<i>Método de luz estructurada</i> .....	53

2.2	<b>Arquitectura de los equipos receptores</b> .....	54
2.3	<b>Comunicación entre la PC y el sistema automático</b> .....	57
2.4	<b>Diseño de la banda para la clasificación de tazas</b> .....	59
2.4.1	<i>Bandas Transportadoras</i> .....	59
2.4.2	<i>Partes principales de una banda Transportadora</i> .....	59
2.4.2.1	<i>Recubrimiento</i> .....	59
2.4.3	<i>Motor</i> .....	60
2.4.4	<i>Los Rodillos</i> .....	63
2.4.5	<i>Tensores de banda</i> .....	65
2.4.6	<i>Pallets</i> .....	65
2.4.7	<i>Sensores</i> .....	66
2.4.8	<i>Diseño final de la banda Transportadora</i> .....	67
2.5	<b>Algoritmo de reconocimiento para comparación y clasificación</b> ....	68
2.5.1	<i>Instalación de Software requerido</i> .....	68
2.5.1.1	<i>Instalación de LibFreenect</i> .....	68
2.5.1.2	<i>Instalación de OpenCV</i> .....	69
2.5.2	<i>Instalación del IDE de Arduino</i> .....	71
2.5.2.1	<i>Descarga del IDE de Arduino</i> .....	71
2.5.2.2	<i>Ejecutar Arduino</i> .....	71
2.5.3	<i>Tratamiento de la Visión del Kinect</i> .....	72
2.5.4	<i>Obtención del dato de profundidad</i> .....	73
2.5.4.1	<i>Fórmula matemática de transformación</i> .....	74
2.5.4.1.1	<i>Caso 0</i> .....	74
2.5.4.1.2	<i>Caso 1</i> .....	75

2.5.4.1.3	<i>Caso 2</i> .....	75
2.5.4.1.4	<i>Caso 3</i> .....	76
2.5.4.1.5	<i>Caso 4</i> .....	76
<b>2.5.5</b>	<b><i>Desarrollo del algoritmo</i></b> .....	<b>76</b>
2.5.5.1	<i>Obtención tazas modelo</i> .....	76
2.5.5.2	<i>Procesado de las imágenes</i> .....	77
2.5.5.3	<i>Algoritmo de Umbralización</i> .....	77
2.5.5.3.1	<i>Proceso de implementación</i> .....	77
<b>2.6</b>	<b>Interfaz de información del estado de los objetos</b> .....	<b>80</b>
<b>CAPÍTULO III</b>		
<b>3</b>	<b>MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN</b> .....	<b>82</b>
<b>3.1</b>	<b>Control de los productos terminados</b> .....	<b>82</b>
<b>3.2</b>	<b>Diagrama de selección del producto</b> .....	<b>85</b>
<b>3.3</b>	<b>Operación de actuadores</b> .....	<b>86</b>
<b>3.4</b>	<b>Resultados del scáner en dimensiones</b> .....	<b>87</b>
<b>3.5</b>	<b>Sistema de control de lazo abierto</b> .....	<b>89</b>
<b>CONCLUSIONES</b> .....		<b>94</b>
<b>RECOMENDACIONES</b> .....		<b>96</b>
<b>BIBLIOGRAFÍA</b>		
<b>ANEXOS</b>		

## ÍNDICE DE ABREVIATURAS

ESPOCH	Escuela Superior Politécnica de Chimborazo.
fps	Frames Por Segundo.
HSV	Hue, Saturación, Value.
IR	Infrarrojo.
m	metros.
cm	centímetros.
mW	mili Watts.
mm	Milímetros.
PC	Computador Personal.
V.A.	Visión Artificial.
3D	Tres Dimensiones.
2D	Dos Dimensiones.
RGB	Red, Green, Blue.
VGA	Video Graphics Array
USB	Universal Serial Bus
VDC	Voltaje Corriente Directa.
lb	Libras
HP	Horsepower
P	Potencia
I	Corriente
V	Voltaje
W	watts
MRF	Markov Random Field.

## ÍNDICE DE TABLAS

<b>Tabla 1-3:</b>	Control de la fase de finalización del producto.....	83
<b>Tabla 2-3:</b>	Valores para modelo 1.....	87
<b>Tabla 3-3:</b>	Valores para modelo 2.....	88
<b>Tabla 4-3:</b>	Análisis de datos de las tazas de prueba.....	90

## ÍNDICE DE ILUSTRACIONES

<b>Figura 1-1:</b>	Proceso general de un sistema de Visión Artificial.....	10
<b>Figura 2-1:</b>	Longitud de onda de colores visibles.....	12
<b>Figura 3-1:</b>	Formato de imagen HSV.....	14
<b>Figura 4-1:</b>	Dilatación binaria.....	16
<b>Figura 5-1:</b>	Erosión binaria.....	17
<b>Figura 6-1:</b>	Apertura.....	18
<b>Figura 7-1:</b>	Imagen tras proceso de cierre.....	18
<b>Figura 8-1:</b>	Instalación de C++ en UBUNTU.....	24
<b>Figura 9-1:</b>	Kinect XBOX 360.....	25
<b>Figura 9-2:</b>	Descripción de motores DC.....	25
<b>Figura 10-1:</b>	Proyección del infrarrojo del kinect.....	28
<b>Figura 11-1:</b>	Rango nominal de visión de profundidad del kinect.....	29
<b>Figura 12-1:</b>	Imagen Depth y RGB en ausencia de luz.....	30
<b>Figura 13-1:</b>	Logo Open Kinect.....	30
<b>Figura 14-1:</b>	Diagramas comunicación serie y paralelo.....	31
<b>Figura 15-1:</b>	Motor de corriente continua 24V.....	33
<b>Figura 16-1:</b>	Carcasa motor de corriente alterna.....	34
<b>Figura 17-1:</b>	Luces piloto de 24V.....	34
<b>Figura 18-1:</b>	Selector de tres posiciones.....	35
<b>Figura 19-1:</b>	Esquema típico de un relé.....	36
<b>Figura 20-1:</b>	Sensor de distancia infrarrojo (Sharp).....	41

<b>Figura 21-1:</b>	Curva de funcionamiento sensor infrarrojo (Sharp).....	41
<b>Figura 22-1:</b>	Cinta Transportadora.....	42
<b>Figura 23-1:</b>	Esquema básico de un proceso de inspección.....	48
<b>Figura 24-1:</b>	Modelo de taza reconstruida digitalmente.....	50
<b>Figura 1-2:</b>	Mesa para Kinect.....	55
<b>Figura 2-2:</b>	Kinect 1.....	55
<b>Figura 3-2:</b>	Kinect 2.....	56
<b>Figura 4-2:</b>	Ubicación Kinect 1.....	56
<b>Figura 5-2:</b>	Ubicación Kinect 2.....	57
<b>Figura 6-2:</b>	Comunicación Serie.....	57
<b>Figura 7-2:</b>	Microcontrolador Arduino uno R3.....	58
<b>Figura 8-2:</b>	Recubrimiento.....	60
<b>Figura 9-2:</b>	Descripción de Motores DC.....	62
<b>Figura 10-2:</b>	Motor de 24VDC.....	63
<b>Figura 11-2:</b>	Sistema de ubicación de Rodillos.....	64
<b>Figura 12-2:</b>	Rodillos.....	64
<b>Figura 13-2:</b>	Tensores de Banda.....	65
<b>Figura 14-2:</b>	Pallet.....	66
<b>Figura 15-2:</b>	Acoplamiento de sensor Sharp.....	66
<b>Figura 16-2:</b>	Vista lateral de la banda.....	67
<b>Figura 17-2:</b>	Vista general de la banda.....	67
<b>Figura 18-2:</b>	Synaptic de Ubuntu 14.04 LTS.....	68
<b>Figura 19-2:</b>	Instalación de paquetes necesarios para el OpenCV.....	69
<b>Figura 20-2:</b>	Compilación del paquete OpenCV.....	69

<b>Figura 21-2:</b>	Configuración del OpenCV.....	70
<b>Figura 22-2:</b>	Configuración de las rutas de las librerías instaladas.....	70
<b>Figura 23-2:</b>	Código de generación de Bash.....	70
<b>Figura 24-2:</b>	Configuración final del PKG_CONFIG_PATH.....	70
<b>Figura 25-2:</b>	Descarga Arduino modo consola .....	71
<b>Figura 26-2:</b>	Ejecutar Arduino modo consola .....	71
<b>Figura 27-2:</b>	IDE de Arduino .....	72
<b>Figura 28-2:</b>	Doble visión del Kinect .....	72
<b>Figura 29-2:</b>	Bits de información DEPTH.....	73
<b>Figura 30-2:</b>	Rango de bits – profundidad del Kinect.....	73
<b>Figura 31-2:</b>	Datos binarios puros DEPTH del Kinect.....	74
<b>Figura 32-2:</b>	Llamada a los datos de los dos videos .....	77
<b>Figura 33-2:</b>	Trackbars de los parámetros .....	78
<b>Figura 34-2:</b>	Imagen Umbralizada .....	78
<b>Figura 35-2:</b>	Cálculo de la posición del objeto .....	79
<b>Figura 36-2:</b>	Código Condición de muestreo en el DEPTH .....	79
<b>Figura 37-2:</b>	Cálculo áreas de las tazas .....	79
<b>Figura 38-2:</b>	Cálculo del valor absoluto y margen de error.....	80
<b>Figura 39-2:</b>	Interfaz áreas de las tazas modelos .....	80
<b>Figura 40-2:</b>	Interfaz tazas similares o distintas .....	81
<b>Figura 41-2:</b>	Interfaz número de tazas probadas .....	81
<b>Figura 42-2:</b>	Interfaz general.....	81
<b>Figura 1-3:</b>	Esmaltado Taza Modelo.....	83
<b>Figura 2-3:</b>	Forma y Modelado de Taza Modelo.....	83

<b>Figura 3-3:</b>	Área de Taza Modelo.....	83
<b>Figura 4-3:</b>	Diagrama de funcionamiento selección del producto.....	85
<b>Figura 5-3:</b>	Diagrama operación del sistema de escaneo.....	86
<b>Figura 6-3:</b>	Resultados prueba scanner modelo 1.....	88
<b>Figura 7-3:</b>	Resultados prueba scanner modelo 2.....	89
<b>Figura 8-3:</b>	Representación sistema de control.....	89
<b>Figura 9-3:</b>	Taza Modelo.....	90
<b>Figura 10-3:</b>	Forma Taza Modelo.....	90
<b>Figura 11-3:</b>	Taza mal estado.....	90
<b>Figura 12-3:</b>	Forma taza sin oreja.....	90
<b>Figura 13-3:</b>	Taza fisurada.....	90
<b>Figura 14-3:</b>	Forma taza fisurada.....	90
<b>Figura 15-3:</b>	Taza con Fisura superior.....	91
<b>Figura 16-3:</b>	Modelo con Fisura superior.....	91
<b>Figura 17-3:</b>	Pallet sin objeto.....	91
<b>Figura 18-3:</b>	Sin forma.....	91
<b>Figura 19-3:</b>	Interfaz.....	92
<b>Figura 20-3:</b>	Medición en eje z.....	92
<b>Figura 21-3:</b>	Presentación de eje z.....	93

## ÍNDICE DE ANEXOS

- ANEXO A.** CARACTERÍSTICAS DE BANDAS COMERCIALES.
- ANEXO B.** UNIDAD DE CONTROL.
- ANEXO C.** RAMPA DE CLASIFICACIÓN DE TAZAS CON FALLAS.
- ANEXO D.** DATASHEET SENSOR GP2Y0A21YK.
- ANEXO E.** DATASHEET SERVOMOTOR SG90.
- ANEXO F.** MICROCONTROLADOR DE ARDUINO UNO R3.
- ANEXO G.** SISTEMA PROTOTIPO DE CONTROL DE CALIDAD.

## RESUMEN

Se desarrolló un sistema automático prototipo para el control de calidad de productos terminados, utilizando un scáner 3D realizado con técnicas de visión artificial y tecnología Kinect en la Escuela de Ingeniería Electrónica en Control y Redes Industriales (EIE-CRI) de la Escuela Superior Politécnica de Chimborazo. Se diseñó la arquitectura más adecuada para la ubicación de los equipos receptores y tener una réplica completa del objeto. Se establece comunicación entre la PC y el prototipo a través del puerto serial mediante la utilización del microcontrolador Arduino, se diseñó e implementó una maqueta a escala de la etapa del proceso de control de calidad de una industria que consta de: banda transportadora accionada por un motor de 24 Voltios Corriente Continua (VDC), un servo motor de 5 Voltios Corriente Continua (VDC) para el desecho de productos en mal estado, sensor óptico para detectar si el objeto está dentro del scáner, dos sensores kinect para escanear las tazas cerámicas, tablero de control para activar y desactivar todo el sistema del scáner 3D. El algoritmo del scáner se programó en el sistema operativo Ubuntu 14.04. y código abierto, para ello se usó el lenguaje de programación C++, trabajando en conjunto con las librerías de visión artificial como: OpenCV y Libfreenect. El dispositivo kinect actúa como scáner permitiendo obtener las características que presenta un objeto, luego compara con un diseño modelo y finalmente decide si el producto es de buena o mala calidad, también consta de una interfaz gráfica por computador que muestra información del estado del producto así como: área, posición y cantidad de muestras. El scáner recepta 100 escenas de una taza cerámica por día, para determinar el buen funcionamiento del prototipo se toman como muestras del 10% dando como margen de error  $\pm 1.13$ . En conclusión el sistema se desenvuelve de una manera eficiente manejando un error bastante bajo. Se recomienda para la implementación en la Empresa Cerámica Andina, implementar el scáner totalmente cubierto para que no tenga alguna incidencia de luz externa que pueda afectar su óptimo desempeño.

**Palabras Clave:** <TÉCNICAS DE VISIÓN ARTIFICIAL> <SISTEMA AUTOMÁTICO>  
<CONTROL DE CALIDAD> <SOFTWARE OPENCV> <SOFTWARE LIBFREENECT>  
<DISPOSITIVO KINETC> <SISTEMA OPERATIVO UBUNTU> <CONTROL Y REDES INDUSTRIALES> <INDUSTRIAS DE CERÁMICAS>

## SUMMARY

An automatic system was developed in order to control the quality of the final products. This system was designed through the use of a 3-D scanner and the artificial and Kinect technology at the Ingenieria Electronica en Control y Redes Industriales (EIE-CRI) of the Escuela Superior Politecnica de Chimborazo. The most suitable architecture was designed to locate the receiving equipment and to have an entire copy of the object. A connection between the PC and the system is established through a serial port and the use of the Arduino microcontroller. A scale model was designed and implemented according to the scale of the process of the quality control of an industry. This industry consist on a transport conveyor activated by a 24 volts engine, Voltage of Continuos Current (VDC), servo motor of 5 volts Continuos Current (VDC) to depose the products in bad conditions, optical sensor to detect if the object is within the scanner, two Kinect sensors to scan the ceramic mugs, control board to active and deactivate the whole system of the 3-D scanner. The algorithm of this scanner was programmed in the language C++ was applied, working together with the libraries of artificial vision like: Open CV and Libfreenect. The device Kinect acts a scanner allowing to get the characteristics that an object presents. After that, it is compared with a designed model and finally it decides if the product is of good or bad quality, it also consist on a plotted interface per computer that shows data of the state of the product like: area, position and amount of samples. The scanner receives 100 scenes of the ceramic mug per day in order to determine the performance of the system. It takes 10% of the samples and gives the margin of  $\pm 1.13$ . In conclusion, the system works in an efficient way and presents a low degree of error. A suggestion for the Empresa Ceramica Andina is to implement the scanner totally covered so it does not have any light influence that will affect its optimal performance.

**Key words:** <ARTIFICIAL VISION TECHNIQUES> <AUTOMATIC SYSTEM>  
<QUALITY CONTROL> <SOFTWARE OPENCV> <SOFTWARE LIBFREENECT>  
<KINETC DEVICE> <UBUNTU> <OPERATING SYSTEM> <INDUSTRIAL CONTROL  
AND NETWORKS> <INDUSTRIAL CERAMICS>

## **INTRODUCCIÓN**

### **SÍNTESIS**

La visión artificial es una rama de la inteligencia artificial, que por sus prestaciones y su potente versatilidad en la implementación de aplicaciones en muchos campos de la ciencia, ha ido ganando su espacio exclusivo dentro del desarrollo actual de la tecnología y las mejoras informáticas.

Desde el nacimiento mismo de la visión artificial esta busca una sola cosa: entender con precisión una escena captada. Entonces las aplicaciones se extienden por casi todas las ciencias existentes. El progreso de la visión artificial debe asemejarse a la del humano.

El crecimiento de esta especialidad va de la mano con las innovaciones tecnológicas tanto de hardware como de software. Podríamos citar desde cámaras con más resolución hasta librerías de visión artificial más potentes y con mejores funcionalidades.

El control de calidad de productos es un ámbito en el cual también se encuentra presente la visión artificial. En la actualidad se puede hacer un control del estado de los productos terminados usando cámaras con las cuales obtenemos datos de los objetos, así como sus dimensiones. El uso de cámaras con datos de profundidad ayudan a que mediante un scáner 3D también sea posible realizar un control de calidad de productos terminados eficaz, pero al mismo tiempo tardará más de lo normal, ya que se adquieren muchos datos y estos deberán ser procesados para la posterior reconstrucción de la imagen.

En el 2011, al abrirse al público un controlador libre para computador del sensor KINECT, lanzado por Microsoft, se dio un gran salto en el avance de cámaras aplicadas a la visión artificial. Las posibilidades de estudio con este dispositivo son variadas y de una cobertura realmente atractiva (Quirarte, *et al.*, 2013).

En el presente trabajo se ha desarrollado un scáner 3D para el control de calidad de productos terminados, con el algoritmo idóneo, usando dos sensores Kinect y código abierto como

plataforma de desarrollo. El proyecto ha puesto hincapié en desarrollar un algoritmo que permita aprovechar las prestaciones necesarias del Kinect.

Muchas herramientas se han desarrollado para el servicio de la visión artificial. Las más importantes son las librerías orientadas. Aunque existen una cantidad considerable de librerías una ha sobresalido entre estas por su versatilidad, potencia y aplicaciones. Además de tener una gran documentación y de que es libre para el público, OpenCV (Marín, 2015).

En orientación del campo industrial el siguiente trabajo ha propuesto un control de calidad mediante un scáner realizado con técnicas de visión artificial y tecnología Kinect. El mismo que servirá para cualquier industria que lo requiera (Quirarte, *et al.*, 2013).

Este documento pretende llegar a ser una herramienta de estudio sobre visión artificial y su aplicación en el campo de las industrias, para los entes académicos de todos los niveles de educación, y que a partir de la misma, se puedan generar muchos trabajos posteriores que revelen la calidad de estudio que tiene la institución educativa a la que pertenece.

## **ANTECEDENTES.**

En la última década se han realizado grandes avances en la captura de la estructura tridimensional de escenas u objetos reales. Las crecientes capacidades de cálculo hacen que hoy en día sea posible procesar volúmenes de datos anteriormente intratables, utilizando y mejorando sus técnicas y haciendo más rápidas y precisas. Esta capacidad de modelar dentro de un computador escenas del mundo real es muy ventajosa en áreas como la industria, el diseño industrial o las artes gráficas.

El reconocimiento de la profundidad en una escena receptada por un dispositivo de captura de imágenes puede ser englobado dentro del ámbito de la visión por computador. La captura de las características tridimensionales es una de las cualidades en las que se puede centrar un proceso de visión por computador. Las técnicas para capturar la geometría tridimensional de una escena y transferirla a un modelo matemático practicable por un computador son varias.

Una de las etapas fundamentales en un proceso productivo es sin duda el control de calidad, que permite realizar una inspección de los productos previo a la salida del mercado. Y con esto se pueden realizar correcciones o directamente del desecho del objeto.

Surge la necesidad en una empresa que su control de calidad sea más eficiente, lo cual se plantea un control preciso de verificación de errores y posteriormente a su clasificación. Actualmente la identificación de daños en piezas finales se realizan mediante la percepción humana y eso ha generado controversia en la finalización de sus procesos, puesto que conlleva de mucho tiempo y mano de obra.

Eventualmente se ha constatado que ciertos objetos disponen de diferentes formas, profundidades y medidas, diferenciándose de las características verdaderas que debe poseer un producto final para la satisfacción del cliente y acreditación de la empresa.

## **JUSTIFICACIÓN**

- **JUSTIFICACIÓN TEÓRICA**

En la actualidad los sistemas de scáner 3D han evolucionado satisfactoriamente y su aplicación se ha centrado específicamente al diseño, sin embargo se planteó implementarlo en una línea de producción de una industria, específicamente al control de calidad de productos terminados, por lo que el sistema automático que se realizó permite comprobar la calidad del producto a partir de un patrón específico. Procedente a esto se puede tomar ciertas medidas con los resultados obtenidos.

El sistema es desarrollado completamente en el sistema operativo Ubuntu 14.04 plataforma Linux, es decir, que no se tendrá que pagar por alguna licencia o software propietario, lo que le da un valor agregado al proyecto, también brinda a los usuarios una ventaja, ya que los programadores de visión artificial en código abierto comparten sus proyectos y sus algoritmos con la comunidad, lo cual ayuda a ser mejorado utilizando el código original.

La visión artificial tiene muchas prestaciones, por ende se ha decidido aplicar a este trabajo: adquisición de datos y características tridimensionales que permita obtener medidas del producto

final y hacer una comparación minuciosa. De tal forma que se implementará un sistema de control en lazo abierto, ya que los artículos denominados en mal estado tendrán que ser desechados y continuar otro proceso de re fabricación.

El sistema de escáner 3D opera simultáneamente con la banda transportadora y el servomotor, realizando un proceso de clasificación automática de los objetos.

- JUSTIFICACIÓN APLICATIVA

El sistema de escáner 3D que se implementó utiliza técnicas de visión artificial para la detección de errores en los productos finales de una industria.

El prototipo consta de un sistema de sensores kinect conectadas a un computador, ubicados estratégicamente de forma que cubran los 360° del objeto a escanear, dichos dispositivos harán la captura de la escena mediante imágenes, obteniendo datos de profundidad, medidas tridimensionales, luego estos datos obtenidos por los dispositivos receptores de imágenes serán procesados, analizados y reconstruidos en 3D mediante modelos matemáticos en la computadora.

Finalmente se realizará una comparación y verificación de fallas que ayudarán a la clasificación de los productos ya sea desecharlos o que continúen su proceso.

Los algoritmos serán desarrollados con la ayuda de las librerías de OpenCV, C++, Openkinect en Linux (Marín, 2015).

Los productos finales son tazas de cerámica. Puesto que existe diferencias en sus respectivas formas, color y sus dimensiones se prestan para el acople de una maqueta. Por lo que se plantea con el sistema automático de control de calidad es verificar las características del producto escaneado y clasificarlo de acuerdo a su estado.

Se utiliza un sistema de banda transportadora, para el desplazamiento de la taza y una palanca accionada por servomotor que desechará al artículo terminado en mal estado. Además se dispone de un sensor óptico para determinar que el objeto se encuentra dentro del scáner.

Para la activación de la banda transportadora se utilizó un motor DC de bajo voltaje, controlado por un computador mediante una etapa de potencia, este trabajo se realizará a pequeña escala, y es preciso aclarar que esta técnica de control de calidad con visión artificial puede ser aplicada a grandes industrias.

## **METODOLOGÍA**

Para realizar este trabajo se utilizó el método experimental, mediante la observación del funcionamiento del proceso actual y realizando pruebas que permiten hallar la mejor alternativa de automatización, analizando tecnologías existentes dentro del ámbito de automatización de procesos industriales.

Se desarrolló un sistema automático para el control de calidad de tazas cerámicas, que consta de una banda transportadora, un sistema de Kinect que realiza la verificación del estado de las tazas, una etapa de control que nos permite activar y desactivar el sistema, un sensor que detecta la presencia de objetos sobre la banda.

Se realizaron diferentes experiencias y modificaciones destinadas a encontrar la mejor solución de automatización del proceso, permitiendo satisfacer los requerimientos presentados, así como mejorar el rendimiento de los elementos y dispositivos que forman parte del mismo.

Una interfaz de monitoreo muestra el número de tazas sujetas al control de calidad. Así como también el número de tazas de buena y mala calidad. Con esto realizamos un conteo del total de tazas producidas.

Por último este trabajo permite alcanzar una comprensión más profunda en cuanto a distintos aspectos de la ingeniería electrónica como son los sistemas de control aplicados a la industria, visión artificial, uso de sensores e instrumentación, enriqueciendo de tal modo los conocimientos que como futuros ingenieros debemos ser capaces de aplicar.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

- Desarrollar un sistema automático para control de calidad de productos terminados, mediante un scáner 3D realizado con técnicas de visión artificial y tecnología Kinect.

### **OBJETIVOS ESPECÍFICOS**

- Investigar la obtención y procesamiento de escenas tridimensionales a través de la tecnología Kinect en sistema Linux.
- Determinar la arquitectura adecuada de los dispositivos receptores, para la correcta adquisición de las características tridimensionales del objeto.
- Establecer una comunicación entre la PC y el sistema automático de control de calidad.
- Diseñar e implementar un sistema de bandas transportadoras para la clasificación de las tazas de cerámica.
- Desarrollar un algoritmo de reconocimiento de patrones para la comparación y clasificación de las tazas cerámicas mediante bandas transportadoras.
- Desarrollar una interfaz que permita informar las características que presentan los objetos en mal estado.
- Desarrollar un estudio de fidelidad y alcance de resultados.

# CAPÍTULO I

## 1. MARCO TEÓRICO.

En el presente capítulo se presenta los conceptos teóricos de las distintas herramientas ocupadas para la consecución nuestra investigación. Además, la actualidad del desarrollo de los avances de la Visión Artificial (en adelante V.A.) con Kinect.

### 1.1 Estado del Arte.

Desde el 2011, cuando el controlador del Kinect fue liberado, las aplicaciones con el mismo se han ido incrementando y han ocupado desarrollos en distintos campos de la ciencia. Es cierto que los trabajos de gran presupuesto llaman la atención y estas le pertenecen a los países que siempre están en la cumbre de la tecnología. Pero también Latinoamérica se ha presentado en la actualidad con grandes proyectos de investigación (Quirarte, *et al.*, 2013).

Al momento de la creación del SDK Kinect de Microsoft se pretendía que solo se use como plataforma de investigación para el dispositivo, dando de baja a las versiones libres existentes, sin embargo las bondades del Open Source han hecho que cualquier persona con un Kinect a la mano y conocimientos de programación pueda inventar aplicaciones usando controladores que no tienen nada que ver con Microsoft.

