



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN MECATRÓNICA

TEMA: “DISEÑO Y CONSTRUCCIÓN DE UN ESCÁNER DIGITAL
3D A PARTIR DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICO
PARA EL LABORATORIO CNC DE LA ESPE-L”

AUTOR: MIGUEL ÁNGEL MOLINA HIDALGO

DIRECTOR: ING. FAUSTO ACUÑA
CODIRECTOR: ING. EDDIE GALARZA

LATACUNGA

2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICADO

ING. FAUSTO ACUÑA (DIRECTOR)
ING. EDDIE GALARZA (CODIRECTOR)

CERTIFICAN

Que el trabajo titulado “DISEÑO Y CONSTRUCCIÓN DE UN ESCÁNER DIGITAL 3D A PARTIR DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICO PARA EL LABORATORIO CNC DE LA ESPE-L” realizado por MIGUEL ÁNGEL MOLINA HIDALGO, ha sido guiado y revisado periódicamente y cumple normas estatuarías establecidas de la Universidad de las Fuerzas Armadas - ESPE.

Latacunga, marzo del 2015

ING. FAUSTO ACUÑA
DIRECTOR

ING. EDDIE GALARZA
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

CARRERA DE INGENIERÍA MECATRÓNICA

DECLARACIÓN DE RESPONSABILIDAD

MIGUEL ÁNGEL MOLINA HIDALGO

DECLARO QUE:

El proyecto de grado denominado “DISEÑO Y CONSTRUCCIÓN DE UN ESCÁNER DIGITAL 3D A PARTIR DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICO PARA EL LABORATORIO CNC DE LA ESPE-L”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, marzo del 2015

MIGUEL ÁNGEL MOLINA HIDALGO

C.C.: 050325576-2

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORIZACIÓN

Yo, MIGUEL ÁNGEL MOLINA HIDALGO

Autorizo a la UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE, la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO Y CONSTRUCCIÓN DE UN ESCÁNER DIGITAL 3D A PARTIR DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICO PARA EL LABORATORIO CNC DE LA ESPE-L”, cuyo contenido, ideas y criterios es de mi exclusiva responsabilidad y autoría.

Latacunga, marzo del 2015

MIGUEL ÁNGEL MOLINA HIDALGO

C.C.: 050325576-2

DEDICATORIA

El presente proyecto se lo dedico a tres mujeres, a las que quiero mucho y que forman parte indispensable en mi vida, en primer lugar a mi Abuelita Antonia que desde el cielo me cuida, me ilumina, me apoya, me bendice y me guía en todas las actividades que realizo o las que tengo que hacer, a mi Madre Beatriz que es la responsable de darme la vida y quien está conmigo mañana, tarde, noche y madrugadas al lado mío viendo que no me falte nada, que este contento y que no me pase nada de malo y a mi Enamorada quien con su apoyo incondicional, sus palabras de aliento, sus risas me ha ayudado a salir adelante, siempre buscando que haga las cosas de forma correcta para ser un hombre de bien tanto dentro como fuera de mi hogar.

Chafas

AGRADECIMIENTO

Dios gracias por darme la vida, por la oportunidad de disfrutar de este objetivo cumplido con toda mi familia, por darles la alegría de que me vean como un Ingeniero, a mis Padres Héctor y Beatriz por darme la vida y cuidarme día a día buscando siempre mi bienestar, dándome lo que he necesitado ya sea poco o bastante pero siempre con mucho amor, a mi Hermano Diego quien fue como mi compañero de tesis, ayudándome con sus conocimientos y capacidades, a mi Hermano Vinicio por estar al pendiente de mí y de las cosas que hago ya sea con una llamada o un mensaje, a mi Hermano Xavier por saber escucharme cuando estoy mal y darme consejos que me servirán para mi futuro, a mi familia que siempre están al pendiente de como estoy o que hago y como me va, así como me quieren yo también les quiero mucho, a mis Ingenieros por sus conocimientos impartidos y a todos mis amigos y amigas que conocí en la Universidad, en el grupo de Danza, en fin gracias a todos.

ÍNDICE DE CONTENIDO

CARÁTULA	i
CERTIFICADO.....	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
RESUMEN	xvi
ABSTRACT	xvii
PRESENTACIÓN.....	xviii
CAPÍTULO I	1
GENERALIDADES.....	1
1.1. Escáner 3D.....	1
1.2.1. Tipos de escáner.....	1
1.2. Tipos de escáner existentes en el mercado	4
1.2.1. Escáner de contacto.....	4
1.2.2. Escáner Móvil.....	4
1.2.3. Escáner giratorio	6
1.2.4. Escáner Manual.....	7
1.2.5. Escáner con Láser	8
1.3. Algoritmo de Canny.....	9
1.3.1. Operador de Canny.....	10
1.3.2. Fases de detección de bordes de Canny	10
CAPÍTULO II.....	13
DISEÑO, SELECCIÓN Y CONSTRUCCIÓN DEL SOFTWARE Y	
HARDWARE.....	13
2.1. Diseño del hardware para ubicar y generar la rotación del objeto.....	13
2.1.1. Parámetros para seleccionar el motor.....	13

2.1.2.	Selección del motor	14
2.2.	Diseño del hardware y software para controlar el motor a pasos	16
2.2.1.	Parámetros para seleccionar el dispositivo electrónico	16
2.2.2.	Selección del PIC	16
2.2.3.	Diseño de la placa de control y potencia para el PIC seleccionado	17
2.2.4.	Programación para la comunicación PC con el motor a pasos	20
2.3.	Diseño del sistema de adquisición de imágenes	27
2.3.1.	Parámetros para la selección de la webcam.....	27
2.3.2.	Selección de la webcam.....	27
2.3.3.	Requerimientos de la PC para la instalación de la Cámara	30
2.4.	Diseño del software para el procesamiento de imágenes y la reconstrucción en 3 dimensiones	30
2.4.1.	Parámetros para la selección del software	30
2.4.2.	Selección del software de programación	31
2.4.3.	Programación en el software seleccionado	31
2.4.4.	Programación para captura, procesamiento y almacenamiento de imágenes.....	37
2.5.	Diseño de la estructura mecánica para el montaje de los elementos.....	55
CAPÍTULO III		59
PRUEBAS Y RESULTADOS.....		59
3.1.	Pruebas.....	59
3.1.1.	Prueba de Escaneado	59
3.1.2.	Prueba de Impresión	78
3.2.	Resultados.....	88
3.3.	Análisis Técnico - Económico	89
3.4.	Validación de la hipótesis	89
CAPÍTULO IV.....		90
CONCLUSIONES Y RECOMENDACIONES		90
4.1.	Conclusiones	90
4.2.	Recomendaciones.....	92

BIBLIOGRAFÍA	94
NETGRAFÍA	94
ANEXOS	97

ÍNDICE DE TABLAS

Tabla 2. 1	Especificaciones del Motor Crouzet.....	14
Tabla 2. 2	Especificaciones del Motor Sanyo Denki	15
Tabla 2. 3	Características técnicas y especificaciones de la Microsoft LifeCam VX-3000.....	28
Tabla 2. 4	Características técnicas y especificaciones de la Genius iSlim 300X	29
Tabla 2. 5	Características técnicas y especificaciones de la Logitech HD C270	29
Tabla 3. 1	Valores de funcionalidad del escáner	60
Tabla 3. 2	Medidas generales del frasco de cerámica.....	85
Tabla 3. 3	Medidas de los detalles del frasco de cerámica.....	85
Tabla 3. 4	Error de las medidas generales del frasco de cerámica	87
Tabla 3. 5	Error de las medidas de los detalles del frasco de cerámica	87
Tabla 3. 6	Materiales utilizados en el proyecto	89

ÍNDICE DE FIGURAS

Figura 1. 1	Método de visión estereoscópica	4
Figura 1. 2	Escaneado de contacto de una obra arqueológica	4
Figura 1. 3	Escáner Móvil en pleno uso.....	6
Figura 1. 4	Escáner Giratorio “MarkerBot Digitizer”.....	6
Figura 1. 5	Escáner Manual OptiNum 3D.....	7
Figura 1. 6	Disposición de los elementos para la captura de escena del objeto a digitalizar	8
Figura 1. 7	Sistema real con todos sus componentes armados y en sus respectivas posiciones.....	9
Figura 1. 8	Diagrama de bloques del algoritmo de Canny.....	10
Figura 1. 9	Asignación de direcciones para la búsqueda del cierre del contorno...12	
Figura 2. 1	Motor Crouzet Unipolar Permanent Magnet Stepper Motor.....	14
Figura 2. 2	Motores Sanyo Denki	15
Figura 2. 3	Diseño de la placa con los elementos a tamaño real	17
Figura 2. 4	Diseño de las pistas con la simbología de los elementos	18
Figura 2. 5	Pistas necesarias para colocarla en la placa de baquelita	18
Figura 2. 6	Placa puesta en el ácido clorhídrico	19
Figura 2. 7	Placa retirada del ácido con todos los elementos soldados	19
Figura 2. 8	Elementos colocados en sus respectivas ubicaciones	20
Figura 2. 9	Programa que permite configurar los descriptores e instaladores .inf.....	21
Figura 2. 10	Configuraciones necesarias para nuestros descriptores e instaladores .inf.....	22
Figura 2. 11	Ventana de asistente para hardware nuevo encontrado.....	22
Figura 2. 12	Selección de la opción para instalar desde una ubicación en la PC	23
Figura 2. 13	Finalización de instalación del hardware.....	23
Figura 2. 14	Sistema intentando instalar el dispositivo de forma automática	24
Figura 2. 15	Software de dispositivo correctamente instalado.....	24
Figura 2. 16	Ingresar al administrador de dispositivos	24
Figura 2. 17	Actualización de archivo .inf con un dispositivo conectado en un nuevo puerto USB.....	25

Figura 2. 18	Buscar el archivo .inf desde una ubicación de la PC	25
Figura 2. 19	Dispositivo correctamente instalado	26
Figura 2. 20	Dispositivo correctamente instalado y listo para usarse	26
Figura 2. 21	Webcam Microsoft LifeCam VX-3000.....	28
Figura 2. 22	Webcam Genius iSlim 300X	28
Figura 2. 23	Webcam Logitech HD C270.....	29
Figura 2. 24	Diagrama de flujo para la captura de las imágenes	34
Figura 2. 25	Diagrama de flujo para la generación de los puntos en el espacio	35
Figura 2. 26	Diagrama de flujo para la generación de los puntos en el espacio	36
Figura 2. 27	LabVIEW 2011	37
Figura 2. 28	Paletas de Vision and Motion	38
Figura 2. 29	Esquema para abrir una cámara	38
Figura 2. 30	Esquema para cambiar la imagen a escala de grises	39
Figura 2. 31	Esquema para binarizar la imagen	39
Figura 2. 32	Esquema de captura, filtrado y almacenamiento de una imagen	40
Figura 2. 33	Matlab R2010a.....	41
Figura 2. 34	Secuencia para crear un archivo .m.....	42
Figura 2. 35	Editor del archivo .m.....	42
Figura 2. 36	Presentación del proyecto	43
Figura 2. 37	Icono de GUI.....	44
Figura 2. 38	Ventana de inicio de GUI	44
Figura 2. 39	Entorno de diseño de GUI	45
Figura 2. 40	Insertar Push Button	45
Figura 2. 41	Edición del campo String	46
Figura 2. 42	Edición del campo Tag	47
Figura 2. 43	Edición del tamaño de la letra.....	47
Figura 2. 44	Botones modificados	48
Figura 2. 45	Insertar campos estáticos de texto.....	48
Figura 2. 46	Campos estáticos de texto modificados.....	49
Figura 2. 47	Insertar campos de edición de texto	50
Figura 2. 48	Campos de edición de texto modificados	50
Figura 2. 49	Axis insertado en la interfaz gráfica	51
Figura 2. 50	Axis 2, 3 y 4 colocados para visualización de las imágenes capturadas, imágenes filtradas y generación de coordenadas	51

Figura 2. 51	Llamada del archivo .m llamado “Programa”	52
Figura 2. 52	Editor de Matlab para colocar las líneas de programación.....	53
Figura 2. 53	Pantalla lista para realizar las pruebas necesarias	53
Figura 2. 54	Interfaz gráfica de usuario llamada Selector_datos.....	54
Figura 2. 55	Interfaz configurada para el proceso	54
Figura 2. 56	Base para colocar el motor a pasos	55
Figura 2. 57	Plato en el cual se va a colocar el objeto a escanear	55
Figura 2. 58	Motor acoplado a la caja y al plato giratorio	56
Figura 2. 59	Platina en forma de C de soporte de la webcam	56
Figura 2. 60	Caja para colocar los elementos creados	57
Figura 2. 61	Ensamblaje final.....	57
Figura 2. 62	Ensamblaje final con todos sus elementos y pintado	58
Figura 3. 1	Resultados estadísticos de la funcionalidad del escáner.....	60
Figura 3. 2	Imagen original de la botella y aplicada el filtro de Canny al inicio del proceso	61
Figura 3. 3	Imagen recortada de la botella para obtener las coordenadas x, y, z.....	61
Figura 3. 4	Imagen original de la botella y aplicada el filtro de Canny durante el proceso.....	62
Figura 3. 5	Imagen recortada de la botella para obtener las coordenadas x, y, z.....	62
Figura 3. 6	Imagen original del frasco de cerámica y aplicada el filtro de Canny al inicio del proceso	63
Figura 3. 7	Imagen recortada del frasco de cerámica para obtener las coordenadas x, y, z.....	63
Figura 3. 8	Imagen original del frasco de cerámica y aplicada el filtro de Canny durante el proceso	64
Figura 3. 9	Imagen recortada del frasco de cerámica para obtener las coordenadas x, y, z.....	64
Figura 3. 10	Imagen original del florero y aplicada el filtro de Canny al inicio del proceso.....	65
Figura 3. 11	Imagen recortada del florero para obtener las coordenadas x, y, z.....	65

Figura 3. 12	Imagen original del florero y aplicada el filtro de Canny durante el proceso	66
Figura 3. 13	Imagen recortada del florero para obtener las coordenadas x, y, z.....	66
Figura 3. 14	Imagen original del frasco de plástico y aplicada el filtro de Canny al inicio del proceso	67
Figura 3. 15	Imagen recortada del frasco de plástico para obtener las coordenadas x, y, z.....	67
Figura 3. 16	Imagen original del frasco de plástico y aplicada el filtro de Canny durante el proceso	68
Figura 3. 17	Imagen recortada del frasco de plástico para obtener las coordenadas x, y, z.....	68
Figura 3. 18	Imagen original de la taza y aplicada el filtro de Canny al inicio del proceso.....	69
Figura 3. 19	Imagen recortada de la taza para obtener las coordenadas x, y, z.....	69
Figura 3. 20	Imagen original de la taza y aplicada el filtro de Canny durante el proceso	70
Figura 3. 21	Imagen recortada de la taza para obtener las coordenadas x, y, z.....	70
Figura 3. 22	Botella escaneada	71
Figura 3. 23	Frasco con detalles de una cara de conejo escaneado	71
Figura 3. 24	Florero en forma de corazón escaneado	72
Figura 3. 25	Frasco de plástico escaneado	72
Figura 3. 26	Taza de café escaneada.....	73
Figura 3. 27	Nube de puntos abierta en SolidWorks	74
Figura 3. 28	Creación del croquis en tres dimensiones.....	74
Figura 3. 29	Superficie a partir de las curvas del croquis	75
Figura 3. 30	Objeto hecho superficie y cerrado completamente	75
Figura 3. 31	Objeto hecho sólido y añadido letras para imprimirlo	76
Figura 3. 32	Botella hecha superficie y cerrada completamente	76
Figura 3. 33	Florero hecho superficie y cerrado completamente	77
Figura 3. 34	Frasco de plástico hecho superficie y cerrado completamente.....	77
Figura 3. 35	Taza hecha superficie y cerrada completamente.....	78

Figura 3. 36	Objeto hecho sólido y con detalles añadidos	78
Figura 3. 37	Archivo .stl abierto en el software CatalystEX.....	79
Figura 3. 38	Objeto orientado en el eje “x” y escalado.....	79
Figura 3. 39	Proceso de colocar material antes que empiece a imprimir.....	80
Figura 3. 40	Objeto en el área permitida para imprimir.....	80
Figura 3. 41	Proceso de impresión.....	81
Figura 3. 42	Tiempo restante de impresión y porcentaje de impresión ya realizado	81
Figura 3. 43	Objeto impreso finalizado.....	82
Figura 3. 44	Objeto real y objeto impreso vista frontal	82
Figura 3. 45	Objeto real y objeto impreso vista lateral	83
Figura 3. 46	Objeto real y objeto impreso vista superior	83
Figura 3. 47	Objeto real y objeto impreso vista isométrica	84
Figura 3. 48	Resultado con el escáner 3D Noomeo.....	84
Figura 3. 49	Gráfico de medidas generales del frasco	86
Figura 3. 50	Gráfico de medidas de los detalles del frasco.....	86

RESUMEN

Hoy en día la tecnología que va apareciendo es muy avanzada en comparación al anterior milenio, por lo que las personas tienen que estar cada vez actualizándose debido a que aparecen dispositivos que ayudan a realizar trabajos que en el pasado eran muy difíciles y que ahora se los desarrolla con mayor facilidad. Uno de estos dispositivos son los escáner 3D, que toman imágenes de un objeto y lo convierten en una imagen tridimensional, la cual se la puede manipular mediante otro software para poder crear réplicas o mejorar algunos detalles y enviarlos a una máquina para su fabricación. Estos avances tecnológicos permiten realizar un sinnúmero de aplicaciones y han sido la base para el desarrollo del presente proyecto, el mismo que da a entender de una manera sencilla, el principio del funcionamiento de un escáner moderno para que los estudiantes puedan crear nuevos y mejores diseños de estos dispositivos. Para tal fin se utilizó una webcam que capturó algunas imágenes de un objeto no mayor de 13 cm. de alto por 13 cm. de ancho y una dimensión mínima de 5 cm., de un peso máximo de 500 gr. y mínimo de 100 gr. Dicho objeto se situó en un plato con un motor, el cual gira un cierto número de grados, dependiendo del objeto a escanear, para obtener la figura completa en tres dimensiones. Con el procesamiento de imágenes se logró encontrar las coordenadas en el espacio para poder ser reconstruidas y visualizadas en otro software de diseño.

PALABRAS CLAVE:

- INGENIERÍA MECATRÓNICA
- COMUNICACIÓN USB-PC
- ESCÁNER 3D

ABSTRACT

Today the technology is appearing is very advanced compared to the previous millennium, so people have to be updated each time because they appear devices that help carry out work in the past were very difficult and now the develops more easily. One such device is the 3D scanner, taking images of an object and make it a three-dimensional image, which it can be manipulated by other software to create replicas or improve some details and send them to a machine for manufacturing. These technological advances allow for a myriad of applications and have been the basis for the development of this project, the same as implied in a simple manner, the principle of operation of a modern scanner to enable students to create new and better designs of these devices. To this end, a webcam that captured some images of an object no larger than 13 cm was used high by 13 cm. wide and a minimum size of 5 cm. with a maximum weight of 500 gr. and at least 100 grammes. Said object is placed in a dish with a motor which rotates a certain number of degrees, depending on the object to be scanned, for the full figure in three dimensions. With image processing it was possible to find the coordinates in space to be reconstructed and displayed in other design software.

KEYWORDS:

- MECHATRONICS ENGINEERING
- COMMUNICATION USB-PC
- 3D SCANNER

PRESENTACIÓN

En el presente proyecto se desarrolla un escáner 3D, el cual mediante la captura de imágenes y el procesamiento de las mismas permite crear una réplica de un objeto en tres dimensiones, además se lo modela en un software CAD para cambiar o añadir algunos detalles al mismo.

El Capítulo I contiene la información acerca de los conceptos básicos que se deben tener en cuenta para poder desarrollar el proyecto como por ejemplo los conceptos de un escáner 3D, los tipos de escáner, los diseños implementados en los escáneres.

El Capítulo II muestra el proceso para definir los parámetros del diseño tanto del hardware como del software, además detalla todo el procedimiento necesario para el desarrollo de todos los elementos del proyecto y toda su implementación.

El Capítulo III indica las pruebas realizadas con diferentes objetos y con todos los elementos ubicados en sus respectivas ubicaciones.

El Capítulo IV detalla las conclusiones y recomendaciones que se dieron durante todo el proceso de diseño, construcción, programación y pruebas del proyecto en pleno funcionamiento.

Además se incluye anexos, referencias bibliográficas y referencias net gráficas que son necesarias para ayudar a la comprensión del proyecto.

CAPÍTULO I

GENERALIDADES

1.1. Escáner 3D

Un escáner 3D es un aparato que captura la forma y características de cualquier tipo de volumen o ambiente y mediante un software específico construye un modelo tridimensional del mismo.

Permite crear una nube de puntos a partir de varias imágenes y luego de un procesamiento con un algoritmo dentro de un software se determina la forma en la cual van a estar unidos a partir de figuras geométricas básicas como pueden ser triángulos, cuadrados, etc., y de esta manera crear el objeto escaneado en tres dimensiones.

Este objeto creado se podrá visualizar de forma tridimensional en algún software capaz de leer los objetos en forma de malla para luego modificarlo conforme se necesite en el proceso.

Dentro de los escáneres 3D se puede destacar que existen algunos que no simplemente pueden obtener información de la distancia a la que se encuentra el objeto sino que también se puede obtener la información de los colores que conforman dicho objeto.

1.2.1. Tipos de escáner

Hay dos tipos de escáner que son los más grandes y engloban a todos los demás que son: los de contacto y los sin contacto.

1.1.1.1 Escáner de contacto

Este tipo de escáner constan de un palpador, el material del que está hecho casi siempre es de acero duro o zafiro, el cual se va desplazando por la superficie del

objeto y mediante sensores internos va registrando la información de la ubicación espacial del palpador y gracias alguna máquina de medición por coordenadas se va visualizando el objeto.

Se usan en su mayoría en control dimensional en procesos de fabricación y pueden conseguir precisiones típicas de 0,01mm. Su mayor desventaja es que requiere el contacto físico con el objeto, por lo que el acto de escanear el objeto quizás lo modifique o lo dañe y además el tiempo de escaneado es muy extenso.

1.1.1.2 Escáner sin contacto

Estos escáneres a diferencia de los anteriores son más rápidos al momento de escanear y por ende no tienen contacto con el objeto en ningún momento sino que se utiliza otras técnicas ya sea con luminosidad o con alguna luz láser. Estos tipos de escáner se dividen en activos y pasivos.

- **Activos**

Los escáneres activos emiten alguna clase de señal y analizan su retorno para capturar la geometría de un objeto o una escena. Se utilizan radiaciones electromagnéticas (desde ondas de radio hasta rayos X) o ultrasonidos.

Tipos de escáner activos:

- Time of flight (Tiempo de vuelo)
- Diferencia de fase
- La holografía conoscópica
- La luz estructurada
- La luz modulada
- Triangulación

- **Pasivos**

El sistema visual humano permite obtener información de profundidad mediante la fusión de dos escenas monoculares, que son las escenas que captan cada uno de nuestros ojos. La visión pasiva se basa entonces en utilizar dos puntos de vista de un mismo objeto para encontrar las coordenadas tridimensionales.

La principal complejidad de este método es la correspondencia de puntos en cada una de las imágenes monoculares. Las técnicas encargadas de realizar esta tarea presentan generalmente un elevado costo computacional.

Tipos de escáner pasivos:

- Estereoscópicos
- Silueta
- Con ayuda del usuario (modelado basado en imagen)

- **Estereoscópicos**

Los sistemas estereoscópicos utilizan el mismo principio de la fotogrametría, utilizando la medida de la paralaje entre dos imágenes para determinar la distancia de cada píxel de la imagen.

Emplean generalmente dos cámaras de video, levemente separadas, mirando a la misma escena. Analizando las diferencias leves entre las imágenes vistas por cada cámara, es posible determinar la distancia en cada punto en las imágenes.

Este método se basa en la visión estereoscópica humana. Dicho sistema se observa en la figura 1.1.



Figura 1. 1 Método de visión estereoscópica

Fuente: <http://es.scribd.com/doc/97305139/Vision-en-3D>

1.2. Tipos de escáner existentes en el mercado

1.2.1. Escáner de contacto

La aplicación más común de este tipo de escáner es para conocer el estado de conservación de las obras. (ejm: Catedral de León), para su reproducción, conservación (como se ha hecho en el parque arqueológico de Alarcos (Ciudad Real)) y hasta para su manipulación virtual. Dicha técnica se observa en la figura 1.2.



Figura 1. 2 Escaneado de contacto de una obra arqueológica

Fuente: <http://blog.educastur.es/restauras/2008/05/28/escaneado-3d/>

1.2.2. Escáner Móvil

El Instituto Tecnológico AIDO y la empresa AGFRA han desarrollado un innovador sistema de inspección en 3D para detectar defectos de piezas de automóviles en la cadena de producción, verificando hasta un 100% de las piezas fabricadas.