Hay que recalcar que también los controladores libres del Kinect no han dejado de desarrollarse a pesar del SDK de Microsoft. Algunos de ellos, como el LibFreenect aún tienen muchas utilidades por abarcar, pero su constitución actual tiene una gran capacidad. Otro controlador es el desarrollado por el Open NI, que también tiene muchos recursos óptimos, aunque en la actualidad este fue adquirido por una gran empresa y es incierto si se continuará su desarrollo (Quirarte, *et al.*, 2013).

Entre algunos trabajos que han conseguido el renombre público podríamos mencionar:

**Turtlebot.-** Es un conocido robot que se moviliza con un robot iCreate y la visión del Kinect, conectado a la laptop. El código de este robot es abierto, así que los desarrolladores pueden modificarlo para intereses más específicos, una de las aplicaciones más conocidas del robot es servir como camarero. El robot se puede adquirir en la página oficial del mismo <http://turtlebot.com>.

**Nuevo asistente de quirófano.-** Su propósito es evitar que al momento de consultar el expediente del paciente, el cirujano no deba usar las manos para evitar cualquier tipo de contagio. Esta labor la realiza con ayuda de los múltiples sensores del Kinect. Este proyecto resultó ganador en un concurso por impulsar el estudio del Kinect (García, 2011).

En Ecuador, la investigación con el Kinect, también ha tenido su apertura por parte de las tesis de grado y de educación de cuarto nivel. Aunque el número de trabajos no es tan grande como en las potencias mundiales, existen grandes proyectos que aprovechan al máximo los recursos del Kinect y demuestran que el mismo puede trabajar en varios campos de estudio.

Entre algunos de estos, podemos mencionar:

**Seguridad.-** SISTEMA DE SEGURIDAD INTELIGENTE BASADO EN RECONOCIMIENTO DE PATRONES MEDIANTE TECNOLOGÍA KINECT PARA RESTRINGIR EL ACCESO NO AUTORIZADO A CONSOLAS DE ADMINISTRACIÓN Y MONITOREO. Ayala Cajas César Andrés, Guerrero Idrovo Rosa Graciela.

**Educación.-** "SISTEMAS DE EDUCACIÓN PARA NIÑOS DE 3 A 5 AÑOS, MEDIANTE UN ROBOT CONTROLADO POR EL SENSOR KINECT" ESPOCH, Rómulo Byron Ilvay Taday (Ilvay, 2014).

**Desarrollo de Controladores.-** "DESARROLLO DE UN PROTOTIPO DE UN API PARA LA INTERACCIÓN DE UN USUARIO CON APLICACIONES CON CONTENIDO 3D UTILIZANDO KINECT", ESPOL, Andrés Estuardo Prieto López (Prieto, 2014).

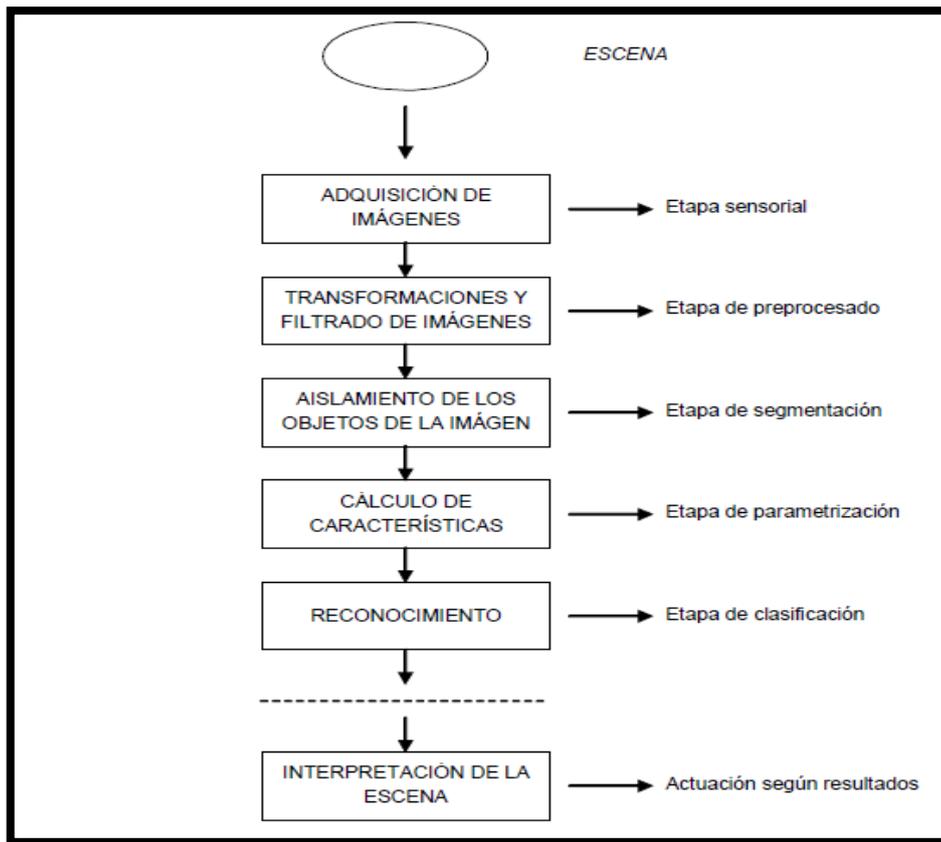
## 1.2 Visión Artificial

Según Nilsson (2001), la visión por computador o V.A. es una rama de la ciencia que pertenece a la Inteligencia Artificial y que se centra en analizar la posibilidad de los sistemas dotados de un dispositivo capaz de captar la escena de su entorno. La V.A. tiene tal versatilidad que puede apoyar el desarrollo de varios campos y aplicaciones como reconocimiento de caracteres, interpretación de imágenes y escenas, robótica, medicina, etc.

La V.A. asocia hardware, software y también desarrollos teóricos con el fin de poder entender de la manera más humana posible una escena propuesta. Por lo que es un trabajo que demanda de muchos recursos de parte del ente aplicativo. Pero sus grandes prestaciones y su capacidad de acoplamiento en torno a tantas ciencias hacen que la investigación sobre este campo sea muy rentable, y además llame la atención tanto de investigadores consagrados como de estudiantes.

Algunas aplicaciones importantes serían: automatización de una línea de ensamblaje, la inspección de circuitos para detectar defectos, teledetección para construir mapas de regiones no fácilmente accesibles, comunicación mediante gestos del ordenador. Y estos abarcan sectores como: auxiliar del automóvil, farmacéutica, embotellado, embalaje etiquetado textil, papel, metalúrgica, semiconductores, cerámica, maderas, equipos electrónicos, etc. (García, 2009).

En general cualquier trabajo de V.A. consta de dos fases: capturar la imagen y procesarla según nuestro objetivo. Sin embargo, un sistema general de V.A. se puede obtener del siguiente gráfico (García, 2009).



**Figura 1-1:** Proceso general de un sistema de Visión Artificial.

**Fuente:** Técnicas y algoritmos básicos de visión artificial, 2006.

En cuanto a las herramientas para todos los proyectos de V.A. se necesitan los mismos elementos básicos de la Figura 1-1. El primero es el que obtenga la imagen ya sea esta analógica o digital. La cámara obtiene la imagen y la deja lista para ser procesada. Algunas veces esto lo puede hacer la cámara automáticamente pero otras se necesita de software y de un código de pre procesado. Seguido de esto, se requiere en la parte de software, un lenguaje de programación y una librería orientada a la V.A.

Por último, y según la aplicación, se necesita una herramienta en donde mostrar y comprobar los resultados. Puede ser un robot, un terminal, hardware orientado o simplemente una pantalla en donde se puede notar una simulación.

La captación de la imagen es en realidad una tarea muy compleja que favorablemente fue resuelta hace ya mucho tiempo. Entonces lo que le resta a la V.A. es implementar los mejores procesos

de interpretación sobre la imagen obtenida. Y aunque esta interpretación, con respecto al ojo humano, es un trabajo del cerebro, si lo aplicamos en un computador necesitaremos un procesador que analice pesos inmensos de información. Para hacer de esta tarea lo menos costosa se modela la escena con técnicas como: fondos homogéneos, variación en la iluminación con respecto a la necesidad y a los favores de hardware a disposición, diferenciar al objetos u objetos importantes de los innecesarios.

El resto del trabajo lo debe realizar el software por lo cual, gran parte de los trabajos en la V.A. está orientada a la eliminación del ruido en una imagen.

El trabajo de reprocesar una imagen tiene como base algoritmos matemáticos que calculan diferentes características, desde nuevas intensidades de iluminación hasta diferencia de pixelado; estos procesos aunque son sencillos, conllevan mucho tiempo de cálculo según la resolución de la imagen. Esto es un gran problema para un ordenador que debe dar respuestas en tiempo real, por lo que se debe buscar la mejor forma de darle la menor carga de trabajo posible, de lo contrario resultaría un trabajo derrochador (González, *et al.*, 2006).

En la actualidad, debido a todos estos antecedentes, la V.A. aprovecha los procesadores de nueva generación que realizar varias acciones al mismo tiempo, las cámaras digitales, los sensores de precisión y muchas otras novedades que la tecnología ha dejado en nuestro tiempo.

### **1.2.1 Imagen Digital**

Se define a una imagen como digital cuando su obtención y grabado lo realizó un aparato o una cámara electrónica, y se construye en unidades de pixel siendo estas finitas y medibles en forma matricial. Es la representación bidimensional de una imagen empleando bits, unidad mínima de información compuesta por dígitos binarios (1 y 0), que se emplea a instancias de la informática y cualquier dispositivo de tipo digital (González, *et al.*, 2006).

Es de mucha importancia para el procesamiento de imágenes en la visión artificial, permitiendo obtener datos en (X, Y).

Las propiedades básicas de una imagen digital son:

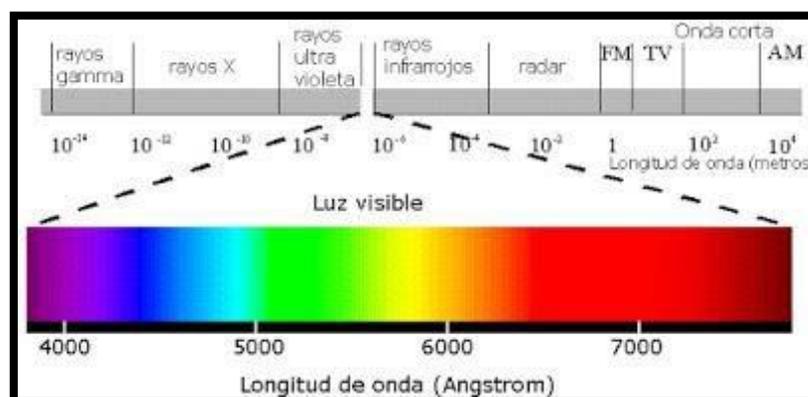
Resolución.- se considera el número matricial de filas x columnas de pixeles que contiene a la imagen.

Definición.- Representa cuan nítida esta una imagen. Depende enteramente del número de bits que se usa para representar un pixel. Aunque se puede representar de forma cuantitativa su cantidad también puede ser cualitativa según un ojo humano o condiciones de software (González, *et al.*, 2006).

### 1.2.1.1 Fundamento de Color

El color no es más que una sentido de percepción que tiene el humano sobre la luz reflejada en las cosas. De allí depende mucho la calidad y cantidad de luz que tiene la escena para la diferencia de los colores. Si un cuarto se llena de luz se puede entender cada uno de los reflejos en los objetos como colores distintos. Mientras que si la luz está totalmente ausente será imposible diferenciarlos, aún percibirlos (González, *et al.*, 2006).

La Figura 2-1, presenta los colores que el humano puede percibir son tan solo una parte de la longitud de onda total que tiene la luz. Ya que hay frecuencias tan altas, o tan bajas que el ojo humano no lo puede percibir, pero que sin embargo si tienen incidencia sobre el objeto que las recibe. Estas ondas están en constante estudio y se cree posibles grandes aplicaciones con ellas (González, *et al.*, 2006).



**Figura 2-1:** Longitud de onda de colores visibles.

Fuente: <http://reflexionesfotograficas.blogspot.com/2011/01/el-espectro-visible-los-colores.html>.

Existen muchos formatos o modelos de colores, los cuales han sido desarrollados con el mejoramiento del software y el hardware. Podemos mencionar: HSI, CMY y RGB.

El RGB es quizá es más utilizado ya que es sencillo de percibir y de maniobrar. Es el modelo que usan las cámaras y los sensores de color primarias.

Si la percepción se basa en tres colores primarios: Rojo, Verde y Azul. De allí su nombre. El estudio indica que a partir de estos colores se puede obtener casi cualquier tono e intensidad de otro color. Incluso el Negro, que reflejaría la ausencia de los colores y el Blanco que significaría la presencia de los tres (González, *et al.*, 2006).

Para representar la cantidad de cada color de forma cuántica se usa un rango igual para todos los colores de 0 a 255.

Ejemplo: Rojo = 255, Verde = 65, Azul = 255

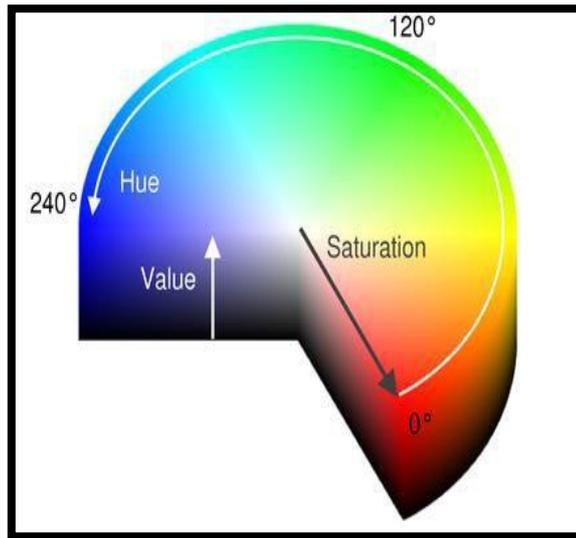
El resultado es el color amarillo.

El formato HSV de la Figura 3-1 en cambio, es un formato de color que es una transformación no lineal del modelo base RGB. Su nombre deriva de las siglas: Hue, Saturation, Value, que significaría respectivamente:

**MATIZ.**- se presenta como un grado de ángulos cuyo valores están en un rango de 0- 360 y con los grados se podría representar el color.

**SATURACIÓN.**- Detona como la distancia al eje básico de brillo blanco-negro. Su rango de medida está en 0-100 %, entonces cuanto más bajo es este valor la imagen tiende a ser más gris y con menos colorido.

**VALOR.**- Es la altura entre el blanco-negro. De la misma manera se puede medir entre 0-100, siendo el cero el negro (González, *et al.*, 2006).



**Figura 3-1:** Formato de imagen HSV.

**Fuente:** <http://stackoverflow.com/questions/11598043/get-slightly-lighter-and-darker-color-from-UIColor/23666926>

### 1.2.2 Adquisición de Imágenes.

Es el proceso de captar una escena del mundo real y guardarlo ya de forma digital o analógica.

Las primeras imágenes adquiridas eran analógicas y su trabajo estaba reservado a verdaderos profesionales del arte, incluso llevaba mucho tiempo llegar a captarlas de forma correcta. Sin embargo, la V.A. hubiera tenido un espacio relativamente nulo debido a la dificultad de este proceso (González, *et al.*, 2006).

En la era digital, como en muchos otros ámbitos, los procesos se hicieron más sencillos y accesibles a todo el público. Hoy, con una cámara web de muy bajo costo se puede incursionar fácilmente en el mundo de la V.A.

Entre las cámaras más populares para la adquisición de imágenes se puede mencionar:

- Cámaras WEB
- Cámaras digitales.

- Kinect

La imagen digital obtenida se mide en píxeles. Es la unidad de medida del tamaño de la imagen, cada píxel se compone de una combinación de los tres colores principales: rojo, verde, azul, Cada píxel tiene una ubicación matricial en el espacio de la imagen (González, *et al.*, 2006).

### **1.2.3      *Procesamiento de Imágenes.***

Para que cualquier imagen digital obtenida pueda ser estudiada se requiere que antes sea procesada. Este procesamiento se puede realizar en la cámara, si es que esta tiene la facultad de hacerlo, o de preferencia con un software vinculado a este trabajo.

El procesamiento básico consiste en acondicionar la imagen a formatos requeridos y aplicar filtros para eliminar ruidos y para tener un mejor enfoque de la escena, con ello harán falta menos recursos para poder analizarla a cabalidad.

En ocasiones, antes de un procesamiento de imágenes se debe pre procesar la imagen, para aumentar la fiabilidad de los resultados, podemos nombrar algunas operaciones del pre procesado como:

Inversión, Adición y sustracción, Producto o división por una constante, Logaritmo y Exponencial, Complementación, And, Or y Xor, Silicing, Clipping, Umbralización, Binarización (González, *et al.*, 2006).

#### **1.2.3.1      *Transformaciones morfológicas.***

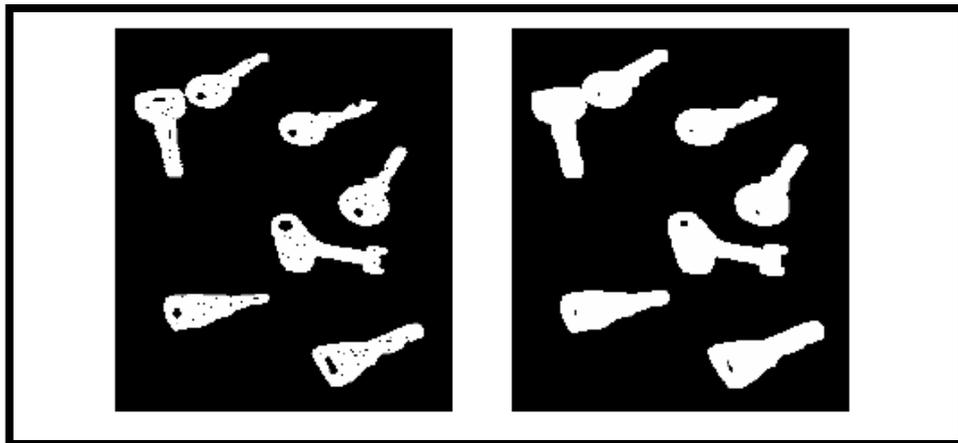
Las transformaciones morfológicas cambian la forma y la estructura de los elementos de una imagen. Permiten modificar estas formas para separar unos objetos de otros, obtener contornos de objetos, volver simples a formas complejas, eliminar ruidos sobre objetos necesarios, etc (González, *et al.*, 2006).

Generalmente estas transformaciones se usan en imágenes previamente binarizadas (en blanco y negro) o también imágenes en niveles de gris.

#### 1.2.3.1.1 Dilatación binaria.

La dilatación, también llamada “crecimiento”, “llenado”, “expansión” produce un efecto de engrosamiento en los bordes en el objeto. Haciendo que pequeños objetos que estén junto al mayor sean absorbidos y ya no se entiendan como ruido (ver Figura 4-1). Existen métodos para la dilatación, por ejemplo sumando los pixeles que rodea a uno central y si supera un umbral se pone a una (González, *et al.*, 2006).

Este efecto es muy valorado para aumentar el contorno de los objetos y unir líneas discontinuas.



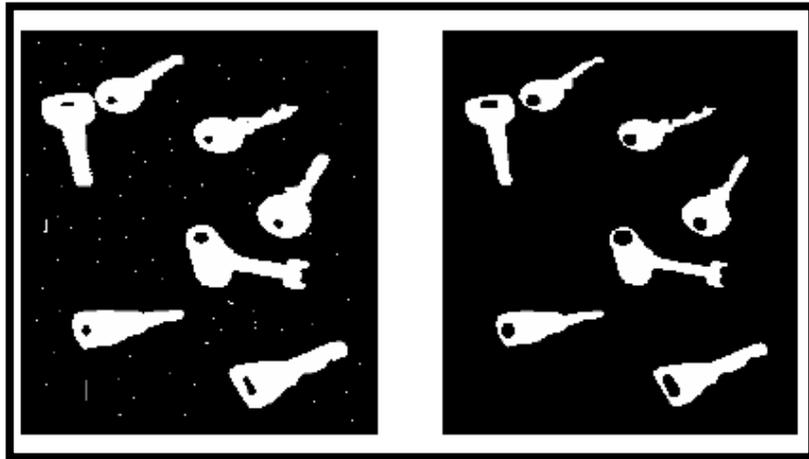
**Figura 4-1:** Dilatación binaria.

Fuente: COMPUTER VISION AND ROBOTIC GROUP.

#### 1.2.3.1.2 Erosión binaria.

Es la función dual de la dilatación, no significa que sea su función inversa. Dicho de otra forma, no significa que si a una imagen primero la dilatamos y luego la erosionamos, obtendremos la imagen original, estos procesos convierten de forma importante y definitiva la imagen. Si la dilatación expandía los bordes y contornos de los objetos, la erosión reduce los contornos de los

objetos (ver Figura 5-1). Se usa para separar pequeños objetos unidos a los objetos más grandes. Igualmente que en la dilatación, existen muchos métodos para erosionar la imagen, uno de ellos se puede parecer mucho a la Umbralización. En la siguiente imagen se puede notar que todos los pequeños objetos que no son reconocidos, son eliminados incluso los bordes los objetos restantes parecen más pálidos y formados (González, *et al.*, 2006).



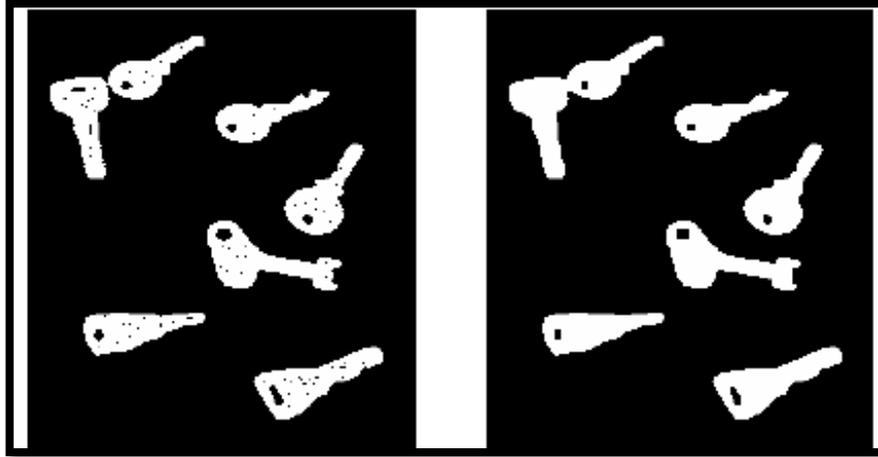
**Figura 5-1:** Erosión binaria.

Fuente: COMPUTER VISION AND ROBOTIC GROUP.

#### 1.2.3.1.3 Apertura.

Consiste en usar una erosión y luego una dilatación. Como se dijo anteriormente, la imagen resultante no va a ser igual a la original (ver Figura 6-1). Si primero eliminamos los ruidos muy pequeños y luego afirmamos los objetos restantes nos quedarán objetos ensanchados pero sin mucho en ruido. Esta transformación es muy usada en aplicaciones como:

- Segmentación de objetos
- Descomposición de objetos en elementos más simples
- Extracción de formas determinadas.
- Elimina salientes estrechos.
- Separación objetos que no están demasiado pegados.
- Aumentar los agujeros que está dentro (González, *et al.*, 2006).

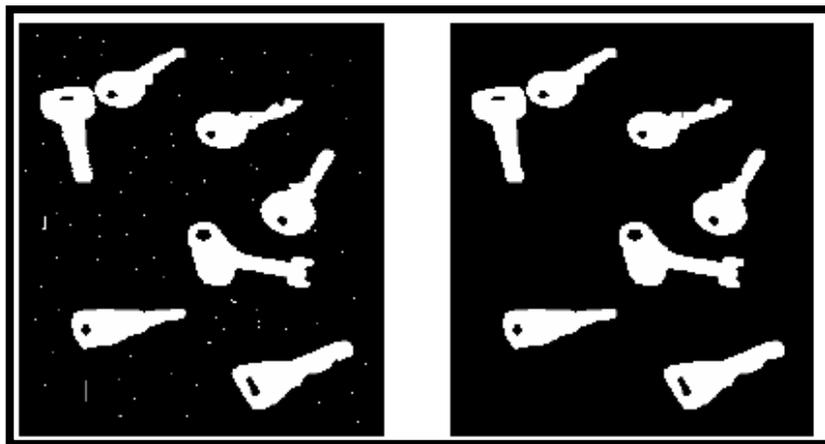


**Figura 6-1:** Apertura.

**Fuente:** COMPUTER VISION AND ROBOTIC GROUP.

#### 1.2.3.1.4 Cierre.

Se realiza primero una dilatación y luego una erosión. Y sus aplicaciones son parecidas a la transformación anterior. Incluso se llega a usar una después de otra para lograr una imagen totalmente libre de ruido (ver Figura 7-1) y dado que el costo computacional de estos procesos son bajos, son una gran idea para que la segmentación sea de calidad (González, *et al.*, 2006).



**Figura 7-1:** Imagen tras proceso de cierre.

**Fuente:** COMPUTER VISION AND ROBOTIC GROUP.

#### **1.2.4 Histograma de una imagen.**

Es una gráfica en un tipo de barras que representa la distribución de los datos de una imagen, muestra: acumulación o tendencia, la variabilidad o dispersión, y como están distribuidas las características de la imagen.

Le aporta al sistema algunas particularidades como:

**Síntesis:** Permite realizar un resumen digerible de los datos.

**Análisis:** analizar los datos pues se puede reconocer.

**Capacidad de comunicación:** comunica la información de forma clara y concisa y se permite procesarla en otro software o hardware (González, *et al.*, 2006).

Los histogramas de imágenes en escala de grises son muy usados debido a su facilidad de comprensión. En general se componen de la siguiente manera: el eje horizontal representa los distintos tonos de gris desde el negro puro al blanco y el eje vertical representa el número de píxeles que contienen la imagen para cada tono.

Una imagen con el histograma bien distribuido es considerada como bien iluminada. Sin embargo, también se pueden estudiar otras situaciones comunes como:

- Tonos apagados
- Sombras o zonas observadas
- Sobreexposición y zonas quemadas
- Sombras pálidas
- Contraluz
- Distribución homogénea (González, *et al.*, 2006).

#### **1.2.5 Segmentación.**

Segmentar una imagen consiste en generar zonas privadas de cada uno de los elementos que forman parte de la escena, diferenciando los objetos con respecto al fondo y a su entorno. En este proceso se pueden encontrar fondos complejos y sencillos siendo los últimos los usados para la investigación y el desarrollo ya que impulsa al resultado primario del estudio (González, *et al.*, 2006).

Una segmentación adecuada debe mostrarnos los objetos diferenciados y listos para el siguiente proceso, esto significa que cada grupo de píxeles debe pertenecer únicamente al objeto y separarse de su entorno.

La etapa de segmentación es crucial para el reconocimiento de formas siendo tan complicado según la complejidad de la escena. No existe relación entre la aplicación la complejidad de la escena ya que siempre se puede adecuar la escena para que no sea tan difícil de segmentar.

Se puede decir que la segmentación es la etapa verdadera del reconocimiento pues el resultado de esta etapa es tener los objetos perfectamente ubicado en la imagen resultante, dependiendo de la técnica utilizada.

#### *1.2.5.1 Umbralización.*

En teoría es el método más simple ya que se basa en clasificar cada píxel de la imagen como parte o no del objeto.

Generalmente existen dos formas de concretar el umbralizado: thresholding, histograming. En cualquiera de las dos se necesita trabajar sobre valores homogéneos de gris, para medir la intensidad de cada píxel (González, *et al.*, 2006).

En el primer caso (thresholding) si los objetos son más claros que el fondo es considerado parte del elemento, de lo contrario serían parte de lo que le resta de la imagen.

En el segundo caso (histograming) se necesita realizar un estudio sobre el histograma de la imagen y se toma un rango a partir del cual el objeto es hallado y el resto de la gráfica no tiene importancia para el estudio.

#### 1.2.5.2 *Segmentación orientada a regiones.*

La segmentación orientada a regiones trabaja con la conectividad de la imagen, que significa que para que un objeto sea reconocido como tal debe tener las condiciones necesarias: compartir características como color, textura, etc.; o, que sean los píxeles conectados dentro de una curva o de líneas de bordes (González, *et al.*, 2006).

#### 1.2.5.3 *Basada en bordes*

Es una técnica de segmentación que acude a los bordes como diferenciadores de un objeto con el fondo o con otro objeto. Requiere un aprendizaje previo y demanda un procesador de imágenes potente para segmentarlo adecuadamente.

Los resultados obtenidos tienden a ser desacertados, debido a la precisión de cálculos con respecto a bordes de objetos que se mueven en tiempo real (González, *et al.*, 2006).

#### 1.2.5.4 *Por colores.*

Otro método ampliamente utilizado por su gran fiabilidad y su versatilidad al combinarse con otros métodos. Tiene muchos caminos al momento de su implementación pero su fin es el mismo: segmentar el objeto que tenga el color elegido. El método más usado y entendible es presentar un color a seguir y evaluar cada píxel para notar su proximidad o igualdad con respecto al color.

Su principal problema es que la escena no debería tener muchos objetos del mismo color pues la segmentación no sería única.

### **1.2.6 Iluminación de la escena.**

Una sección importante en la construcción de la escena es la iluminación, ya que a partir de las condiciones de luz se pueden tener colores más vivos u opacos, estructuras más definidas, y fondos más determinados.

Un entorno debidamente iluminado es imprescindible para obtener unas condiciones de partida óptimas. En cambio, si la luz no es adecuada, los efectos sobre los resultados pueden ser incluso nocivos. Aunque existen muchas cámaras y sensores preparados para diversas condiciones de luz, no se le puede confiar la totalidad de este punto a la máquina.

La iluminación de la escena consta de tres sectores importantes: el tipo iluminación, el fondo, la posición de la cámara (González, *et al.*, 2006). De estas tres la más importante es el tipo de iluminación.

#### **1.2.6.1 Iluminación frontal.**

En este tipo, la luz incide directamente sobre el objeto ya sea de forma oblicua o difusa, vertical u horizontalmente. También influye mucho el color de la lámpara de iluminación, si bien una iluminación amarilla es más potente y tiene un mayor campo de efecto, la luz blanca es ideal para reconocer colores.

#### **1.2.6.2 Iluminación por retroalimentación.**

Donde se ilumina una pantalla de forma cuadrada o circular de tal modo que lo que se busca es el contorno del objeto como una sombra particular. Este objeto puede estar detrás o delante del fondo. Este tipo de iluminación puede variar en otros dos tipos: Con la iluminación delante del fondo y la iluminación detrás del fondo.

### **1.3 Software.**

Existen decenas de entornos y programas para aplicaciones de V.A., pero para los fines propios y por las ventajas de experiencia y la documentación se han usado los siguientes sistemas operativos, lenguajes y librerías.

#### ***1.3.1 Linux UBUNTU 14.04 LTS.***

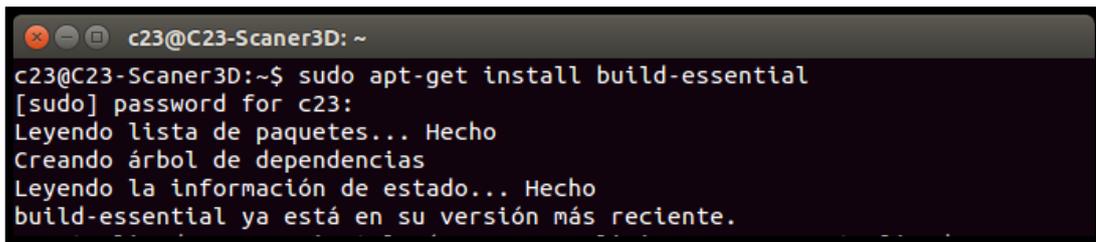
Es un sistema operativo basado en Unix, perteneciente a Linux. El sistema fue desarrollado y completado por varios investigadores desde los años 60. Con el tiempo su premisa de Open Source ha sido preferida por muchos investigadores y aficionados al software. También ha servido de plataforma de varias aplicaciones, librerías y lenguajes que hacen de Linux es rincón de la libertad del software (Borja, 2008).

#### ***1.3.2 Lenguaje C++.***

Es un lenguaje de programación ampliamente usado desde su creación en 1983. De principio fue pensado como una extensión del C, se llamaba C with class, pero en 1983 lo renombraron magistralmente C++. Debido a su gran difusión y éxito se llegó a estandarizar (Deitel & Deitel, 2004).

Ocupa el primer lugar de uso entre los distintos lenguajes debido a su éxito entre los programadores, sus potentes herramientas y que muchos recursos están escritos en el mismo lenguaje. Tiene las ventajas del C original pero con expresiones, flexibilidad y mejor concisión y eficiencia (Deitel & Deitel, 2004).

Tiene varias evoluciones como la programación orientada a objetos. La Figura 8-1 presenta la forma de instalación en Ubuntu 14.04.



```
c23@C23-Scanner3D: ~  
c23@C23-Scanner3D:~$ sudo apt-get install build-essential  
[sudo] password for c23:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
build-essential ya está en su versión más reciente.
```

**Figura 8-1:** Instalación de C++ en UBUNTU.

**Realizado por:** Logroño C. y Mishqui K., 2016

### 1.3.3 *OpenCV*

OpenCV Es una potente librería de visión artificial y machine learning, tiene licencia BSD, lo que permite utilizar y modificar el código.

Está escrito en C/C++ lo que la hace más propensa a seguir en desarrollo, prueba de ello es que tiene una gran comunidad, casi 50000, y más de 7 millones de descargas (Bradsky & Kaebler, 2008).

Sus funciones principales son identificar objetos, caras, clasificar acciones humanas, tracking de objetos, modelos 3d.

Una mayoría de aplicaciones de Open Source de V.A. han elegido a OpenCV para su desarrollo puesto su gran capacidad y su documentación amplia. Entre sus aplicaciones más populares tenemos:

- Detector de movimiento
- Detección de parámetros en dos y tres dimensiones.
- Segmentación y reconocimiento.
- Detección de Objetos.
- Calibración de cámaras (Bradsky & Kaebler, 2008).

Sus funciones básicas son:

**Core:** Se definen las estructuras de datos básicas que usan el resto de los módulos.

**Imgproc:** módulo procesamiento de imágenes como: filtrado lineal / no lineal, transformaciones afines, conversión del espacio de color, histogramas.

**Video:** módulo de análisis de video que incluye algoritmo para la estimación del movimiento. Extracción de fondo.

**Calib3d:** Algoritmos básicos de visión múltiple como calibración de cámaras estéreo y otras cámaras.

**Features 2d:** detectores de características.

**Objdetect:** detección de objetos instancias o clases predefinidas como: caras, ojos, gente, coches.

**Highgui:** Todo lo relacionado a la interfaz gráfica de OpenCV. Permiten importar imágenes y video.

**GPU:** algoritmos acelerados x hardware para distintos módulos OpenCV (Bradsky & Kaebler, 2008).

#### 1.3.4 CMake.

Es un sistema de compilación de código abierto. Es desarrollado por Kitware. Se compone de una familia de herramientas para construir y probar el paquete de software. Se usa para controlar el proceso de compilación. CMake es una familia de herramientas diseñadas para crear, probar y el paquete de software. CMake se utiliza para controlar el proceso de compilación de software utilizando la plataforma simple y archivos de configuración del compilador independiente.

CMake genera makefiles nativos y espacios de trabajo que se pueden utilizar en el entorno del compilador de su elección (Cabrera, 2009).

#### 1.4 Kinect.



**Figura 9-1:** Kinect XBOX 360.

**Fuente:** <http://www.xbox.com/en-US/xbox-360/accessories/kinect>

Kinect es un sensor compuesto desarrollado por Microsoft con la firme intención de aumentar el uso de la videoconsola XBOX (ver Figura 9-1). Se lanzó en el 2010 y a partir de la liberación de su código en el 2011 ha sido la plataforma de muchas investigaciones respecto a la V.A. Su éxito se debe a su bajo costo y a sus prestaciones completas. Se conoce al Kinect como un sensor, pero en realidad Kinect es un sistema de sensores, independientes que trabajan en conjunto según la necesidad (Quirarte, *et al.*, 2013).

Al notar que muchos investigadores usaban el Kinect en la PC con los controladores libres que algunos grupos venían desarrollando, la Microsoft pensó en demandarlos y cortar desde la cabeza todos estos desarrollos "piratas" sobre su producto.

Sin embargo lanzó al público su propio controlador de Kinect que funciona en Windows 7 y 8. No tiene aún SDK para Linux. Tiene muchas ventajas ya que se puede usar como una clase o una función en el lenguaje orientado a objetos C#, facilitando así, la construcción de aplicaciones (Velasco, 2013).

Sin embargo, los controladores libres aún se siguen desarrollando y son muchos los programadores que se vuelcan a usarlo. En parte, porque el sistema operativo Linux se ha convertido en sinónimo de desarrollo Open Source, y también por el espíritu mismo de esta tendencia.

Entre las librerías más usadas están desarrolladas por Open NI y Open Kinect (Github, 2016).

#### **1.4.1 Menú de componentes.**

##### Generales

- Infrarrojo (IR) CMOS camera.
- Protector Infrarrojo -830, 60mW diodo laser.

##### Audio

- Micrófonos

## Controladores

- Motor
- Acelerómetro (3-axes)

## Procesador y memoria.

- Prime Sense Chip PSI080 -A2
- 64 MB DDR2 SDRAM

## Kinect

- Color Camera 640 x 480; 640x480 @fps output
- IR Camera: 1280 x 1024 , 640x480 @30 fps output
- Rango de operación (Sensor Depth). Rango= 0,8 - 3,5
- Resolución espacial (@ 2m distancia) = 3mm
- Resolución Depth (@ 2m distancia) = 1cm (Sebastian, 2010).

Además Kinect soluciona muchos problemas de comunicación con el computador gracias al tipo de adaptador USB para la PC.

También tiene la facultad de ajustar la cámara según una inclinación deseada pues tiene un motor integrado en su base. Posee unos 27° de trabajo verticales.

### **1.4.2** *Visión del Kinect.*

Los elementos más usados del Kinect son su cámara RGB y su sensor de profundidad llamado Depth.

#### **1.4.2.1** *Visión RGB.*

La cámara RGB es una cámara clásica que podemos hallar en cualquier dispositivo de la obtención de imágenes. Esta posee una resolución de ocho bits de tipo VGA. La diferencia con las demás se halla tan solo en la resolución y los frames por segundo (fps) de la cámara. Ya que esta tiene unos 30 fps. Los FPS son la cantidad de imágenes que se recoge por segundo, entre más FPS se pueda obtener más definido se notará el video (Quirarte, *et al.*, 2013).

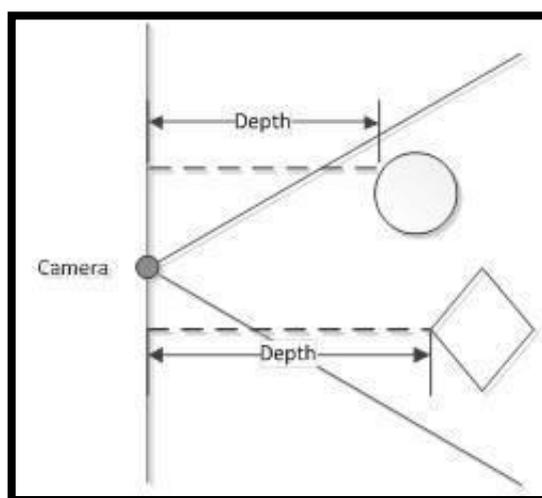
A simple vista se puede notar como una cámara más potente que las convencionales, pero cabe recalcar los siguientes elementos propios de la misma:

- Las cámaras CMOS tienen un menor costo y son menos afectadas por los cambios de luz, pues tiene una estructura simple lo cual desemboca en trabajos del mismo calibre.
- La resolución de la cámara también se puede modificar a través de código FUENTE, y de preferencia esta debe ser igual al del sensor infrarrojo.

#### 1.4.2.2 Imagen DEPTH.

La imagen de profundidad (Depth) está completamente unificada al llamado sensor 3D, esta se compone de dos partes: un proyector de rayos IR (ver Figura 10-1) y un sensor CMOS monocromático. El sensor que percibe los rayos infrarrojos puede capturar datos de video en 3D bajo casi cualquier estado de la luz. Tiene una resolución de (640 x 480) con 16-bit de profundidad a 30 fps. Y unos 2048 niveles de sensibilidad (Quirarte, *et al.*, 2013).

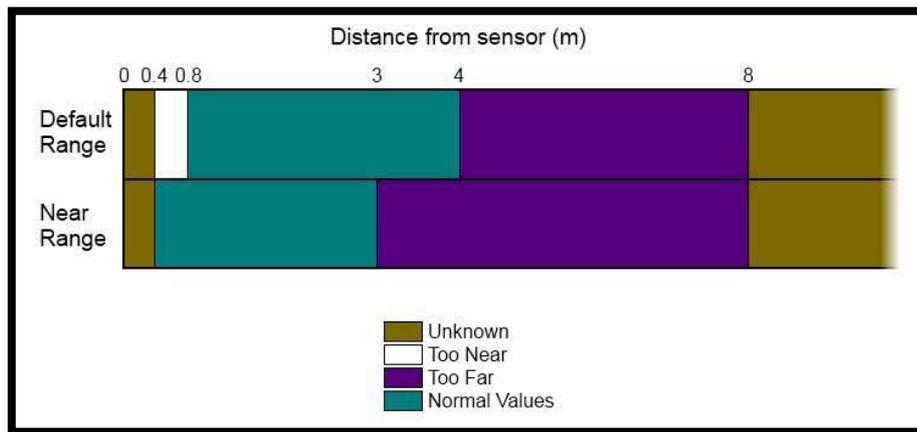
La técnica que Kinect usa para obtener la información de la profundidad fue desarrollada por la dueña de su chip Prime Sense. La técnica se llama luz estructurada. La misma técnica ha venido siendo utilizada en aplicaciones donde se necesita mucha precisión, y por lo tanto los sensores usados tienden a tener costos muy elevados.



**Figura 10-1:** Proyección del infrarrojo del Kinect.

Fuente: <https://msdn.microsoft.com/en-us/library/hh973078#CommunityContent>

Aunque el paquete Kinect, contiene especificaciones básicas para su uso, solo el investigador tienen la última palabra en cuanto a la distancia que se necesita estudiar. En la Figura 11-1, podemos verificar los rangos de acción. Pero, hay que recalcar que la forma de devolver información de parte del sensor no es la misma en cada uno de los rangos. Las medidas varían bastante mientras el objeto es cercano y varían muy poco cuando la distancia es demasiado extensa. Por ellos, algunas fórmulas de conversión del dato devuelto por el sensor a distancia real, no son confiables.



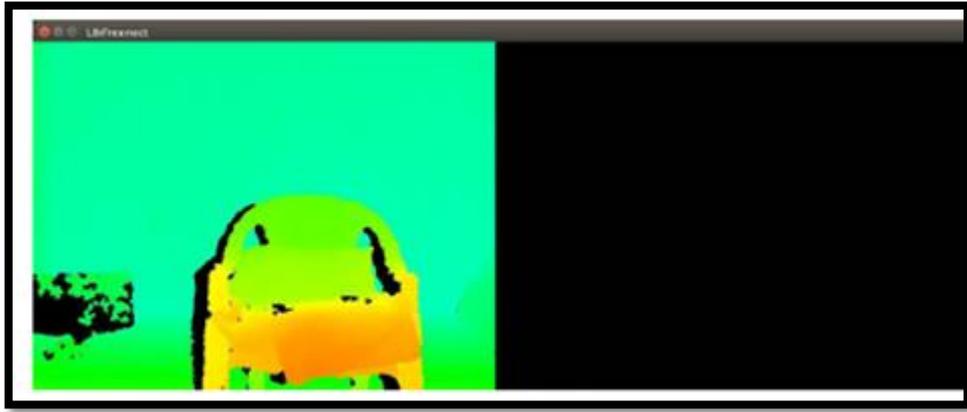
**Figura 11-1:** Rango nominal de visión de profundidad del Kinect.

Fuente: <https://msdn.microsoft.com/en-us/library/hh973078#CommunityContent>

#### 1.4.2.2.1 Luz Estructurada.

La luz estructurada consiste en que a partir de una FUENTE de luz, o de un haz de luz se la proyecte sobre una escena, los objetos situados en la misma serán impactados por el haz y lo deformarán según su posición y según su forma. Esta deformación puede ser observada y recogida por otra cámara estratégicamente situada. Una vez recogida la información se puede saber la distancia a la que se encuentra el objeto.

En la Figura 12-1, podemos ver la imagen Depth de una sencilla escena. La parte derecha está completamente negra pues la escena se tomó con ausencia total de luz. Comprobándose así la explicación de luz estructurada.



**Figura 12-1:** Imagen Depth y RGB en ausencia de luz.

**Realizado por:** Logroño C. y Mishqui K., 2016

### 1.4.3 *LibFreenect.*

Es el controlador para el Kinect, desarrollado como Open Source por la comunidad Open Kinect.



**Figura 13-1:** Logo Open Kinect.

**Fuente:** [http://openkinect.org/wiki/File:Logo\\_openkinect\\_shaded.svg](http://openkinect.org/wiki/File:Logo_openkinect_shaded.svg)

Open Kinect, es una comunidad con más de 2000 miembros, que ha desarrollado varios controladores, demos y ejemplos para Kinect, que se pueden instalar en Linux, Apple y Mac. Sus funciones más populares son las de la imagen RGB y Depth, además de ciertos ejemplos de compilación con OpenCV, y el uso del motor y los leds del Kinect (Github, 2016).

Todo código está bajo la disposición Apache 2.0 o licencias GPL2 opcionales. Tiene ejemplos desarrollados para Python, C++, C#, Java JNI.

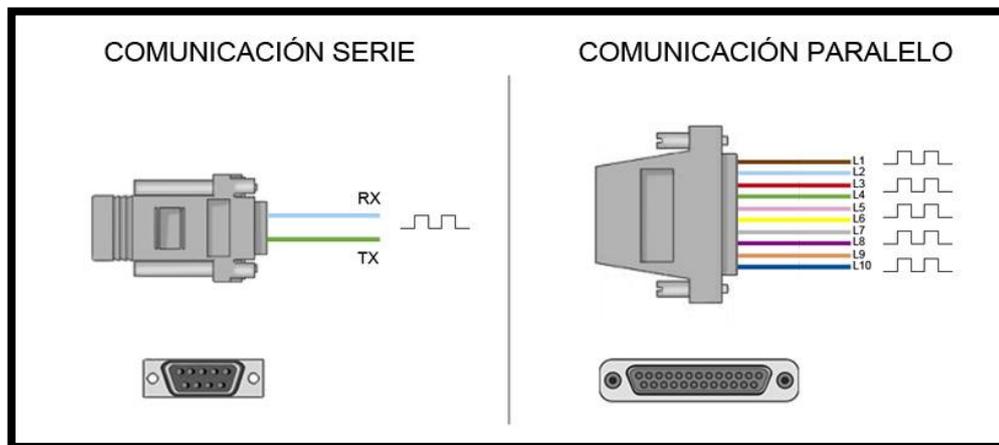
## 1.5 Comunicación Serial.

### 1.5.1 Puerto serie.

Un puerto es el nombre genérico con que denominamos a las interfaces físicas o virtuales, que permiten la comunicación entre dos ordenadores o dispositivos.

Un puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión) (ver Figura 14-1). No obstante, pueden existir otros conductores para referencia de tensión, sincronismo de reloj, etc.

Por el contrario, un puerto paralelo envía la información mediante múltiples canales de forma simultánea. Para ello necesita un número superior de conductores de comunicación, que varían en función del tipo de puerto. Igualmente existe la posibilidad de conductores adicionales además de los de comunicación (Espinosa, *et al.*, s.f.).



**Figura 14-1:** Diagramas comunicación serie y paralelo.

Fuente: <http://www.luisllamas.es/2014/04/arduino-puerto-serie/>

Históricamente ambos tipos de puertos han convivido en los ordenadores, empleándose los puertos paralelos en aquellas aplicaciones que requerían la transmisión de mayores volúmenes de datos. Sin embargo, a medida que los procesadores se hicieron más rápidos los puertos de serie fueron desplazando progresivamente a los puertos paralelos en la mayoría de aplicaciones.

Un ordenador convencional dispone de varios puertos de serie. Los más conocidos son el popular USB (universal serial port) y el ya casi olvidado RS-232 (el de los antiguos ratones). Sin embargo, dentro del ámbito de la informática y automatización existen una gran cantidad adicional de tipos de puertos serie, como por ejemplo el RS-485, I2C, SPI, Serial Ata, Pcie Express, Ethernet o FireWire, entre otros.

En ocasiones veréis referirse a los puertos de serie como UART. La UART (universally asynchronous receiver/transmitter) es una unidad que incorporan ciertos procesadores, encargada de realiza la conversión de los datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada (Espinosa, *et al.*, s.f.).

Por otro lado, también podéis oír el término TTL (transistor-transistor logic). Esto significa que la comunicación se realiza mediante variaciones en la señal entre 0V y Vcc (donde Vcc suele ser 3.3V o 5V). Por el contrario, otros sistemas de transmisión emplean variaciones de voltaje de -Vcc a +Vcc (por ejemplo, los puertos RS-232 típicamente varían entre -13V a 13V).

## **1.6 Componentes Eléctricos.**

### **1.6.1 Motores Eléctricos.**

#### *1.6.1.1 Funcionamiento.*

El motor es una máquina, la cual transforma energía eléctrica en mecánica con el objetivo de producir movimiento, que sirve para accionar diferentes mecanismos, dispositivos, que son usados en varias áreas de la industria, comunicaciones, electrodomésticos.

El funcionamiento de este tipo de máquinas se basa en hacer circular corriente por un conductor, el cual produce un campo magnético explicado por la ley de Faraday, produciendo una relación entre la corriente y el magnetismo, que produce el movimiento de la parte móvil del motor o rotor por consecuencia de la acción de las fuerzas de atracción y repulsión generadas en la parte fija del motor o estator (Calloni, 2003).

### 1.6.1.2 Clasificación de los motores eléctricos.

#### 1.6.1.2.1 Motores corriente continua.



**Figura 15-1:** Motor de corriente continua 24V.

**Fuente:** <http://www.directindustry.es/prod/smart-motor-devices/product-72050-693251.html>.

Su funcionamiento es parecido a un motor de corriente alterna, pero poseen una ventaja que es la posibilidad de poder regular su velocidad, controlando las siguientes variables:

- El flujo producido por la corriente de excitación en cada polo, a menor flujo mayor será la velocidad de rotación.
- La tensión de alimentación aplicada al motor, a mayor tensión mayor velocidad y viceversa.
- La resistencia del circuito inducido, a través de la conexión en serie a este devanado de una resistencia, que producirá un aumento o disminución de la velocidad según se aumente o reduzca el valor de la resistencia de manera directamente proporcional.

Este tipo de motores se clasifican de acuerdo al tipo de excitación en:

- Motores con excitación independiente.
- Motores con excitación serie.
- Motores con excitación derivación.
- Motores con excitación compuesta (Calloni, 2003).

#### 1.6.1.2.2 Motores corriente alterna.

Son máquinas eléctricas que convierten la energía eléctrica en mecánica, este tipo de motores son los más usados en la industria, ya que su alimentación es con voltaje alterno, el cual es suministrado por la red.



**Figura 16-1:** Carcasa motor de corriente alterna.

Fuente: <http://electronicatecnica9412.blogspot.com>

#### 1.6.2 Luz piloto.

Es un elemento de tipo indicador, usado para representar el estado de una variable, elemento, que forme parte de un proceso, mediante el encendido de un foco, de alimentación eléctrica variada. Se usan principalmente del tipo led, y por su variedad de colores (ver Figura 17-1) son muy usadas en proceso de producción, permitiendo la distinción de los diferentes estados del mismo asignándolos a algún color, para su selección se debe tener en cuenta el color y su tensión de alimentación (Paco, 2014).



**Figura 17-1:** Luces piloto de 24V.

Fuente: [http://www.servitrara.com/vercatalogo.php?id\\_producto=39044&TB\\_iframe=1&width=800&height=475](http://www.servitrara.com/vercatalogo.php?id_producto=39044&TB_iframe=1&width=800&height=475)

### **1.6.3**      *Selector.*

Los selectores son elementos de accionamiento que permiten elegir entre varios modos de funcionamiento un elemento, variable, o un proceso en sí, a través del cambio en la posición (ver Figura 18-1) en la que se encuentran activados los contactos que lo componen (Paco, 2014).



**Figura 18-1:** Selector de tres posiciones.

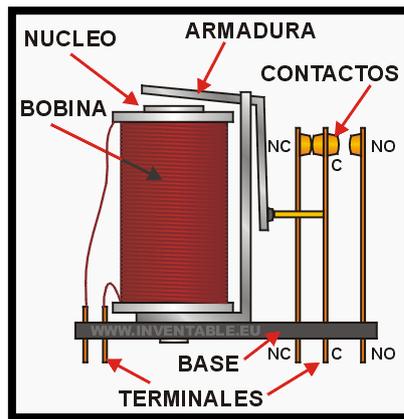
**Fuente:** <https://www.grainger.com/product/SCHNEIDER-ELECTRIC-Non-Illuminated-Selector-Switch>

Un selector puede tener varias posiciones, a cada una de las cuales se les puede asignar el accionamiento o corte de algún tipo de contacto, sea este normalmente abierto o cerrado, y la elección de su posición puede ser activada de forma manual sin ninguna restricción o con algún tipo de protección como el uso de una llave.