Fabricar un coche que aporte el máximo de calidad en cada una de sus piezas y, a la vez, que tenga un precio atractivo para el usuario, es todo un reto. Por eso, la industria automovilística necesita contar con tecnologías que aplicadas a sus procesos de producción sean capaces de aumentar el valor añadido de los vehículos y disminuir los costes. El mercado exige satisfacer al conductor con coches cuyos componentes sean cada vez más perfectos y, para lograr este objetivo, es fundamental detectar con la máxima precisión cualquier defecto que pudiera existir en las piezas que configuran un vehículo.

El Instituto Tecnológico AIDO y la empresa AGFRA han desarrollado WLIP (White Light Inspection Project), un sistema revolucionario que permite realizar un control tridimensional para detectar defectos de las piezas de automóviles en la cadena de producción.

Esta tecnología está basada en la digitalización 3D por luz blanca estructurada, que permite la captura masiva de puntos en la pieza (hasta 2.000.000 de puntos por toma) en tiempos muy reducidos (0,3 segundos). Así se obtiene un modelo global de las zonas a inspeccionar capaz de generar información tridimensional comparable con los modelos CAD de fabricación.

Como resultado, el proceso de inspección se acelera y simplifica, aumentando el valor añadido de las piezas producidas, abaratando costes y mejorando la calidad del producto final. Además, esta herramienta pionera consigue incrementar notablemente la productividad al tratarse de un sistema automático que actúa en tiempo real, ininterrumpidamente, durante las 24 horas del día; dicho sistema se observa en la figura 1.3.

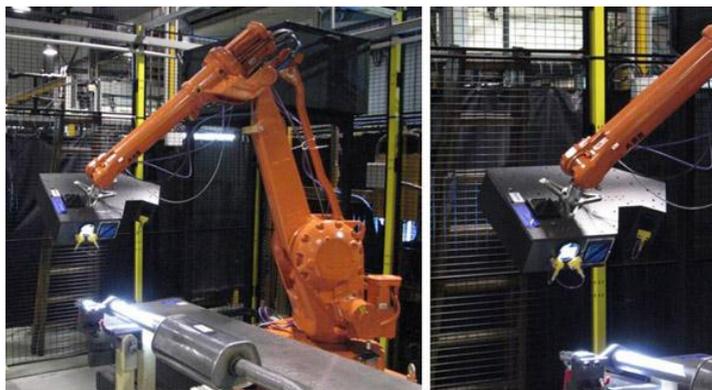


Figura 1. 3 Escáner Móvil en pleno uso

Fuente: <http://www.agfra.es/proyectos/hola-mundo/>

1.2.3. Escáner giratorio

El MarkerBot Digitizer se presenta como una máquina que tarda unos 12 minutos en escanear cada objeto. ¿Cómo funciona? Dos haces láser escanean la geometría de la superficie del objeto que está girando en el plato central y posteriormente convierte la información obtenida en un modelo 3D fácilmente replicable. Así mismo, también presenta varias limitaciones. En primer lugar, es necesario llevar a cabo el proceso en una habitación con una buena iluminación para que el resultado sea el más óptimo.

Por otro lado, hay que tener en cuenta que los objetos a escanear no pueden superar los 3 kilogramos de peso. Dicho sistema se observa en la figura 1.4.



Figura 1. 4 Escáner Giratorio “MarkerBot Digitizer”

Fuente: <http://imapas.net/2013/08/23/digitizer-el-escaner-3d-de-makerbot-se-pone-la-venta-por-1-400-dolares/>

En la actualidad, uno de los mayores problemas entorno a este campo radica en la dificultad de crear un modelo 3D para posteriormente imprimirlo. Se necesitan ciertos conocimientos técnicos para ser capaz de utilizar un software 3D y no siempre resulta obvio adquirirlos.

1.2.4. Escáner Manual

OptiNum 3D es el escáner 3D adecuado para aquellos que buscan una alta precisión adaptable a cualquier entorno de trabajo. Optinum pesa menos de 2kg. y es muy compacto. Al tener una batería incorporada le proporciona horas de trabajo junto a su ordenador portátil. Adecuado para realizar trabajo de campo en cualquier situación, simplemente desplácese al lugar deseado y comience a escanear. Una vez capturados los datos, procéselos cómodamente desde su oficina o laboratorio.

Nuestro sistema de auto posicionamiento evita tener que preparar el objeto o instalar un sistema de referencia. Además, no es necesario colocar pegatinas o etiquetas evitando el deterioro del objeto y la pérdida de tiempo.

Las nubes de puntos generadas por nuestro escáner Optinum tienen una precisión de hasta +/- 100 micras y son adecuados para un amplio rango de objetos cuyo volumen esté entre 10cm³ y 1m³. Cada disparo del escáner captura hasta 500.000 puntos de información. Es la mayor cantidad de información obtenida por captura en este tipo de escáner 3D. En la figura 1.5 se observa el escáner Optinum en funcionamiento.



Figura 1. 5 Escáner Manual OptiNum 3D

Fuente: <http://www.cinpi.es/blog/2011/10/optinum-escaner-3d-digitalizacion-3d-inspeccion-e-ingenieria-inversa/>

1.2.5. Escáner con Láser

Uno de los tipos de escáner más comunes es el que utiliza una cámara y un láser, la cámara ayuda a ver la proyección del láser en el objeto y de esta manera se obtiene la información necesaria para lograr la reconstrucción de los objetos en tres dimensiones.

Este escáner consta de un software para la adquisición de las imágenes y su posterior procesamiento y reconstrucción. Al culminar el escaneo se tiene un tipo de archivo que es compatible con el software de modelado de objetos en tres dimensiones para realizar otras actividades como por ejemplo: imprimirlo en tres dimensiones o el mecanizado en una fresadora o torno CNC. El funcionamiento del sistema se muestra en la figura 1.6.

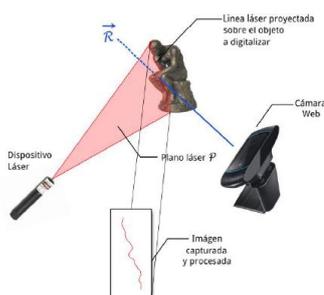
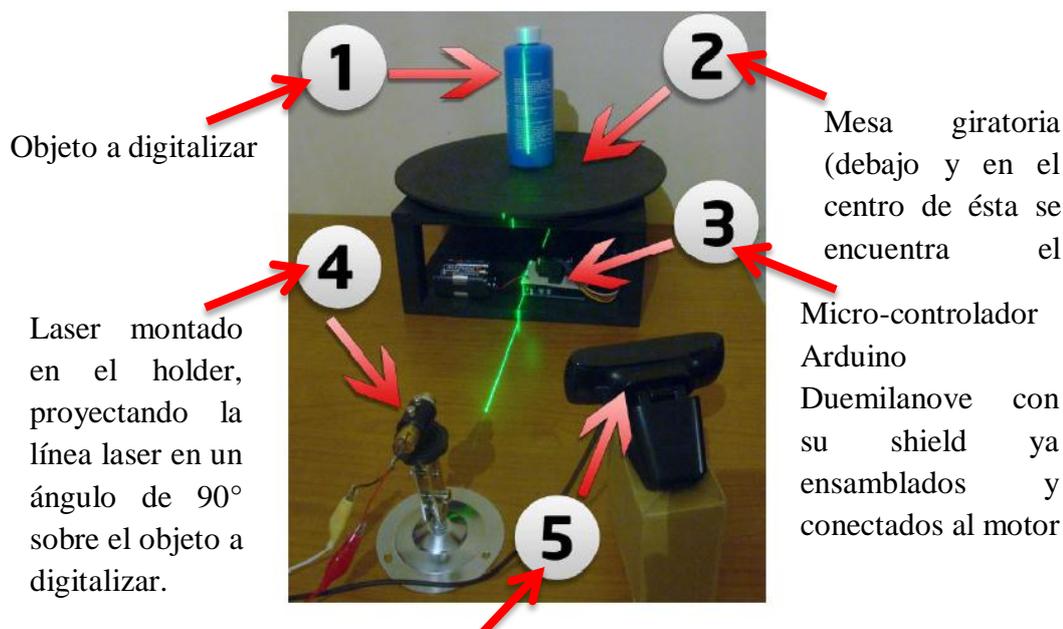


Figura 1. 6 Disposición de los elementos para la captura de escena del objeto a digitalizar

Fuente: http://www.researchgate.net/publication/220006751_Sistema_para_digitalizacin_y_modelado_3D_de_objetos_mediante_proyeccion_laser_utilizando_hardware_de_arquitectura_abierta

En la figura 1.7 se observa todos los componentes que conforman dicho sistema en sus respectivas ubicaciones.



Cámara Logitech HD Pro Webcam C910 capturando imágenes de la escena y dispuesta en un ángulo de aproximadamente $16^\circ - 20^\circ$ con respecto al láser.

Figura 1. 7 Sistema real con todos sus componentes armados y en sus respectivas posiciones

Fuente: http://www.researchgate.net/publication/220006751_Sistema_para_digitalizacin_y_modelado_3D_de_objetos_mediante_proyeccion_laser_utilizando_hardware_de_arquitectura_abierta

1.3. Algoritmo de Canny

El operador de detección de bordes de Canny fue desarrollado por un catedrático de la universidad de Berkeley (EEUU) en 1986 y se basa en un algoritmo de múltiples fases para detectar un amplio rango de bordes. Es sin duda el operador más utilizado en la detección de bordes.

El objetivo de Canny era descubrir el algoritmo óptimo para la detección de bordes. Un detector óptimo significará una buena detección, el algoritmo debe marcar tantos bordes reales como sea posible, una buena localización, los bordes marcados deben estar lo más cerca posible del borde en la imagen real, y por último una mínima respuesta, es decir, un borde dado debe ser marcado sólo una vez y donde sea posible el ruido presente en la imagen no debería crear falsos bordes.

1.3.1. Operador de Canny

Esta técnica, que se caracteriza por estar optimizada para la detección de bordes diferenciales, consta de tres fases pero algunos autores agregan una última fase, siendo en total cuatro fases.

1.3.2. Fases de detección de bordes de Canny

- Obtención del gradiente
- Supresión no máxima al resultado del gradiente
- Histéresis de umbral a la supresión no máxima
- Cierre de contornos abiertos

1.3.2.1. Obtención del gradiente

En la figura 1.8 se observa el diagrama de bloques de las fases que están involucradas dentro del funcionamiento del algoritmo de Canny.

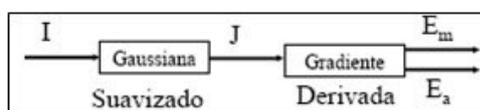


Figura 1. 8 Diagrama de bloques del algoritmo de Canny

Fuente: <http://oefa.blogspot.com/2009/04/deteccion-de-bordes-algoritmo-de-canny.html>

Para obtener el gradiente se tiene que suavizar la imagen:

- Aplicar a la imagen I un suavizado gaussiano G o filtro gaussiano (también se puede aplicar otros filtros) $J = I * G$
- Para calcular el kernel gaussiano G se utiliza la Ec. 1.1

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{Ec. 1.1}$$

Enseguida calcular el gradiente:

Para cada pixel de la imagen $J(i,j)$

- Calcular las componentes del gradiente J_x y J_y
- Calcular la magnitud de los bordes con la Ec. 1.2

$$e_s(i, j) = \sqrt{J_x^2(i, j) + J_y^2(i, j)} \quad \text{Ec. 1.2}$$

- Estimar la orientación de la normal de los bordes con la Ec. 1.3

$$e_o(i, j) = \arctan \frac{J_y}{J_x} \quad \text{Ec. 1.3}$$

1.3.2.2. Supresión no máxima al resultado del gradiente

El objetivo de este paso es obtener bordes del pixel de grosor al considerar únicamente píxeles cuya magnitud es máxima en bordes gruesos y descartar aquellos cuyas magnitudes no alcancen ese máximo.

- Para todo punto se obtiene la dirección más cercana dk a 0° , 45° , 90° y 135° en $Ea(i,j)$
- Si $Em(i,j)$ es menor que uno de sus dos vecinos en la dirección dk , $IN(i,j)=0$. Si no $IN(i,j)=Em(i,j)$

1.3.2.3. Histéresis de umbral a la supresión no máxima

Permite eliminar máximos procedentes de ruido, etc.

- Entrada IN , Ea , y dos umbrales $T1$ y $T2$ ($T2 > T1$)
- Para todo punto en IN , y explorando en un orden:
 - Localizar el siguiente punto tal que $IN(i,j) > T2$
 - Seguir las cadenas de máximos locales a partir de $IN(i,j)$ en ambas direcciones perpendiculares a la normal al borde siempre que $IN > T1$. Marcar los puntos explorados.

- La salida es un conjunto de bordes conectados de contornos de la imagen, así como la magnitud y orientación.

1.3.2.4. Cierre de contornos abiertos (Algoritmo de Deriche y Cocquerez)

- La imagen de entrada es una imagen de contornos binarizada (1= borde; 0=no borde)
- Para cada punto de borde de un extremo abierto se le asigna un código que determina las direcciones de búsqueda para el cierre del contorno, dicha asignación se la observa en la figura 1.9.

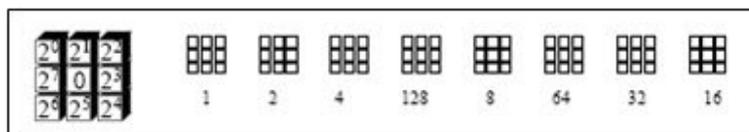


Figura 1. 9 Asignación de direcciones para la búsqueda del cierre del contorno

Fuente: <http://oefa.blogspot.com/2009/04/deteccion-de-bordes-algoritmo-de-canny.html>

- Para los píxeles marcados con este código se marca como píxel de borde el de máximo gradiente en las tres direcciones posibles.
- Se repiten los pasos hasta que se cierren todos los contornos.

CAPÍTULO II

DISEÑO, SELECCIÓN Y CONSTRUCCIÓN DEL SOFTWARE Y HARDWARE

Según el numeral 1.2 donde se analizó los tipos de escáner que existen se optó por realizar un escáner sin contacto ya este no se demora mucho tiempo en escanear, que tenga una base giratoria y una webcam pero que no tenga un láser debido a que la luz del ambiente causaría cambios en el escaneado final, además se puso como límites máximos para los objetos a escanear los siguientes:

Alto = 13 cm.

Ancho = 13 cm.

Peso máximo = 500 gr.

Peso mínimo = 100 gr.

2.1. Diseño del hardware para ubicar y generar la rotación del objeto

Para iniciar este proceso se seleccionó un sistema adecuado que logre girar el objeto, el mismo que ayudó a realizar las capturas necesarias. Con lo antes expuesto se vio la necesidad de buscar un motor que a su vez se acople a una base sobre la cual se colocará el objeto a escanear.

2.1.1. Parámetros para seleccionar el motor

A continuación se detallará los parámetros necesarios para elegir el motor apropiado a utilizarse en el desarrollo del proyecto:

- Tamaño no más de 10cm por lado
- Que sea de fácil posicionamiento en un cierto número de grados
- Que su giro sea total y no tenga algún tipo de truncamiento
- Que no funcione con corrientes ni voltajes altos
- Que posea un buen torque para mover objetos como adornos de cerámica

2.1.2. Selección del motor

Con los parámetros antes mencionados se encontró que el tipo de motor adecuado es el motor a pasos, ya que la característica principal de este es la facilidad que tiene para posicionarse un cierto número de grados de acuerdo a la cantidad de pulsos que reciba. Para escoger el motor a pasos ideal se buscó entre las marcas que existen en el mercado como son: Schneider Electric, Crouzet, Johnson Electric, Aerotech, Sanyo Denki, etc. y sus diversos modelos. En la figura 2.1 se observa el motor Crouzet y en la tabla 2.1 sus características.



Figura 2. 1 Motor Crouzet Unipolar Permanent Magnet Stepper Motor

Fuente: <http://uk.rs-online.com/web/p/stepper-motors/0440307/>

Tabla 2. 1

Especificaciones del Motor Crouzet

Current Rating	0.19A
Depth	43mm
Holding Torque	240mNm
Number of Pole Pairs	2
Number Of Wires	6
Resistance Per Phase	26.7 Ω
Shaft Diameter	2mm
Shaft Length	13mm
Step Angle	7.5°
Stepper Motor Type	Permanent Magnet
Voltage Rating	12.7 Vdc
Winding Arrangement	Unipolar

En la figura 2.2 se observa los tipos de motores Sanyo Denki, de todos estos se escogió uno pequeño y de buen torque, las características se muestran en la tabla 2.2.



Figura 2. 2 Motores Sanyo Denki

Fuente: <http://www.tme.eu/es/Document/48ac9f8512124ec9a791d3759c517a64/sanyo-motor.pdf>

Tabla 2. 2

Especificaciones del Motor Sanyo Denki

Basic Step Angle	Permanent Magnet
Bipolar Parallel Current	1,8° ± 0.09°
Unipolar Current	0.42
Resistance	0.6
Inductance	6.7
Bipolar Holding Torque	5.4
Unipolar Holding Torque	19
Rotor Inertia	14.5
Theoretical Acceleration	30
Back E.M.F.	18
Mass	0.2

Teniendo en cuenta estas características se decidió seleccionar el motor a pasos de la marca Sanyo Denki 103-547-52500 que se mostró en la figura 2.2 que consta con 5 cables y es bipolar lo que facilita el control del movimiento y además tiene un ángulo de paso más pequeño en comparación al de la marca Crouzet lo que permite dar pasos más pequeños y poder tomar más fotos, en el Anexo A se encuentra la ficha técnica del motor seleccionado.

2.2. Diseño del hardware y software para controlar el motor a pasos

Para el control del motor a pasos es necesario entablar una comunicación con la PC para enviar y recibir información, la misma que ayudará a que el motor gire con el objeto para obtener todas las imágenes para la reconstrucción, en este proceso es necesario la utilización de algún dispositivo, en este caso es un elemento electrónico, con su respectiva programación.

2.2.1. Parámetros para seleccionar el dispositivo electrónico

A continuación se detallará los parámetros para seleccionar el elemento electrónico capaz de realizar la comunicación entre software y hardware:

- Que posea entradas y salidas para envío y recepción de datos
- Que pueda conectarse vía USB
- Que no tenga un lenguaje muy complejo al momento de realizar la programación

Teniendo en cuenta estas características el elemento que las cumple es el PIC, pero este cuenta con algunas familias para lo cual se va a buscar el que mejor se acople al proyecto.

2.2.2. Selección del PIC

Dentro de todas las familias que tiene el PIC se encontró con la familia 18Fxx5x los cuales tienen módulos USB, estos permiten la comunicación sin ningún problema con la laptop, los PIC de esa familia son los de numeración PIC18F2455/2550/4455/4550, para este proyecto se va a utilizar el 18F2550 el cual consta de 28 pines, este fue seleccionado debido a que en el proceso no se necesita muchas salidas para enviar los datos, en el Anexo B se encuentra el datasheet del PIC seleccionado.

2.2.3. Diseño de la placa de control y potencia para el PIC seleccionado

Una vez realizada la programación en Matlab y la comunicación con el PIC se procedió a realizar un circuito de control y potencia para manipular los datos que se envía desde la PC y así poder controlar el motor a pasos.

Para realizar la parte de control se buscó la configuración que se utiliza para cualquier PIC como por ejemplo la colocación del botón del reset, el oscilador, etc., para la parte de potencia se diseñó un circuito con transistores para el control de la corriente necesaria y mover el motor a pasos y para separar la parte de control con potencia se utilizó un opto transistor para que no se afecte las dos partes de la placa.

En la realización de estos dos circuitos se realizó el diseño en el programa PCBWizard con las medidas de los elementos a tamaño real como se muestra en la figura 2.3.

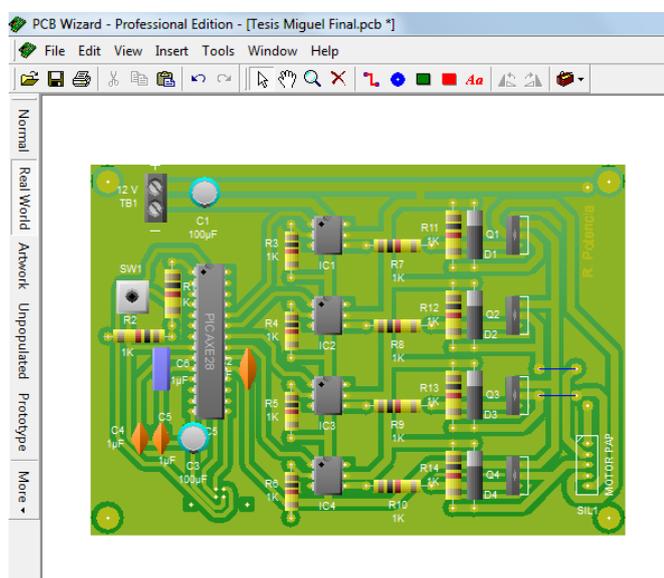


Figura 2. 3 Diseño de la placa con los elementos a tamaño real

Una vez concluido la conexión de todos los elementos en el programa, se procedió a generar las pistas con la simbología de los elementos como se muestra en la figura 2.4 y en la figura 2.5 se observa solo las pistas que se traspasaron a la placa de baquelita.

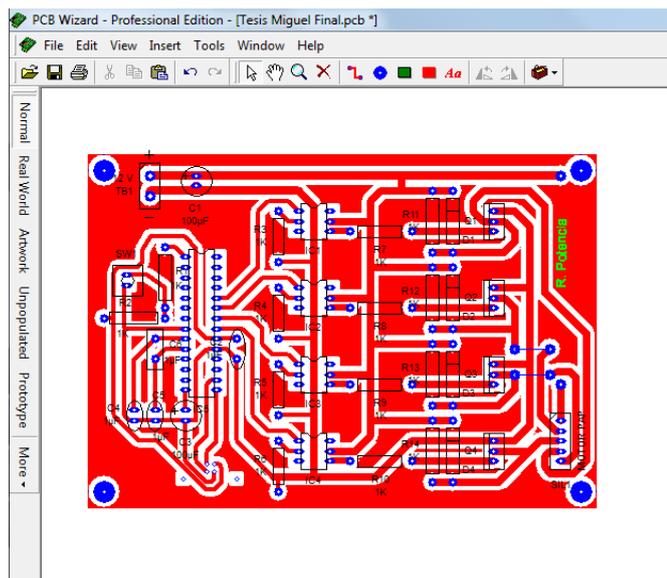


Figura 2. 4 Diseño de las pistas con la simbología de los elementos

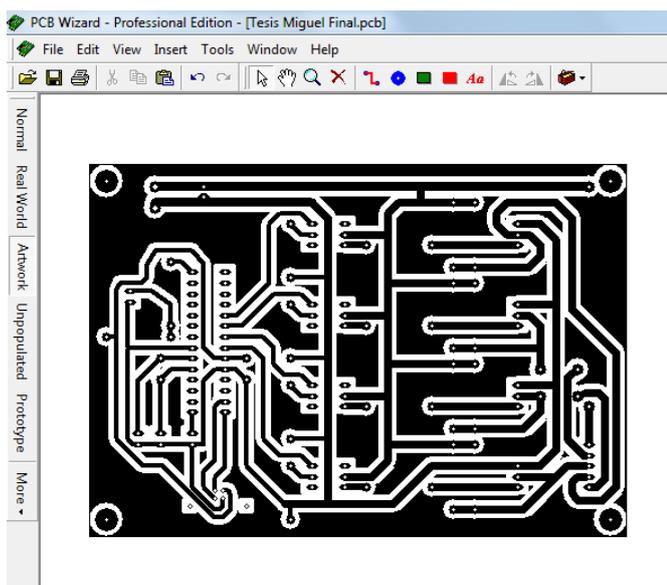


Figura 2. 5 Pistas necesarias para colocarla en la placa de baquelita

Una vez que ya se pasó las pistas a la placa de baquelita y se hizo las perforaciones en cada uno de los pines donde van colocados los elementos, se colocó la placa en un recipiente con agua hervida y una funda de ácido clorhídrico para retirar el cobre de la placa como se muestra en la figura 2.6.



Figura 2. 6 Placa puesta en el ácido clorhídrico

Concluido este paso se procedió a la soldadura de todos los elementos en sus respectivas ubicaciones, después de este se hizo una caja para colocar la placa, el material de la caja es mica transparente, ya que las pistas deben estar aisladas de cualquier contacto ya sea del metal o el agua para que no exista un corto circuito en la misma, la placa soldada se muestra en la figura 2.7.

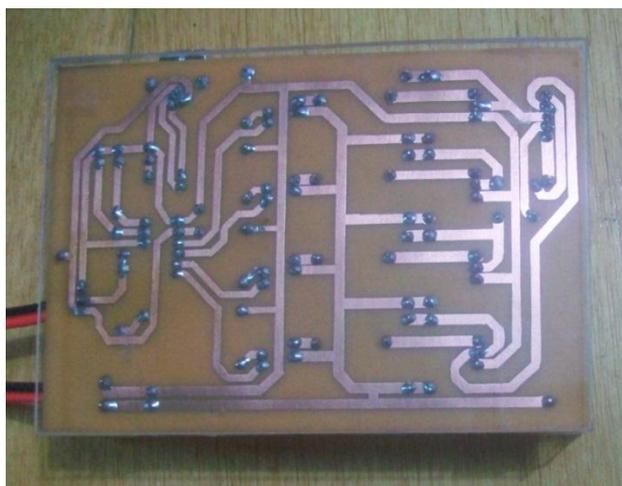


Figura 2. 7 Placa retirada del ácido con todos los elementos soldados

En la figura 2.8 se muestra la placa desde la parte superior con todos los elementos soldados en sus respectivas ubicaciones.