### **1.6.4**      *Relé.*

#### **1.6.4.1**      *Funcionamiento.*

Un relé es un interruptor accionado por un electroimán (ver Figura 19-1), el cual está formado por una barra de hierro llamada núcleo, rodeada por una bobina de hilo de cobre, al pasar una corriente eléctrica por la bobina el núcleo de hierro se magnetiza, lo que le convierte en un imán que atrae unos contactos en su interior permitiendo el paso de corriente, el relé consta de una línea común, un normalmente cerrado y un normalmente abierto (Castela, 2012).



**Figura 19-1:** Esquema típico de un relé.

Fuente: <http://creativecommons.org/licenses/by-nc-sa/3.0/es/deed.es>

#### 1.6.4.2 Tipos de relés.

Hay relés que están compuestos por un contacto móvil y dos contactos fijos, cuando no hay corriente o se encuentra en esta de reposo, el móvil se encuentra en contacto con un fijo lo que se conoce como normalmente cerrado, en el momento que existe corriente por el bobinado, el núcleo atrae el móvil hacia el otro fijo el cual es denominado como normalmente abierto (Castela, 2012).

Existen diferentes tipos de relés en base a sus contactos:

- SPST.- Relé con un solo interruptor normal.
- SDT.- Relé con un solo conmutador de dos vías.
- DPST.- Relé con dos interruptores normales.
- DPDT.- Relé con dos conmutadores de dos vías.

#### 1.6.5 Sensores.

Un sensor es un dispositivo que recibe señales provenientes de magnitudes físicas o químicas del medio ambiente, y otorga una señal de salida que está en función de la variable medida. Sensor y transductor son términos usados a veces como sinónimos, ya que ambos términos hacen referencia a una transformación de señales.

Se diferencian en que un sensor se encuentra en contacto con la variable a medir, de manera constante, adquiriendo valores que por su naturaleza o magnitud no pueden ser distinguidas de forma directa por los sentidos (Pallas, 2003).

#### 1.6.5.1 *Funcionamiento de los sensores.*

Los siguientes términos pueden definir el funcionamiento de los sensores en métodos de medición.

- Rango y margen: El rango define los límites entre los cuales pueden variar las entradas, el margen es la diferencia que existe entre el valor máximo de la entrada con el valor mínimo.

$$R = V_{\max} - V_{\min}$$

Donde:

R = rango

$V_{\max}$  = valor máximo

$V_{\min}$  = valor mínimo

- Error: Error es la diferencia entre el valor medido y el valor real que se mide, su fórmula es muy sencilla.

$$E = V_m - V_r$$

Donde:

E = error

$V_m$  = valor medido

$V_r$  = valor real

Exactitud: La exactitud es el grado hasta que un valor producido por un sistema de medición podría estar equivocado, se puede decir que es igual a todas las sumas de errores posibles más el error en la exactitud de la calibración del transductor.

$$\text{Exactitud} = \pm \text{Valor } \%$$

- Sensibilidad: La sensibilidad es la relación que indica qué tanta salida se obtiene por unidad de entrada es decir es la relación de salida/entrada.

$$S1= S / E$$

Donde :

S1 = sensibilidad

S = salida de resultados

E = entrada de datos

- Error por histéresis: Los transductores pueden producir distintas salidas de la misma magnitud que se mide, si dicho valor se obtuvo mediante un incremento o reducción continua a este efecto se le conoce como histéresis.

- Error por no linealidad: Para muchos sensores que supone que en su rango de funcionamiento la relación que existe entre la entrada y la salida es lineal, no todo transductor tiene una linealidad exacta por lo que existe un error que se puede definir como la desviación máxima respecto a la línea recta correspondiente.

- Repetitividad: El término repetitividad se utilizan para describir la capacidad del transductor de producir la misma salida después de aplicar varias veces un mismo valor en la entrada.

- Estabilidad: Es la capacidad de producir la misma salida cuando se emplea para medir una entrada constante en un período, el cambio en la salida que ocurre en ese tiempo se le conoce como deriva. Este se puede expresar como un porcentaje del rango de salida total.

- Banda /tiempo muerto: La banda muerte es cuando en rangos de valores de entrada del transductor no hay valores de salida.

- Resolución: Cuando la entrada varía continuamente en todo el rango, las señales de salida pueden cambiar a pequeños intervalos.

- Impedancia de Salida: Cuando un sensor produce una salida eléctrica se vincula con un circuito electrónico, lo que es necesario conocer su impedancia de salida ya que se puede conectar en serie o paralelo con dicho circuito (Bolton, 2013).

### 1.6.5.2 Tipos de sensores.

Existen una gran variedad de magnitudes que pueden ser medidas, por lo cual el número de sensores que existen es elevado, razón por la cual se los dividen de acuerdo a varios criterios según:

- El aporte de energía: se dividen en modulares o activos y generadores o pasivos.
- La señal de salida: se dividen en analógicos o digitales.
- Modo de funcionamiento: pueden ser de deflexión o comparación.
- Tipo de relación entrada-salida: pueden ser de orden cero, primer, segundo y de orden superior.
- Magnitud medida: esta clasificación es amplia, y tiene en cuenta la magnitud física o química a ser adquirida, puede ser temperatura, presión, humedad, caudal, velocidad, aceleración, fuerza, par, etc.
- El parámetro variable: puede ser resistencia, capacidad, inductancia.

En el área de la electrónica la clasificación más usada es de acuerdo con el parámetro variable (Pallas, 2003).

#### 1.6.5.2.1 Sensores inductivos.

Su funcionamiento se basa en la detección de objetos metálicos, a través de corrientes inducidas por campos magnéticos.

El sensor está compuesto por un núcleo inductor, usado para generar un campo magnético de alta frecuencia, el cual genera en algún objeto metálico cercano, una circulación de corriente, que a su vez genera otro campo que se opone al del núcleo, causando un cambio en la inductancia de éste, la detección del metal se realiza mediante la medición de la inductancia del sensor, el objeto debe estar a una distancia cercana comúnmente centímetros (Rodríguez, 2004).

Para una correcta elección de este tipo de sensores se debe tener en cuenta lo siguiente:

- Magnitud de Desplazamiento: Se puede decir que es la distancia que existe entre el sensor y el objeto a detectar, puede ser en milímetros o metros.

- Si el desplazamiento es lineal o angular

Los sensores lineales sirven para monitorear el grosor u otras dimensiones en forma de hoja, si es angular se utiliza en desplazamiento angular de ejes.

- La resolución que se necesita
- La exactitud que se necesita
- El costo
- El material que está hecho el objeto

Algunos funcionan con materiales ferromagnéticos, otros sólo metales y algunos solo aislantes (Bolton, 2013).

#### *1.6.5.2.2 Sensores de contacto.*

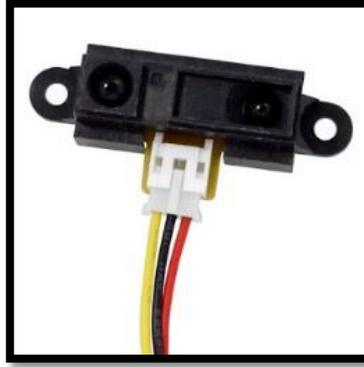
Son interruptores también conocidos como Switchs, emplean para su funcionamiento contactos que pueden ser normalmente abiertos o cerrados, los cuales cierran o abren un circuito eléctrico, según su configuración, como consecuencia del empuje del objeto o máquina que se encuentran detectando.

Existen varios tipos de sensores de contacto entre los principales se encuentran:

- Finales de carrera
- Interruptores manuales y contactores (Rodríguez, 2004).

#### *1.6.5.2.3 Sensores ultrasónicos.*

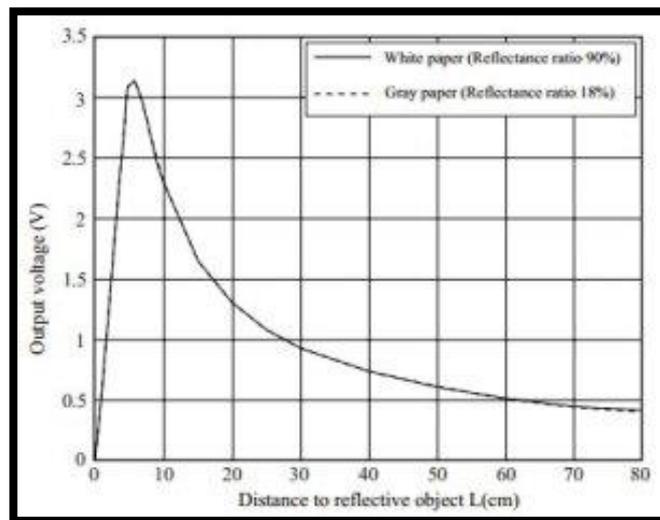
La Figura 20-1 presenta un sensor Sharp tradicional. Su funcionamiento es basado en la emisión de sonidos con frecuencias superiores a las que puede percibir el oído humano, que permiten detectar presencia o medir distancias, a través de la medición del tiempo transcurrido, entre la emisión de la señal, y la recepción del eco correspondiente. La señal que se emite puede ser de tipo pulso, onda, etc.



**Figura 20-1:** Sensor de distancia infrarrojo (Sharp).

**Fuente:** <http://soloarduino.blogspot.com/2015/09/sensor-medidor-infrarrojos-de.html>

Su correcto funcionamiento se basa en el conocimiento de la velocidad del sonido en el aire, dato a través del cual se puede conocer la distancia del objeto que refleja el sonido emitido (ver Figura 21-1), aplicando la ecuación  $d=0.5.v.t$  en donde  $d$  es la distancia a la que se encuentra el objeto, y  $t$  es el tiempo transcurrido, la velocidad de la señal emitida en el medio es de 344 m/s, para el sonido en aire a 20°C.



**Figura 21-1:** Curva de funcionamiento sensor infrarrojo (Sharp).

**Fuente:** <https://minibots.wordpress.com/2013/11/30/utilizacion-del-sensor-de-distancia-sharp-2y0a21/>

El problema fundamental de este tipo de sensores radica en las reflexiones especulares, las cuales son reflexiones que se producen sobre elementos u objetos, con un ángulo determinado que

ocasionan que el tiempo de retorno del eco hacia el receptor sea un poco más alto, provocando una estimación superior de la distancia medida.

Actualmente existen métodos para prevenir este tipo de errores los cuales ya vienen implementados en los productos destinados a cumplir la función de un sensor ultrasónico, su uso es mayormente en medición de nivel en tanques, silos, etc. (Rodríguez, 2004).

### 1.7 Bandas Transportadoras.

Una cinta transportadora como la de la Figura 22-1, es un sistema de transporte continuo formado por una banda continua que se mueve entre dos tambores.



**Figura 22-1:** Cinta Transportadora.

**Fuente:** <http://lonera-sanjavier.com.ar/portfolio/cintas-transportadoras/>

Por lo general, la banda es arrastrada por la fricción de sus tambores, que a la vez este es accionado por su motor. Esta fricción es la resultante de la aplicación de una tensión a la banda transportadora, habitualmente mediante un mecanismo tensor por husillo o tornillo tensor. El otro tambor suele girar libre, sin ningún tipo de accionamiento, y su función es servir de retorno a la banda. La banda es soportada por rodillos entre los dos tambores. Denominados rodillos de soporte (Hinojosa, 2002).

Debido al movimiento de la banda el material depositado sobre la banda es transportado hacia el tambor de accionamiento donde la banda gira y da la vuelta en sentido contrario. En esta zona el material depositado sobre la banda es vertido fuera de la misma debido a la acción de la gravedad y/o de la inercia.

Las cintas transportadoras se usan principalmente para transportar materiales granulados, agrícolas e industriales, tales como cereales, carbón, minerales, etcétera, aunque también se pueden usar para transportar personas en recintos cerrados (por ejemplo, en grandes hospitales y ciudades sanitarias) (Hinojosa, 2002).

### ***1.7.1 Aplicación.***

Los transportadores son utilizados como componentes en la distribución automatizada y almacenamiento. En combinación con manejo equipos computarizados para de tarimas permiten que se realice eficientemente el almacenamiento, manufactura y distribución de materiales en la industria. Es considerado además como un sistema que minimiza el trabajo que permite que grandes volúmenes sean movidos rápidamente a través de procesos, permitiendo a las empresas embarcar o recibir volúmenes más altos con espacios de almacenamiento menores con un menor gasto. El uso de las cintas transportadoras está aplicado especialmente al procesamiento de productos industriales, agroindustriales, agrícolas, mineros, automotrices, navales o farmacéuticos. Esto, al mismo tiempo, puede ser de forma vertical, horizontal o inclinada, dependiendo de los materiales que se quieran transportar (Hinojosa, 2002).

### ***1.7.2 Sistemas de transportadores de bandas.***

Los transportadores de banda son comúnmente utilizados para transportar objetos que tienen una superficie de fondo irregular, pequeños objetos que puedan caerse de entre rodillos o bolsas con producto que pueda atorarse entre los rodillos.

Los transportadores de banda son construidos generalmente de la misma forma: con un bastidor metálico, con rodillos en los extremos y una cuna de deslizamiento sobre chapa o plástico de baja fricción. En aplicaciones donde el producto es demasiado pesado, la cama metálica es sustituida por rodillos. Los rodillos permiten que los objetos sean transportados reduciendo la fricción generada sobre la banda. Los transportadores de banda pueden ser fabricados con secciones

curvas. Estos sistemas de transportadores de banda son comúnmente utilizados en oficinas postales o en los aeropuertos para manejo del equipaje de pasajeros (Hinojosa, 2002).

### **1.7.3 Ventajas.**

Las ventajas que tiene la cinta transportadora son:

- Permiten el transporte de materiales a gran distancia
- Se adaptan al terreno
- Tienen una gran capacidad de transporte
- Permiten transportar un gran variedad de materiales
- Es posible la carga y la descarga en cualquier punto del trazado
- Se puede desplazar
- No altera el producto transportado
- Aumenta la cantidad de producción
- Ausencia de articulaciones de rápido desgaste (Hinojosa, 2002).

## **1.8 Arduino.**

### **1.8.1 Definición.**

Es una plataforma de prototipos de electrónica de código abierto basada en hardware y software flexibles fácil de usar, pensado para artistas, diseñadores y cualquier interesado en crear objetos o entornos interactivos. El microcontrolador de la placa se programa en Arduino programming language, los proyectos pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (Enríquez, 2009).

### **1.8.2 Ventajas de usar Arduino.**

- Barato: Las placas de Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras.
- Multiplataforma: Se ejecuta en sistemas operativos Windows, OSX, y GNU/Linux.
- Entorno de programación simple: El entorno es fácil de usar para principiantes, pero

flexible para que usuarios avanzados puedan aprovecharlo.

- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores avanzados.
- Código abierto y hardware extensible: El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel, incluso usuarios relativamente inexpertos pueden construir la versión de la placa para entender cómo funciona (Enríquez, 2009).

### **1.8.3 Hardware.**

Hay múltiples versiones de la placa Arduino, la mayoría usa el ATmega168 mientras que las más antiguas utilizan las ATmega8.

#### ➤ Placas de E/S.

- Diecimila: Es la más popular, se conecta al computador con cable USB estándar y contiene todo lo que necesitas para programar y usar la placa.
- Nano: Es una placa compacta para el uso como tabla de pruebas, su conector al ordenador es USB Mini-B.
- Bluetooth: El arduino BT contiene un módulo que permite comunicación y programación sin la necesidad de cables.
- LilyPad: Diseñada para aplicaciones lista para llevar con un estilo sublime.
- Mini: Esta es la placa más pequeña de los arduinos, sirve para aplicaciones que necesiten ahorro de espacio.
- Serial: Es una placa básica que usa RS232 como un interfaz con el ordenador para programación y comunicación (Enríquez, 2009).

#### ➤ Estructura de placa Arduino.

- Vista General: Arduino es una placa microcontroladora basado en el ATmega168, está compuesto de 14 pines de entrada/salida digital, 6 pines de entradas analógicas, un oscilador a 16 MHz, un conector USB, un conector para alimentación y un botón de reset. Estas placas generalmente funcionan con un voltaje de operación de 5V, su voltaje de entrada es máximo de 20 V la que no es muy recomendada (Enríquez, 2009).

- Memoria: El ATmega168 tiene 16 KB de memoria flash para almacenar código de los cuales 2 KB son reservados para el bootloader, 1 KB de SRAM y 512 bytes de EEPROM (Enríquez, 2009).
- Entradas y Salidas: Cada uno de los 14 pines pueden ser usados como entradas o salidas, usando funciones como `pinMode ()`, `digitalWrite ()` y `digitalRead ()`, pueden recibir y proporcionar un máximo de 40 mA. Existen funciones especiales como:
  - ✓ Serial: 0 (Rx) y 1 (Tx)
  - ✓ Interruptores externos 2 y 3
  - ✓ PWM: 3, 5, 6, 9, 10 y 11
  - ✓ SPI: 10 (SS), 11 (MOSI), 12 (MOSI), 13 (SCK)
  - ✓ LED: conectado internamente al pin 13
- Comunicación: El Arduino tiene la opción de comunicarse con un ordenador, con otro Arduino, u otros microcontroladores. Posee una comunicación serie UART TTL (5V), el cual está disponible en los pines 0 y 1 conocidos como (Rx) y (Tx) respectivamente (Enríquez, 2009).
- Programación: El Arduino puede ser programado con el software Arduino el cual está disponible en código abierto en la Web. La placa Arduino está incluido con un bootloader pregrabado el cual permite subirle nuevo código sin usar un programador hardware externo (Enríquez, 2009).
- Protección de Sobrecarga del USB: El Arduino tiene un fusible reseteable que protege tus puertos USB del ordenador de cortes y sobrecargas, el fusible proporciona una capa de protección extra. Si más de 500 mA. se aplican al puerto se romperá la conexión hasta que el corte o la sobrecarga sean eliminados (Enríquez, 2009).

## **1.9 Control de calidad.**

El control de calidad de los productos es el proceso por el cual se establecen y se cumplen normas que aseguran el cumplimiento de las especificaciones del producto.

### ***1.9.1 Etapas para el control de calidad.***

- Establecer el objeto del control. Elegir una unidad de medida.
- Establecer el valor normal o estándar de la especificación a controlar.

- Establecer un instrumento de medida.
- Realizar la medición de la magnitud en la unidad elegida.
- Interpretar las diferencias entre el valor real y el valor normal o estándar.
- Actuar sobre las diferencias encontradas.

El control de calidad no se aplica únicamente al producto final, sino que se realiza a lo largo de todo el proceso de producción. Es decir: en la recepción de materias primas, en el proceso de fabricación, en los productos semielaborados y en el propio producto final.

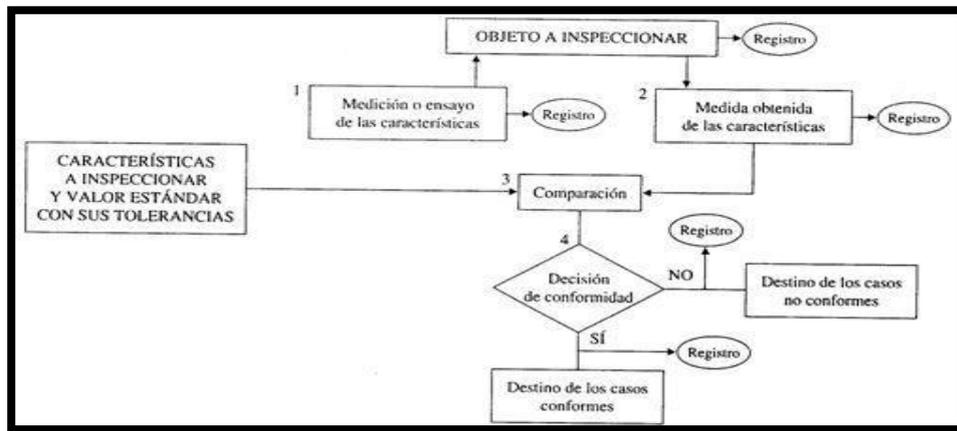
Para realizar el control de calidad, se realiza la inspección y el ensayo de determinadas características de los productos a controlar (Montoya, 2014).

### **1.9.2 Inspección.**

La inspección es la evaluación de la calidad de alguna característica del producto. Las actuaciones para realizar una inspección son las siguientes:

- Determinación de la característica del producto a inspeccionar.
- Medición o ensayo de la calidad de esa característica.
- Comparación de la característica especificada o estándar con la medida obtenida y determinación de la conformidad.
- Decisión sobre el destino de los casos conformes.
- Decisión sobre el destino de los casos no conformes.
- Registro de los datos obtenidos.
- En el proceso de producción existen diversos puntos de inspección. Estos puntos de inspección se determinan según un plan preestablecido. Para cada punto de inspección es necesario determinar todas las actuaciones del proceso de inspección (Montoya, 2014).

### 1.9.3 Esquema de un proceso de inspección.



**Figura 23-1:** Esquema básico de un proceso de inspección.

Fuente: [http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/indata/Vol6\\_n2/pdf/sistema.pdf](http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/indata/Vol6_n2/pdf/sistema.pdf)

### 1.9.4 Modalidades de inspección.

La inspección puede ser visual, mediante la medida de una magnitud física sencilla o mediante ensayos de laboratorio. Dependiendo de la naturaleza del producto, la inspección puede ser de diversos tipos:

- Inspección de unidades discretas de producto, es decir, de elementos independientes, como tornillos, platos, tazas, bolas de rodamiento.
- Inspección de partes de una masa de la que se ensaya una muestra, como, por ejemplo, una colada de acero o una cuba de vino.

Normalmente los productos se inspeccionan en forma de lote, que es un conjunto de unidades del producto elaboradas bajo condiciones homogéneas.

La intensidad de la inspección puede tomar diversos grados:

- No inspeccionar, cuando existen garantías suficientes por el propio dominio del proceso de fabricación de que el producto cumple las especificaciones.
- Inspección por muestras pequeñas, pues existe un gran nivel de dominio del proceso que únicamente requiere comprobaciones para verificar el mantenimiento de las condiciones.
- Inspección por muestras amplias, mediante muestreo aleatorio.
- Inspección al 100% cuando se trata de productos críticos, especialmente complejos o con procesos de fabricación muy variables en los que el muestreo estadístico es insuficiente.

La Inspección de unidades discretas de producto se hará énfasis en nuestro proyecto de titulación (Montoya, 2014).

### **1.10 Indicadores de calidad.**

Un indicador es una medida cuantitativa que puede usarse como guía para controlar y valorar la calidad de los diferentes productos. Es decir, la forma particular (numérica) en la que se mide o evalúa cada uno de los criterios. Los indicadores de calidad se construyen a partir de la experiencia, del conocimiento sobre el área en el que se trabaja.

Para este trabajo se ha establecido como indicador de calidad al área del objeto escaneado, la forma y el color (Montoya, 2014).

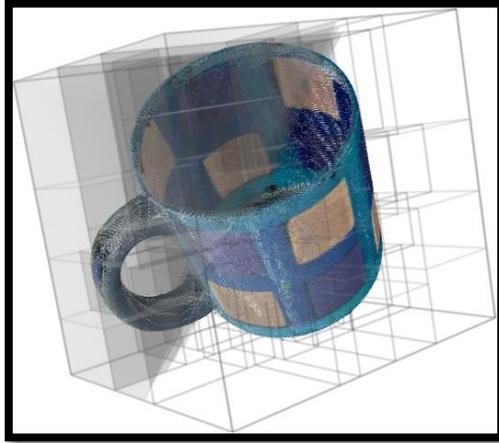
#### **1.10.1 *Sistema de medición.***

El mecanismo sistemático y permanente de monitoreo de resultados y alcance de la operación diaria de productos finales, se realizara a través de la adquisición de datos del sensor Kinect para evaluar el cumplimiento del criterio de calidad, a través de indicadores y metas.

- Genera información vital para la toma de decisiones en la instrumentación de acciones que apoyarán el logro del mejoramiento.
- Permite ubicar el nivel de eficacia y eficiencia de la industria
- Permite la difusión de logro de objetivos, tanto al interior como al exterior (Montoya, 2014).

#### **1.10.2 *Criterio de Calidad del producto.***

Los criterios se definen como aquella condición que debe cumplir una determinada actividad, actuación o proceso para ser considerada de calidad. Es decir qué se persigue, cuál es el objetivo, qué se pretende teniendo en cuenta aquellas características que mejor representan (siempre que pueden medirse) lo que se desea lograr (Montoya, 2014).



**Figura 24-1:** Análisis de taza modelo.

**Fuente:** [http://docs.pointclouds.org/trunk/group\\_\\_kdtree.html](http://docs.pointclouds.org/trunk/group__kdtree.html)

### ***1.10.3 Indicadores de calidad.***

- Área
- Color
- Forma

## CAPÍTULO II

### 2 MARCO METODOLÓGICO.

En el presente capítulo describiremos los métodos que se utilizaron para la realización y cumplimiento de los diferentes objetivos propuestos en el presente trabajo.

#### 2.1 Obtención y procesamiento de escenas Tridimensionales.

En el campo de la V.A. existen diferentes formas de obtener escenas tridimensionales con la ayuda de distintos equipos. Detallaremos algunos métodos para la obtención de estas escenas.

##### 2.1.1 Método RGB

Los métodos basados en RGB estarían clasificados dentro de los pasivos. Estos procedimientos confían en la existencia de luz visible para determinar la forma del objeto. Existen numerosos métodos que se pueden incluir en esta categoría.

Vouzounaras propuso un método para determinar la geometría tridimensional de un objeto a partir de una única imagen RGB usando la detección de los puntos de fuga. Este método cuenta con el conocimiento que se tiene de las estructuras geométricas tales como los planos ortogonales. Requiere la presencia del suelo, del techo y de las paredes para que el algoritmo funcione. Son capaces de reconstruir objetos de geometría sencilla, pero sin embargo, no pueden hacer lo mismo con objetos esféricos (Vouzounaras, *et al.*, 2010)

Saxena llevó a cabo una investigación sobre la reconstrucción de escenas en 3D a partir de una o muy pocas imágenes tomadas con una cámara de RGB. Lo primero que hacían era sobre segmentar la imagen en pequeños parches llamados superpíxeles. Simultáneamente intentan

encontrar la posición y orientación tridimensional de cada uno de esos superpíxeles. Para hacer esto utilizan el campo aleatorio de Markov (MRF) (Saxena, *et al.*, 2007).

Weiss estudió la posibilidad del reconocimiento de objetos a partir de una imagen en 2D usando invariantes geométricas. Esas invariantes las encontraban al hacer ciertas suposiciones para modelos tridimensionales de los objetos. Esas hipótesis podían ser para un modelo particular o para uno más general. Después, a través de ciertas relaciones algebraicas eran capaces de describir modelos invariantes en 3D (Weiss & Ray, 2001).

Analizando la documentación sobre los métodos basados en RGB podemos comprobar que existe una gran desventaja. Todos estos procedimientos requieren un paso computacional adicional para la detección de la profundidad. También pueden necesitar ayuda física extra, como las cuatro cámaras y la mesa rotatoria. Además, la gran mayoría de estos métodos solo detectan ciertos tipos de objetos, normalmente los que son planos, mientras que los esféricos los pasan por alto.

### **2.1.2 Método Estereográfico.**

Los métodos estereográficos están categorizados en el mismo grupo que los de RGB, no necesitan contacto y son pasivos. De todos modos, existe una gran diferencia entre los dos. El estereográfico necesita dos cámaras de RGB posicionadas de cierta manera y previamente calibradas. Debido a la facilidad de adquirir este tipo de cámaras en la actualidad, existen multitud de investigaciones acerca de la construcción de un mapa de profundidad a partir de las mismas.

Huang propone un método para capturar la imagen tridimensional usando una cámara estereográfica y utilizando el teorema que permite obtener la forma del objeto a partir de su sombra (shape-from-shading) (Huang & Chen, 2011). Como ya usara Zheng, se usa una mesa rotatoria para capturar al objeto desde todas las perspectivas. El núcleo de su investigación se basa en la detección de los puntos clave del objeto, como las esquinas, lo que resulta ineficaz para los objetos esféricos (Zheng, *et al.*, 2011).

Hub hizo un buen trabajo sobre el reconocimiento y el rastreo de objetos móviles para ayudar a los invidentes. Utilizan una cámara estereográfica acoplada a la cabeza del usuario de forma que

el sistema va informando de la existencia de objetos, y que clase de objetos se tratan, si se pueden mover o por el contrario hay que evitarlos (Hub, *et al.*, 2006).

### **2.1.3 Método Time-of-flight.**

Los métodos basados en time-of-flight (tiempo de vuelo) calculan la distancia del objeto a partir del tiempo que tarda un pulso de radiación emitida de luz infrarroja, en ir y volver. Por este motivo, estos procedimientos se incluyen en el grupo de sin contacto activo.

Pheat describe un método en el que usa dos diodos láser y una cámara digital. También usan una mesa rotatoria para poder ver al objeto desde todos los ángulos. Normalmente necesitan entre 200 y 400 escaneos para reconstruir un simple objeto (Pheatt, *et al.*, 2005).

Hagebeuker introduce una nueva generación de cámaras de tipo time-of-flight en. Este nuevo tipo de cámaras, llamadas PMD (Photonic Mixer Device), tienen las ventajas de ser más rápidas que sus predecesoras y tener mayor resolución lateral además de seguir aportando información fiable de profundidad (Hagebeuker, 2007).

### **2.1.4 Método de luz estructurada.**

Las cámaras de profundidad que usan el método de luz estructurada normalmente contienen una cámara RGB normal junto a un emisor y receptor de luz infrarroja. A este tipo de cámaras se las llama habitualmente RGB-D y cada vez son más fáciles de conseguir para los consumidores. Tenemos un gran ejemplo con el dispositivo Kinect de Microsoft, el núcleo de nuestro proyecto. Las cámaras RGB-D estarán incluidas dentro del grupo de sin contacto activo.

Existe numerosa documentación acerca de la adquisición de datos a través de la Kinect. En el artículo publicado por Cruz podemos aprender sobre los últimos avances llevados a cabo con dicho sensor. También podemos leer acerca de los retos que se presentan así como la multitud de aplicaciones que se pueden llevar a cabo con este dispositivo (Cruz, *et al.*, 2012).

Izadi propone un sistema para capturar ciertos objetos dentro de una escena a través del sensor Kinect. En este caso, primero se reconstruye la imagen por completo, después el usuario mueve el objeto que queremos identificar, lo que consigue separar el objeto del resto de la escena (Izadi, *et al.* 2011).

Otro ejemplo de captura de datos a través de una cámara RGB-D lo encontramos en Henry (2010). Hacen un estudio acerca del uso que se puede dar de estas cámaras en el ámbito de la robótica, especialmente para hacer mapas tridimensionales de entornos interiores. Con esos mapas podemos desarrollar aplicaciones de navegación de robots.

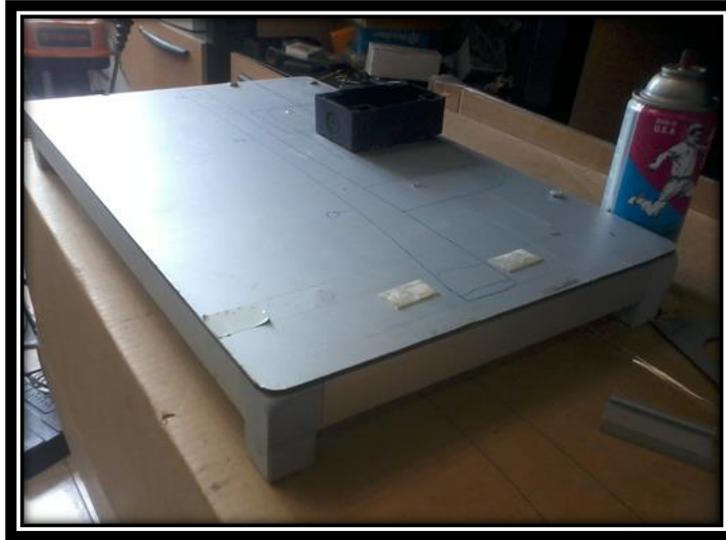
## **2.2 Arquitectura de los equipos receptores.**

El sensor Kinect es un dispositivo electrónico desarrollado por la empresa Microsoft, diseñado para videojuegos en Xbox 360, actualmente se están utilizando para experimentos tecnológicos, su funcionalidad es amplia ya que dispone de tres sensores incluidos: RGB, profundidad, infrarrojo, también dispone de un micrófono y un motor que permite girar la cámara 60 grados horizontalmente.

Se ha optado la utilización de este dispositivo, para así extraer datos del mundo real y procesarlos mediante algoritmos computacionales y desplegar resultados.

Los sensores kinect reciben datos a partir de los 40 cm de longitud respecto al objeto, lo cual se ha previsto adoptar una arquitectura específica ajustable para tener resultados exactos.

En la Figura 1-2 se observa la mesa que soporta el prototipo, el material es de aluminio ligero para ser transportado, luego se instala los receptores tecnológicos.



**Figura 1-2:** Mesa para Kinect.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Se han ubicado dos sensores a 45 grados respecto del objeto, situados a izquierda y derecho respectivamente. En la Figura 2-2 y 3-2 se observa el apoyo del Kinect entornillado a la mesa de la maqueta, su altura es paralelo de la banda transportadora.



**Figura 2-2:** Kinect 1.

**Realizado por:** Logroño C. y Mishqui K., 2016.



**Figura 3-2:** Kinect 2.

**Realizado por:** Logroño C. y Mishqui K., 2016.

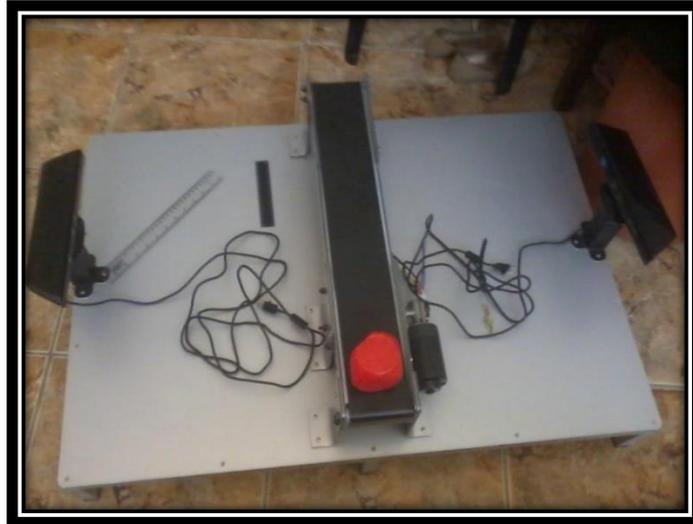
Mediante esta arquitectura se tienen los mejores resultados, puesto que estos sensores toman datos a partir de los 40 cm hasta los 8 metros, fuera de este rango el Kinect entra en zona desconocida.

En la Figura 4-2 se observa la ubicación de los dos Kinect, para el lado izquierdo y derecho, es decir recibe datos de ambas caras del objeto. La ubicación de un tercer dispositivo tiene que ser estratégico para la captura de un tercer lado o el interior de la taza cerámica, o se puede definir mediante el producto a analizar.



**Figura 4-2:** Ubicación Kinect 1.

**Realizado por:** Logroño C. y Mishqui K., 2016.



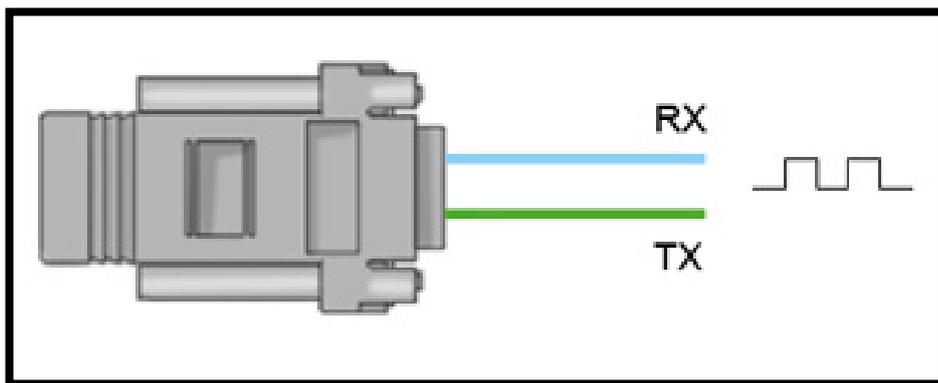
**Figura 5-2:** Ubicación Kinect 2.

Realizado por: Logroño C. y Mishqui K., 2016.

### 2.3 Comunicación entre la PC y el sistema automático.

Una interfaz física o virtual permite la transmisión y recepción de datos entre dos ordenadores o microcontroladores.

El puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión) (ver Figura 6-2).



**Figura 6-2:** Comunicación Serie.

Fuente: Luis Llamas, 2014, <http://www.luisllamas.es/2014/04/arduino-puerto-serie/>

El microcontrolador va a detectar estados de un dispositivo y el envío de señales al ordenador. El equipo hará la interpretación de todos estos valores, y el envío de mensajes de un lado para alterar el comportamiento del Arduino (ver Figura 7-2).



**Figura 7-2:** Microcontrolador Arduino uno R3.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Se necesita las siguientes librerías para los puertos USB.

```
#include <studio.h>  
#include <iostream>
```

Con la siguiente línea de código se abre el puerto ttyACM0 y se guarda el valor digital en la variable "W".

```
file = fopen ("/dev/ttyACM0","w");
```

Entonces, se procede a la programación en el microcontrolador:

Inicializar la comunicación a 9600 baudios.

```
Serial.begin (9600);
```

Leer el serial (1, 0) y almacena en la variable lectura.

```
lectura = Serial.read();
```

Imprime en pantalla la variable receptada.

```
Serial.println (lectura);
```

## **2.4 Diseño de la banda para la clasificación de tazas.**

### **2.4.1 Bandas Transportadoras.**

- Características Generales del material a transportar.

Las características de los materiales son esencialmente determinantes para el diseño del sistema de transporte, puesto que la cinta seleccionada debe cumplir con los requisitos confiables de vida útil para el tipo de material a transportar.

Se debe tener en cuenta las siguientes propiedades para realizar un buen diseño por ejemplo:

- a) Peso específico: 10 kg
- b) Tamaño: 10 cm cuadrados
- c) Forma: Taza cerámica
- d) Material de construcción: cerámico
- e) Resistencia: mínima

Se debe tener en cuenta si la temperatura del material a transportar para así seleccionar un recubrimiento adecuado.

### **2.4.2 Partes principales de una banda Transportadora.**

#### **2.4.2.1 Recubrimiento.**

Los recubrimientos de goma sirven para unir los elementos constitutivos de la carcasa y constan de dos partes, la superior y la inferior.

El espesor del recubrimiento de la carcasa está en función del tipo de aplicación de la banda y de la anchura de esta.

La goma es el elemento básico de los recubrimientos; tomando en consideración las propiedades mecánicas de resistencia, alargamiento y abrasión.

Como se observa en la Figura 8-2, el recubrimiento se ha fabricado de material de carpas de caucho, ideal para cintas transportadoras.



**Figura 8-2:** Recubrimiento.

Realizado por: Logroño C. y Mishqui K., 2016.

- Ancho de banda : 10 cm
- Longitud de banda: 71 cm
- Grosor de banda: 3 mm
- Material: goma

### 2.4.3 *Motor.*

Para proyectar una cinta transportadora, lo importante es las características del material a transportar, teniendo en cuenta la gran diversidad de materiales existentes. Para este trabajo se va clasificar objetos que no superan los 10cm y los 5kg de peso.

Determinar la velocidad del transportador.

Como datos iniciales tenemos que 5 tazas x Banda x minuto

$$\frac{5 \text{ unidades}}{1 \text{ minuto}}$$
$$\frac{5 \text{ unidades}}{1 \text{ minuto}} \times 0.7 \text{ m}$$
$$3,5 \frac{\text{metros}}{\text{min}}$$

$$3,5 \frac{\text{metros}}{\text{min}} + 30\% = 14.92 \frac{\text{pies}}{\text{min}}$$

Velocidad aproximada necesaria:

$$15 \frac{\text{pies}}{\text{min}}$$

Determinar capacidad de carga necesaria. 5 kilogramos en 0.7 metros que tiene nuestra banda.

$$7.145 \frac{\text{kg}}{\text{m}} = 4.79 \frac{\text{lb}}{\text{pies}}$$

Cálculo de la potencia del motor:

Para el cálculo de la unidad del transportador (potencia necesaria para mover la carga), tomando en cuenta el coeficiente de fricción dado por el tipo de banda a utilizar y tipo de transportador a utilizar obtenido en el Anexo A, tenemos el tipo:

- FEBOR 15 NF PVC color negro 01
- Ancho de 3000mm
- Peso 2.6kg/m<sup>2</sup>
- El coeficiente de fricción es de 0.4

Para determinar la potencia del motor necesaria se utiliza la siguiente ecuación:

$$HP = \frac{(W + w) * f * s}{33000}$$

Donde:

W= peso de carga (lb)

w= peso de banda (lb)

f= coeficiente de fricción

s= velocidad

$$HP = \frac{(61,81 + 188.62) * 0.4 * 15}{33000}$$

Multiplicar por el factor de servicio y aproximamos a valores nominales existentes en el mercado.

$$HP = 0.048 * 1.5 = 0.06829$$

$$P = 52 W \approx 90 W$$

En la Figura 9-2, el modelo del motor que se utilizó es: Z5D90-24.

Descripción de Producto									
motor eléctrico del engranaje de la C.C. de 12v/24v 90watt									
Parametrización para la optimización del tratamiento del motor									
Tipo modelo	Voltaje	Energía	Parámetros sin carga		Parámetros de la carga			Vida del cepillo H	Peso del motor Kilogramo
	V	W	Velocidad r/min	Actual A	Velocidad r/min	Esfuerzo de torsión mN.m	Actual A		
Z5D90-12	12	90	3100	2.0Max.	2600	330	12.0	2000	2.2
Z5D90-24	24	90	2800	1.0Max.	2800	307	5.0	2000	2.2
Z5D90-90	90	90	2900	0.3Max.	2900	307	1.4	2000	2.2

**Figura 9-2:** Descripción de Motores DC.

Realizado por: Logroño C. y Mishqui K., 2016.

Se ha visto la necesidad de utilizar el siguiente motor de 24v DC, 4 amperios (ver Figura 10-2).

$$P = V * I$$

$$P = 24 * 4$$

$$P = 96 watt = 0,13 hp$$

Dimensionar correctamente el conductor principal de alimentación para el motor eléctrico: 14 AWG, puesto que se requiere 4 Amperios y una sola carga.



**Figura 10-2:** Motor de 24VDC.

**Realizado por:** Logroño C. y Mishqui K., 2016.

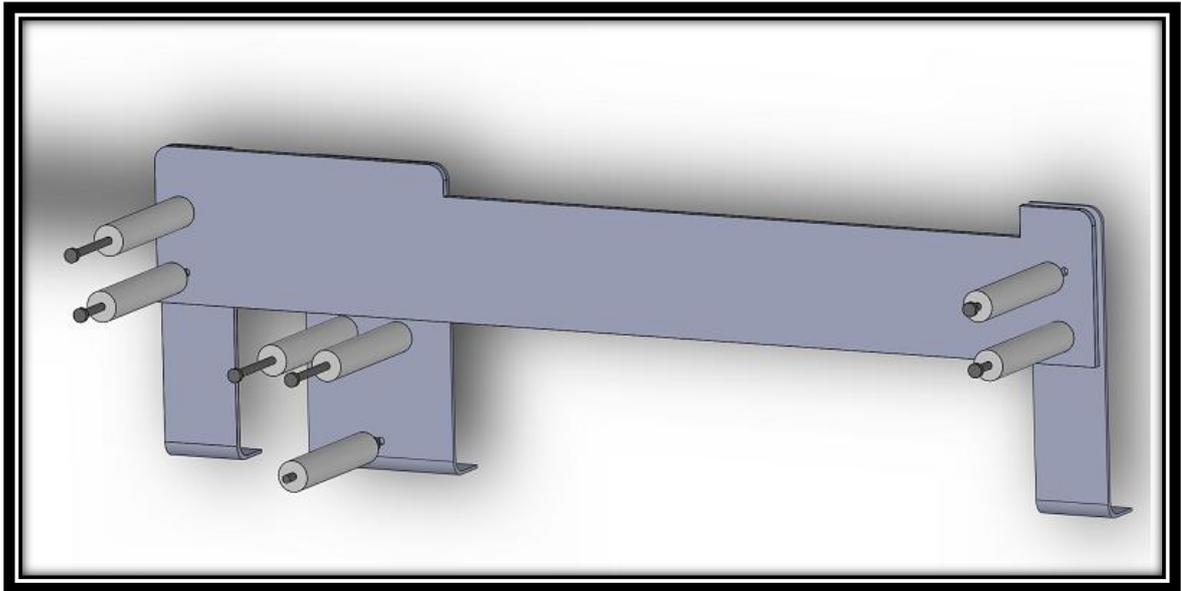
- Fuente de voltaje: 24 DC, 5 amperios
- Cable conductor: 14 AWG

#### **2.4.4**     *Los Rodillos.*

Los Rodillos son cilíndricos con la superficie exterior e interior lisa, para un mejor deslizamiento del recubrimiento.

- Lograr el adecuado contacto entre la banda y el tambor motriz.
- Evitar derrames de material en las proximidades de los puntos de carga, motivados por falta de tensión en la banda.
- Compensar las variaciones de longitud producidas en la banda, estas variaciones son debidas a cambios de tensión en la banda.
- Mantener la tensión adecuada en el ramal de retorno durante el arranque.

Como se observa en la Figura 11-2, un diseño previo de piezas en 3d desarrollada en Solidwork, la ubicación de los rodillos en la banda transportadora permiten mover la cinta de goma.



**Figura 11-2:** Sistema de ubicación de Rodillos.

Realizado por: Logroño C. y Mishqui K., 2016.

- Longitud del rodillo: 9,8 cm
- Diámetro externo de rodillo: 1,5 cm
- Diámetro interno de rodillo: 0,5 cm
- Número de rodillo: 7 unidades

Luego del diseño en Solidwork se implementó la banda Transportadora como se observa en la Figura 12-2, sus dimensiones son acorde a los planos, para no tener inconvenientes con el deslizamiento de la cinta.



**Figura 12-2:** Rodillos.

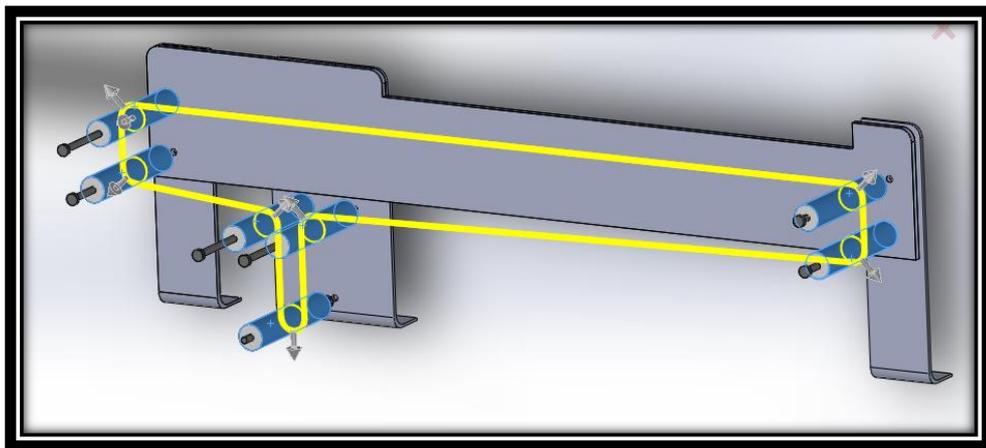
Realizado por: Logroño C. y Mishqui K., 2016.

#### 2.4.5 *Tensores de banda.*

En la Figura 13-2, se observa cómo actúan los rodillos con la cinta transportadora, el rodillo tensor con la flecha vertical hacia abajo permite ajustar la banda y así tener mejor adherencia, el tensor actúa con el motor de 24 voltios.

Los Dispositivos de tensado cumplen las siguientes funciones:

- Lograr el adecuado contacto entre la banda y el tambor motriz.
- Evitar derrames de material en las proximidades de los puntos de carga, motivados por falta de tensión en la banda.
- Compensar las variaciones de longitud producidas en la banda, estas variaciones son debidas a cambios de tensión en la banda.
- Mantener la tensión adecuada en el ramal de retorno durante el arranque.



**Figura 13-2:** Tensores de Banda.

Realizado por: Logroño C. y Mishqui K., 2016.

- Por su forma constructiva: De lazo múltiple.
- Por la forma de aplicar la fuerza tensora: Fija.
- Por el equipo mecánico que aplica la fuerza: eléctrico automático.

#### 2.4.6 *Pallets*

Un armazón de material plástico empleado en el movimiento de carga, facilitando el levantamiento, manejo y dirección del objeto a transportar.

La figura 14-2 muestra el pallet fabricado de material alocubond, permite evitar fisuras al momento de transportar la taza cerámica.



**Figura 14-2:** Pallet.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- Dimensiones: 7cm largo x 7cm ancho
- Alto: 7 mm

#### **2.4.7 Sensores.**

Como se observa la imagen 15-2 el sensor Sharp para la detección de objetos y activación de la banda transportadora, están ubicados en aluminio.

Este sensor detecta la presencia de la taza cerámica en la banda transportadora, luego da inicio al scanner.

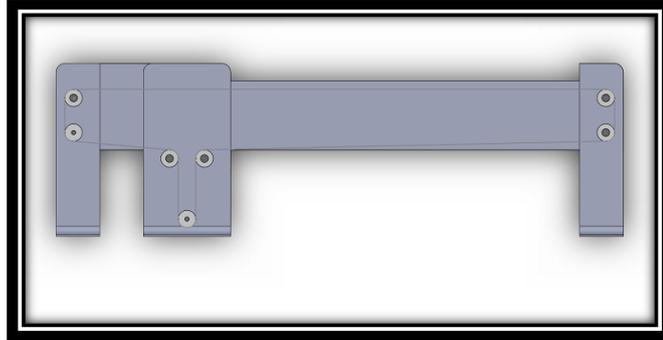


**Figura 15-2:** Acoplamiento de sensor Sharp.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- Sharp: Sensor Sharp 2Y0A21
- Voltaje: 5v dc 20 mA
- Distancia de detección: 18 cm

**2.4.8** *Diseño final de la banda Transportadora.*

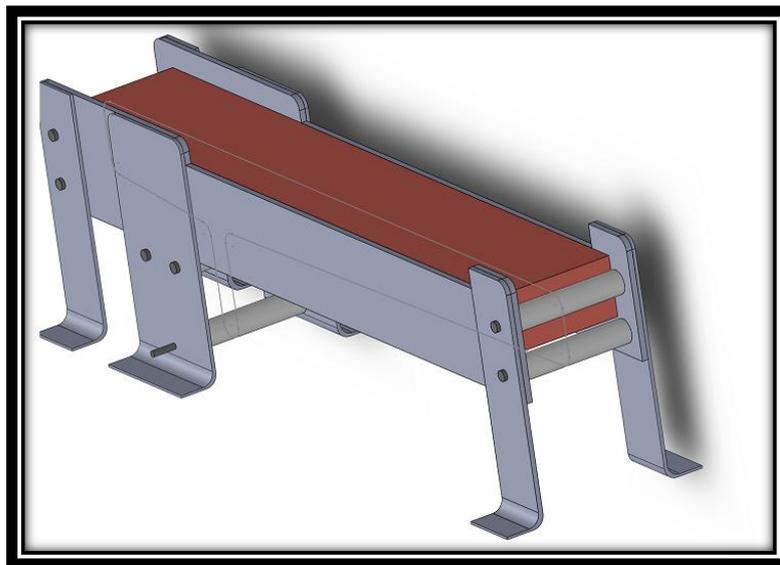


**Figura 16-2:** Vista lateral de la banda.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- Longitud: 70 cm
- Altura: 25 cm
- Ancho : 13 cm

Como se observa la Figura 17-2, el diseño final en Solidwork de la banda Transportadora, nos permite verificar el buen funcionamiento de todos los rodillos, y el acople de todos los materiales para fabricar el prototipo.



**Figura 17-2:** Vista general de la banda.

**Realizado por:** Logroño C. y Mishqui K., 2016

## 2.5 Algoritmo de reconocimiento para comparación y clasificación.

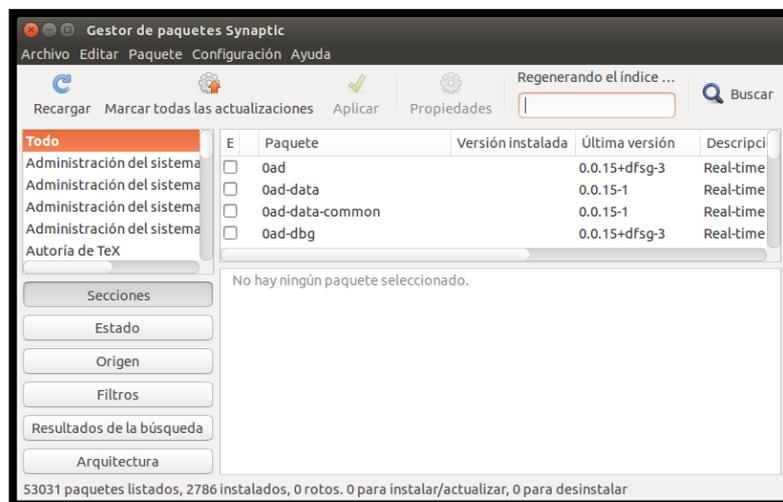
El algoritmo para la comparación y clasificación de las tazas se lo realizó en el lenguaje de programación C++, con la ayuda de las librerías de OpenCV, LibFreenect

### 2.5.1 *Instalación de Software requerido.*

Debido a la singularidad del trabajo investigativo en Open Source fue necesario incluir un apartado en donde se muestre la instalación de las herramientas más importantes que intervinieron en el presente trabajo.