Figura 2. 8 Elementos colocados en sus respectivas ubicaciones

2.2.4. Programación para la comunicación PC con el motor a pasos

En la actualidad los dispositivos que se conectan a la PC son vía USB ya que las laptop no constan de puerto serie ni de puerto paralelo, por lo que en este proyecto se decidió usar una comunicación USB y utilizando un PIC lograr el control de nuestro motor a pasos desde la PC.

Para realizar esto se tuvo que descargar de la página de Microchip los archivos `mpusbapi.dll` y `_mpusbapi` los cuales permiten entablar la comunicación entre el PIC y la PC, estos archivos se los guardó en una carpeta.

Para el PIC se utilizó la programación en C y a su vez para poder compilar el programa PIC C Compiler de la empresa CCS el cual genera el archivo `.hex` necesario para grabar en el PIC, en el Anexo C se encuentra la programación para que el PIC se pueda comunicar vía USB con la PC.

Ahora para que el compilador funcione de acuerdo al PIC se descargó el programa “Automatización de datos Enumeración USB” que permite configurar los descriptores y los archivos instaladores `.inf` necesarios para enumerar los PIC 18Fxx5x y hacer posible las transferencias a través del puerto serie universal USB, el programa a utilizarse se muestra ejecutado en la figura 2.9.

Es un programa que permite introducir datos como Nombre, VID, PID, Fabricante, Descripción, entre otros para configurar el instalador que necesita el Windows cuando se conecte el dispositivo por primera vez.

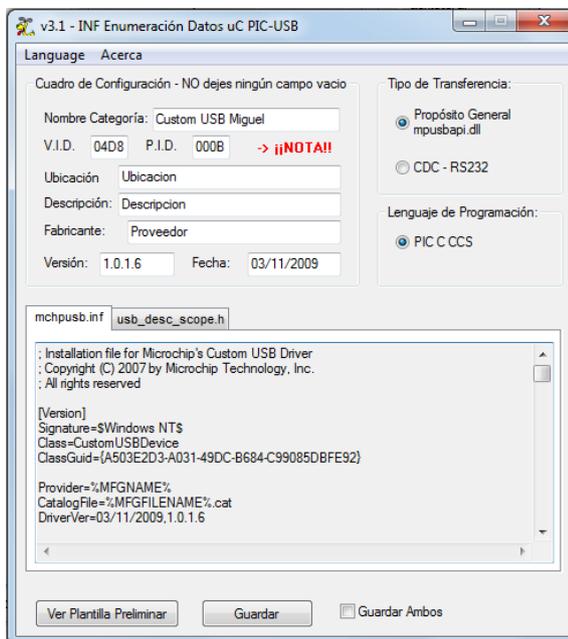


Figura 2. 9 Programa que permite configurar los descriptores e instaladores .inf

Aquí se va poder cambiar los datos antes ya mencionamos en donde se encuentran los espacios de VID y PID en los cuales se va a colocar los valores de VID=04D8 y PID=000B estos son los que indican al programa que el PIC es de la compañía Microchip.

Hay dos modos de creación de drivers, el modo de transmisión BULK para controlar las transacciones con MCHPUSB.DLL y mediante la clase CDC (que servirá para la emulación RS-232).

En este proyecto se utilizó el modo BULK, para lo cual en este programa se eligió en la parte Tipo de Transferencia la opción que dice Propósito General para utilizar la mpushbapi.dll y a su vez se seleccionó la opción de Guardar ambos en la parte inferior del programa como se muestra en la figura 2.10.

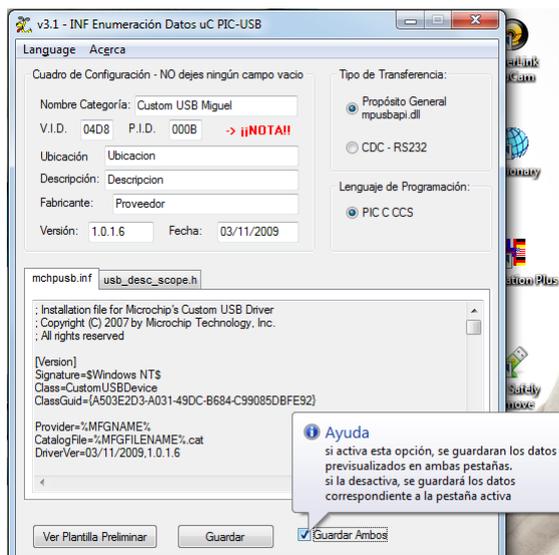


Figura 2. 10 Configuraciones necesarias para nuestros descriptores e instaladores .inf

Una vez que se conecta el PIC a la PC por primera vez lo reconoció como un dispositivo USB nuevo y se desplegó la ventana de asistente para hardware nuevo encontrado como se muestra en la figura 2.11.



Figura 2. 11 Ventana de asistente para hardware nuevo encontrado

En este caso se seleccionó que no se desea que Windows se conecte a Windows Update sino que se escogió No por el Momento y luego en siguiente.

Una vez que se realizó este paso en la siguiente pantalla se marcó Instalar desde una lista o ubicación específica como se muestra en la figura 2.12.

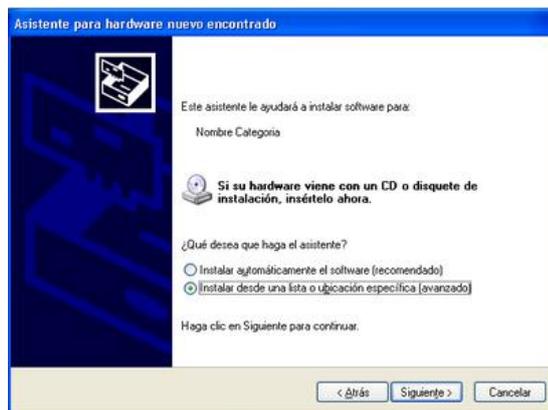


Figura 2. 12 Selección de la opción para instalar desde una ubicación en la PC

El archivo que se buscó es el que se generó en el programa de “Automatización de datos Enumeración USB”, es el mchpusb.inf y luego en siguiente, de ahí el sistema comenzó a instalar lo necesario para que nuestro dispositivo sea reconocido sin ningún problema, el asistente se muestra en la figura 2.13.



Figura 2. 13 Finalización de instalación del hardware

En caso de que se conecte el dispositivo a otro puerto USB dentro de la misma PC o en otra PC el sistema por sí solo intentará instalar el dispositivo de forma automática como se muestra en la figura 2.14.

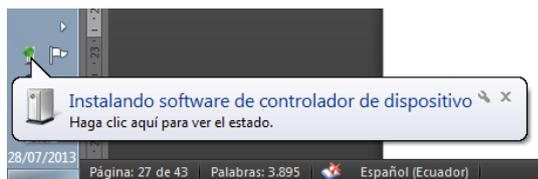


Figura 2. 14 Sistema intentando instalar el dispositivo de forma automática

Luego de esto el dispositivo apareció que está correctamente instalado como se muestra en la figura 2.15.

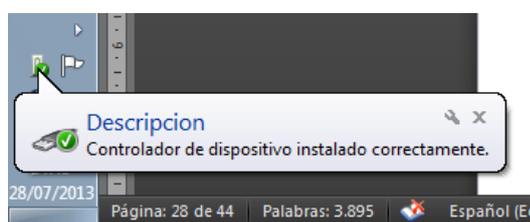


Figura 2. 15 Software de dispositivo correctamente instalado

Para verificar que está correctamente instalado se ingresó a MI PC o Equipo y se dio clic derecho y se entró en Propiedades y de ahí en administrador de dispositivos como se muestra en la figura 2.16.

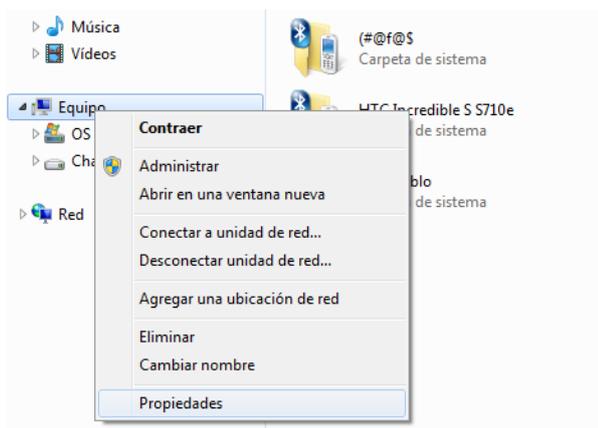


Figura 2. 16 Ingresar al administrador de dispositivos

La aparición de un triángulo amarillo indicará que todavía no está listo para utilizar, para esto se va a dar clic derecho en el dispositivo y luego en Actualizar software de controlador como se muestra en la figura 2.17.

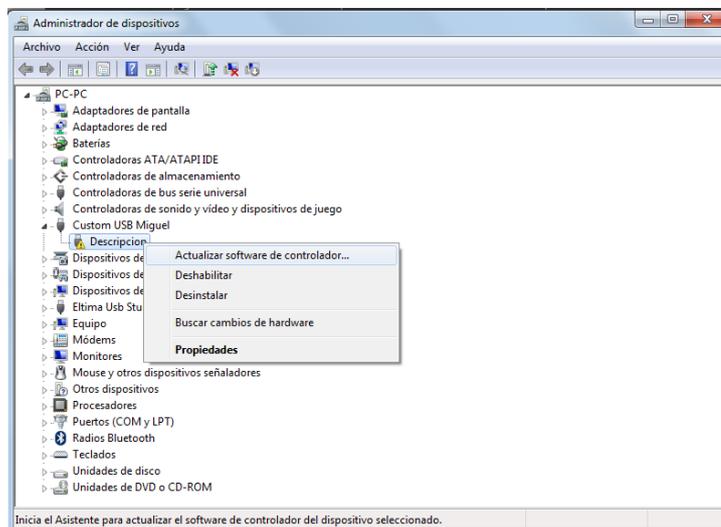


Figura 2. 17 Actualización de archivo .inf con un dispositivo conectado en un nuevo puerto USB

A continuación se realizó el procedimiento de cargar el archivo .inf, este paso se observa en la figura 2.18.

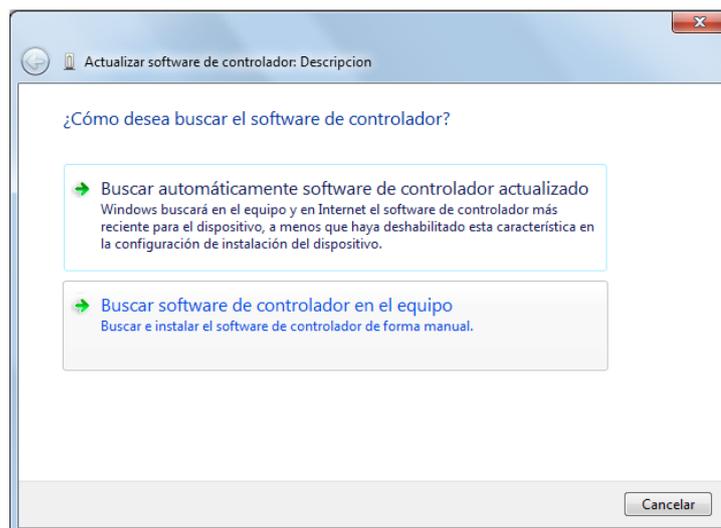


Figura 2. 18 Buscar el archivo .inf desde una ubicación de la PC

Ahí el sistema instaló el software necesario para que el dispositivo sea reconocido exitosamente y salió un mensaje como se observa en la figura 2.19.

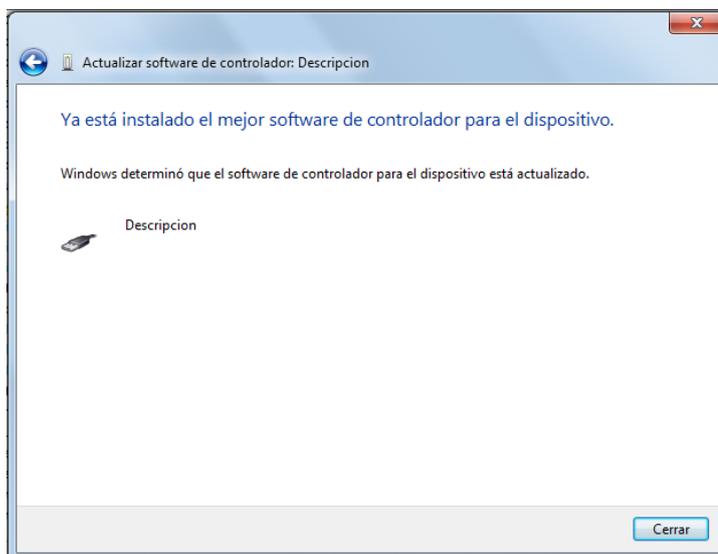


Figura 2. 19 Dispositivo correctamente instalado

Una vez realizado este paso se procedió a retirar el cable USB y se colocó de nuevo y se verificó si el dispositivo está listo para usarse como se muestra en la figura 2.20.

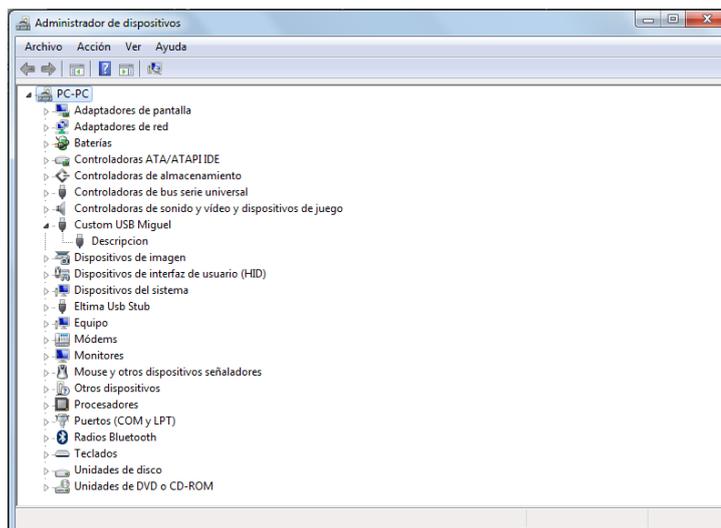


Figura 2. 20 Dispositivo correctamente instalado y listo para usarse

2.3. Diseño del sistema de adquisición de imágenes

Como en el diseño se va a utilizar una webcam, se debe tener en cuenta las existentes en el mercado y las características de las mismas, pero siempre comparando cada una con las necesidades que se tiene dentro del diseño.

2.3.1. Parámetros para la selección de la webcam

En la actualidad se dispone de muchas marcas entre las cuales se destacan Genius, Microsoft, Logitech, Philips, D-Link, Cisco, Axis, Nilux y dentro de estas existen diversos modelos, pero para el proyecto se buscó una webcam que tenga las siguientes características:

- Que posea conexión USB
- Resolución de imágenes mínimo VGA (640 x 480 píxeles)
- Que posea un excelente ángulo de visión de al menos 100°
- Capacidad para tomar y enviar fotos a través de un software

2.3.2. Selección de la webcam

Teniendo en cuenta las características antes mencionadas se buscó algunos modelos de webcam de diferentes marcas con sus respectivas especificaciones técnicas para poder elegir la que mejor se adapte al proyecto, a continuación se detallan los modelos de las marcas más comercializadas y conocidas en el mercado y que son aceptadas por la mayor parte de usuarios a nivel mundial.

En las figuras 2.21, 2.22 y 2.23 se observa diferentes modelos de webcam que existen en el mercado y en las tablas 2.3, 2.4 y 2.5 se detallan las características de cada una respectivamente.



Figura 2. 21 Webcam Microsoft LifeCam VX-3000

Fuente: <http://es.kioskea.net/download/descargar-2562-driver-de-webcam-de-microsoft-lifecam-vx-3000>

Tabla 2. 3

Características técnicas y especificaciones de la Microsoft LifeCam VX-3000

Imagen	Los parámetros principales	Además
Matriz: 0,3 millones de píxeles, CMOS	Conexión: USB 2.0	Ángulo de visión: 55 grados
Resolución (video): 640x480	Micrófono: incorporado	Construcción: Monte el monitor
La frecuencia de cuadros máxima: 30 Hz	Enfoque: Manual	La longitud del cable: 1,8 m
Resolución (foto): 640x480	Zoom Digital: 2x función de seguimiento facial: Sí	Medidas (WxHxD): 53x55x74 mm
Resolución interpolada en megapíxeles (foto): 1,3 millones de píxeles	Sistemas operativos compatibles: Windows Vista/XP SP2	Peso: 95 g



Figura 2. 22 Webcam Genius iSlim 300X

Fuente: <http://www.geniusnet.com/Genius/wSite/ct?xItem=19461&ctNode=1302>

Tabla 2. 4

Características técnicas y especificaciones de la Genius iSlim 300X

Imagen	Los parámetros principales	Además
Matriz: 0,3 millones de píxeles, CMOS	Conexión: USB 1.1	Ángulo de inclinación: 100 grados
Resolución ([vídeo]: rdug1.640x480, 3200x2400] [lduga interpolados)	Micrófono: no	Construcción: Monte el monitor
La frecuencia de cuadros máxima: 30 Hz	Enfoque: Manual	Medidas (WxHxD): 51x65x69 mm
Apoyos: 640x480 @ 30 Hz	Sistemas operativos compatibles: Windows 7/Vista/XP SP2	Peso: 64 g



Figura 2. 23 Webcam Logitech HD C270

Fuente: <http://www.logitech.com/en-hk/product/hd-webcam-c270>

Tabla 2. 5

Características técnicas y especificaciones de la Logitech HD C270

Imagen	Los parámetros principales	Además
Resolución (video): 1280x720	Conexión: USB 2.0	Construcción: Monte el monitor
Resolución interpolada en megapíxeles (foto): 3 millones de píxeles	Micrófono: incorporado	Micrófono integrado con tecnología RightSound
	Sistemas operativos compatibles: Windows XP/Vista/7, Mac OS X 10.4.9 o superior	Información adicional: Las llamadas de vídeo en el formato H.720p

Tomando en cuenta las especificaciones anteriores se optó por utilizar la Genius iSlim 300X que se observó en la figura 2.2 ya que tiene un buen ángulo de inclinación lo que permitió capturar fotos desde diferentes posiciones dependiendo el tamaño del objeto.

Una vez seleccionada la webcam y para que funcione correctamente, a continuación se muestra los requerimientos necesarios que debe cumplir la PC.

2.3.3. Requerimientos de la PC para la instalación de la Cámara

- Procesador Pentium 4 o superior
- Sistema operativo Windows Vista/XP/2000
- DirectX 9.0 o superior
- 512 MB de RAM
- 1 GB de espacio libre en el disco duro
- Puerto USB disponible
- Unidad de CD/DVD-ROM para instalar software
- Altavoces

2.4. Diseño del software para el procesamiento de imágenes y la reconstrucción en 3 dimensiones

Para lograr la reconstrucción en 3 dimensiones hay que tener en cuenta que se necesita de la ayuda de algún tipo de software y de algoritmos que permitan el procesamiento de cada una de ellas para luego juntar toda esa información y unir las al final del proceso.

2.4.1. Parámetros para la selección del software

A continuación se detallan los parámetros básicos para seleccionar el software adecuado para desarrollar el proyecto:

- Adquirir, procesar y guardar imágenes en formato .jpg

- Realizar operaciones matemáticas con los valores que vienen dentro de cada imagen como por ejemplo las coordenadas exactas de cada píxel
- Reconstruir imágenes en 3 dimensiones
- Permitir una comunicación vía USB entre el software de la pc y hardware

Dentro de todo lo que se refiere a plataformas o software está en seleccionar uno en el que se pueda procesar imágenes y además realizar cálculos matemáticos con los valores que se obtienen de cada imagen, también debe ser capaz de entablar algún tipo de comunicación desde la PC hacia dispositivos físicos como un PIC, un motor a pasos, entre otros, de forma automática.

2.4.2. Selección del software de programación

Con lo anteriormente indicado se encontró que destacan los siguientes softwares: LabVIEW y Matlab debido a su flexibilidad en el uso de comandos para realizar cualquier tipo de operación, en los cuales se involucran imágenes y además permiten una comunicación fácil entre software y hardware.

2.4.3. Programación en el software seleccionado

Antes de realizar la programación se elaboró ecuaciones para determinadas etapas del sistema, como son: la cantidad de capturas que puede realizar el escáner, creación de la coordenada Z en función del ángulo de giro.

Con el motor seleccionado que tiene un ángulo de giro de $1,8^\circ$ por paso, se utilizó la Ec. 2.1 para determinar cuántos pasos tiene que dar para que dé una vuelta completa.

$$\# \text{ de pasos} = 360^\circ / 1,8^\circ \quad \text{Ec. 2.1}$$

Luego se realizó el cálculo del ángulo que debe girar el motor para cada paso de los 200 que dio como resultado de la anterior ecuación, estos valores se generaron con la Ec. 2.2.

$$\text{ángulo entre capturas} = 1,8^\circ * \text{paso} \quad \text{Ec. 2.2}$$

Finalmente para determinar cuántas capturas se deben realizar se aplicó la Ec. 2.3.

$$\text{capturas} = 360^\circ / \text{ángulo entre capturas} \quad \text{Ec. 2.3}$$

Con las anteriores ecuaciones se generó una tabla con 200 valores, la tabla completa se muestra en el Anexo D, y se optó por los valores exactos en cuanto al ángulo de posicionamiento en función del número de fotos y número de pasos exactos del motor, los valores se indican en la tabla 2.6.

Tabla 2. 6

Valores con el número de imágenes necesarias para capturar

CAPTURAS	ÁNGULO ENTRE CAPTURAS	PASOS
1	360	200
2	180	100
4	90	50
5	72	40
8	45	25
10	36	20
20	18	10
40	9	5

Con los resultados de la tabla anterior se descartaron los dos primeros debido a que no hay como reconstruir un objeto con tan solo una o dos imágenes, los tres valores siguientes también se los descartan porque no serviría reconstruir un objeto con tan pocas imágenes, de ahí los tres valores siguientes son válidos para el proceso.

De la imagen se encontró el perfil del objeto y se almacenó los datos de forma ordenada como se muestra en la matriz A, en la primera columna se guardó los datos de la coordenada Y, mientras que en la segunda los de la coordenada X.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \end{bmatrix}$$

A partir de esta matriz se creó una segunda del mismo tamaño como se muestra en la matriz B.

$$B = \begin{bmatrix} b_{11} & k_{12} \\ b_{21} & k_{22} \\ \vdots & \vdots \\ b_{m1} & k_{m2} \end{bmatrix}$$

Donde k es el valor de la mitad de la imagen que es constante para los n valores del perfil.

Teniendo estas dos ecuaciones se calculó el desplazamiento de la coordenada X con la Ec. 2.4 y con la Ec. 2.5 se encontró la coordenada Z.

$$X = \cos \left[\left(\frac{\theta\pi}{180^\circ} \right) (B - A) \right] \quad \text{Ec. 2.4}$$

$$Z = \sin \left[\left(\frac{\theta\pi}{180^\circ} \right) (B - A) \right] \quad \text{Ec. 2.5}$$

Con la sintaxis correcta se creó una matriz llamada “puntos” en la cual se colocó las coordenadas X, Y, Z encontradas anteriormente.

Para continuar la programación, se realizó diagramas de flujo en los que se muestran la lógica de programación de las tres etapas en las que está dividido todo el sistema.

En la figura 2.24 se muestra el diagrama de la primera parte del proceso de escaneado, en donde se indica los pasos que se debe seguir para la captura de un cierto número de imágenes para luego guardarlas en una dirección específica con su nombre respectivo.