#### 2.5.1.1 *Instalación de LibFreenect.*

- Abrir el gestor de paquetes synaptic (ver Figura 18-2).



**Figura 18-2:** Synaptic de Ubuntu 14.04 LTS.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Buscar y descargar:

- git core
- CMake
- freeglut3-dev

- pkg - config
- build - essential
- libxmu -dev
- libxi -dev
- libusb -1.0-0-dev

Abrir el terminal y ejecutar:

- git clone https://github.com/OpenKinect/libfreenect.git

Aparece el directorio LibFreenect. Se crea una carpeta build y en ella se compilan los ejecutables que vienen como demos en el LibFreenect, los cuales se encontrarán en la carpeta bin.

### 2.5.1.2 Instalación de OpenCV.

El proceso resumido es el siguiente:

- Instalar los paquetes necesarios para el requerimiento del funcionamiento del OpenCV (ver Figura 19-2).

```
c23@c23-Scanner3D:~$ sudo apt-get install build-essential libgtk2.0-dev libjpeg-dev libtiff4-dev libjasper-dev libopenexr-dev cmake python-dev python-numpy python-tk libtbb-dev libeigen2-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev libqt4-dev libqt4-opengl-dev sphinx-common texlive-latex-extra libv4l-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev
```

**Figura 19-2:** Instalación de paquetes necesarios para el OpenCV.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- Una vez tenido esto, descargamos el directorio que contiene al OpenCV (ver Figura 20-2).

```
c23@c23-Scanner3D:~$ cmake -D WITH_TBB=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON ..
CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

**Figura 20-2:** Compilación del paquete OpenCV.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Entonces el momento de la configuración indispensable para que el OpenCV trabaje correctamente.

```
c23@C23-Scanner3D:~$ sudo gedit /etc/ld.so.conf.d/opencv.conf
```

**Figura 21-2:** Configuración del OpenCV.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- El código de la Figura 21-2 abre un editor de texto, en el cual se debe guardar lo siguiente (ver Figura 22-2).



```
opencv.conf x  
1 /usr/local/lib
```

**Figura 22-2:** Configuración de las rutas de las librerías instaladas.

**Realizado por:** Logroño C. y Mishqui K., 2016.

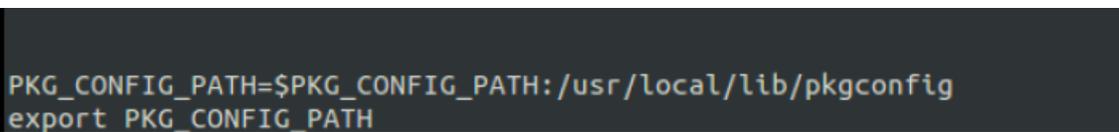
- Escribimos el siguiente código para abrir otro editor de textos (ver Figura 23-2).

```
c23@C23-Scanner3D:~$ sudo gedit /etc/bash.bashrc
```

**Figura 23-2:** Código de generación de Bash.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Y al final de lo ya escrito añadimos lo siguiente (ver Figura 24-2):



```
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig  
export PKG_CONFIG_PATH
```

**Figura 24-2:** Configuración final del PKG\_CONFIG\_PATH.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Ahora tenemos el OpenCV totalmente configurado y se puede comprobar probando alguno de los ejemplos demo (Montabobe, 2014).

## 2.5.2 *Instalación del IDE de Arduino.*

Para empezar con nuestro Arduino necesitaremos de su IDE. A continuación se explicara como instalarlo en una Ubuntu 14.04 (Montabobe, 2014).

### 2.5.2.1 *Descarga del IDE de Arduino.*

A continuación podemos descargar el IDE oficial de Arduino desde su página web oficial, o bien desde los repositorios oficiales de Ubuntu (ver Figura 25-2). Se instalara la versión de 64 bits para GNU/Linux.

```
c23@C23-Scanner3D:~$ sudo apt-get install arduino-core
```

**Figura 25-2:** Descarga Arduino modo consola.

**Realizado por:** Logroño C. y Mishqui K., 2016.

### 2.5.2.2 *Ejecutar Arduino.*

A continuación probaremos que el Arduino está instalado y funcionando correctamente en nuestro GNU/Linux.

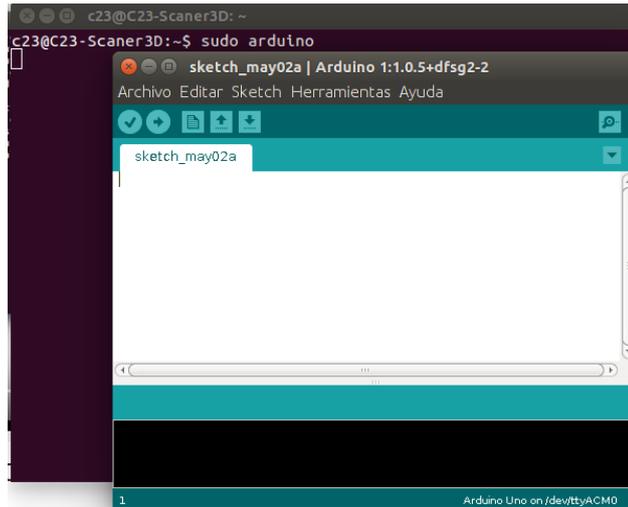
Primero escribimos el siguiente comando en la consola para ejecutar Arduino y darle permisos de súper usuario (ver Figura 26-2).

```
c23@C23-Scanner3D:~$ sudo arduino
```

**Figura 26-2:** Ejecutar Arduino modo consola.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Luego de ejecutar el comando nos aparecerá una ventana que nos muestra el IDE de Arduino, es decir la pantalla de programación (ver Figura 27-2). Ahora ya podemos utilizar y conectar el Arduino.

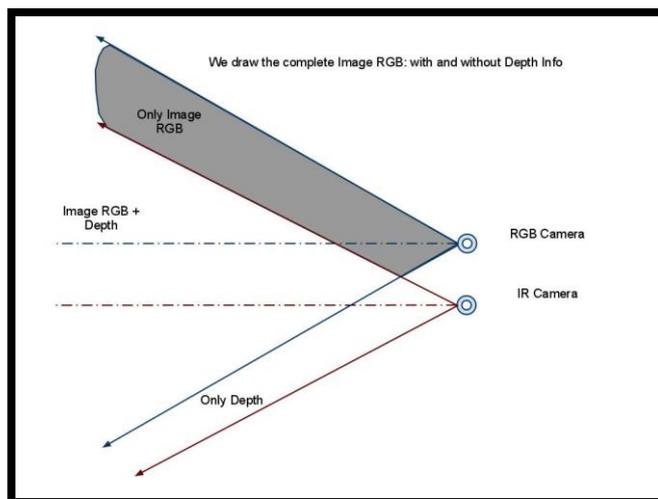


**Figura 27-2:** IDE de Arduino.

**Realizado por:** Logroño C. y Mishqui K., 2016.

### 2.5.3 *Tratamiento de la Visión del Kinect.*

La "doble" visión del Kinect (ver Figura 28-2) es el baluarte más apreciado de su construcción. Sin embargo, dentro de todas sus virtudes, es quizá el único problema que tiene, después de lograr su completo funcionamiento por código, ya que por razones de construcción la visión RGB y la Depth no son exactamente las mismas en cuanto a dimensión y posición. Ya que por defecto en las dos cámaras tienes distintas resoluciones y eso puede hacer caer en el error de los cálculos finales.



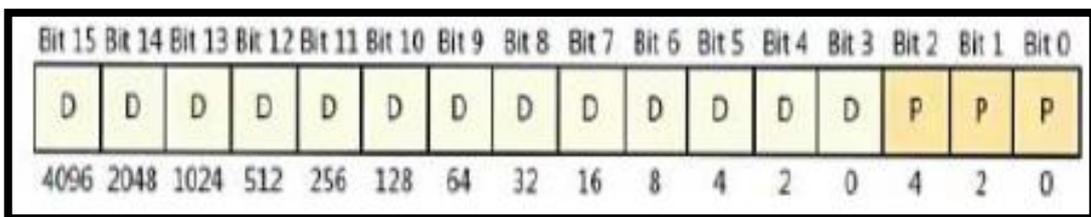
**Figura 28-2:** Doble visión del Kinect.

**Fuente:** 2011, <https://landerpfc.files.wordpress.com/2011/03/verticalblackwideline.jpg>

La calibración del Kinect se debe realizar de una forma similar a como lo hacen las cámaras convencionales, o más bien orientar el espacio de trabajo con un detalle capaz de no equivocar los cálculos generados. O más bien los resultados se deben trabajar teniendo en cuenta esta peculiaridad.

### 2.5.4 Obtención del dato de profundidad.

Ha sido el punto más importante del Trabajo. Pero el proceso que se ha llevado a cabo es el mismo en todas las ocasiones.



**Figura 29-2:** Bits de información DEPTH.

**Fuente:** usando SDK Kinect.

Como se observa de la Figura 29-2, la información que devuelve Kinect de su sensor infrarrojo tiene 16 bits de tamaño short. De los cuales los tres primeros son usados por el SDK Kinect para aplicaciones de reconocimiento esquelético. Por lo que han quedado otros 13 bits. Sin embargo, en la práctica solo se necesitó los próximos 5 bits puesto que su información es de la suficiente profundidad necesaria para la investigación.

Fue necesario, entonces, analizar el comportamiento del Kinect en cada uno de aquellos rangos. La Figura 30-2, muestra resultados aproximados sobre pruebas de distancia.

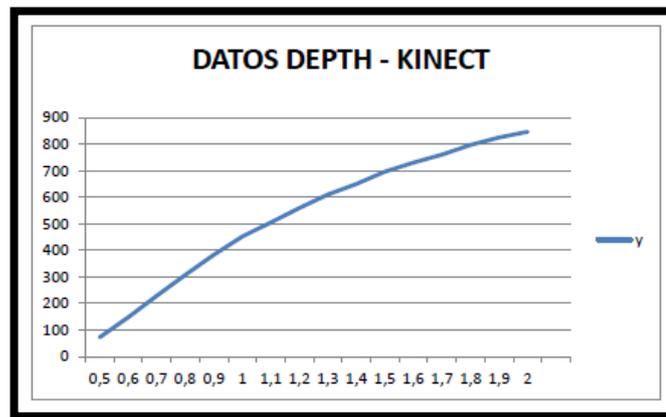
4	3	2	1	0
RANGO DE BITS		DISTANCIAS APROXIMADAS(cm)		
0 [0 - 256]		[ 0 - 74 ]cm		
1 [256 - 512]		[ 74 - 110] cm		
2 [512 - 768]		[110 - 172]cm		
3 [768 - 1024]		[172 - 328]cm		
4 [1024 - ?]		-		

**Figura 30-2:** Rango de bits – profundidad del Kinect.

**Realizado por:** Logroño C. y Mishqui K., 2016.

#### 2.5.4.1 Fórmula matemática de transformación.

Se debió tener en cuenta que la información de profundidad tiene una variación más notoria en distancias cortas mientras que en distancias largas esta diferencia se torna lenta. Por lo que describe una curva no lineal (ver Figura 31-2) a la que un simple análisis no se puede englobar en una sola fórmula.



**Figura 31-2:** Datos binarios puros DEPTH del Kinect.

**Realizado por:** Logroño C. y Mishqui K., 2016.

La solución menos costosa y sencilla, ha sido generar una fórmula matemática para cada uno de los grupos de bits separados en la tabla anterior. Se usó las fórmulas de ecuación de línea dado que cada grupo de bits, muestra cierta linealidad. Los métodos numéricos no fueron ocupados pues su consecución ha presentado más errores que aciertos. El dato de profundidad será conocido como *pval*.

##### 2.5.4.1.1 Caso 0.

Es cuando el objeto está más cercano al origen de visión. Por supuesto esto también tiene un rango inicial puesto que se ha preferido comenzar desde cierta distancia a contar el muestreo. Su fórmula es la siguiente:

$$\text{Cero} = (0,130434 * pval) + 40,6086.$$

Donde:

Cero = valor del eje Z del rango (0-256)

Pendiente = 0,130434

pval= número de bits dentro del rango ( 0-256)

Distancia limite = 40,6086

#### 2.5.4.1.2 Caso 1.

Es el rango continuo. Su fórmula es la siguiente:

$$\text{Cero1} = (0,140625 * pval) + 38.$$

Donde:

Cero1 = valor del eje Z del rango (256 -512)

Pendiente = 0,140625

pval= número de bits dentro del rango ( 256-512)

Distancia limite = 38

#### 2.5.4.1.3 Caso 2.

Las distancias en este rango son más cortas entre el principio y el fin. Su fórmula es la siguiente:

$$\text{Cero2} = (0,242187 * pval) - 14.$$

Donde:

Cero2 = valor del eje Z del rango (512-768)

Pendiente = 0,242187

pval= número de bits dentro del rango ( 512-768)

Distancia limite = 14

#### 2.5.4.1.4 Caso 3.

Es el último rango del que se ha realizado fórmula ya que su visión excede los tres metros de profundidad, superando nuestro espacio de trabajo escogido. Su fórmula es la siguiente:

$$\text{Cero3} = (0,531120332 * pval) - 235,9.$$

Donde:

Cero3 = valor del eje Z del rango (768 -1024)

Pendiente = 0,531120332

pval= número de bits dentro del rango (768-1024)

Distancia limite = 235,9

#### 2.5.4.1.5 Caso 4.

Es un caso *default* si el rango es irreconocible, pinta el pixel de color negro.

### 2.5.5 **Desarrollo del algoritmo.**

Para la programación del scáner se utilizó el editor de texto vim, el cual nos permite realizar todo el código mediante la consola de Ubuntu con OpenCV (Marín, 2015).

#### 2.5.5.1 *Obtención tazas modelo.*

Lo primero que hacemos es obtener las imágenes de ambos Kinect, las mismas que serán nuestras imágenes modelo de las tazas. Estas imágenes una vez capturadas son almacenadas y posteriormente leídas para su comparación.

### 2.5.5.2 *Procesado de las imágenes.*

Una vez obtenidas las imágenes están tienen que ser sometidas a un proceso para mejorar la imagen. Para que funcione este proceso las tazas tienen que ser de un color determinado ya que necesitamos ubicar la taza en el entorno. Para este caso se eligió las tazas de color rojo.

### 2.5.5.3 *Algoritmo de Umbralización.*

En esta sección se detallará los puntos importantes en la implementación del presente algoritmo.

#### 2.5.5.3.1 *Proceso de implementación.*

- **Apertura de la cámara.-** Es el mismo proceso de llamado en todos los algoritmos. Se realiza haciendo el llamado a las clases que apertura el video y la imagen Depth. El siguiente código permite abrir las cámaras y hacerlas funcionar, pero para tener el resultado de lo que esas cámaras proyectan se necesita llamar esas respuestas en imprimir en ventanas.

```
device.startVideo();
```

```
device.startDepth();
```

- **Invocación de la imagen RGB y Depth.-** Se llama a las dos Fuentes de información y se alista una segunda imagen para los resultados (ver Figura 32-2).

```
void VideoCallback(void* _rgb, uint32_t timestamp) {  
    m_rgb_mutex.lock();  
    //Mutex::ScopedLock lock(m_rgb_mutex);  
    uint8_t* rgb = static_cast<uint8_t*>(_rgb);  
  
void DepthCallback(void* _depth, uint32_t timestamp) {  
    m_depth_mutex.lock();
```

**Figura 32-2:** Llamada a los datos de los dos videos.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- **Parámetros.-** Se crean y establecen parámetros movibles mediante el uso de Trackbars (ver Figura 33-2). La ventana puede variar los número dentro de cada rango en el HSV, aunque en la primera expresión el rango ya está definido para seguir un solo color, el cual solo puede ser variado así mismo.



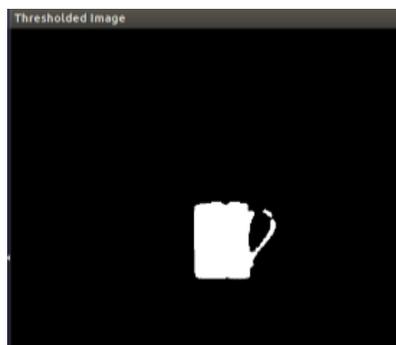
**Figura 33-2:** Trackbars de los parámetros.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- **Conversión.-** Parte importante en donde se convierte la imagen RGB a HSV (ver Figura 39-2) y dados los parámetros anteriores se procede a obtener una nueva imagen.

**cvtColor(rgbMat,imgHSV,COLOR\_BGR2HSV);**

- **Umbralización y transformaciones Morfológicas.-** Mediante código se procede a realizar la Umbralización bajo los parámetros de los Trackbars propuestos (ver Figura 34-2). Las transformaciones morfológicas se realizan para eliminar todos los ruidos que podría aún tener la imagen anterior.



**Figura 34-2:** Imagen Umbralizada.

**Realizado por:** Logroño C. y Mishqui K., 2016.

- **Cálculos de Posición X y Y.**

```
Moments oMoments = moments (imgThresholded);
double dM01 = oMoments.m01;
double dM10 = oMoments.m10;
double dArea = oMoments.m00;
posX = dM10/dArea;
posY = dM01/dArea;
```

**Figura 35-2:** Cálculo de la posición del objeto.

Realizado por: Logroño C. y Mishqui K., 2016.

- **Cálculos de posición z.** El siguiente código (ver Figura 36-2) expresa la condición básica para obtener la información del Depth necesaria. Si el programa envía los datos del Depth de cada píxel se tomaría un tiempo exagerado y el gasto computacional subiría en demasía. Por lo que se genera la condición mostrada en el código, para solo pedir el dato de la coordenada (X, Y). Entonces solo responde con pocos datos y el programa puede controlarse sin problemas.

```
if((j>posY && j<posY+12)&&
(k>posX && k<posX+12))
```

**Figura 36-2:** Código Condición de muestreo en el DEPTH.

Realizado por: Logroño C. y Mishqui K., 2016.

- **Áreas de las tazas**

Obtenemos las áreas (ver Figura 37-2) de ambas imágenes modelo, este valor nos servirá para realizar la comparación entre las áreas y su distancia en z.

```
Moments oMoments5 = moments (imgThresholded);
double dM05 = oMoments.m01;
double dM15 = oMoments.m10;
double dArea5 = oMoments.m00;
amodelo1 = dArea5/20000;
```

**Figura 37-2:** Cálculo áreas de las tazas.

Realizado por: Logroño C. y Mishqui K., 2016.

- *Comparación y clasificación.*

Las imágenes obtenidas, almacenadas y procesadas de las tazas modelo las comparamos con las imágenes de las tazas en tiempo real. Para esto tenemos que sacar el valor absoluto de las áreas obtenidas previa y actualmente, y decidir si la taza está o no en mal estado (ver Figura 38-2).

También se toma en cuenta la distancia obtenida en z para poder iniciar con el proceso de escaneo. Ya que al momento de adquirir los datos de las tazas modelo también obtenemos su distancia. Y se establece un margen de error para las áreas y distancia z.

```
int areat = abs(amodelo1-aprueba1);
int areat2 = abs(amodelo2-aprueba2);
int dist = abs(distancia-dis2);

//margen de error
if ((areat<=2) && (areat2<=2)&&(dist<2))
```

**Figura 38-2:** Cálculo del valor absoluto y margen de error.

**Realizado por:** Logroño C. y Mishqui K., 2016.

## 2.6 *Interfaz de información del estado de los objetos.*

La interfaz de usuario es normalmente el medio mediante el cual un usuario puede comunicarse con un equipo, máquina, pc. Mediante ésta se puede realizar el control y monitorización de los diferentes equipos conectados.

Normalmente estas interfaces suelen ser amigables y sencillas de entender por parte del usuario.

En el presente trabajo se diseñó una interfaz amigable y funcional para el usuario final. La interfaz está desarrollada en la consola de Ubuntu, en la cual nos muestra las áreas de las tazas a ser comparadas, denominadas modelo 1 y modelo 2 (ver Figura 39-2) las mismas que son extraídas por el Kinect 1 y Kinect 2 respectivamente.

```
modelo1 : 89 modelo2 : 100
```

**Figura 39-2:** Interfaz áreas de las tazas modelos.

**Realizado por:** Logroño C. y Mishqui K., 2016.

Después de haber obtenido los modelos de las tazas el algoritmo desarrollado procede a realizar la comparación, si es similar la taza escaneada la pantalla nos muestra un mensaje de **\*\*SIMILAR\*\*** seguido del número 1 y si la taza tiene fallas nos presenta **\*\*DISTINTA\*\*** seguido del número 0 (ver Figura 40-2). Además de la diferencia de áreas en cuanto a las tazas.

```

*****SIMILAR*****, 1
dif1:1          dif2:1

```

**Figura 40-2:** Interfaz tazas similares o distintas.

**Realizado por:** Logroño C. y Mishqui K., 2016.

También en la pantalla se muestra el número de tazas escaneadas, el número de tazas similares y distintas, realizando un conteo de las mismas (ver Figura 41-2). Además la interfaz presenta la posición de la taza en coordenadas de z.

```

*****SIMILAR*****, 1
dif1:1          dif2:1
TAZAS PROBADAS:1      TAZAS SIMILARES:1      TAZAS DISTINTAS:0
z-> 49.35

```

**Figura 41-2:** Interfaz número de tazas probadas.

**Realizado por:** Logroño C. y Mishqui K., 2016.

La interfaz desarrollada es casi completamente de visualización (ver Figura 42-2) salvo un detalle, existe una posibilidad de manipulación ya que cuando se inicia el proceso procedemos a tomar un modelo de taza con lo que pulsamos una letra, la cual no se mostrará en pantalla pero internamente es aceptada por el Sistema para iniciar.