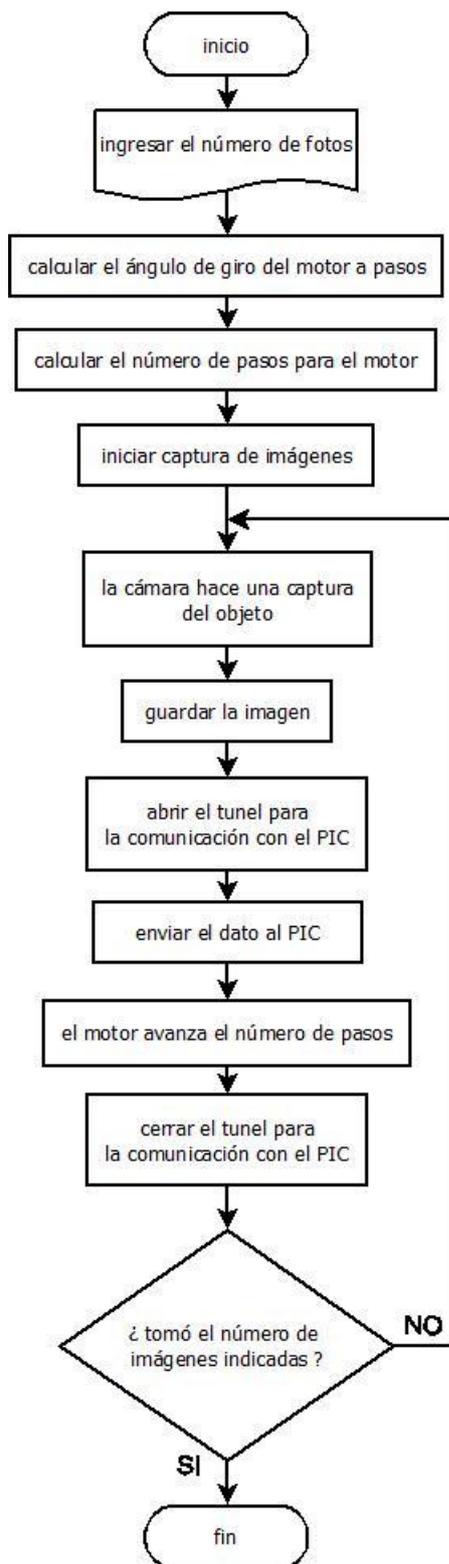


Figura 2. 24 Diagrama de flujo para la captura de las imágenes

En la figura 2.25 se aprecia el diagrama donde se realiza el proceso de aplicar el filtro de Canny, el mismo que cambia a escala de grises a cada una de las imágenes capturadas en el anterior paso.



Figura 2. 25 Diagrama de flujo para la generación de los puntos en el espacio

En la figura 2.26 se observa los pasos a seguir para la apertura de las imágenes antes guardadas, a las cuales se las recorta para su procesamiento y determinar las coordenadas del perfil de cada una, luego se procedió al cálculo de la coordenada Z y a su vez se crea un archivo .txt que es una nube de puntos, un v.obj y un f.obj que son archivos de malla.

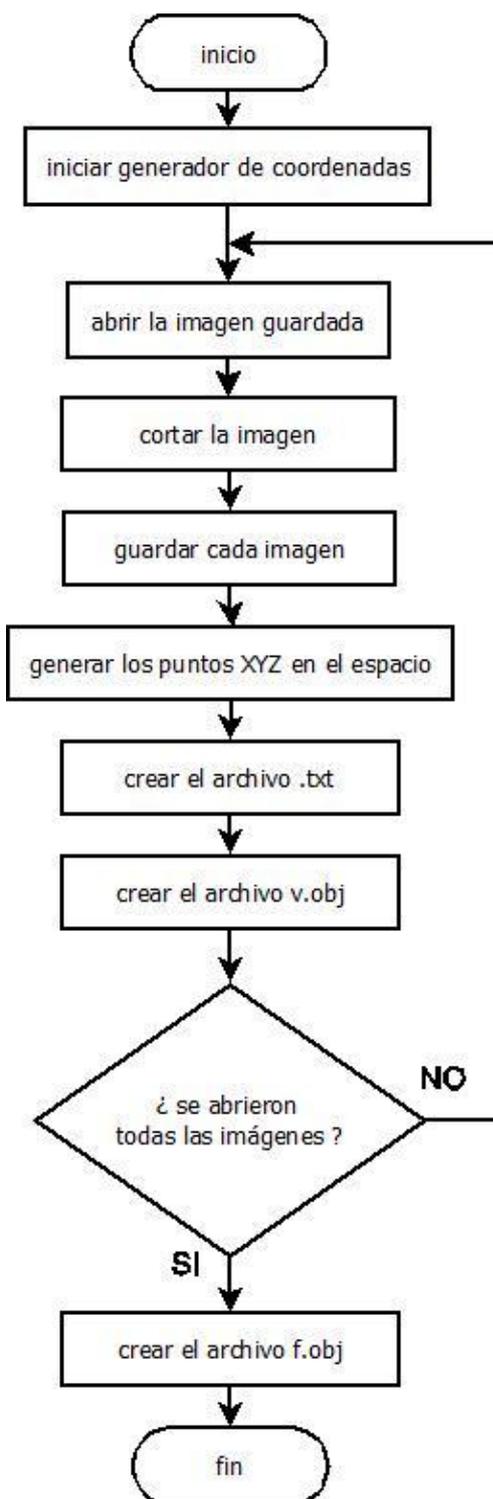


Figura 2. 26 Diagrama de flujo para la generación de los puntos en el espacio

Una vez que se tiene los dos archivos en .obj se procedió a unir el segundo en el primero y con esto se tuvo todos los datos necesarios para poder visualizar el objeto escaneado en tres dimensiones.

Inicialmente se realizó dos pruebas: la primera que consta de la programación en LabVIEW 2011 para 64 bits y la segunda con Matlab R2010a para 32 bits.

2.4.4. Programación para captura, procesamiento y almacenamiento de imágenes

Primero se utilizó LabVIEW ya que no utiliza líneas de comandos, sino más bien se trata de una programación gráfica y se utiliza simplemente íconos los cuales se van uniendo unos con otros para realizar cualquier operación. En la figura 2.27 se observa el software ejecutado.

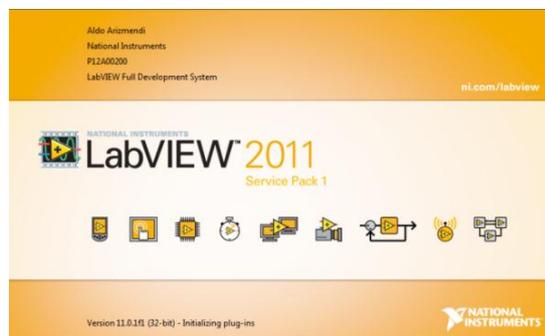


Figura 2. 27 LabVIEW 2011

Dentro de las librerías con las que cuenta LabVIEW está la de IMAQ Vision, con esta librería se puede tomar imágenes por medio de cámaras externas al sistema, también procesarlas dependiendo la utilización que se la vaya a dar, se puede convertirlas, cambiarlas, combinarlas, etc.

Esta es una herramienta muy útil que conjuntamente a todas las aplicaciones y funcionalidades con las que cuenta el software se puede realizar cualquier tipo de procesamiento de imágenes.

Luego de ejecutado el programa se creó un nuevo VI en el cual se va a realizar el filtrado de imágenes, para esto se utiliza las herramientas de Vision and Motion en el Diagrama de Bloques. En la figura 2.28 se observa las herramientas de las paletas de Vision and Motion.

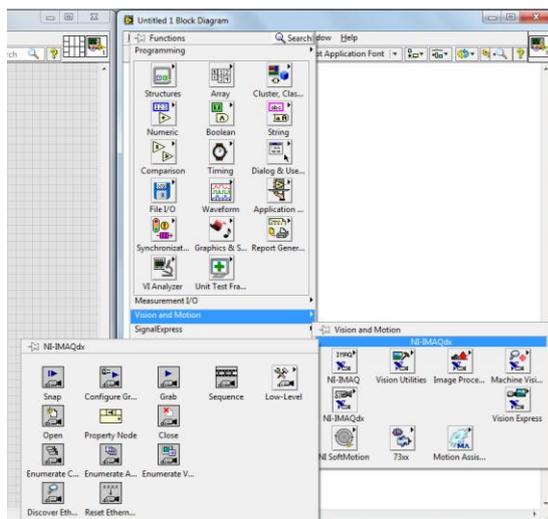


Figura 2. 28 Paletas de Vision and Motion

Antes de empezar a programar se abrió un lazo WHILE LOOP y con la paleta antes mencionada se procedió a realizar el esquema para poder visualizar cada una de las cámaras como se observa en la figura 2.29.

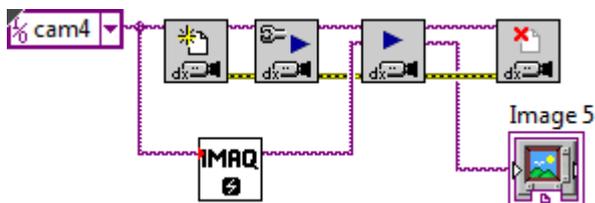


Figura 2. 29 Esquema para abrir una cámara

Luego de capturar la imagen se procedió a cambiarla a escala de grises para tener a la imagen en valores de 0 a 255 para poder manipularla de mejor manera, el diagrama para realizar este paso se observa en la figura 2.30.

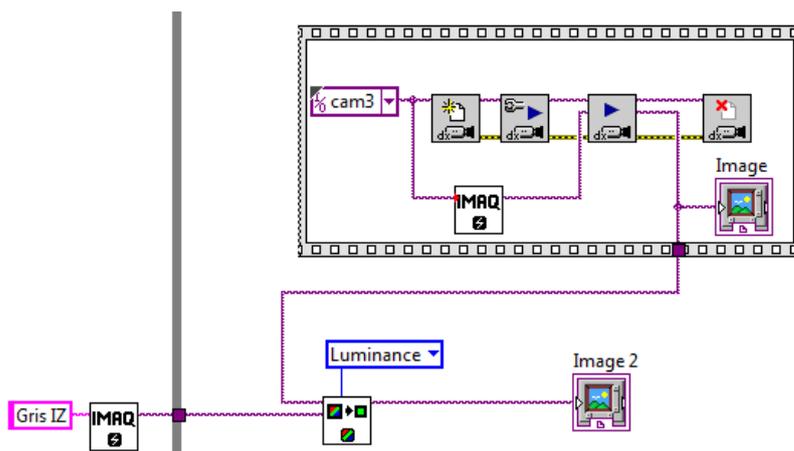


Figura 2. 30 Esquema para cambiar la imagen a escala de grises

Una vez que se realizó este paso se binarizó la imagen para tenerla en valores de 0 y 1, estos valores dependen de los rangos en que se va a colocar, para este caso los rangos son: si cada píxel tiene un valor entre 0 y 128, entonces coloque un 0 y si tiene un valor entre 129 y 255 entonces coloque un 1. El diagrama de binarización se muestra en la figura 2.31, el mismo que está adjunto al diagrama anterior.

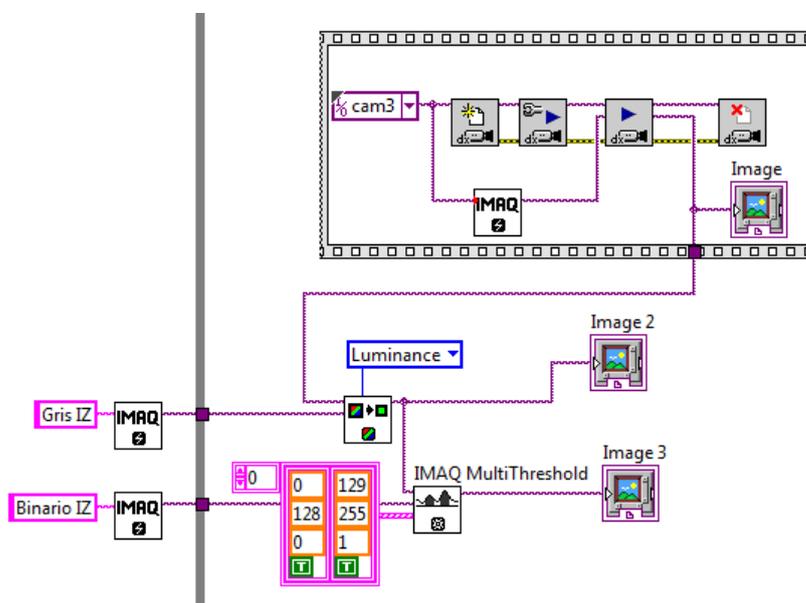


Figura 2. 31 Esquema para binarizar la imagen

Terminado este paso ahora se aplicó el filtro de Canny, el cual detecta los bordes de la imagen que ya paso por algunos procesos.

Después se guardó la imagen en una dirección específica dentro de la PC con su respectivo formato. Dichos procesos se adjuntan al anterior diagrama y se muestra en la figura 2.32.

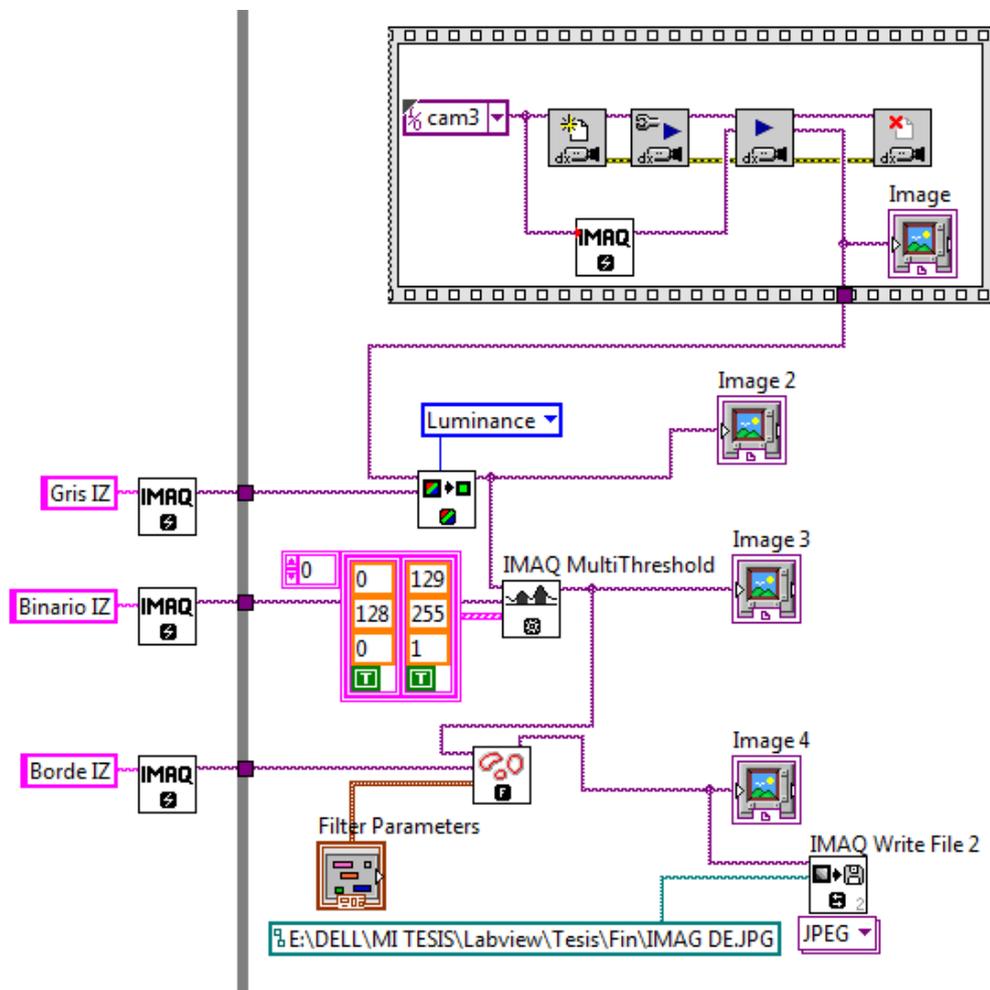


Figura 2. 32 Esquema de captura, filtrado y almacenamiento de una imagen

Con LabVIEW existió algunas complicaciones luego de que no se pudo manipular la información de cada imagen para realizar operaciones matemáticas y encontrar las coordenadas en el espacio y por ende la reconstrucción en tres dimensiones como estaba destinado en el proyecto, por lo cual se tuvo que buscar la manera de como poder llegar al objetivo de escanear un objeto con tan solo tomar imágenes.

Para esto se tuvo que utilizar el otro software seleccionado, el mismo que a más de procesar las imágenes también puede manipular la información que contiene cada una de ellas de manera fácil y rápida.

El otro software seleccionado es Matlab, ya que este por su capacidad de procesamiento de imágenes y de realizar cálculos matemáticos nos da una gran facilidad de crear nuevas imágenes no solo en dos dimensiones sino también en tres dimensiones.

Aquí también se puede crear entornos gráficos, los mismos que ayudan al operario durante el proceso de escaneado para tener buenos resultados.

A diferencia de LabVIEW, Matlab es un software que se basa en la programación por líneas, en donde el usuario tiene que ingresar el respectivo código para realizar cualquier tipo de operación.

Para iniciar la programación se procedió a instalar y ejecutar Matlab R2010a para 32 bits en la PC, el software ejecutado se muestra en la figura 2.33.

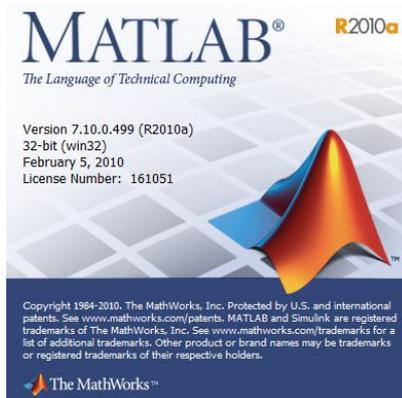


Figura 2. 33 Matlab R2010a

Para iniciar la programación se creó un archivo .m en el cual se escribió todas las líneas de programación del proyecto.

En la pantalla principal de Matlab se dio clic en la pestaña File luego en New y en Script o con el atajo Ctrl+N, dicho paso se observa en la figura 2.34.

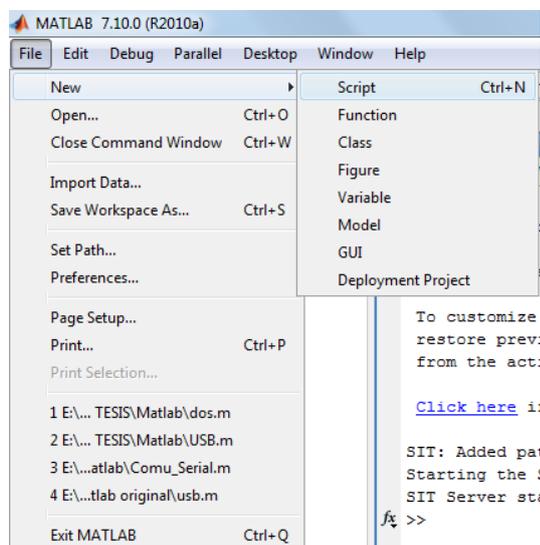


Figura 2. 34 Secuencia para crear un archivo .m

Una vez que se realizó el paso anterior, apareció una nueva ventana como se observa en la figura 2.35.

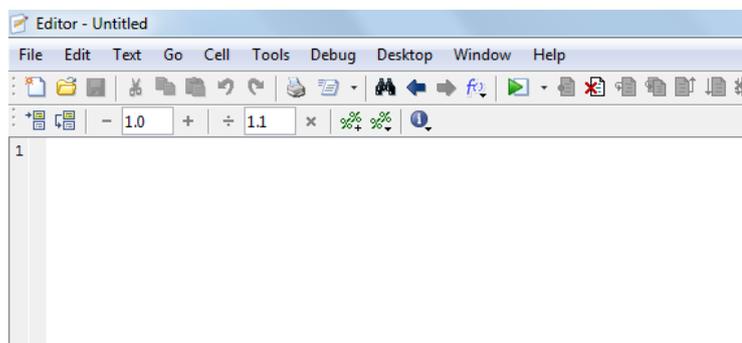


Figura 2. 35 Editor del archivo .m

A continuación se creó una pantalla de presentación del proyecto en la cual se colocó el nombre de la institución, el sello, etc. como se observa en la figura 2.36.

Para poder correrlo se da clic en el ícono  dentro del editor o aplastando F5 y de ahí se va configurando hasta obtener el objetivo deseado.



Figura 2. 36 Pantalla de presentación del proyecto

En dicha presentación se colocó un botón llamado CONTINUAR, para la siguiente parte del escaneado.

Ahora se creó una interfaz gráfica de usuario o GUIDE, la misma que ayuda a crear un entorno de programación visual para realizar y ejecutar programas que necesitan ingreso continuo de datos. Tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++.

Para realizar la interfaz existen dos maneras:

- La primera es dar clic en el ícono  dentro de la pantalla principal de Matlab como se muestra en la figura 2.37.

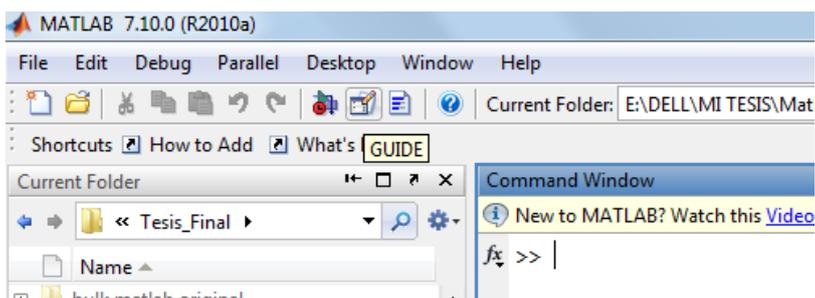


Figura 2. 37 Icono de GUI

- La segunda es escribir la palabra `>> guide` en la ventana de comandos.

Una vez que se realizó la ejecución del GUIDE apareció una ventana como se muestra en la figura 2.38.

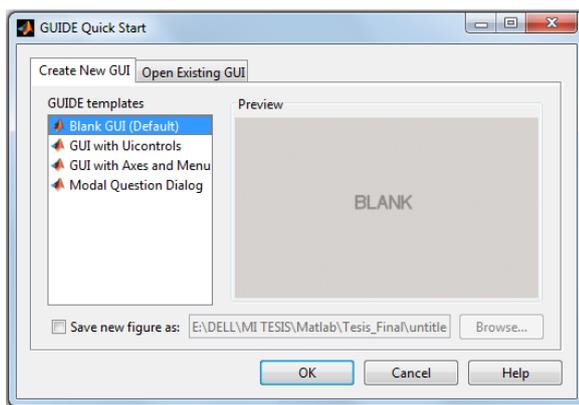


Figura 2. 38 Ventana de inicio de GUI

En esa ventana se escogió la opción de Blank GUI y se dio clic en OK para crear una interfaz en blanco.

Una vez creada la nueva interfaz apareció la ventana que se muestra en la figura 2.39.

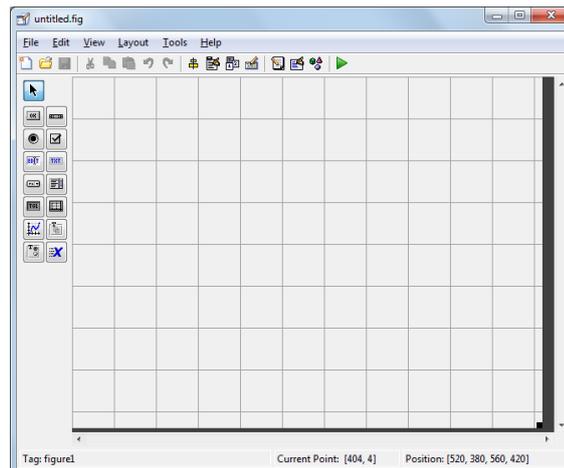


Figura 2. 39 Entorno de diseño de GUI

Esta interfaz se la guardó con el nombre de Programa, dicho archivo se crea en una extensión .fig, la cual es llamada en la pantalla de Presentación, en esta pantalla se insertó los Push Button para modificarlos cada uno.

Para sacar los Push Button se buscó el ícono en la paleta izquierda y se le arrastró en cualquier parte de la ventana ya abierta como se muestra en la figura 2.40.

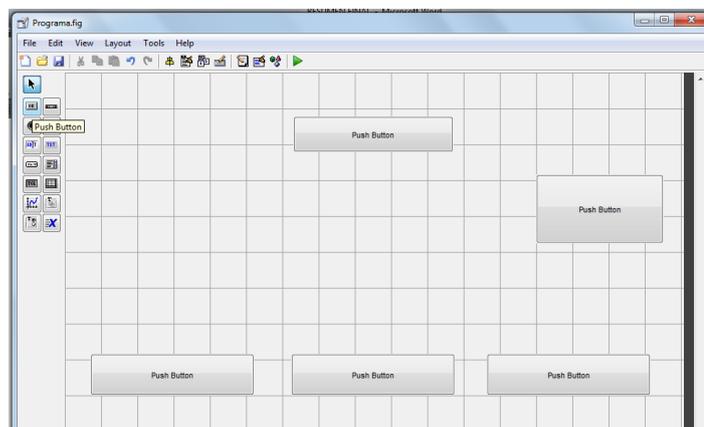


Figura 2. 40 Insertar Push Button

Para poder modificar cada uno de los botones se dio doble clic izquierdo en el Push Button y se desplegó una nueva ventana en la cual se modificó los campos String y Tag.

En el campo String se escribió la palabra que se visualizará en el botón, mientras que el campo Tag se colocó el nombre con el que se va a conocer nuestro botón para poder utilizarlo dentro del código, en este campo el nombre va sin espacios o puesto un guion bajo entre las palabras. Además se puede modificar el tamaño de la letra para una mejor visualización en la opción FontSize.

De ahí si se desea cambiar alguna otra característica pues se va cambiando cada uno de los campos que se encuentran en la ventana de configuración de cada Push.

En la figura 2.41 se observa como modificar el campo String.

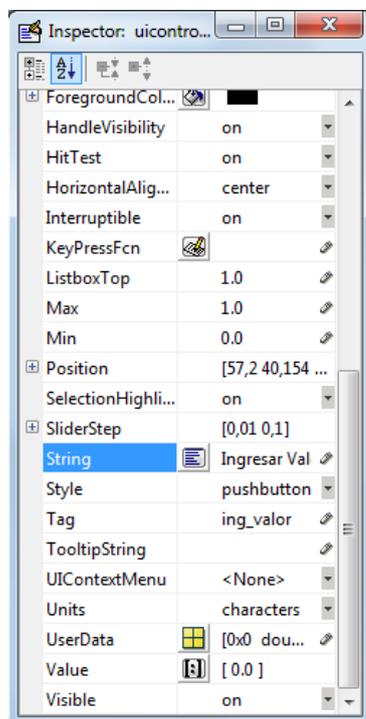


Figura 2. 41 Edición del campo String

En la figura 2.42 se observa como modificar el campo Tag.

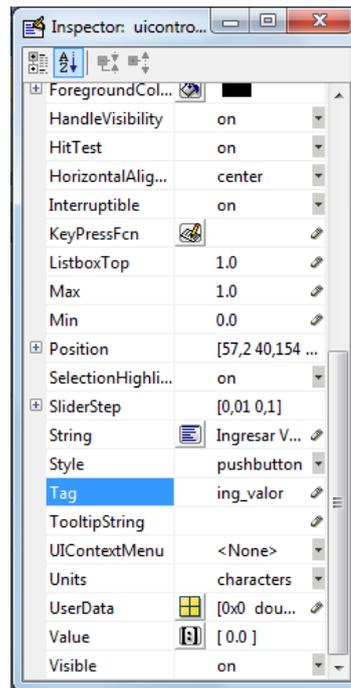


Figura 2. 42 Edición del campo Tag

En la figura 2.43 se observa como modificar el campo FontSize.

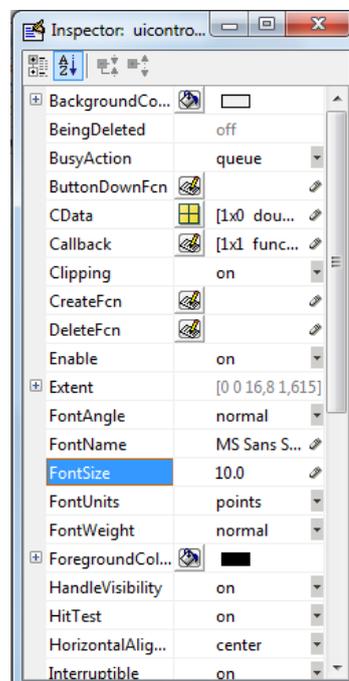


Figura 2. 43 Edición del tamaño de la letra

En la figura 2.44 se observa los Push Button ya modificados.

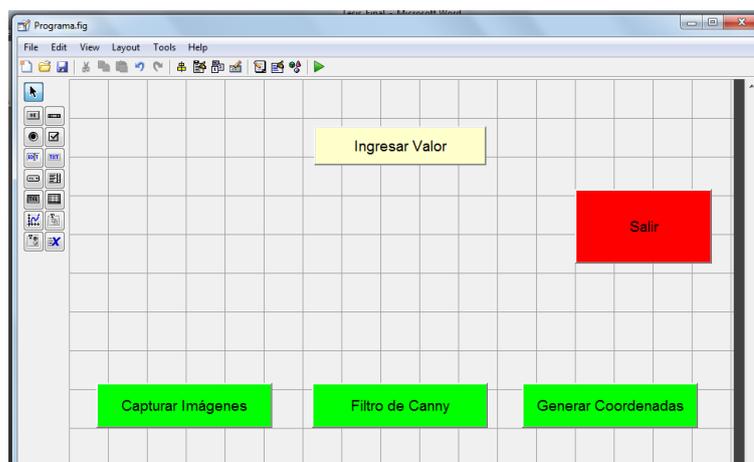


Figura 2. 44 Botones modificados

Este proceso se realizó para los demás botones que se encuentran en la interfaz gráfica de usuario.

A continuación se procedió a crear campos estáticos de texto, de igual manera se buscó el ícono en la paleta izquierda y se le arrastró en cualquier parte de la ventana ya abierta como se muestra en la figura 2.45.

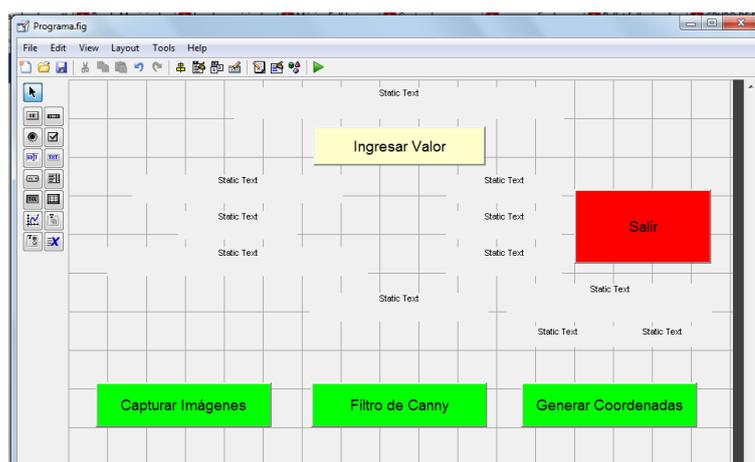


Figura 2. 45 Insertar campos estáticos de texto

De igual forma que los Push Button se dio doble clic izquierdo en los campos de edición de texto para editarlos y utilizarlos en el código.

A excepción de ciertos campos de edición de texto no se los tiene que modificar de ahí simplemente se realiza en los que realmente sea necesario, la edición se realizó en los campos String y no los campos Tag debido a que solo queremos visualizar una frase más no utilizar el botón dentro de la programación.

En los cuadros restantes se modificó solo los campos Tag y se escribió numero_fotos, angulo_giro y conteo_fotos entre otros, además se puede cambiar de colores a cada cuadro de texto y Push Button como se observa en la figura 2.46.

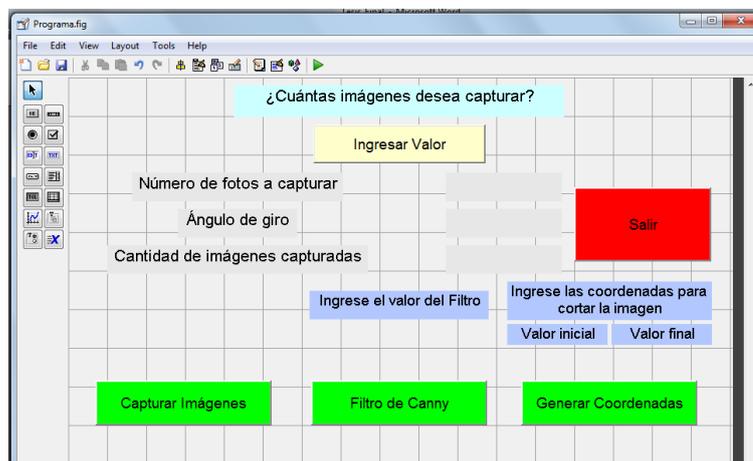


Figura 2. 46 Campos estáticos de texto modificados

A continuación para ingresar datos se colocó campos de edición de texto en los cuales se puede ingresar valores tanto numéricos como alfabéticos como se muestra en la figura 2.47.

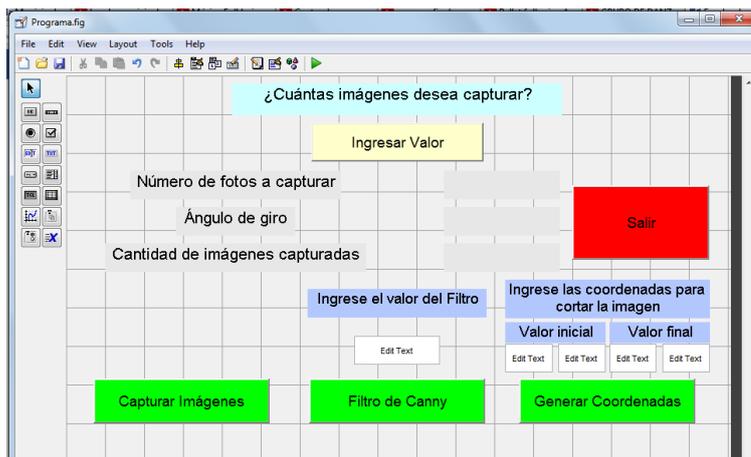


Figura 2. 47 Insertar campos de edición de texto

De igual manera para modificar se dio doble clic izquierdo en cada bloque para cambiar la letra, el color, el campo String se va a dejar vacío y en el campo Tag se colocan las diferentes palabras con las cuales se va a utilizar dentro de la programación como se muestra en la figura 2.48.

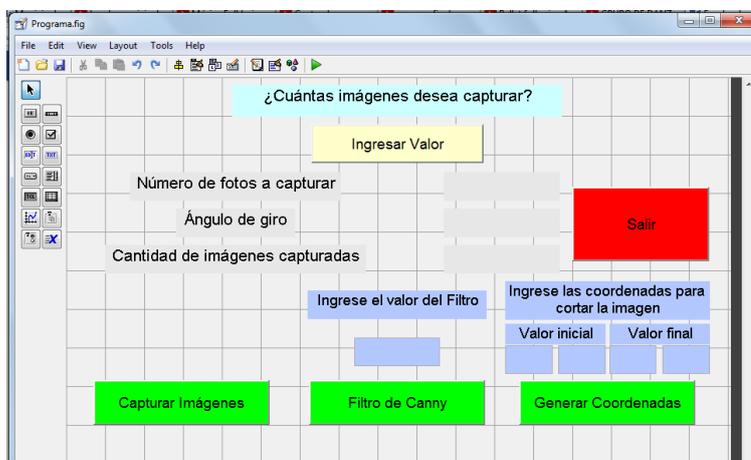


Figura 2. 48 Campos de edición de texto modificados

Ahora se colocó una imagen del sello de la carrera para lo cual se coloca un axes en la parte posterior de todos los botones como se muestra en la figura 2.49.

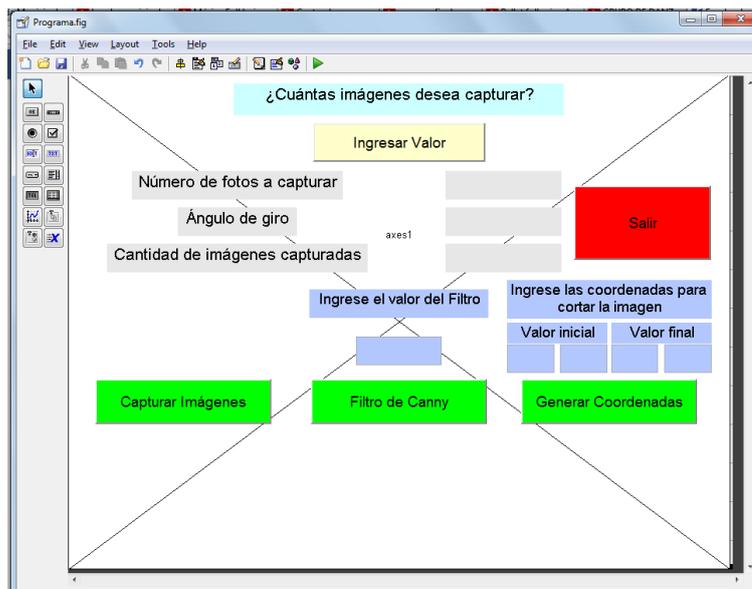


Figura 2. 49 Axes insertado en la interfaz gráfica

Para lograr la visualización de las imágenes capturadas se colocó un axes en la parte inferior del botón Capturar Imágenes, otro debajo del botón Filtro de Canny y uno más debajo del botón de Generar Coordenadas. Este paso se observa en la figura 2.50.

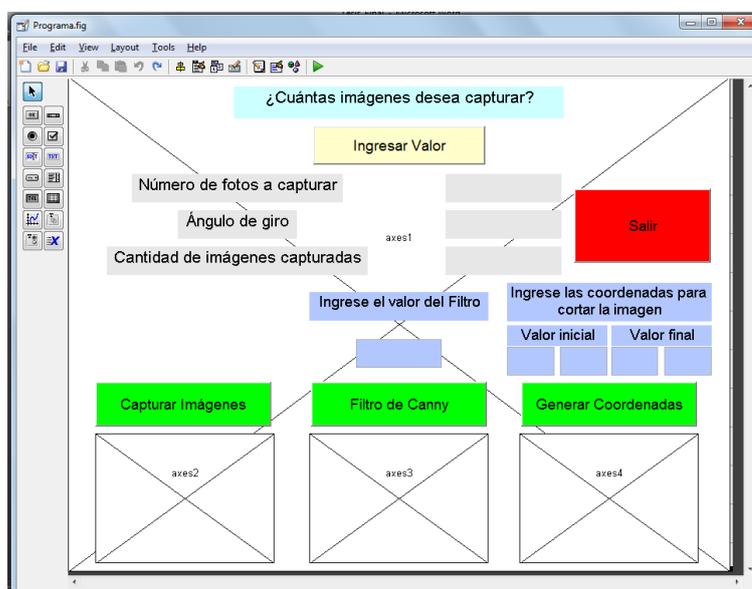


Figura 2. 50 Axes 2, 3 y 4 colocados para visualización de las imágenes capturadas, imágenes filtradas y generación de coordenadas

Después de haber colocado todos los push button, cuadros de textos y los axes respectivos se va a editó la distancia entre los mismos, además se añadió y se eliminó botones conforme sea la necesidad, para verificar como va quedando se aplasta F5.

Para realizar la programación de cada Push Button se dio clic derecho en cada uno, luego en View Callback y finalmente en Callback como se muestra en la figura 2.51.

Con esto aparecerá el cuadro de edición de Matlab con lo que a su vez se creó un archivo .m correspondiente al GUIDE creado.

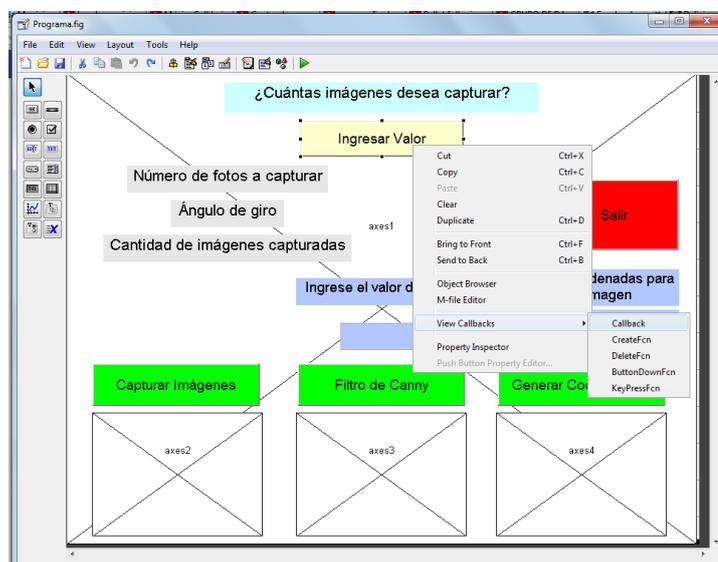


Figura 2. 51 Llamada del archivo .m llamado “Programa”