```

modelo1 : 89          modelo2 : 100
z-> 49.48
*****SIMILAR*****, 1
dif1:1          dif2:1
TAZAS PROBADAS:1      TAZAS SIMILARES:1      TAZAS DISTINTAS:0
z-> 49.35

```

**Figura 42-2:** Interfaz general.

**Realizado por:** Logroño C. y Mishqui K., 2016.

## CAPÍTULO III

### **3 MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN.**

En el presente capítulo mostraremos los resultados obtenidos durante el desarrollo del presente trabajo, además de su análisis.

Partimos de la calidad de productos que es el grado de aceptación o satisfacción que proporciona un producto o servicio a partir de un patrón específico. Esto ha permitido ingeniar un proceso tecnológico que ayude al mejoramiento de sus productos y más aún minimizar el tiempo de trabajo.

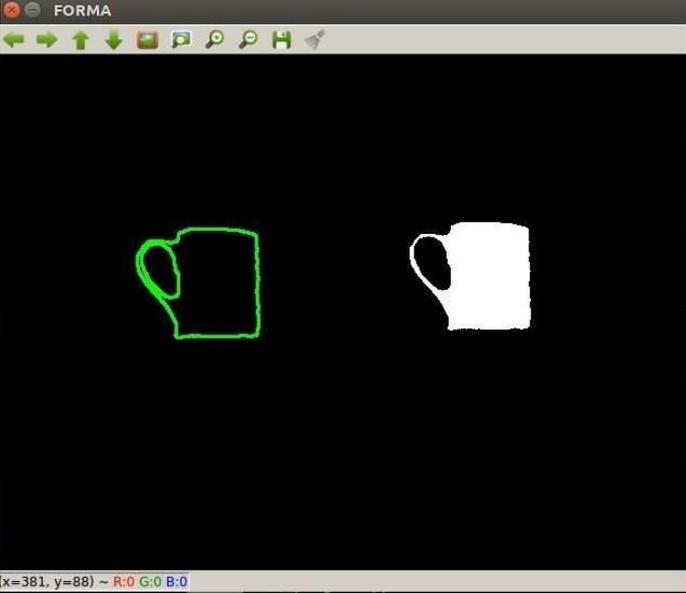
No obstante dejar pasar las características necesarias que debe tener la taza cerámica: área, color, forma. De ahí que su control de calidad varía en las empresas, debido a su producto fabricado.

Mediante el prototipo tecnológico se aspira motivar a las empresas hacer uso de la Visión Artificial en sus procesos de fabricación.

#### **3.1 Control de los productos terminados.**

A continuación se presenta las características específicas que debe tener una taza en buen estado, cabe recalcar que existen más factores que determinan su calidad como es el peso, material de fabricación, etc. En este trabajo se hace hincapié a detalles visibles y tratables por la visión artificial.

**TABLA 1-3: Control de la fase de finalización del Producto**

<p>Esmaltado</p>	<p>Verificación de la correcta decoración del producto.</p>  <p><b>Figura 1-3: Esmaltado Taza Modelo.</b></p> <p>Realizado por: Logroño C. y Mishqui K., 2016.</p>
<p>Forma y moldeado</p>	<p>Observación de las deformaciones y despostillados.</p>  <p><b>Figura 2-3: Forma y Modelado de Taza Modelo.</b></p> <p>Realizado por: Logroño C. y Mishqui K., 2016.</p>
<p>Dimensiones</p>	<p>Verificación de superficie acorde al modelo de fabricación.</p>  <p><b>Figura 3-3: Área de Taza Modelo.</b></p> <p>Realizado por: Logroño C. y Mishqui K., 2016.</p>

Realizado por: Logroño C. y Mishqui K., 2016.

Es necesario indicar las falencias más comunes dentro de una línea de fabricación, ya sea para piezas robóticas, superficies metálicas, plásticas, etc. o las mismas tazas cerámicas.

Para hacer una selección juiciosa del objeto, es útil entender la terminología de las propiedades necesarias, para obtener el dato específico que servirá para la posterior decisión.

La fase de finalización del producto se ha generalizado en los siguientes puntos:

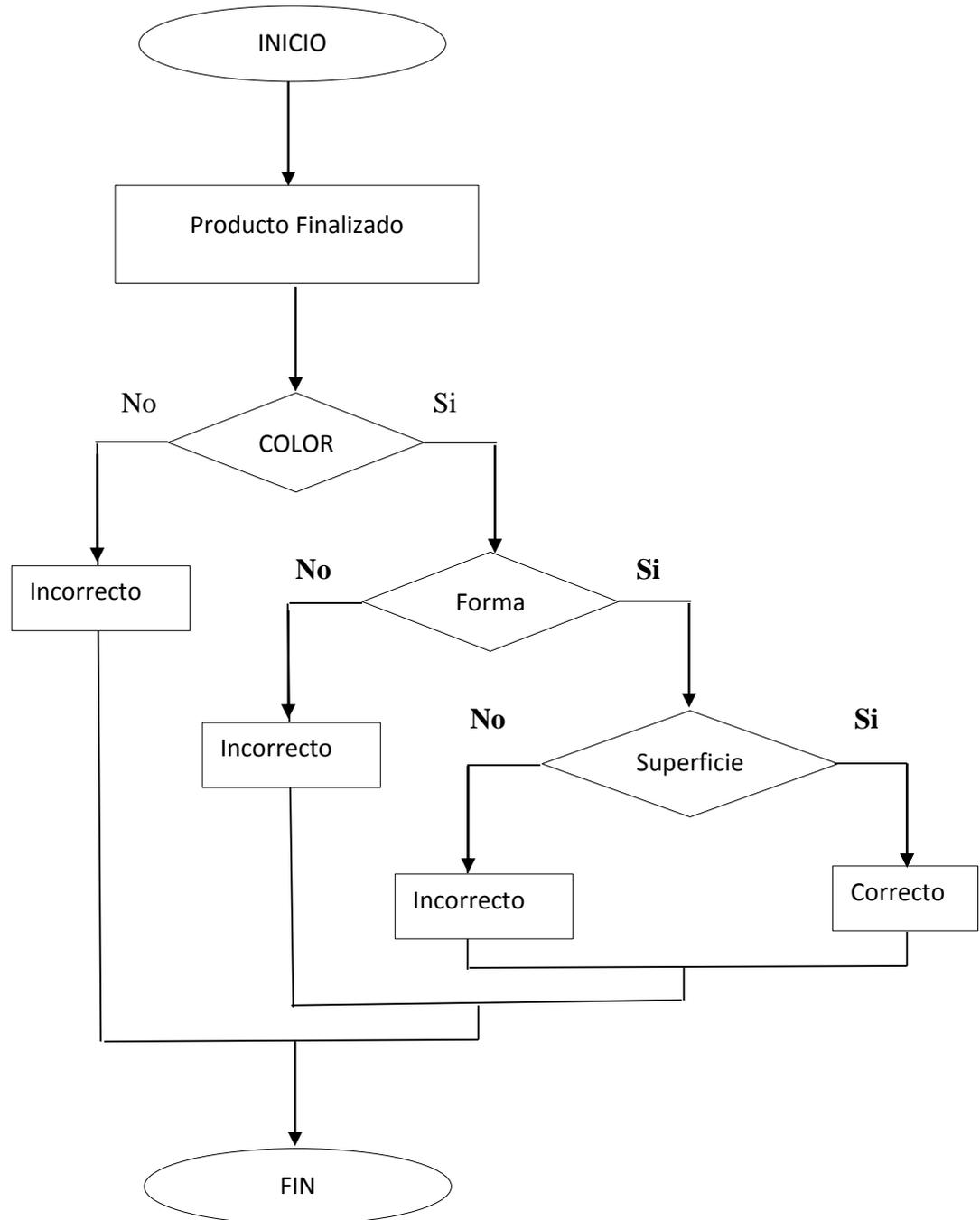
**ESMALTADO:** Color deseable para dar una mejor presentación de la taza, lo cual nos permite ajustar a diferentes colores dependiendo de la demanda.

**FORMA Y MODELADO:** Permitirá divisar la forma correcta que debe tener, de ser así sigue su proceso de empaquetado caso contrario será desechado, una de los errores en esta fase podría ser el desgaste de los moldes o la brusca manipulación que ocasione fisuras en el producto.

**DIMENSIONES:** Viene a la par con la forma y modelado, pero su análisis es mucho más profundo, puesto que se representará en números la superficie existente y mediante algoritmos matemáticos se analizará inmediatamente.

### 3.2 Diagrama de selección del producto

Acorde a sus características de análisis, se presenta la lógica utilizada en las tazas terminadas (ver Figura 4-3).

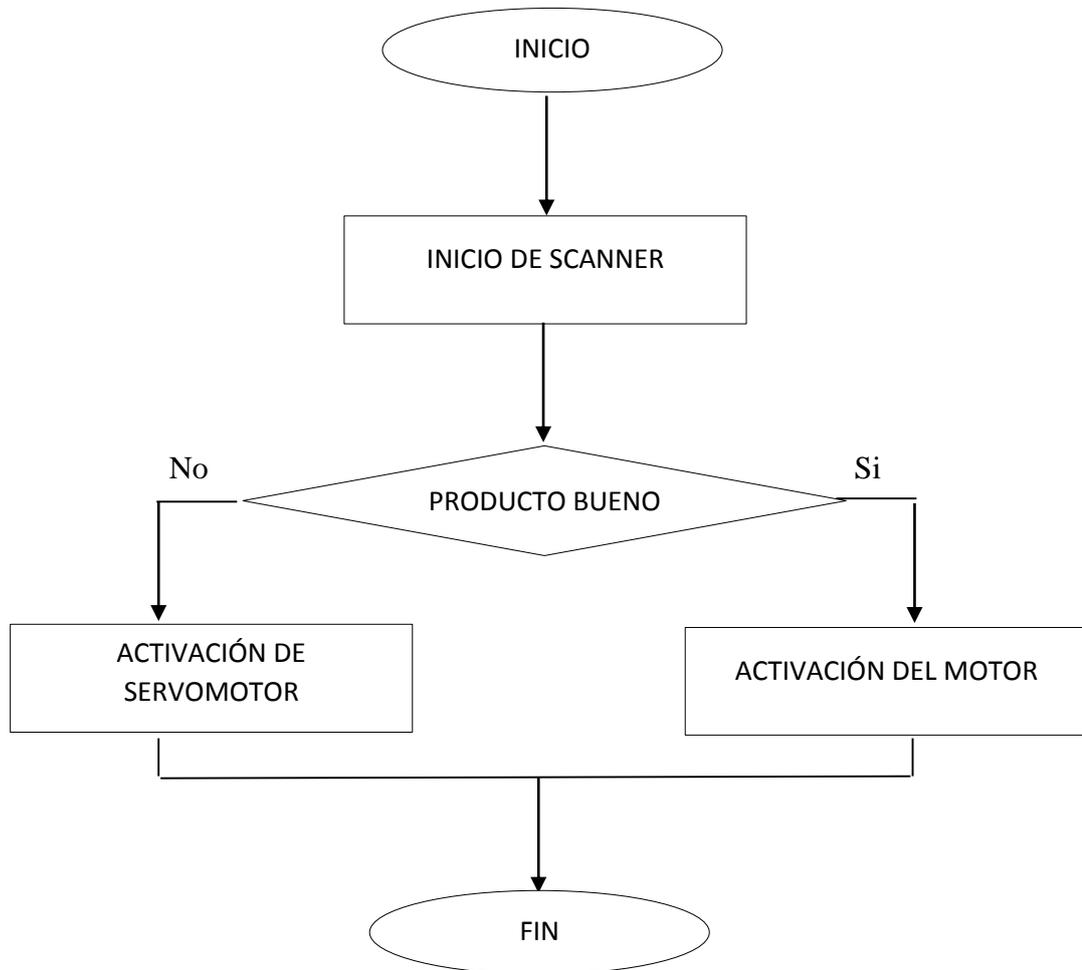


**Figura 4-3:** Diagrama de funcionamiento selección del producto.

Realizado por: Logroño C. y Mishqui K., 2016.

### 3.3 Operación de actuadores.

Dentro de este grupo se hace mención a: relés, motor 24 V DC, servomotor 6 V DC, 20 mA, su funcionamiento se basa acorde a la respuesta del scanner (ver Figura 5-3).



**Figura 5-3:** Diagrama operación del sistema de escaneo.

Realizado por: Logroño C. y Mishqui K., 2016.

A través de la Figura 5-3 se muestra el funcionamiento del scáner: si el producto es de buena calidad, la taza cerámica seguirá en el proceso mediante la activación del motor 24 dc, caso contrario será desechado mediante la activación del servo motor que basará su función empujando el objeto y desechándolo.

### 3.4 Resultados del scáner en dimensiones.

Para el cumplimiento de este objetivo que permite determinar la fiabilidad del scáner y el sistema automático, el scáner 3D recepta 100 escenas de un objeto, expresadas en cm<sup>2</sup>. Por lo cual necesitamos analizar la variación de estos números.

Entonces se toma el 10% de los datos totales del scáner debido a su repetitividad y suficientes para entrar en estudio. Cuyos números corresponden a una taza modelo de buena calidad, al igual necesitamos saber la operación con una taza en mal estado.

Para finalmente realizar una comparación y determinar el comportamiento de estos datos mediante una gráfica en Excel.

El scanner está compuesto por 2 sensores Kinect que capturan cada lado de la taza cerámica, por lo tanto se tiene 2 áreas distintas.

**TABLA 2-3:** Valores para modelo 1.

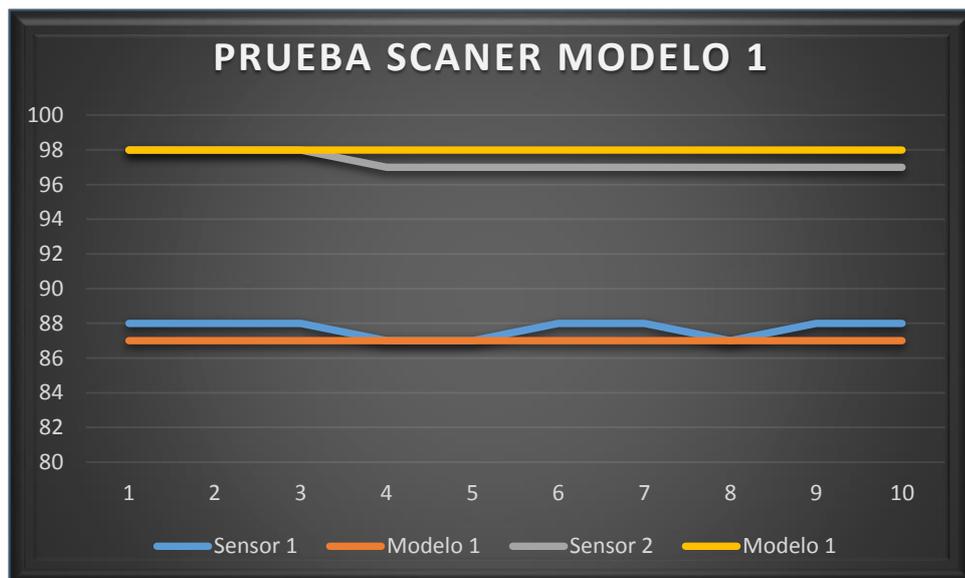
N° Pruebas	Sensor 1	Modelo 1	Sensor 2	Modelo 1
1	88	87	98	98
2	88	87	98	98
3	88	87	98	98
4	87	87	97	98
5	87	87	97	98
6	88	87	97	98
7	88	87	97	98
8	87	87	97	98
9	88	87	97	98
10	88	87	97	98

**Realizado por:** Logroño C. y Mishqui K., 2016.

**TABLA 3-3:** Valores para modelo 2.

N° Pruebas	Sensor 1	Modelo 2	Sensor 2	Modelo 2
1	82	82	94	94
2	82	82	94	94
3	82	82	94	94
4	82	82	94	94
5	82	82	94	94
6	81	82	93	94
7	82	82	94	94
8	82	82	94	94
9	82	82	94	94
10	82	82	94	94

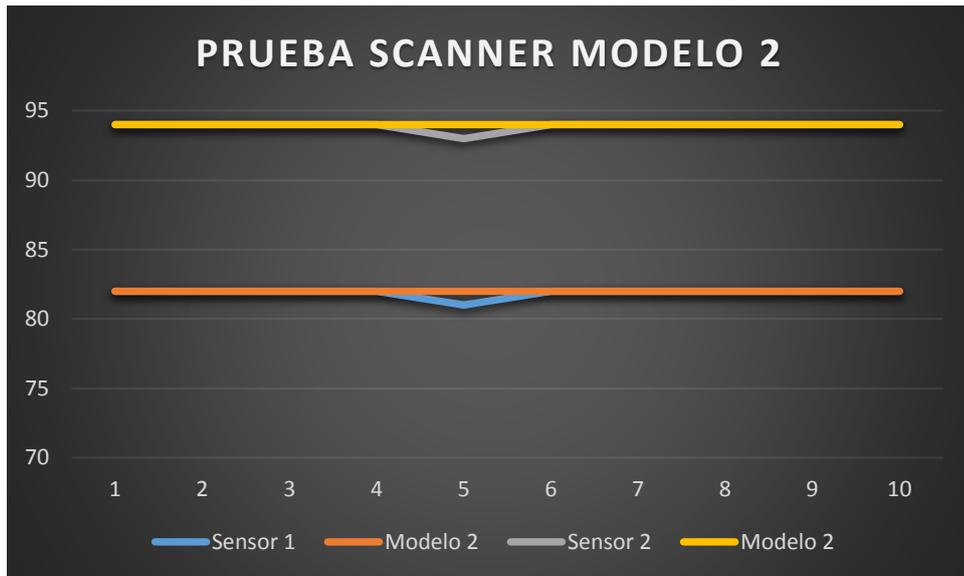
Realizado por: Logroño C. y Mishqui K., 2016.



**Figura 6-3:** Resultados prueba scanner modelo 1.

Realizado por: Logroño C. y Mishqui K., 2016.

- Mediante el gráfico de la Figura 6-3, se puede deducir que no existe mayor a 1 cm<sup>2</sup> en variación entre valores del producto final y taza modelo
- El sensado 1 en la prueba 6 sus áreas difieren en una unidad, para los siguientes casos su dimensión se mantiene
- El sensado 2 mantiene similitud, hasta la prueba 5 del producto final.



**Figura 7-3:** Resultados prueba scanner modelo 2.

**Realizado por:** Logroño C. y Mishqui K., 2016.

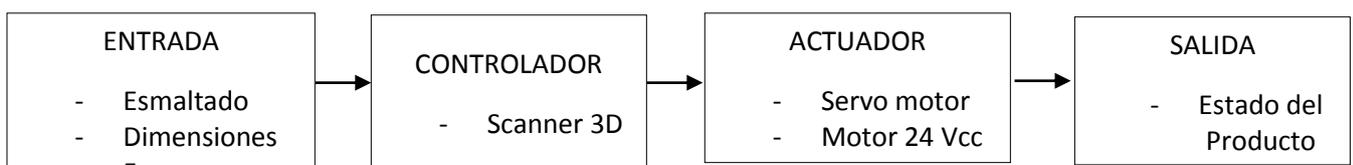
Mediante este modelo 2 se puede apreciar que (ver Figura 7-3):

- El sensado 1 y 2 actúan como en la anterior taza modelo.
- El sensado 1 su área es igual al área modelo hasta la prueba 6, es decir tuvo una sola variación durante su proceso de verificación de tazas.
- El sensado 2 sigue manteniendo su diferencia de  $1 \text{ cm}^2$ , esto puede ser por las variaciones de iluminación dentro del scanner

Error del sistema scanner 3d:

- Mediante el análisis de estos valores se establece una exactitud de  $\pm 1,13$ .

### 3.5 SISTEMA DE CONTROL DE LAZO ABIERTO.

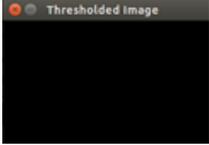


**Figura 8-3:** Representación sistema de control.

**Realizado por:** Logroño C. y Mishqui K., 2016.

**TABLA 4-3:** Análisis de datos de las tazas de prueba.

TAZAS DE PRUEBA	CARACTERISTICAS
 <p><b>Figura 9-3:</b> Taza Modelo. Realizado por: Logroño C. y Mishqui K., 2016.</p>	<p>-Esmaltado: Rojo -Forma:</p>  <p><b>Figura 10-3:</b> Forma Taza Modelo. Realizado por: Logroño C. y Mishqui K., 2016. -Área 1: 87 - Área 2: 98</p>
 <p><b>Figura 11-3:</b> Taza mal estado. Realizado por: Logroño C. y Mishqui K., 2016.</p>	<p>-Esmaltado: Rojo -Forma:</p>  <p><b>Figura 12-3:</b> Forma taza sin oreja. Realizado por: Logroño C. y Mishqui K., 2016. -Área 1: 81 -Área 2: 94</p>
 <p><b>Figura 13-3:</b> Taza fisurada. Realizado por: Logroño C. y Mishqui K., 2016.</p>	<p>-Esmaltado: Rojo y Blanco -Forma:</p>  <p><b>Figura 14-3:</b> Forma taza fisurada. Realizado por: Logroño C. y Mishqui K., 2016. -Área 1: 71 -Área 2: 94</p>

 <p><b>Figura 15-3:</b> Taza con Fisura superior.</p> <p><b>Realizado por:</b> Logroño C. y Mishqui K., 2016.</p>	<p>-Esmaltado: Rojo y Blanco</p> <p>-Forma:</p>  <p><b>Figura 16-3:</b> Modelo con Fisura superior.</p> <p><b>Realizado por:</b> Logroño C. y Mishqui K., 2016.</p> <p>-Área 1: 87</p> <p>-Área 2: 90</p>
 <p><b>Figura 17-3:</b> palet sin objeto.</p> <p><b>Realizado por:</b> Logroño C. y Mishqui K., 2016.</p>	<p>-Esmaltado: ninguno</p> <p>-Forma: ninguna</p>  <p><b>Figura 18-3:</b> sin forma.</p> <p><b>Realizado por:</b> Logroño C. y Mishqui K., 2016.</p> <p>-Área 1: 0</p> <p>-Área 2: 1</p>

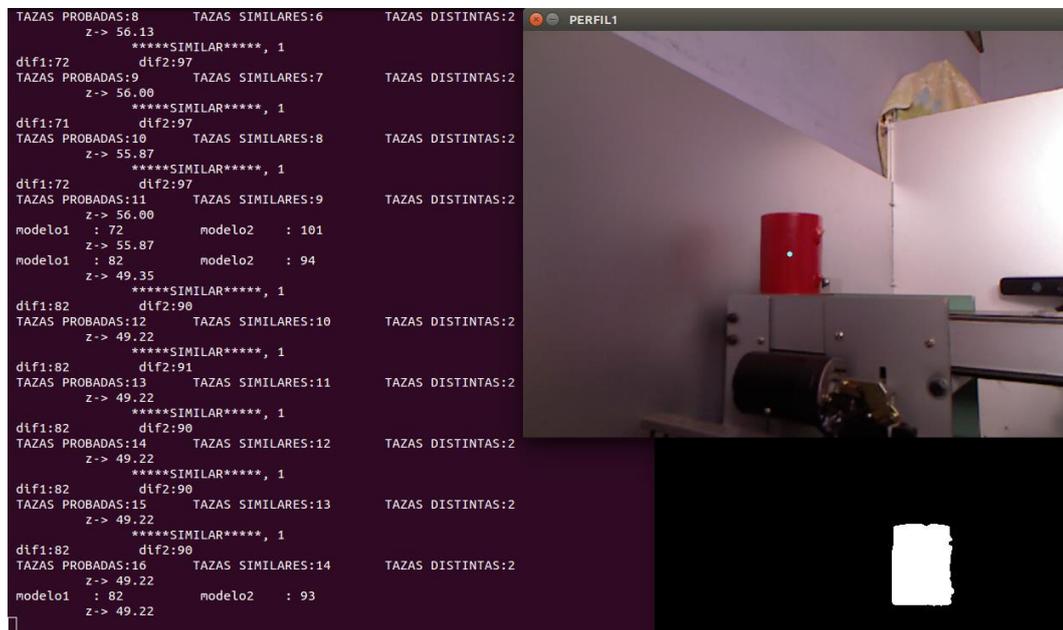
**Realizado por:** Logroño C. y Mishqui K., 2016.

continúa.

Mediante la Tabla 4-3, se representan los datos que reflejan sus características de producto bueno o malo, cabe indicar que siempre se debe partir de un objeto modelo para realizar las respectivas comparaciones, los datos que se reciben son, color de esmaltado, área en cm cuadrados, y su forma o moldeado.

El nivel de iluminación hace que su censado sea diferente, es por eso que al iniciar el proceso es necesario realizar una calibración de los sensores receptores esto se lo puede realizar en la programación mediante filtros de thresholding disminución de ruido y aumento de contornos.

En la Figura 18-3, del Thresholded image se puede ver que toma lecturas de área 1 = 0 y área 2= 1 se establece que ya está calibrado y listo para empezar el censo de productos terminados.



**Figura 19-3:** Interfaz.

Realizado por: Logroño C. y Mishqui K., 2016.

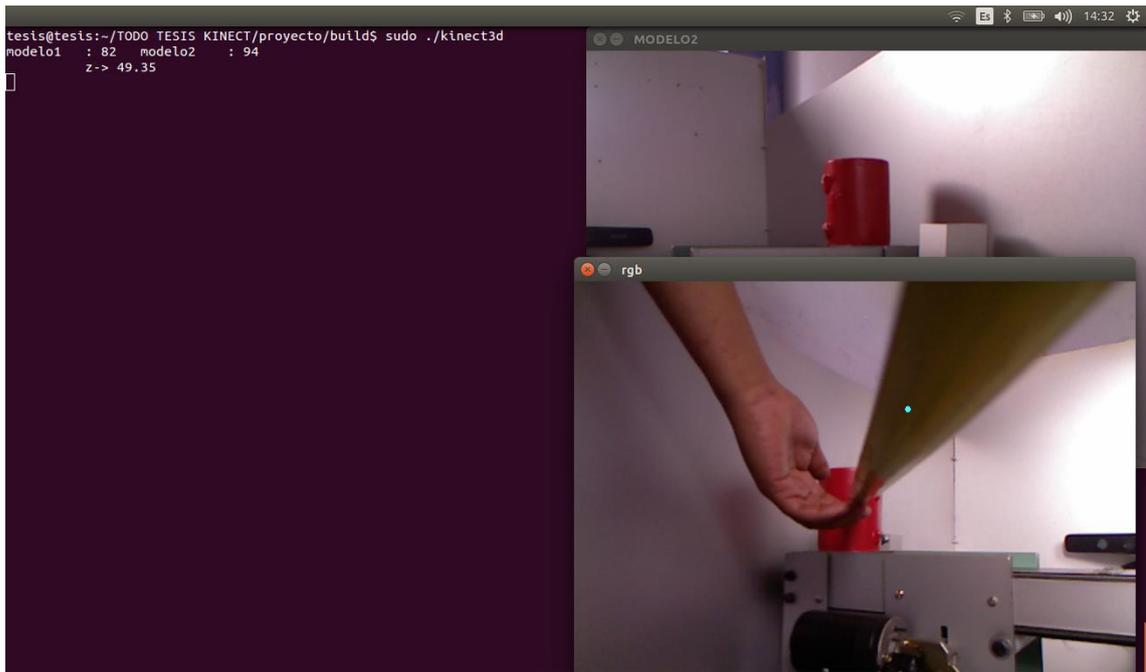
A través del sensor RGB podemos determinar imagen en las coordenadas X, Y, pero el sensor de profundidad propio del Kinect nos permite obtener el eje Z, lo que hace a este proyecto trabajar tridimensionalmente.

El eje Z nos sirve para determinar la localización del objeto en la banda transportadora y su correcta ubicación, de ser así continua su proceso de sensado.



**Figura 20-3:** Medición en eje z.

Realizado por: Logroño C. y Mishqui K., 2016.



**Figura 21-3:** Presentación de eje z.

**Realizado por:** Logroño C. y Mishqui K., 2016.

## CONCLUSIONES.

- Se determinó que existen diversas maneras de adquisición de escenas tridimensionales, entre ellas se utilizó el dispositivo Kinect como sensor para lectura de datos del mundo real permitiendo interpretarlo a través de modelos matemáticos por computador programados en Open Source con plataforma Linux Software Libre, librerías de C++ y LibFreenect para su correcta funcionalidad.

- La distancia mínima que adquieren datos los Kinect son de 40 cm por lo cual se ha concluido a ubicarlos de acuerdo a esta distancia con un Angulo de 45° respecto al objeto.

- La transmisión serial resulta fácil de implementar, además de que es segura y de bajo coste. En el caso de un microcontrolador como interfaz hardware de intercambio con cualquier sistema de cómputo, es viable considerar las propuestas realizadas en este trabajo, ya que se adaptan a cualquier aplicación similar, debido a que la unidad básica de comunicación serie es la misma en todo momento.

- Los microcontroladores actuales, por lo general, cuenta con un módulo funcional de comunicación serie USART/SCI (Universal Synchronous Asynchronous Receiver Transmitter/Serial Communication Interface) o en otro caso éste puede ser implementado en la mayoría de sus funciones mediante software. Este módulo es ampliamente utilizado en sistemas de cómputo en general, el medio de transmisión para la interfaz USART requiere cuando menos de dos vías por lo que la complejidad, costos y el uso de pines del microcontrolador se reducen; esta interfaz tiene dos modos de comunicación: asíncrona o síncrona. La utilizada en este Proyecto es full duplex con una vía RX (Recepción) y TX (Transmisión).

- Las bandas transportadoras se distinguen por su proceso de fabricación, los sistemas de transporte continuo son los más utilizados en la industria, varían su diseño de acuerdo a su línea de fabricación, para este proyecto se ha diseñado una banda horizontal accionada por un motor DC, con recubrimiento de caucho para dar un mejor acoplamiento al objeto. El mismo fue diseñado tomando en cuenta características exactas de los productos que se van a transportar, incorporando un sistema de rampa con un servomotor para el desecho de los productos de mala calidad.

- Para el reconocimiento de patrones del producto final, se utilizó el sensor infrarrojo; para la localización en el eje Z del producto, el sensor RGB; mediante el código HSV se obtuvo el color del esmaltado de la taza. Mediante estas prestaciones del Kinect se desarrolló el algoritmo para la detección de errores en las tazas.

- Una interfaz amigable hacia el operador es de mucha importancia, ya que permite tomar decisiones a futuro, A través de la consola de Linux el Sistema de control automático informa el desempeño del proceso, leyendo variables como: dimensión en X, Y, Z, Superficie de la taza, un registro de cantidad de tazas producidas en buen y mal estado, y el análisis con respecto al producto modelo en cuestión.