Al momento de realizar este paso se comenzó a programar en nuestro editor cada botón y cada cuadro de texto que están vacíos y cada uno de los axes para mostrar las diferentes imágenes. El editor de Matlab que saldrá se observa en la figura 2.52.

```

1 function varargout = Secuencia(varargin)
2
3
4 gui_Singleton = 1;
5 gui_State = struct('gui_Name',mfilename,'gui_Singleton',gui_Singleton,...
6                 'gui_OpeningFcn',@Secuencia_OpeningFcn,...
7                 'gui_OutputFcn',@Secuencia_OutputFcn,...
8                 'gui_LayoutFcn',[],'gui_Callback',[]);
9
10 if nargin && ischar(varargin{1})
11     gui_State.gui_Callback = str2func(varargin{1});
12 end
13
14 if nargin
15     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
16 else
17     gui_mainfcn(gui_State, varargin{:});
18 end
19
20 %Abrir imagen de fondo
21 function Secuencia_OpeningFcn(hObject, eventdata, handles, varargin)
22
23 scrsz = get(0, 'ScreenSize');
24 pos_act = get(gcf, 'Position');
25 xr = scrsz(3) - pos_act(3);
26 xp = round(xr/2);
27 yr = scrsz(4) - pos_act(4);
28 yp = round(yr/2);
29 set(gcf, 'Position', [xp yp pos_act(3) pos_act(4)]);
30
31 clc
32 imagfon = imread('Logo_Amarillo.jpg'); %Leer la imagen
33 axes(handles.axes1) %Ubicación donde se va a mostrar la imagen
34 image(imagfon) %Mostrar la imagen antes leída
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura 2. 52 Editor de Matlab para colocar las líneas de programación

Una vez realizada la programación se procedió a ejecutar el programa y se fue verificando si está correctamente hecha la programación, caso contrario se la corrigió, el programa ya ejecutado con los Push Button, Static Text y Axes modificados se muestra en la figura 2.53.

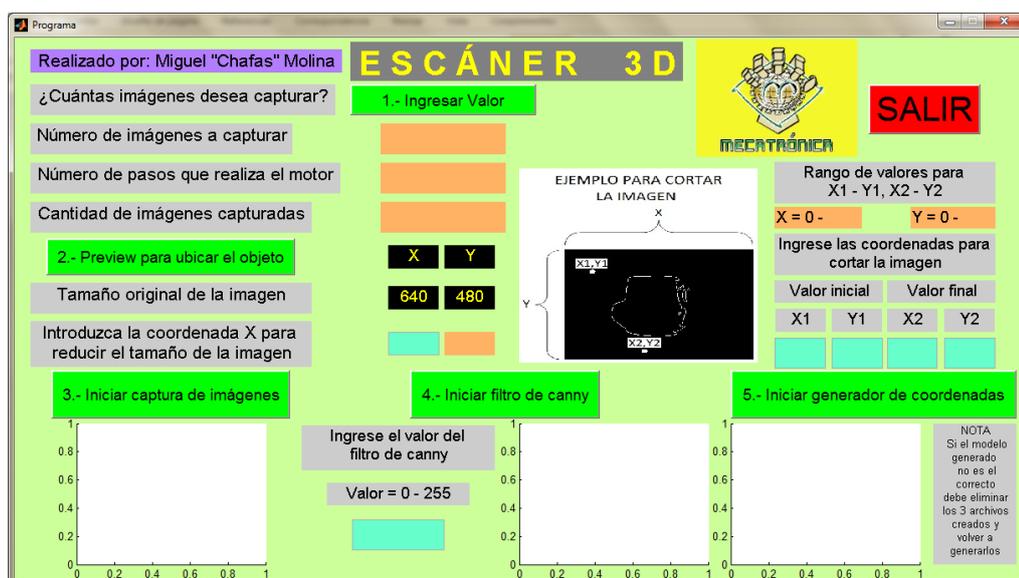


Figura 2. 53 Pantalla lista para realizar las pruebas necesarias

Una vez finalizado esta etapa se procedió a crear otra interfaz gráfica de usuario a la cual se la llamó selector_datos la misma que será llamada cuando demos clic en el botón de Ingresar Valor en la anterior GUIDE, en dicha interfaz se creó un menú con los valores de la cantidad de fotos que va a capturar nuestro sistema.

La opción para realizar este proceso se llama Pop-up Menu, además se añadió una imagen de fondo y un cuadro de edición de texto, la nueva interfaz se muestra en la figura 2.54.

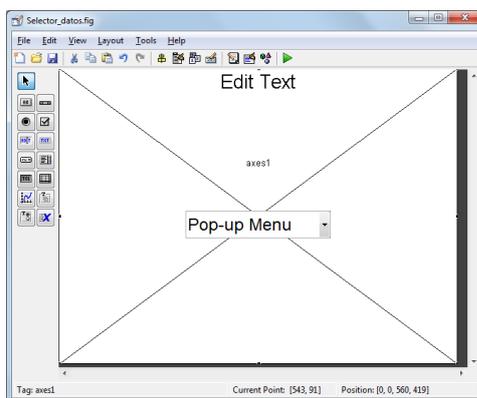


Figura 2. 54 Interfaz gráfica de usuario llamada Selector_datos

De igual manera se programó solo la parte String en el campo de edición de texto y el campo Tag no se hace nada, mientras que en el Pop-up Menu se editó los dos campos, en el String se puso las opciones que se utilizó para el procesamiento de imágenes, dicha pantalla ya modificada se la observa en la figura 2.55.



Figura 2. 55 Interfaz configurada para el proceso

En el Anexo E se encuentra toda la programación de las pantallas creadas en Matlab.

Una vez culminado con la adquisición de imágenes, se realizó el diseño de la estructura donde se van los elementos

2.5. Diseño de la estructura mecánica para el montaje de los elementos

Para realizar el diseño de la estructura se utilizó un software en el cual se pueda tener una visualización en tres dimensiones de todos los elementos del proyecto, el seleccionado fue SolidWorks versión 2014.

Para que el motor pueda girar se diseñó una base y a su vez un plato donde va el objeto como se observa en las figuras 2.56 y 2.57.

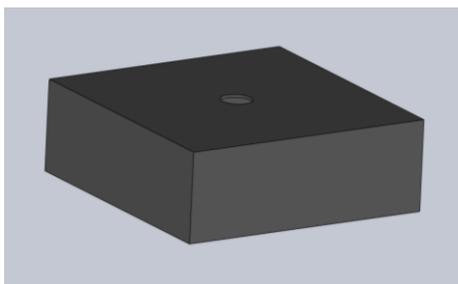


Figura 2. 56 Base para colocar el motor a pasos

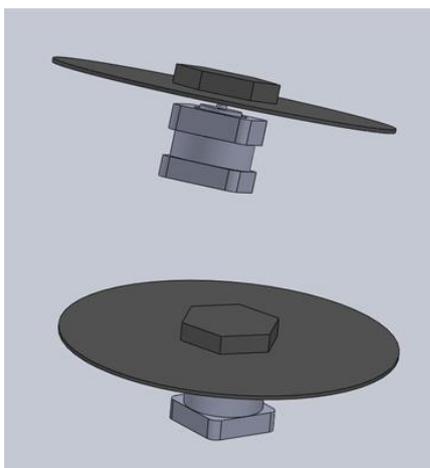


Figura 2. 57 Plato en el cual se va a colocar el objeto a escanear

Una vez construida la caja y el plato se procedió a acoplar los dos elementos para montarlos a la estructura principal como se muestra en la figura 2.58.

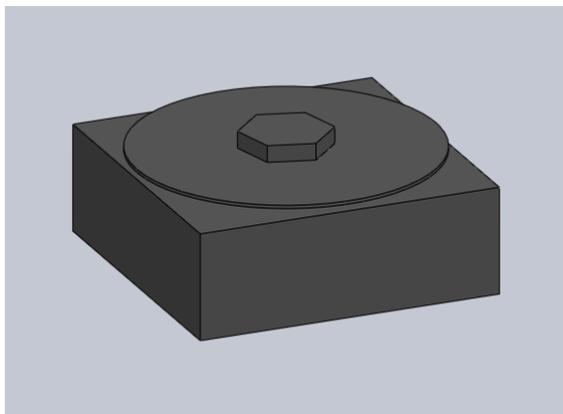


Figura 2. 58 Motor acoplado a la caja y al plato giratorio

A continuación se procedió a diseñar un soporte para la webcam, el cual tiene una forma de C, esto ayuda a tener diferentes posiciones en donde ubicar la webcam dependiendo el tamaño del objeto a escanear, dicho sistema se encuentra en la figura 2.59.

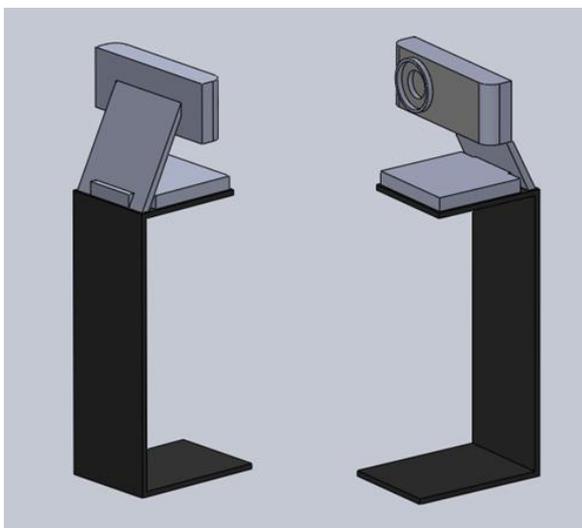


Figura 2. 59 Platina en forma de C de soporte de la webcam

A continuación se realizó el diseño de la caja en la cual va a ir colocados todos los elementos antes ya creados como se muestra en la figura 2.60.

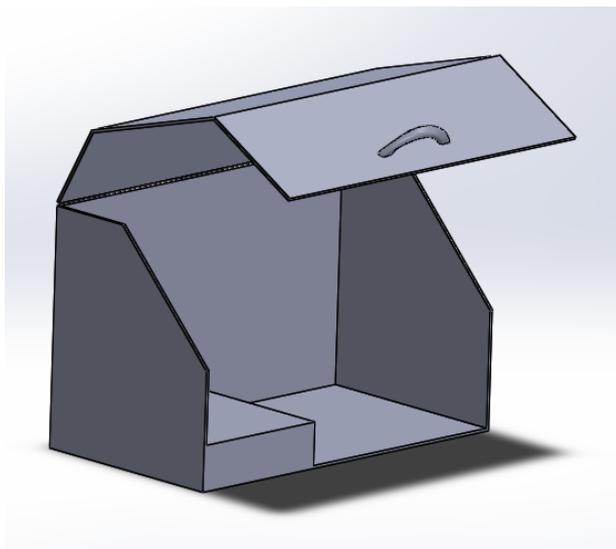


Figura 2. 60 Caja para colocar los elementos creados

Una vez concluido este paso se creó el ensamblaje de la caja con todos los elementos anteriores como se muestra en la figura 2.61.

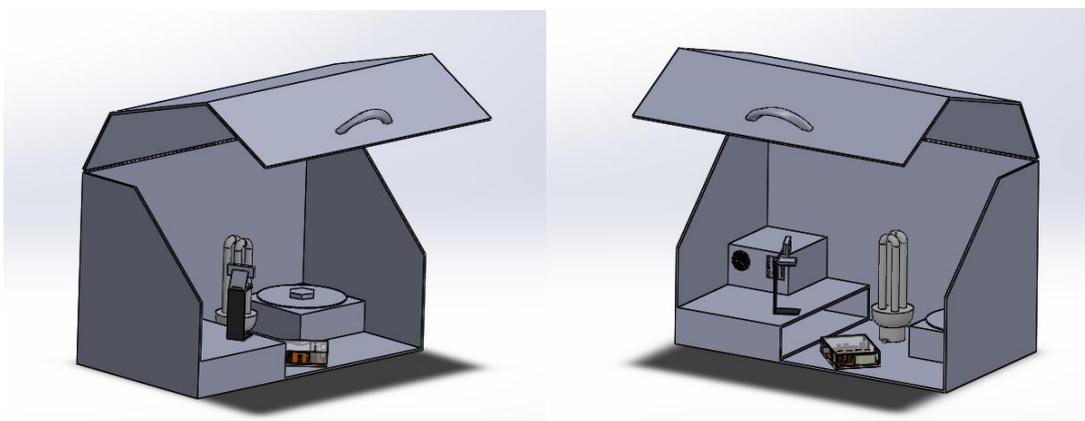


Figura 2. 61 Ensamblaje final

Finalmente se añadió los colores a la estructura con sus elementos como se muestra en la figura 2.62.

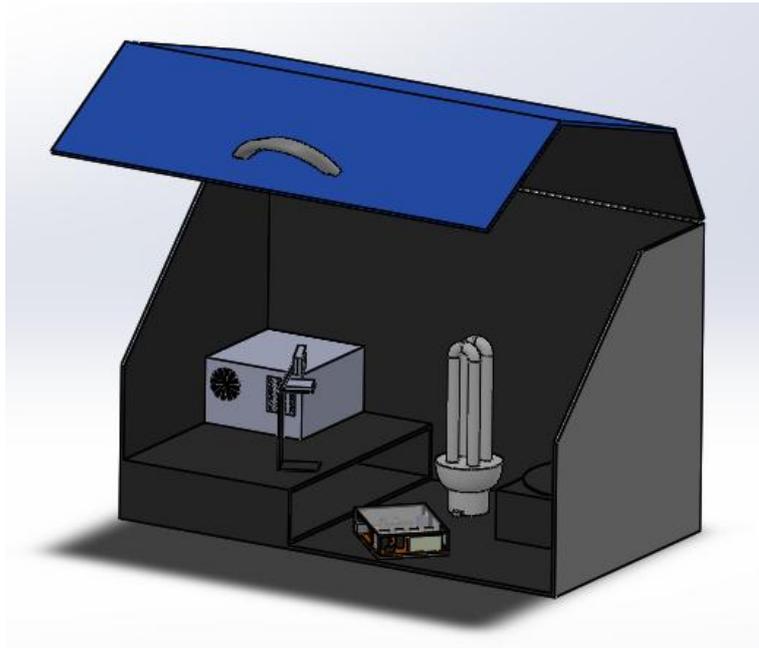


Figura 2. 62 Ensamblaje final con todos sus elementos y pintado

CAPÍTULO III

PRUEBAS Y RESULTADOS

3.1. Pruebas

Con el dispositivo ya construido se procedió a realizar las pruebas pertinentes sobre el escaneado que es su única función, al momento de terminar con la captura de las imágenes, el proceso de filtrado y la obtención de las coordenadas en X, Y e Z, el programa generó tres archivos, los dos primeros en extensión .obj y el tercero en extensión .txt, con los primeros creados se unió ambos y se abrió en un software que pueda leer mallas y se pudo visualizar el objeto, con el otro archivo en extensión .txt se procedió a utilizarlo en un software que trabaja con nube de puntos, en este caso se utilizó SolidWorks versión 2014 y se realizó el respectivo procesamiento para convertir dicha nube en una superficie, luego la superficie en sólido y realizar una prueba de impresión en tres dimensiones gracias a la máquina con la que cuenta el Laboratorio CNC, además se realizó una prueba de escaneo con el dispositivo que posee la universidad y así se comparó los resultados con el escáner fabricado.

3.1.1. Prueba de Escaneado

Para verificar que el escáner funcione de manera correcta fue necesario utilizar algunos objetos de diferentes formas, tamaños, peso y material para determinar las restricciones del sistema.

Los objetos utilizados fueron los siguientes:

- Una botella de madera
- Un frasco de cerámica con detalles de una cara de conejo
- Un florero de cerámica en forma de corazón y con pintura brillante
- Un frasco de plástico
- Una taza de café de cerámica con pintura brillante

Con las pruebas realizadas se obtuvo los datos que se muestran en la tabla 3.1 y cuadro estadístico de la figura 3.1 en cuanto a funcionalidad se refiere.

Tabla 3. 1

Valores de funcionalidad del escáner

# de repeticiones del programa funcionando	% de Funcionalidad
1	10
2	40
3	75
4	80
5	85
6	90

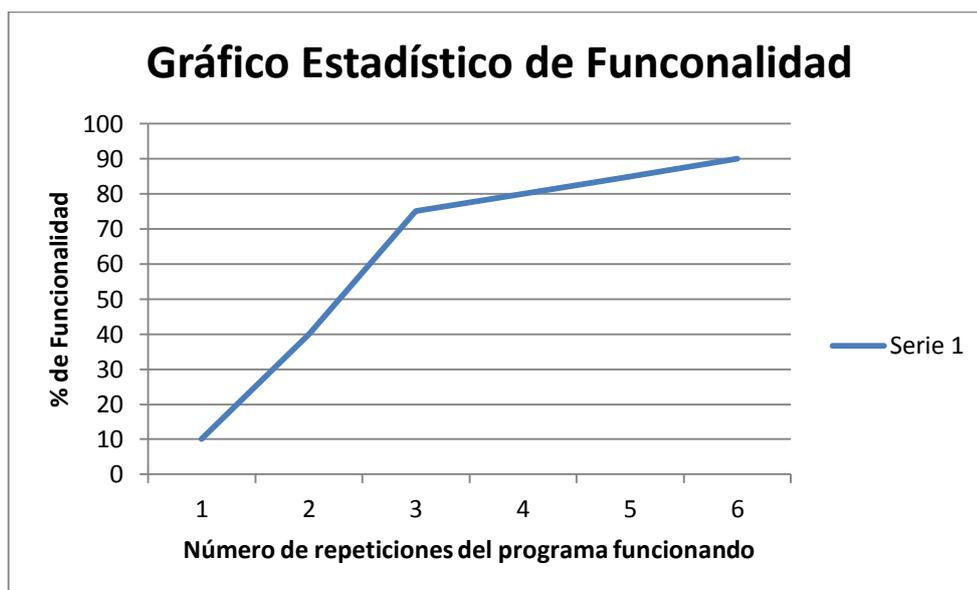


Figura 3. 1 Resultados estadísticos de la funcionalidad del escáner

Con los resultados obtenidos se determinó que el sistema necesita de al menos tres ejecuciones para que se estabilice y pueda girar una vuelta completa y obtener un escaneado completo y continuo.

Los escaneados de los cinco objetos se muestran a continuación:

Prueba 1: Botella de madera

En la figura 3.2 se observa en el lado izquierdo la imagen que la webcam capturó de la botella, mientras que en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.

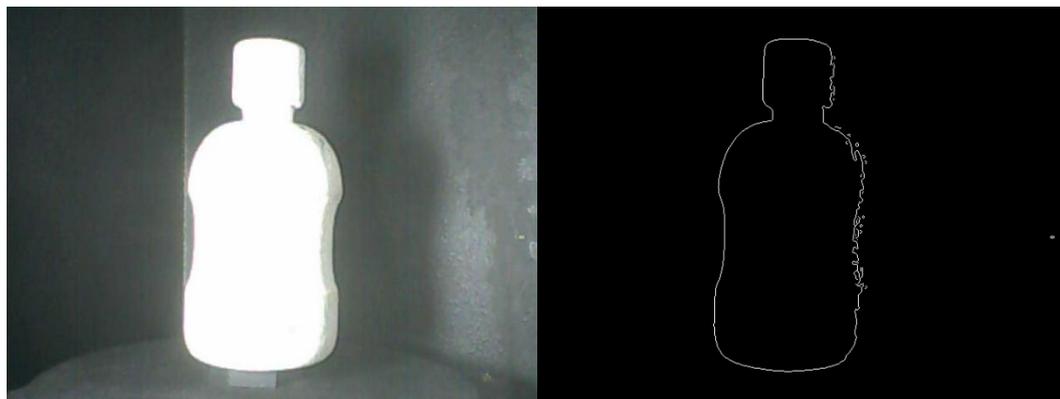


Figura 3. 2 Imagen original de la botella y aplicada el filtro de Canny al inicio del proceso

En la figura 3.3 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

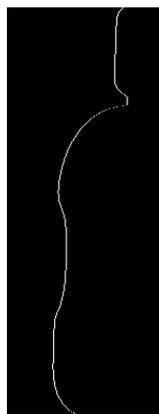


Figura 3. 3 Imagen recortada de la botella para obtener las coordenadas x, y, z

En la figura 3.4 se observa en el lado izquierdo la imagen que la webcam obtuvo pero cuando el objeto ya ha girado un cierto número de grados y en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.

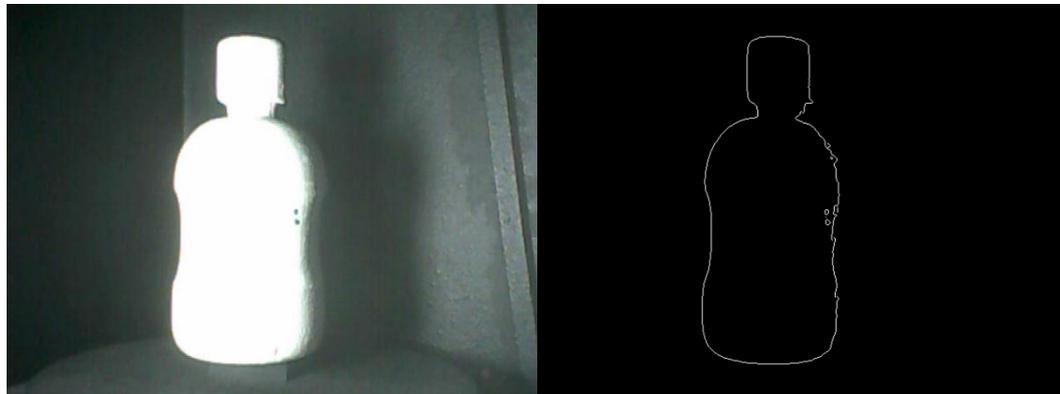


Figura 3. 4 Imagen original de la botella y aplicada el filtro de Canny durante el proceso

En la figura 3.5 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

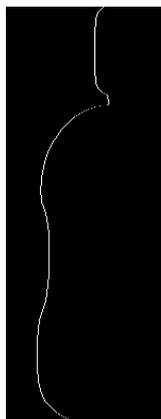


Figura 3. 5 Imagen recortada de la botella para obtener las coordenadas x, y, z

Prueba 2: Frasco de cerámica con detalles de una cara de conejo

En la figura 3.6 se observa en el lado izquierdo la imagen que la webcam capturó del frasco de cerámica, mientras que en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.

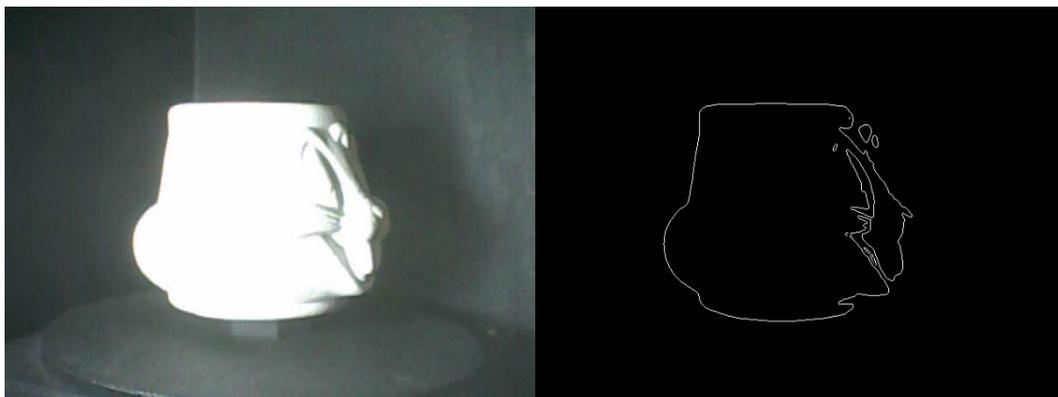


Figura 3. 6 Imagen original del frasco de cerámica y aplicada el filtro de Canny al inicio del proceso

En la figura 3.7 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

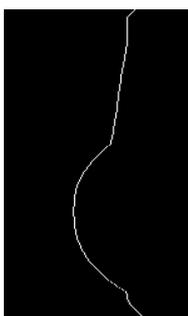


Figura 3. 7 Imagen recortada del frasco de cerámica para obtener las coordenadas x, y, z

En la figura 3.8 se observa en el lado izquierdo la imagen que la webcam obtuvo pero cuando el objeto ya ha girado un cierto número de grados y en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.

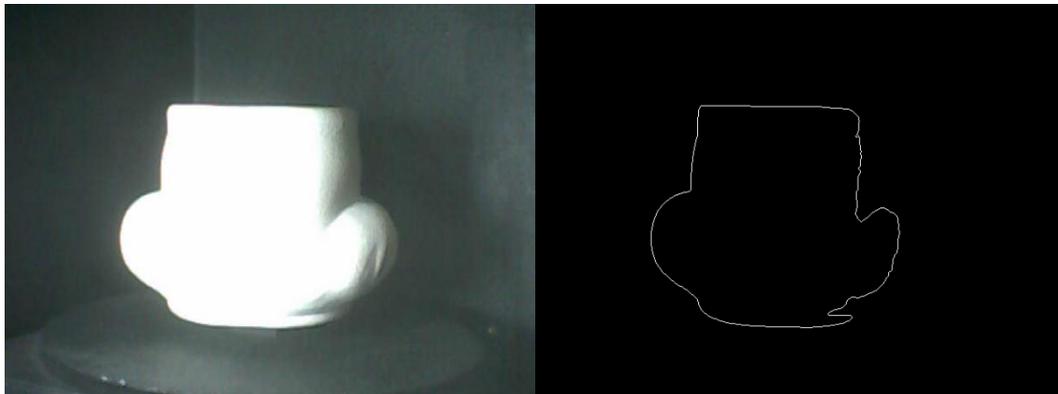


Figura 3. 8 Imagen original del frasco de cerámica y aplicada el filtro de Canny durante el proceso

En la figura 3.9 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

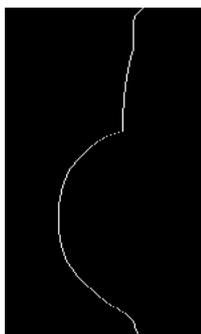


Figura 3. 9 Imagen recortada del frasco de cerámica para obtener las coordenadas x, y, z

Prueba 3: Florero de cerámica en forma de corazón y con pintura brillante

En la figura 3.10 se observa en el lado izquierdo la imagen que la webcam capturó del florero, mientras que en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.

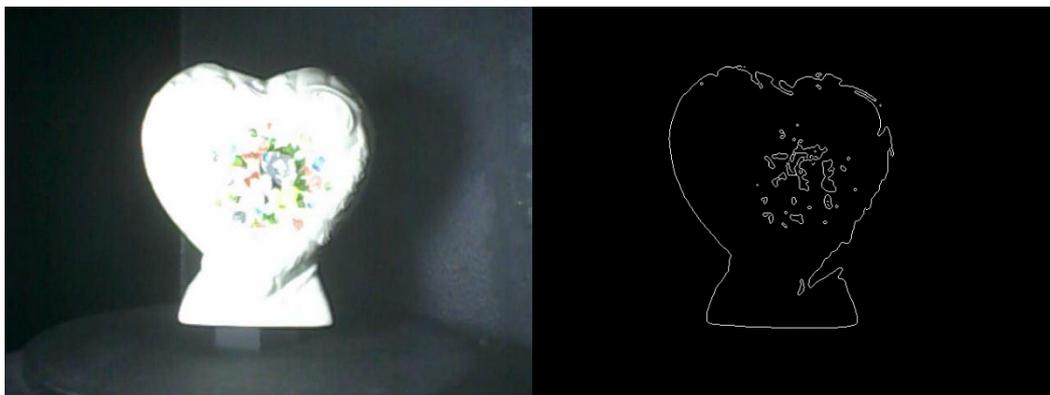


Figura 3. 10 Imagen original del florero y aplicada el filtro de Canny al inicio del proceso

En la figura 3.11 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.



Figura 3. 11 Imagen recortada del florero para obtener las coordenadas x, y, z

En la figura 3.12 se observa en el lado izquierdo la imagen que la webcam obtuvo pero cuando el objeto ya ha girado un cierto número de grados y en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.

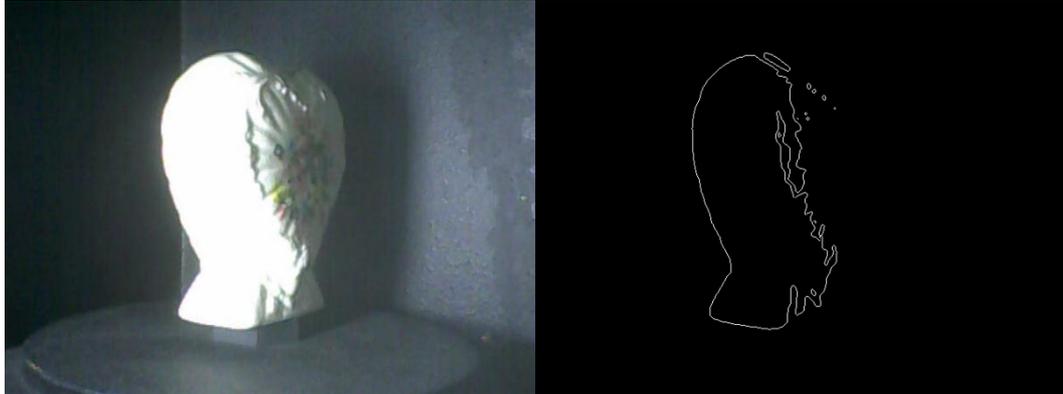


Figura 3. 12 Imagen original del florero y aplicada el filtro de Canny durante el proceso

En la figura 3.13 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

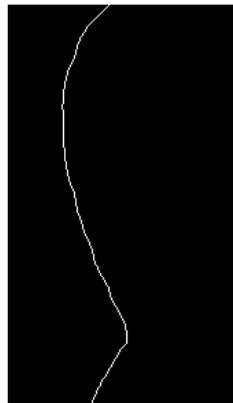


Figura 3. 13 Imagen recortada del florero para obtener las coordenadas x, y, z

Prueba 4: Frasco de plástico

En la figura 3.14 se observa en el lado izquierdo la imagen que la webcam capturó del frasco de plástico, mientras que en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.



Figura 3. 14 Imagen original del frasco de plástico y aplicada el filtro de Canny al inicio del proceso

En la figura 3.15 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

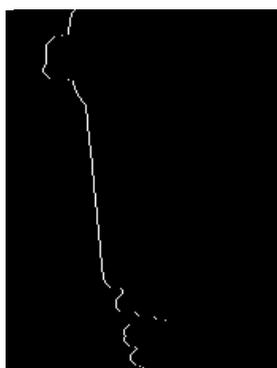


Figura 3. 15 Imagen recortada del frasco de plástico para obtener las coordenadas x, y, z

En la figura 3.16 se observa en el lado izquierdo la imagen que la webcam obtuvo pero cuando el objeto ya ha girado un cierto número de grados y en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.



Figura 3. 16 Imagen original del frasco de plástico y aplicada el filtro de Canny durante el proceso

En la figura 3.17 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

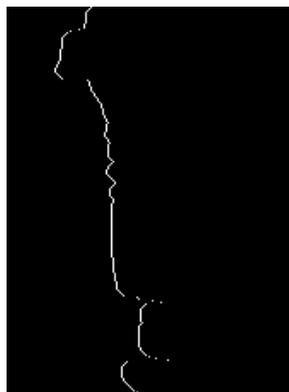


Figura 3. 17 Imagen recortada del frasco de plástico para obtener las coordenadas x, y, z

Prueba 5: Taza de café de cerámica con pintura brillante

En la figura 3.18 se observa en el lado izquierdo la imagen que la webcam capturó de la taza de café, mientras que en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.



Figura 3. 18 Imagen original de la taza y aplicada el filtro de Canny al inicio del proceso

En la figura 3.19 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

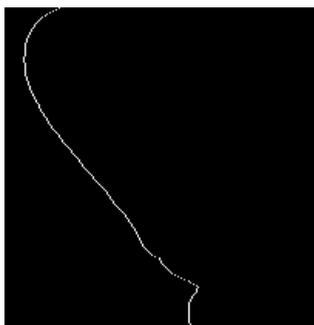


Figura 3. 19 Imagen recortada de la taza para obtener las coordenadas x, y, z

En la figura 3.20 se observa en el lado izquierdo la imagen que la webcam obtuvo pero cuando el objeto ya ha girado un cierto número de grados y en la parte derecha se observa la misma imagen pero aplicado el filtro de Canny.



Figura 3. 20 Imagen original de la taza y aplicada el filtro de Canny durante el proceso

En la figura 3.21 se observa la imagen que está aplicada el filtro de Canny recortada simplemente con el área donde se encuentran los puntos que forman el contorno del objeto a escanear.

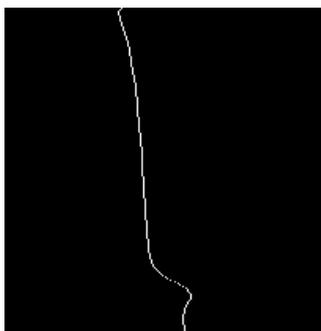


Figura 3. 21 Imagen recortada de la taza para obtener las coordenadas x, y, z

Con las fotos capturadas, procesadas y guardadas con la información de las coordenadas de los puntos en él espacio, se tuvo que buscar un software que pueda abrir archivos .obj, el seleccionado para realizar este proceso fue el programa llamado David Laserscanner.

A continuación se muestra los resultados de los cinco objetos escaneados.

En la figura 3.22 se observa la imagen en tres dimensiones de la botella.

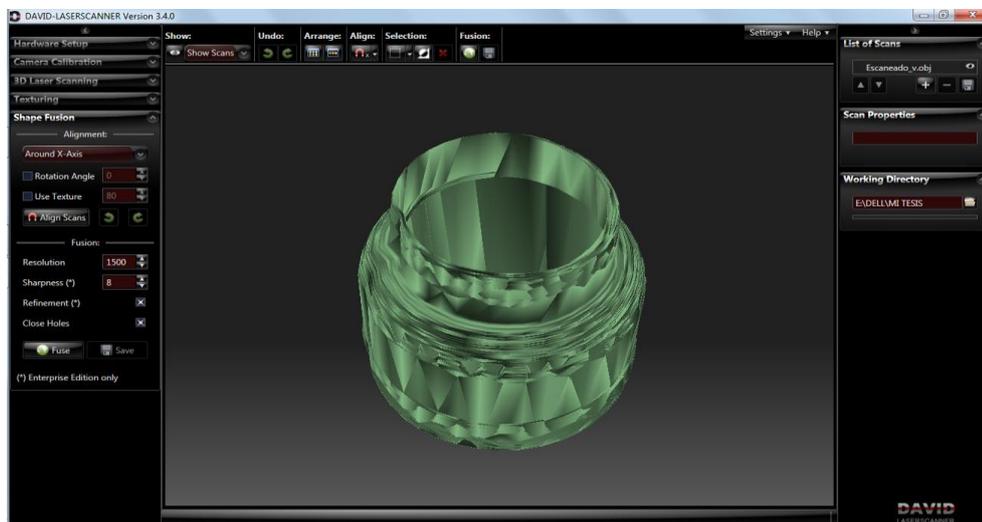


Figura 3. 22 Botella escaneada

En la figura 3.23 se observa la imagen en tres dimensiones del frasco con detalles de una cara de conejo.

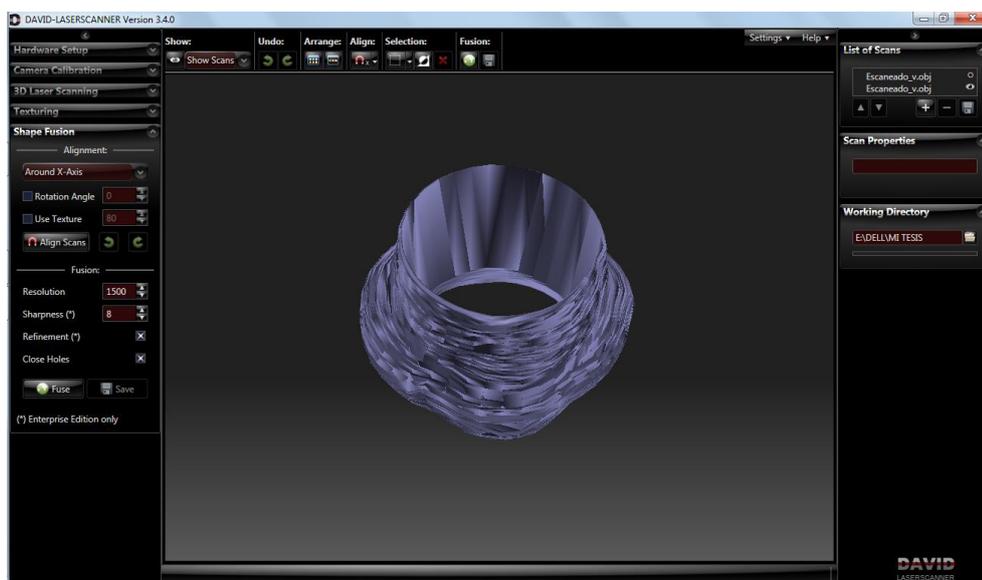


Figura 3. 23 Frasco con detalles de una cara de conejo escaneado

En la figura 3.24 se observa la imagen en tres dimensiones del florero en forma de corazón.

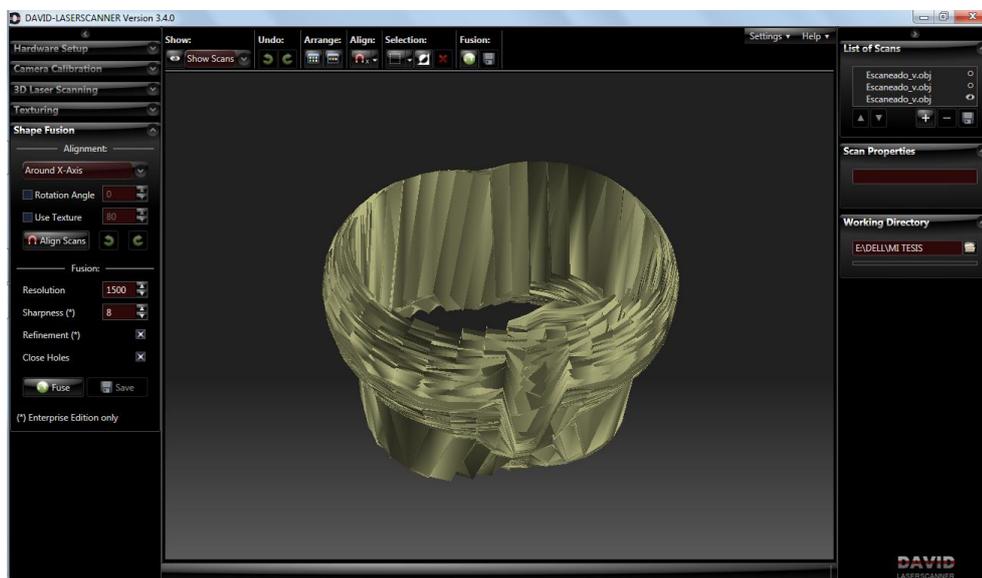


Figura 3. 24 Florero en forma de corazón escaneado

En la figura 3.25 se observa la imagen en tres dimensiones del frasco de plástico.

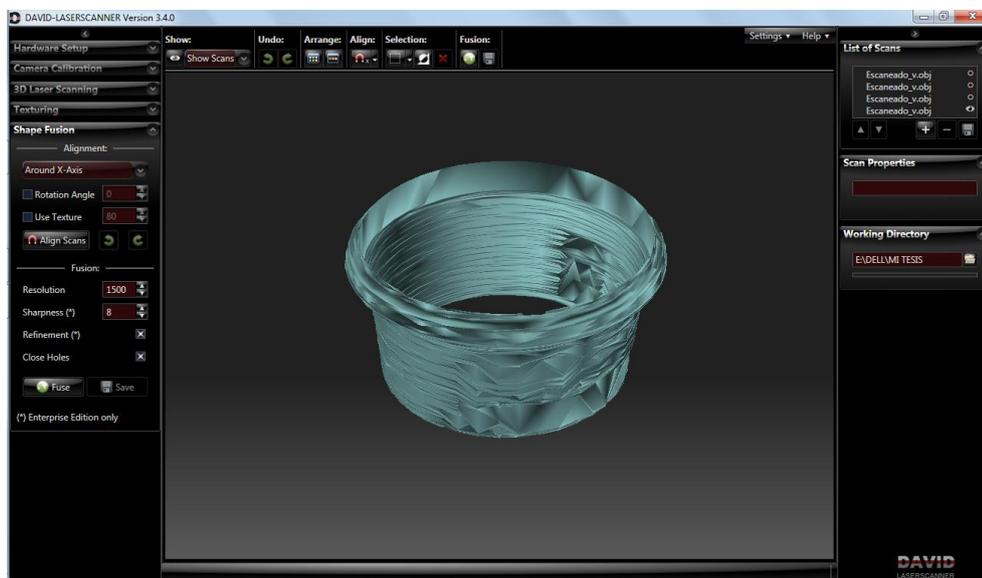


Figura 3. 25 Frasco de plástico escaneado

En la figura 3.26 se observa la imagen en tres dimensiones de la taza de café.

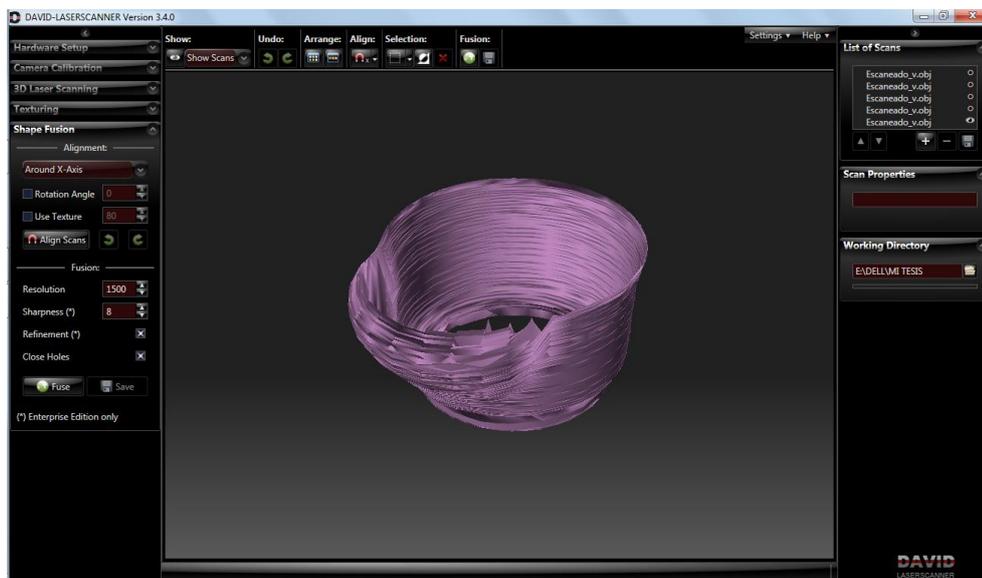


Figura 3. 26 Taza de café escaneada

Con el segundo documento que nos creó el dispositivo en extensión .txt se procedió a abrirlo en SolidWorks para realizar el proceso de creación de una superficie a partir de una nube de puntos.

Para esto primero se abrió dicho archivo en el software para ver la información que contiene el mismo, como muestra se tomó el frasco de cerámica con cara de conejo debido a que tiene mayor número de detalles, en el archivo se encontró la nube de puntos que se generó en el programa de Matlab y que se muestra en la figura 3.27.

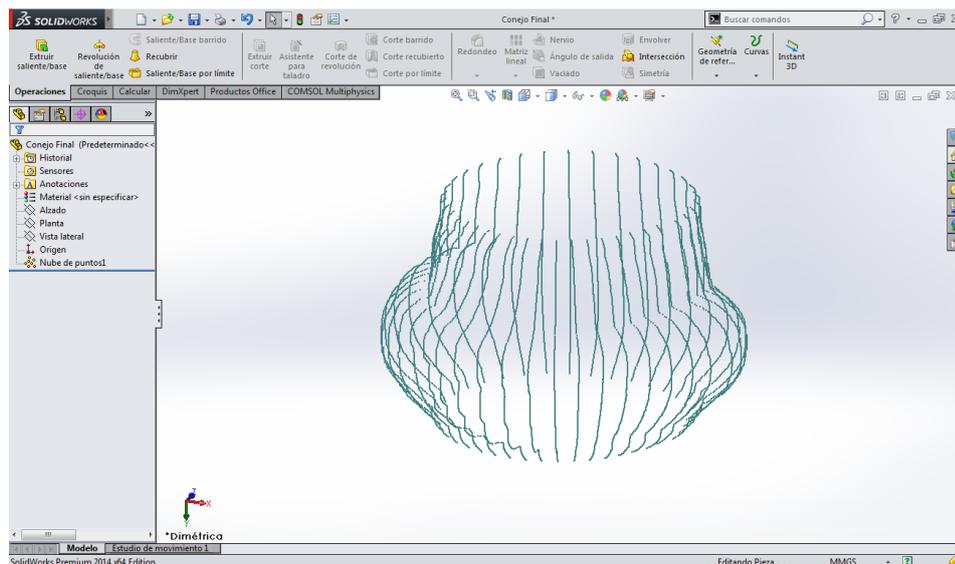


Figura 3. 27 Nube de puntos abierta en SolidWorks

A partir de la nube de puntos y con la ayuda de la herramienta de crear curvas se generó un cierto número de las mismas siguiendo la dirección de los puntos y dando forma al objeto escaneado como se muestra en la figura 3.28.

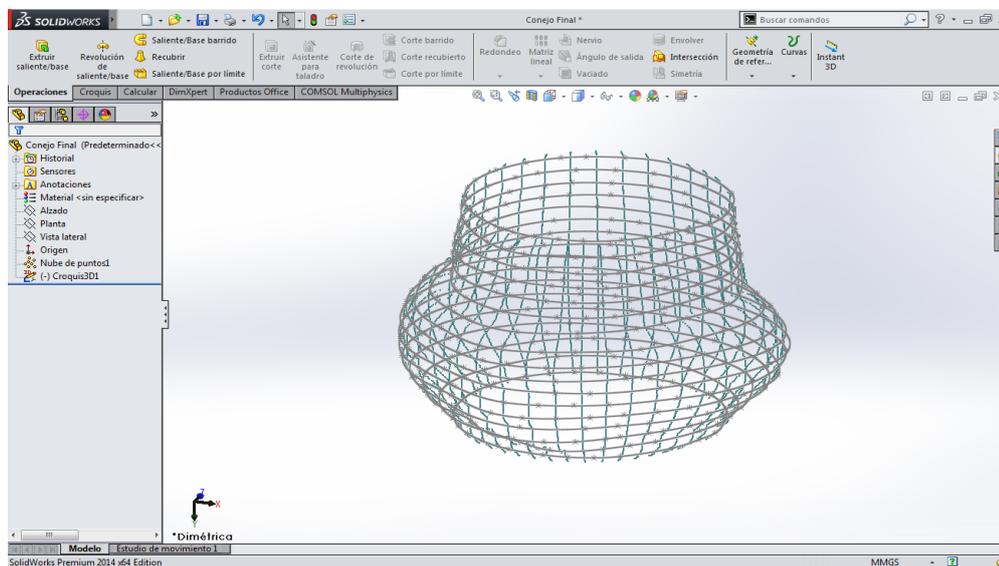


Figura 3. 28 Creación del croquis en tres dimensiones

Con este croquis se rellenó todas las curvas con la herramienta de crear superficie a partir de curvas y además se ocultó la nube de puntos como se muestra en la figura 3.29.

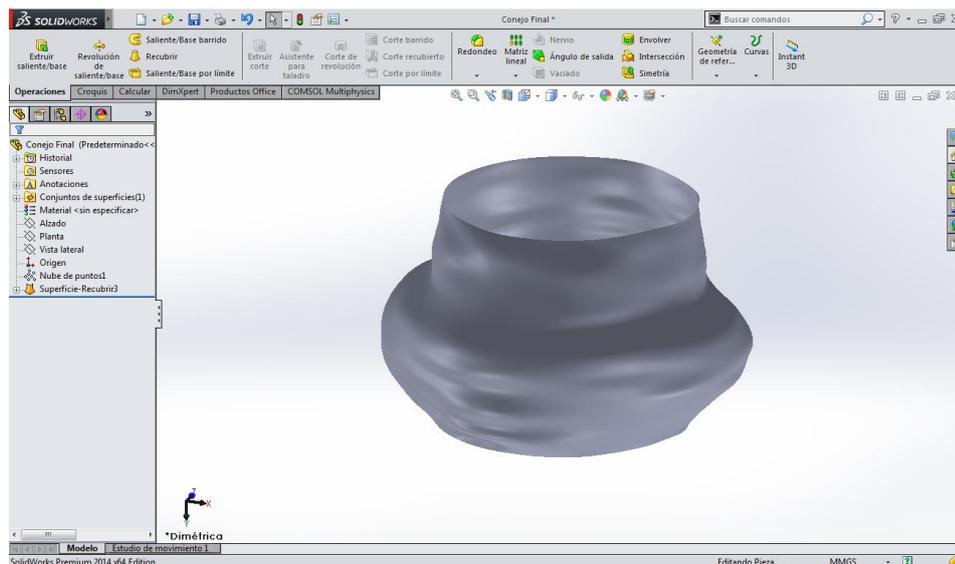


Figura 3. 29 Superficie a partir de las curvas del croquis

Luego se creó dos superficies, una en la parte superior y otra en la inferior para tapar completamente el objeto como se indica en la figura 3.30.

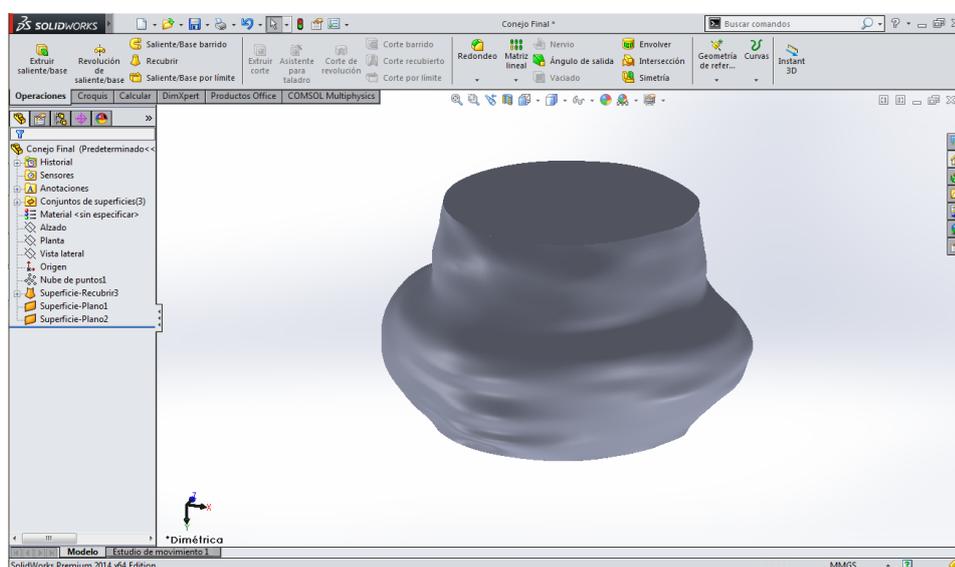


Figura 3. 30 Objeto hecho superficie y cerrado completamente

Este objeto ya se lo pudo utilizar de mejor manera que la nube de puntos, por lo que se lo convirtió en un objeto sólido y además se añadió algún tipo de detalle para imprimirlo en tres dimensiones, el objeto modificado se observa en la figura 3.31.

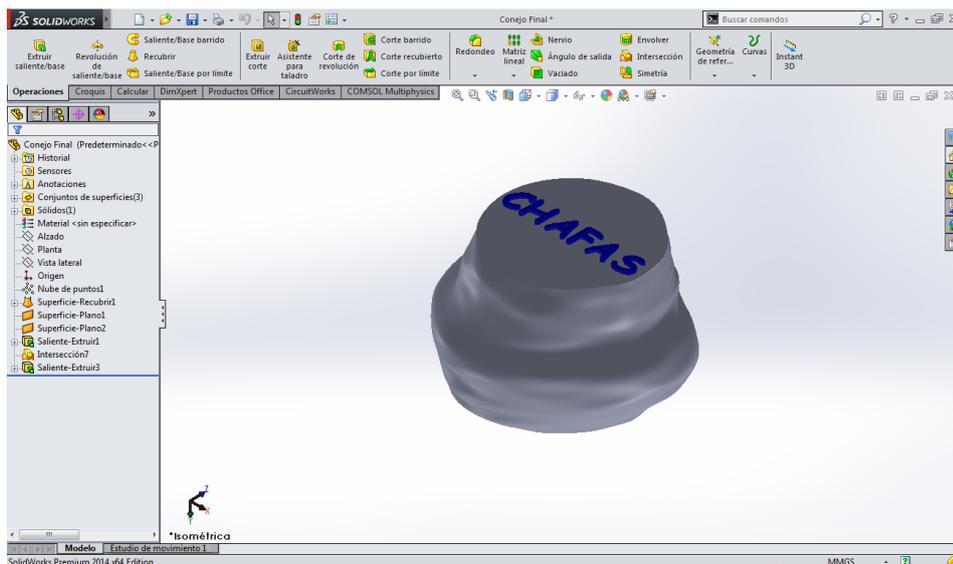


Figura 3. 31 Objeto hecho sólido y añadido letras para imprimirlo

Dicho procedimiento se realizó para los otros cuatro objetos escaneados, los mismos que se muestran en las figuras 3.32, 3.33, 3.34 y 3.35.

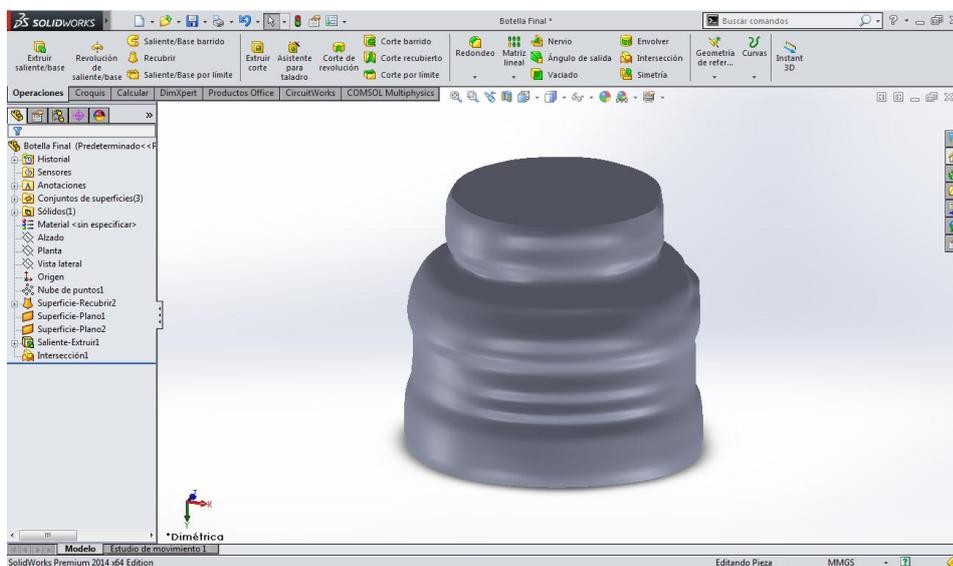


Figura 3. 32 Botella hecha superficie y cerrada completamente

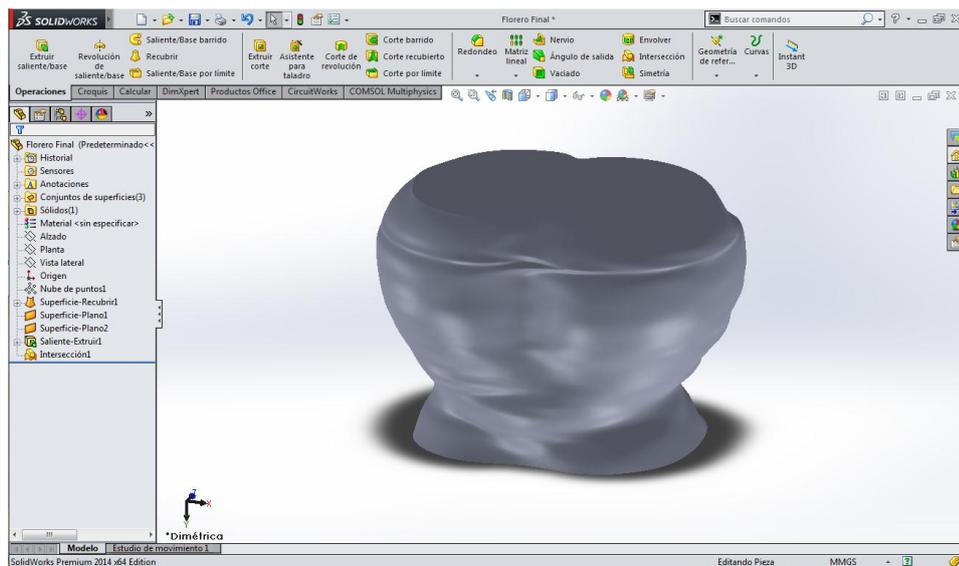


Figura 3. 33 Florero hecho superficie y cerrado completamente

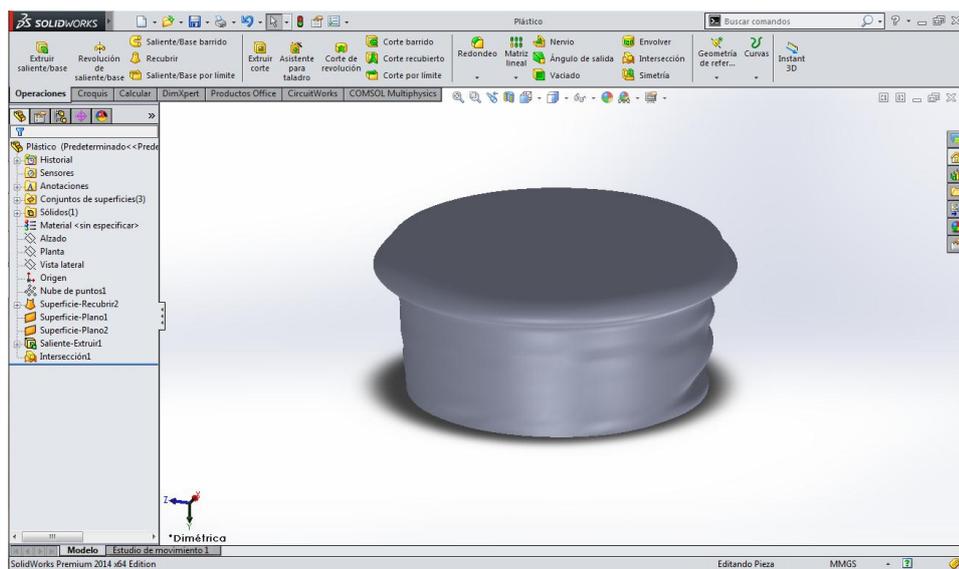


Figura 3. 34 Frasco de plástico hecho superficie y cerrado completamente

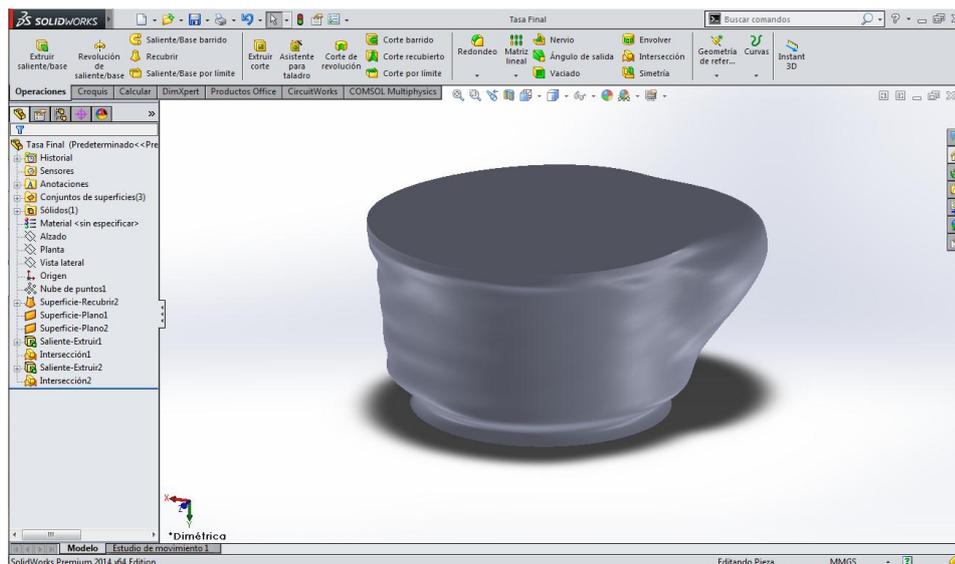


Figura 3. 35 Taza hecha superficie y cerrada completamente

3.1.2. Prueba de Impresión

Para esta prueba se utilizó el archivo antes ya creado en forma de sólido pero se lo guardó en extensión .stl ya que es el tipo de archivo que reconoce una impresora tridimensional.

La máquina utilizada fue la CatalystEX con su respectivo software, existente en el Laboratorio CNC de la ESPE-L, la misma que se observa en la figura 3.36.

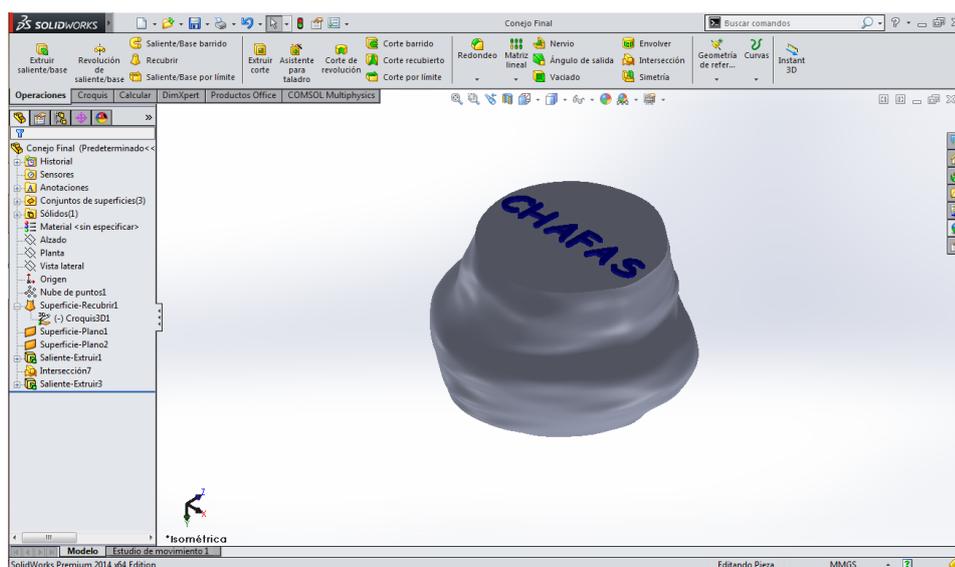


Figura 3. 36 Objeto hecho sólido y con detalles añadidos

A continuación se abrió el documento en el software de la impresora como se muestra en la figura 3.37.

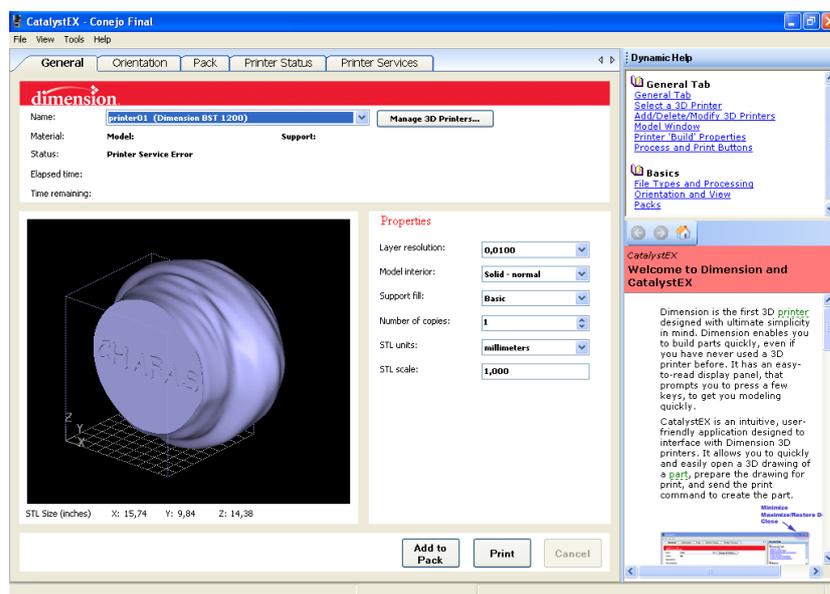


Figura 3. 37 Archivo .stl abierto en el software CatalystEX