- Este Sistema de control automático de lazo abierto, permite la verificación de los productos finales de una línea de producción, aceptando los objetos en buen estado de acuerdo a las variables de análisis (esmaltado, dimensión, forma), y desechando los productos que contienen daños. Su operación tiene resultados satisfactorios, obteniendo dimensiones con un margen de error de  $\pm 1.13\%$ .

## **RECOMENDACIONES.**

- Implementar las escenas tridimensionales para los diferentes procesos de fabricación en una industria, ya que permite obtener múltiples datos para analizarlos a través de modelos matemáticos.
- Para los procesos en una industria se recomienda utilizar una arquitectura cubierta que no permita el acceso de iluminación, ya que afecta a la recepción de datos del scanner 3D.
- Verificar la comunicación serial TX, RX, debido a las ocasiones de desconfiguración y cambios de puertos, establecer sincronismo en el algoritmo de visión artificial y del microcontrolador.
- Dentro del ámbito industrial se recomienda utilizar bandas transportadoras referentes al proceso, ya que estos pueden variar en dimensiones y fuerza debido a que se puede tener productos pesados y trayectos más largos.
- La visión artificial y su reconocimiento de patrones implementado en un Scanner 3D se puede utilizar para múltiples aplicaciones y más aún se pretende implantar para el uso del control de calidad recogiendo datos en X, Y, Z y los compara, esto es de mucha importancia.
- Verificar los resultados desplegados en la interfaz, estén acorde a los resultados obtenidos por computador, se recomienda ajustar los receptores para que los ejes X, Y, Z sean más exactos.
- Realizar todas las pruebas necesarias, que permitan verificar el buen funcionamiento del sistema, mediante la prueba de uno o varios objetos dentro del trabajo del sistema configurado.

## **BIBLIOGRAFÍA.**

Bradsky, G., Kaebler, A. (2008). *Learning OPenCV*, O'Reilly Media, Inc. USA. 1<sup>st</sup> Ed. pp. 25-50.

Bolton, W. (2010). *Mecatrónica, Sistemas de Control Electrónico en Ingeniería Mecánica y Eléctrica*. 2<sup>a</sup> ed. Bogota-Colombia. Alfaomega. pp. 17-22

Borja, S. (2008). *Introducción a Ubuntu*.

Recuperado de: [www.ardinova.com/cursos/ubuntu/introduccion\\_a\\_ubuntu.pdf](http://www.ardinova.com/cursos/ubuntu/introduccion_a_ubuntu.pdf)

Cabrera, D. (2009). *Tutorial de CMake*.

Recuperado de: <http://servicio.uca.es/softwarelibre/publicaciones/cmake-material.pdf>

Calloni, J. (2003). *Mantenimiento eléctrico y mecánico para pequeñas y medianas empresa pymes*. Buenos Aires-Argentina. Nobuko, pp. 155-159

Castela, P. (2012). *El Relé*.

Recuperado de: <http://platea.pntic.mec.es/~pcastela/tecno/documentos/apuntes/rele.pdf>.

COMPUTER VISIÓN AND ROBOTIC GROUP., Operaciones Morfológicas, Universitat de Girona, pp. 41- 59.

Cruz, L., Lucio, D., & Velho, L. (2012). *Kinect and RGB images: Challenges and applications*. SIBGRAPI Tutorial.

Deitel, H., Deitel, P. (2004). *Cómo programar en C/C++ y Java*. Pearson Education, Inc., México, S.A. de C.V. 4<sup>a</sup> ed. pp. 7-10

Enríquez, R. (2009). *Guía de usuario de arduino*.

Recuperado de:.

[http://www.uco.es/aulasoftwarelibre/wpcontent/uploads/2010/05/Arduino\\_user\\_manual\\_es.pdf](http://www.uco.es/aulasoftwarelibre/wpcontent/uploads/2010/05/Arduino_user_manual_es.pdf).

Espinosa, J., Cano, J., Sepúlveda, A. (s.f.). *Comunicación serial*.

Recuperado de: <http://juandeg.tripod.com/comserial.htm>

García, E. (2011). *Aplicaciones Quirófano con Kinect*.

Recuperado de: <http://www.meristation.com/xbox-360/noticias/quiroyfanos-con-menos-infecciones-con-un-proyecto-de-kinect/1533531/1671007>

García, Eloi. (2009). *Visión Artificial*.

Recuperado de: <https://pdf.yt/d/6ofH1oITrxHb9jvQ/download>

GitHub, Inc. (2016). *OpenKinect/libfreenect*.

Recuperado de: <https://github.com/OpenKinect/libfreenect>

González, A., Martínez, F., Pernía, A., Alba F., Castejón, M., Ordieres, J., Vergara, E. (2006). Técnicas y Algoritmos Básicos de Visión Artificial. *Grupo de Investigación EDMANS*, Universidad de la Rioja, España. Pp. 13-80.

Hagebeuker, D.I.B. (2007). *A 3d time of flight camera for object detection*.

Henry, P., Krainin, M., Herbst, E., Ren, X. and Fox, D. (2010). *Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments*. In the 12th International Symposium on Experimental Robotics (ISER).

Hinojosa, H. (2002). *Software para Diseño de Transportadores de Banda*. Tesis de Grado, Ing. Mecánica, ESPOL., pp. 30-39

Recuperado de: <https://www.dspace.espol.edu.ec/handle/123456789/4477>

Huang, Y. & Chen, M. (2011). *3d object model recovery from 2d images utilizing corner detection*. In System Science and Engineering (ICSSE), 2011 International Conference on, pages 76–81. IEEE.

Hub, A., Hartter, T., & Ertl, T. (2006). *Interactive tracking of movable objects for the blind on the basis of environment models and perception-oriented object recognition methods*. In Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility, pages 11–118. ACM.

Ilvay, R. (2014). *Sistema de Educación para niño de 3-5 años, mediante un robot controlado por el sensor Kinect.*, Tesis Ing. Electrónica., Riobamba-Ecuador., Escuela Superior Politécnica de Chimborazo., pp. 17-36

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. (2011). *Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera*. In Proceedings of the 24th annual ACM symposium on User interface software and technology, pages 559–568. ACM.

Marín, C. (2015). *Introducción a OpenCV*.

Recuperado de: <http://acodigo.blogspot.com/2013/05/introduccion-opencv.html>

Montabobe, S. (2014). *Installing OpenCV 2.4.9 in Ubuntu 14.04 LTS*.

Recuperado de: <http://www.samontab.com/web/2014/06/installing-opencv-2-4-9-in-ubuntu-14-04-lts/>

Montoya, J. (2014). *Control de Calidad, Aseguramiento de Calidad* [en línea].

Recuperado de: <https://prezi.com/rzqv3gtvm8xl/control-de-calidad-aseguramiento-de-calidad-control-estadi/>

Nilsson, N. (2001). *Inteligencia Artificial. Una nueva síntesis*. 1<sup>a</sup> .ed. Madrid, España: McGraw-Hill Interamericana de España. pp. 33-62, 75-99.

Paco. (2014). *Botón pulsador con luz piloto*.

Recuperado de: <http://coparoman.blogspot.com/2014/02/boton-pulsador-con-luz-piloto.html>

Pallas, R. (2003). *Sensores y acondicionadores de señal*. 4<sup>a</sup> ed. Barcelona-España. Marcombo SA. pp. 3-8

Pheatt, C., Ballester, J., & Wilhelmi, D. (2005). *Low-cost three-dimensional scanning using range imaging*. *Journal of Computing Sciences in Colleges*, 20(4):13–19.

Prieto, A. (2013). *Desarrollo de un prototipo de un api para la interacción de un usuario con aplicaciones con contenido 3d utilizando Kinect*. Tesis Ing. Ciencias Computacionales., Guayaquil - Ecuador. pp. xii -12.

Quirate, A., Merckel G., Pedroza, B., Fuentes, F. & López, J. (2013). *Introducción a Kinect*.

Recuperado de: <http://hacedores.com/introduccion-a-kinect/>

Rodriguez, F. & Soria, M. (2004). *Control y robótica en agricultura* [en línea]. Almería-España. Universidad de Almería servicio de publicaciones. pp. 102-105 [Consulta: 15 de Enero 2016].  
Recuperado de: <https://books.google.com.ec>

Saxena, A., Sun, M. & Ng, A.Y. (2007). *3-d reconstruction from sparse views using monocular vision*. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE.

Sebastian, J. (2010). *Todo sobre el Kinect de Xbox 360 – Características y Precio* .

Recuperado de: <http://www.comusoft.com/todo-sobre-el-kinect-de-xbox-360-caracteristicas-y-precio>

Velasco, J. J. (2013). *Kinect un dispositivo más allá del ocio*.

Recuperado de: <http://blogthinkbig.com/kinect-mas-alla-del-ocio/> Velas

Vouzounaras, G., Perez-Moneo Agapito, J.D., Daras, P., & Srinivas, M.G. (2010). *3d reconstruction of indoor and outdoor building scenes from a single image*. In Proceedings of the 2010 ACM workshop on Surreal media and virtual cloning, pages 63–66. ACM.

Weiss, I. & Ray, M. (2001). *Model-based recognition of 3d objects from single images*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(2):116–128.

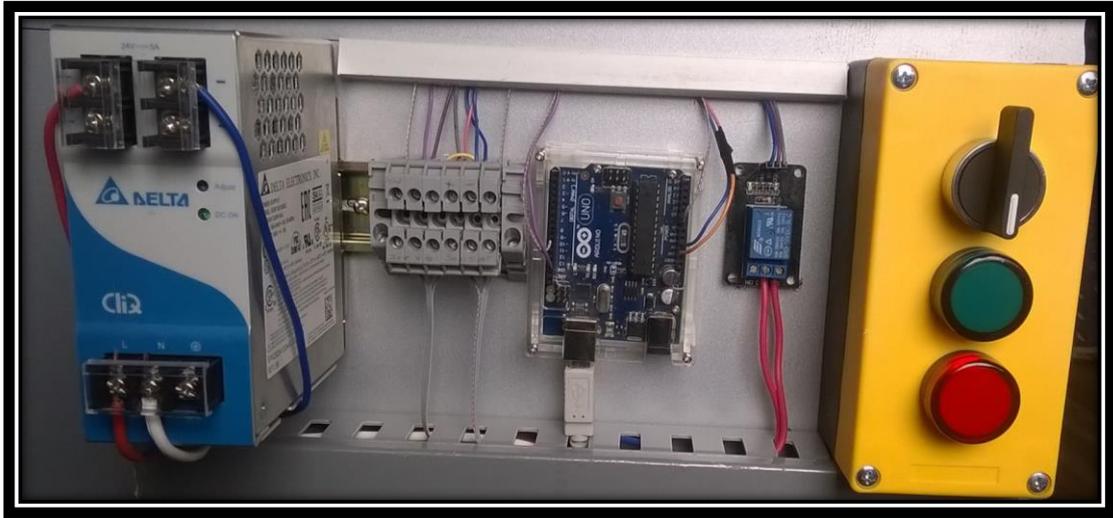
Zheng, H., Yuan, J., & Gu, R. (2011). *A novel method for 3d reconstruction on uncalibrated images*. In Proceedings of the Third International Conference on Internet Multimedia Computing and Service, ICIMCS '11, pages 138–141, New York, NY, USA. ACM.

ANEXOS.

ANEXO A. CARACTERISTICAS DE BANDAS COMERCIALES

Tipo	Cobertura superior					Impregnación Cobertura inferior			Características especiales	Temperatura en continuo (puntual) del producto transportado °C	Tejidos		Espesor banda mm	Peso banda Kg/m <sup>2</sup>	a 20°C		Ancho fabric. mm		
	Material	Dureza 5HA	Color	Esesor mm	Acabado	Material	Color	Esesor mm			Nº de telas	Trama			Ømm	Ømm		Carga trabajo al 1% alargam. N/mm	
FEBOR 15NF	PVC	82	Negro 01	0.50	Mate	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	2.10	2.60	40	60	15	3000
FEBOR 19NF	PVC	82	Negro 01	0.90	Mate	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	2.50	3.00	40	60	17	3000
FEBOR 20NF	PVC	82	Negro 01	1.00	Mate	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	3.00	3.60	60	80	17	3000
FEBOR 21NF	PVC	82	Negro 01	0.90	Mate	RC	Negro 00	0,10	W	W	- 10 (-15) + 80 (100)	2	Flexible	2.70	3.30	55	75	20	3000
FEBOR 22FF	RC	-	Negro 00	0.10	Impregn.	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	2.40	2.85	60	60	14	3000
ASTER 15QF	PVC	55	Negro 02	1.70	Grabado Q	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	3.20	3.50	50	60	15	2-3000
ASTER 15GZF	PVC	55	Negro 02	3.70	Grabado GZ	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	5.50	4.20	45	70	15	2000
ASTER 15WTF	PVC	65	Negro 02	6.00	Grabado W1	RC	Negro 00	0,10	S	W	- 10 (-15) + 80 (100)	2	Rígida	8.75	4.80	80	100	10	1250

## ANEXO B. UNIDAD DE CONTROL



## ANEXO C. RAMPA DE CLASIFICACIÓN DE TAZAS CON FALLAS.



# ANEXO D. DATASHEET SENSOR GP2Y0A21YK.

**SHARP**

## GP2Y0A21YK Optoelectronic Device

### FEATURES

- Digital Output
- LED Pulse Cycle Duration: 32 ms
- Range: 10 to 80 cm
- Typical response time: 39 ms
- Typical start up delay: 44 ms
- Average Current Consumption: 30 mA
- Detection Area Diameter @ 80 cm: 12 cm

### DESCRIPTION

The GP2Y0A21YK is a wide-angle distance measuring sensor.

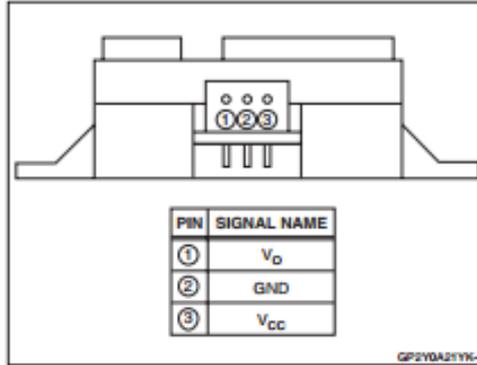


Figure 1. Pinout

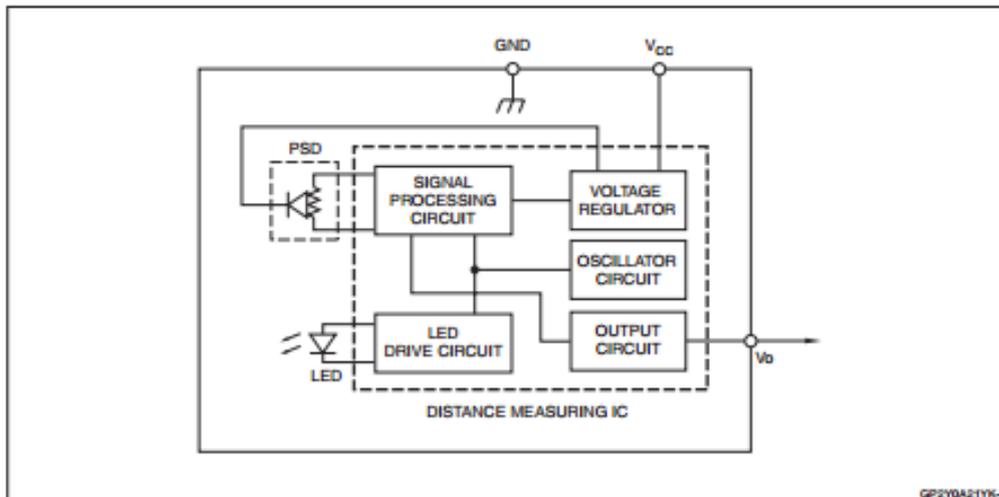


Figure 2. Block Diagram

## ELECTRICAL SPECIFICATIONS

## Absolute Maximum Ratings

$T_a = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{ VDC}$

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	$V_{CC}$	-0.3 to +7	V
Output Terminal Voltage	$V_O$	-0.3 to ( $V_{CC} + 0.3$ )	V
Operating Temperature	$T_{opr}$	-10 to +60	$^\circ\text{C}$
Storage Temperature	$T_{stg}$	-40 to +70	$^\circ\text{C}$

## Operating Supply Voltage

PARAMETER	SYMBOL	RATING	UNIT
Operating Supply Voltage	$V_{CC}$	4.5 to 5.5	V

## Electro-optical Characteristics

$T_a = 25^\circ\text{C}$ ,  $V_{CC} = 5\text{ VDC}$

PARAMETER	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT	NOTES
Measuring Distance Range	$\Delta L$		10	-	80	cm	1, 2
Output Terminal Voltage	$V_O$	$L = 80\text{ cm}$	0.25	0.4	0.55	V	1, 2
Output Voltage Difference	$\Delta V_O$	Output change at $\Delta L$ (80 cm - 10 cm)	1.65	1.9	2.15	V	1, 2
Average Supply Current	$I_{CC}$	$L = 80\text{ cm}$	-	30	40	mA	1, 2

## NOTES:

- Using reflective object: white paper (made by Kodak Co. Ltd.) gray cards R-27, white face, reflective ratio: 90%
- $L$  = Distance to reflective object.

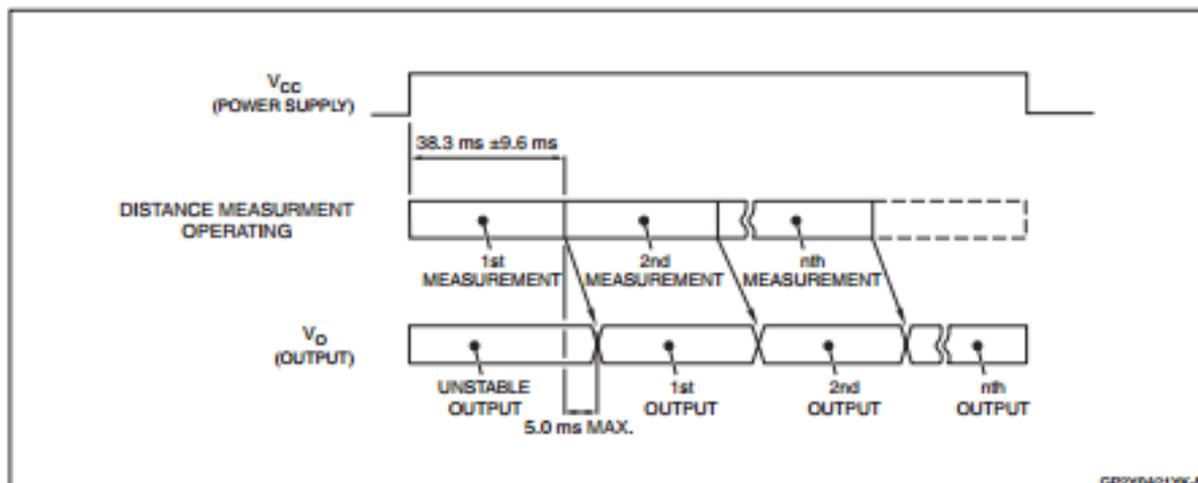


Figure 3. Timing Diagram

GP2Y0A21YK-5

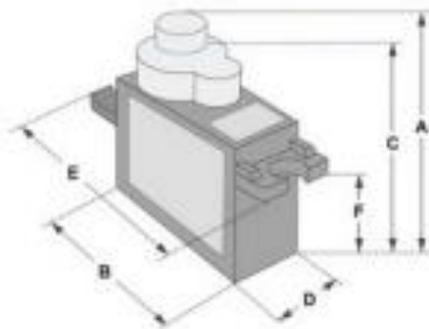
## ANEXO E. DATASHEET SERVOMOTOR SG90.

### SERVO MOTOR SG90

### DATA SHEET



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

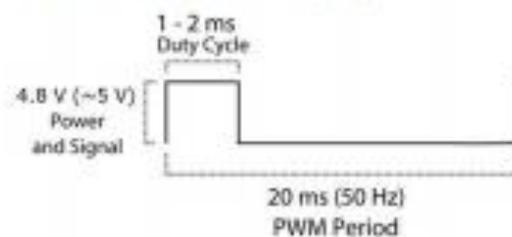


#### Dimensions & Specifications

A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Position 0° (1.5 ms pulse) is middle, 90° (~2ms pulse) is middle, is all the way to the right, 180° (~1ms pulse) is all the way to the left.

PWM=Orange (⏏)  
Vcc = Red (+)  
Ground=Brown (-)



## ANEXO F. MICROCONTROLADOR DE ARDUINO UNO R3.



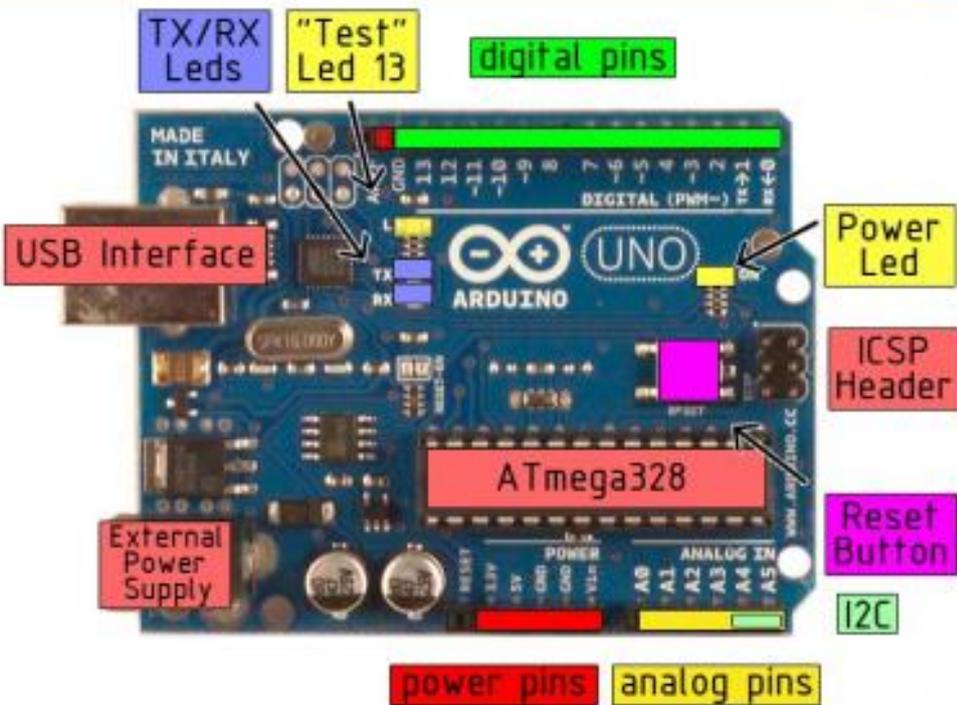
# Technical Specification

EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

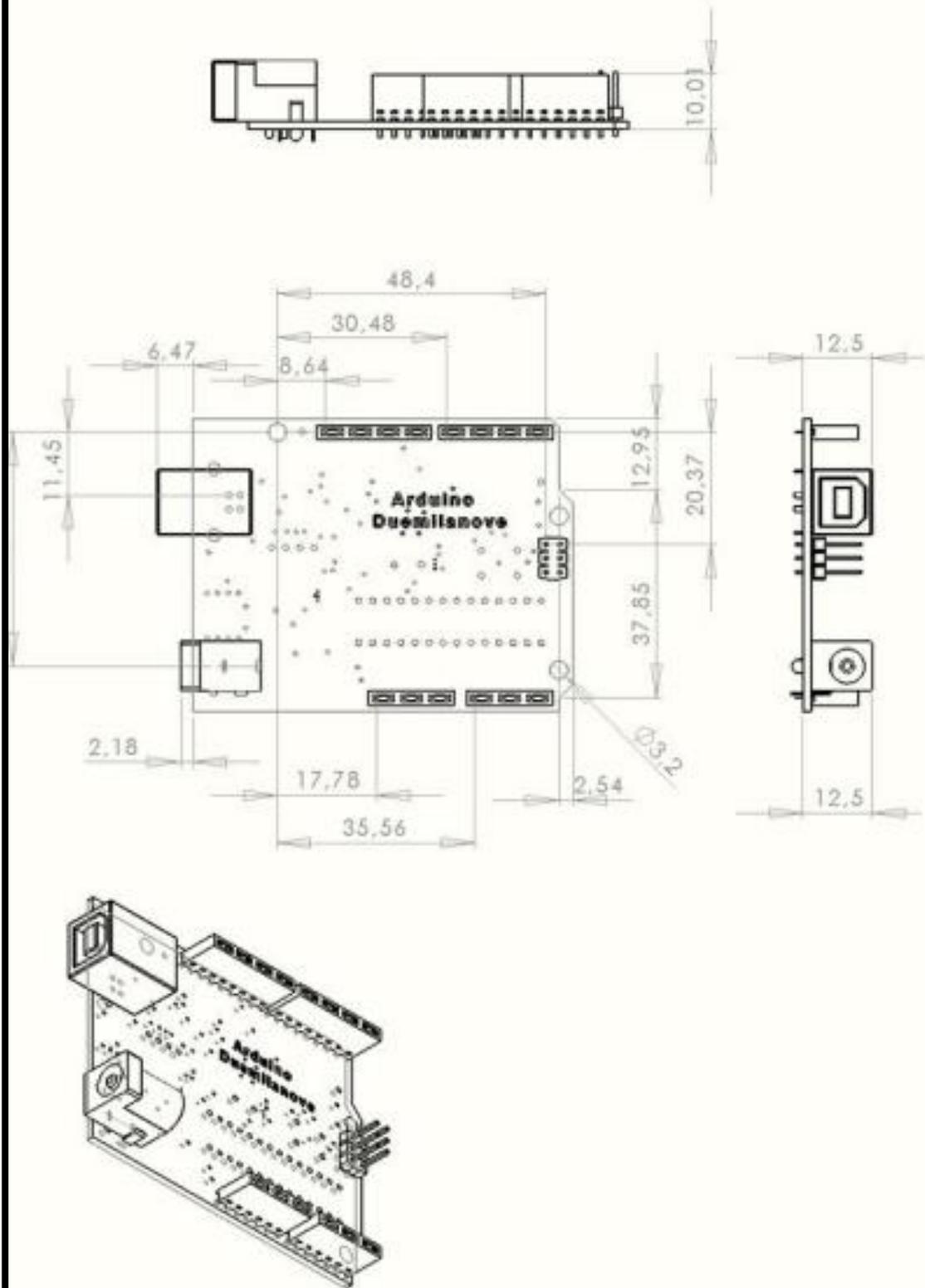
## the board



The diagram shows an Arduino Uno R3 board with the following components labeled:

- TX/RX Leds**: Located near the USB interface.
- "Test" Led 13**: A red LED on the digital pin header.
- digital pins**: The top row of pins on the digital pin header.
- USB Interface**: The USB Type-B connector.
- ATmega328**: The microcontroller chip.
- External Power Supply**: The DC power jack.
- power pins**: The 5V, GND, and GND pins on the power header.
- analog pins**: The A0 through A5 pins on the analog pin header.
- Power Led**: A green LED on the digital pin header.
- ICSP Header**: The In-Circuit Serial Programming header.
- Reset Button**: A push-button for resetting the microcontroller.
- I2C**: The I2C interface pins on the digital pin header.

# Dimensioned Drawing



radiospares

RADIONICS



**ANEXO G. SISTEMA PROTOTIPO DE CONTROL DE CALIDAD.**