Una vez abierto el archivo se pudo modificar la escala para que pueda ser impreso dentro del área permitida, así como también la orientación del objeto como se muestra en la figura 3.38.

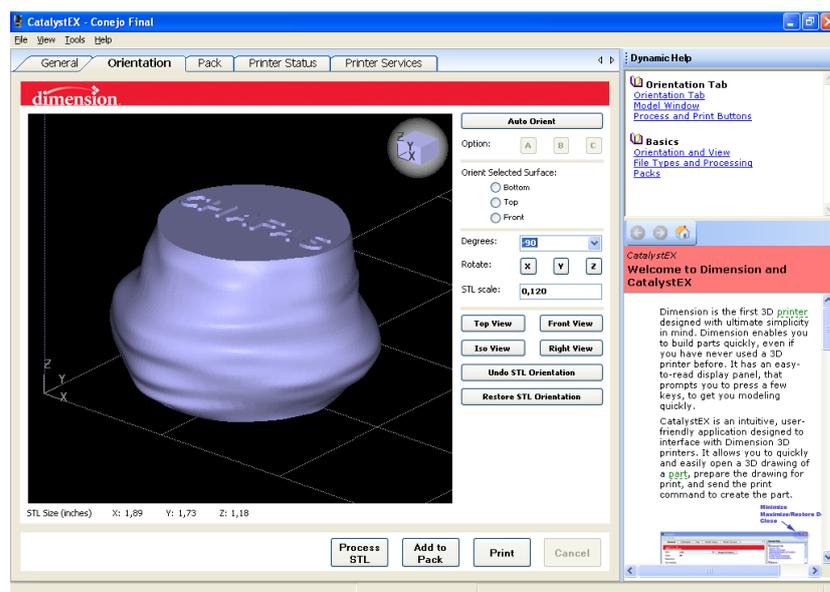


Figura 3. 38 Objeto orientado en el eje “x” y escalado

Luego se dio clic en el botón Add to Pack y automáticamente el software añadió el material de soporte y el material que va a ser impreso, además nos calculó el tiempo de impresión y la cantidad de material a utilizarse, el objeto añadido los materiales se muestra en la figura 3.39.

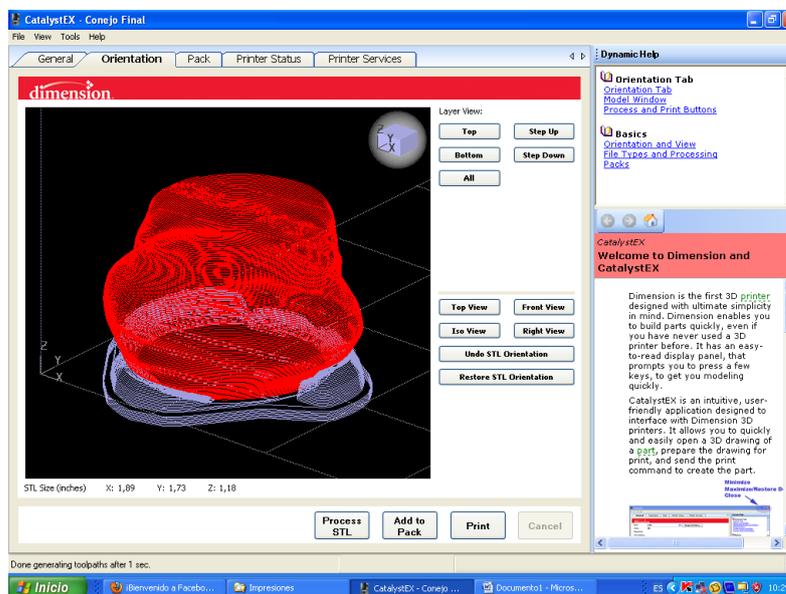


Figura 3. 39 Proceso de colocar material antes que empiece a imprimir

En la figura 3.40 se observa el objeto en el área de impresión.

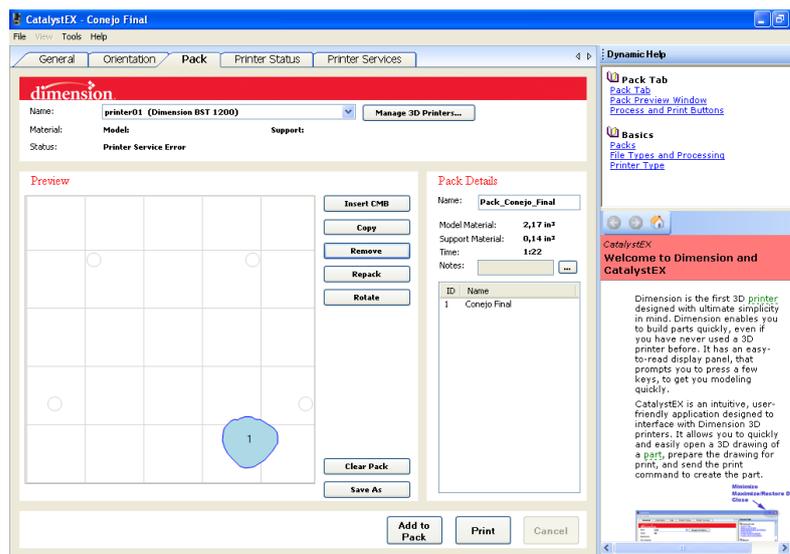


Figura 3. 40 Objeto en el área permitida para imprimir

Una vez concluido este paso se dio clic en print y el software cargó el archivo en la máquina, en la misma se aplastó el botón para que empiece a funcionar.

El proceso de impresión se muestra en la figura 3.41.



Figura 3. 41 Proceso de impresión

En la figura 3.42 se observa el nombre del archivo, el tiempo de duración y el porcentaje de impresión.



Figura 3. 42 Tiempo restante de impresión y porcentaje de impresión ya realizado

Una vez finalizado toda la impresión se obtuvo como resultado el modelo que se muestra en la figura 3.43.



Figura 3. 43 Objeto impreso finalizado

En las figuras 3.44, 3.45, 3.46 y 3.47 se observa el modelo original y el modelo impreso con sus respectivas vistas como son: la frontal, la lateral, la superior y la isométrica.



Figura 3. 44 Objeto real y objeto impreso vista frontal



Figura 3. 45 Objeto real y objeto impreso vista lateral



Figura 3. 46 Objeto real y objeto impreso vista superior



Figura 3. 47 Objeto real y objeto impreso vista isométrica

Luego de terminada la impresión se realizó un escaneado del mismo objeto con el escáner 3D NOOMEO que posee la universidad para realizar las comparaciones tanto en medidas generales como en detalles entre el objeto original, el impreso y los dos escaneados, el resultado escaneado se puede observar en la figura 3.48.

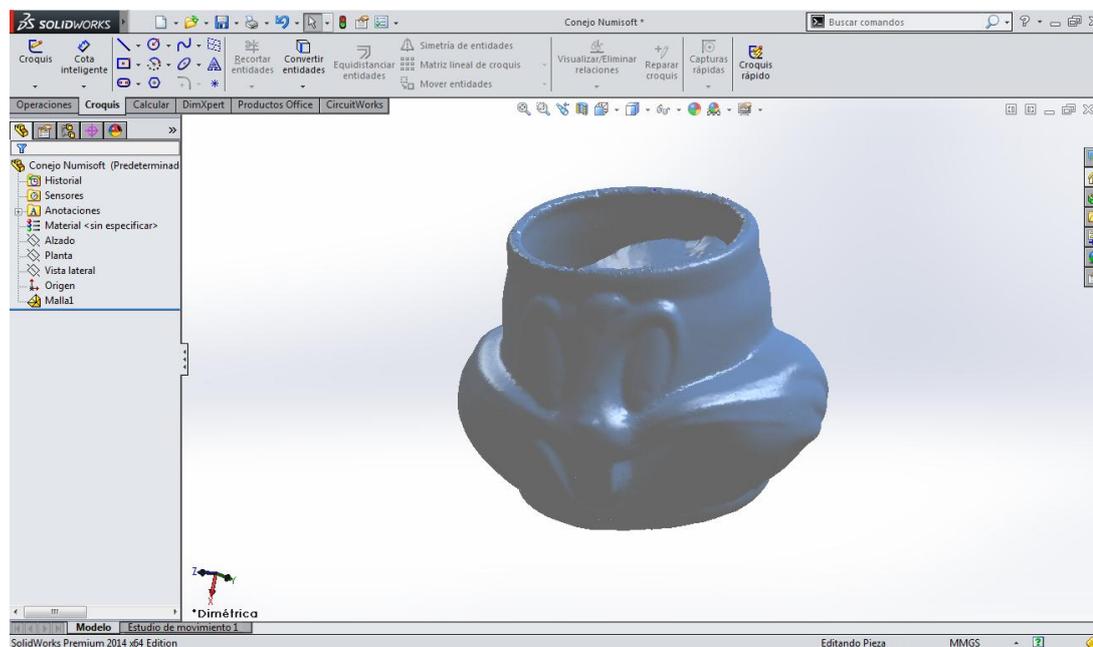


Figura 3. 48 Resultado con el escáner 3D Noomeo

En las medidas del modelo impreso se tuvo que rebajar la escala a 0,4 con respecto al original para ahorrar material, con esta reducción de la escala se modificaron las medidas tanto generales como las de los detalles, los valores se ve en las tablas 3.2 y 3.3.

Tabla 3. 2

Medidas generales del frasco de cerámica

	Alto (mm)	Ancho (mm)	Diámetro del hueco (mm)
Original	87	115	76
Escáner creado	110	140	100
Escáner 3D Noomeo	80	115	76
Impreso a escala 0.4	45	65	45

Tabla 3. 3

Medidas de los detalles del frasco de cerámica

	Altura de los párpados (mm)	Altura de la nariz (mm)	Altura de los dientes (mm)	Saliente de las mejillas (mm)	Ancho de las mejillas (mm)
Original	6	15	13	15	42
Escáner creado	8	18	15	18	57
Escáner 3D Noomeo	6	15	13	15	35
Impreso a escala 0.4	3	7	6	7	23

Con los valores de las tablas anteriores se realizó el cuadro estadístico de cada una, dándonos los resultados que se muestran en las figuras 3.49 y 3.50.

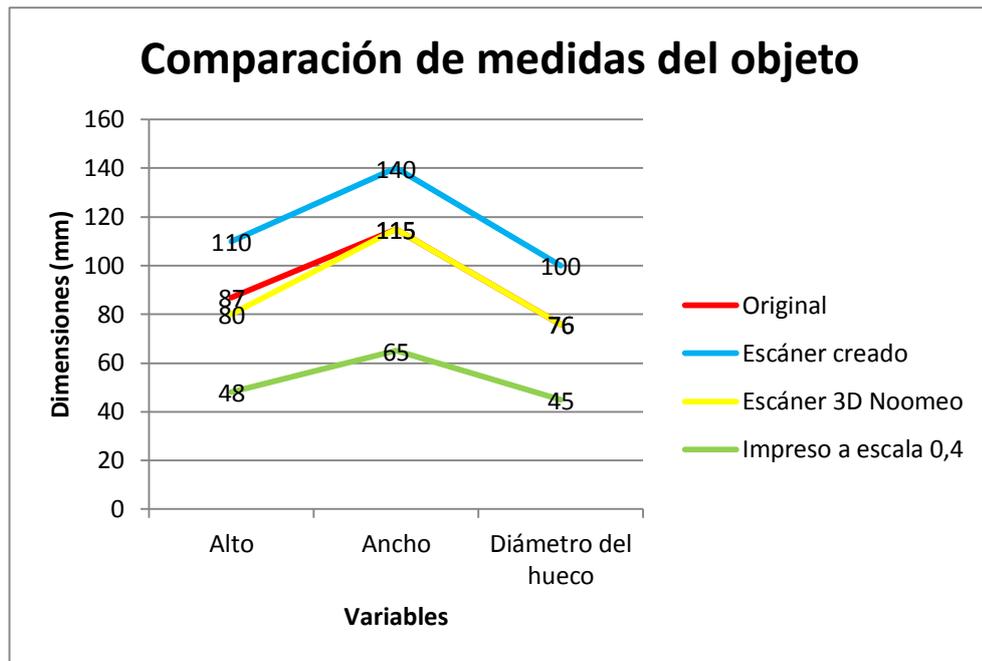


Figura 3. 49 Gráfico de medidas generales del frasco

En el gráfico se observa que el escáner creado sobrepasa las medidas en relación al original a diferencia del 3D Noomeo, el mismo que generó una réplica casi similar al original, ya que solo la medida del alto es diferente y las medidas del impreso son menores debido a la reducción de la escala al momento de imprimirlo.

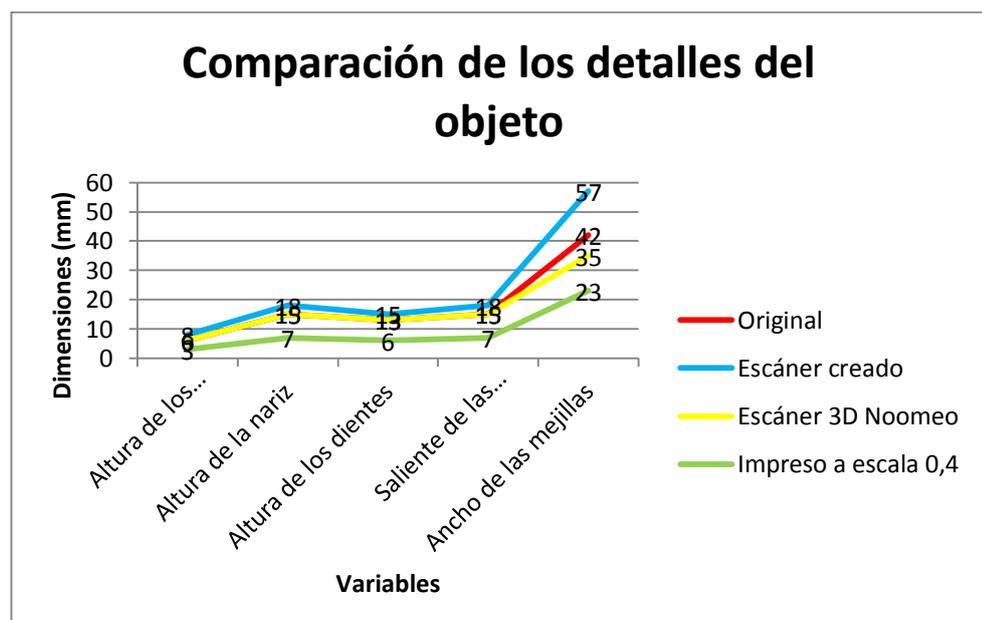


Figura 3. 50 Gráfico de medidas de los detalles del frasco

En el gráfico se observa que sucede lo mismo que en el anterior, el escáner creado tiene medidas más altas a las originales, en cuanto al 3D Noomeo es casi similar ya que el ancho de las mejillas varía y de igual manera el impreso es más pequeño debido a la escala.

Con las medidas obtenidas en los gráficos anteriores entre la figura original y la escaneada, se realizó el cálculo para obtener el error que existe en el escáner creado, en la tabla 3.4 se muestra el error en cuanto a medidas generales y en la tabla 3.5 se observa el error en cuanto a medidas de detalle.

Tabla 3. 4

Error de las medidas generales del frasco de cerámica

	Alto (mm)	Ancho (mm)	Diámetro del hueco (mm)	Promedio
Original	87	115	76	92,7
Escáner creado	110	140	100	116,7
ERROR (Ec - O)	23	25	24	24

Tabla 3. 5

Error de las medidas de los detalles del frasco de cerámica

	Altura de los párpados (mm)	Altura de la nariz (mm)	Altura de los dientes (mm)	Saliente de las mejillas (mm)	Ancho de las mejillas (mm)	Promedio
Original	6	15	13	15	42	18,2
Escáner creado	8	18	15	18	57	23,2
ERROR (Ec - O)	2	3	2	3	15	5

Con los datos obtenidos de las tablas anteriores se determina el error relativo de la medición, la ecuación a utilizar es la Ec. 3.1.

$$E_r = \frac{\Delta x}{x} \quad \text{Ec. 3.1}$$

Para el error relativo porcentual, se multiplicará por 100, teniendo la Ec. 3.2 y la Ec. 3.3.

$$E_{\%} = 100 * \frac{\Delta x}{x} \quad \text{Ec. 3.2}$$

$$E_{\%} = 100 * \frac{|Valor\ medido - Valor\ verdadero|}{Valor\ verdadero} \quad \text{Ec. 3.3}$$

3.2. Resultados

Calculado el error porcentual de las mediciones se obtiene los siguientes resultados: 25,89% en cuanto se refiere a medidas generales y 27,47% en cuanto se refiere a las medidas de los detalles que tiene el frasco de cerámica.

El operario puede reducir estos dos errores desde la pantalla de inicio, introduciendo un valor que permita la reducción del tamaño de la imagen original y generar el archivo que contiene la nube de puntos, luego de esto procesar el mismo y una vez que esté hecho sólido medirlo y determinar si el tamaño es apropiado o no, caso contrario tendría que ingresar otro valor de reducción y volver a repetir el proceso.

En las medidas realizadas al objeto se observó que los resultados obtenidos con el escáner 3D Noomeo genera un archivo a escala 1:1 en relación al objeto real, a diferencia del escáner creado generó un archivo a escala 1:26, esto sucede debido a que al momento de cortar la imagen se lo realizó a la mitad de los valores iniciales, eso dependerá del operario si desea hacerlo más grande o más pequeño, ya que el sistema si permite cambiar las dimensiones y lograr que el objeto pueda acercarse a la escala 1:1 que es la óptima; el escáner 3D Noomeo escaneó los detalles tal y como es en la realidad sin importar la forma, tamaño o complejidad de los mismos en el objeto, a diferencia del escáner creado que no se visualizó los que son muy pequeños, para esto se realizó el análisis del error relativo porcentual con las medidas de los detalles más relevantes que tiene el frasco (mejillas, nariz, dientes, párpados) y se determinó que todas estas se incrementan un poco más que el 25% a las originales y no se pueden observar de una manera bien definida, lo que si puede ver es el contorno que forman los mismos y que pueden ser medidos y comparados con el objeto escaneado para ver si se asemejan a los originales.

3.3. Análisis Técnico - Económico

La máquina ha sido implementada con satisfacción cumpliendo su función en el Laboratorio CNC; por lo cual se hace necesario cuantificar su valor económico para comparar con el dispositivo existente en la Universidad.

Para la realización del proyecto se utilizó materiales eléctricos, electrónicos, mecánicos, softwares que se detallan en la tabla 3.6.

Tabla 3. 6

Materiales utilizados en el proyecto

ORDEN	DETALLE	TOTAL USD
1	WEBCAM	20
2	MOTOR	20
3	SOFTWARE	60
4	ADQUISICIÓN DE MATERIALES	150
5	INTERNET	50
6	VARIOS	150
	TOTAL	450

El costo del escáner creado fue de 450 dólares que es un valor extremadamente pequeño en cuanto al que posee la Universidad que esta alrededor de 42000 dólares, por lo tanto se puede realizar una inversión para mejorar el dispositivo fabricado y obtener mejores resultados, incluso poder fabricarlos a gran escala.

3.4. Validación de la hipótesis

La hipótesis planteada para el proyecto fue:

¿Con el procesamiento de imágenes obtenidas mediante 2 webcams, se logrará obtener un objeto en 3D que será luego procesado en software CAD?

Se pudo obtener un objeto en 3D con el procesamiento de imágenes y se logró procesarlo en un software CAD. El proyecto será útil para prácticas en el Laboratorio CNC de la Universidad de las Fuerzas Armadas - ESPE extensión Latacunga.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Se diseñó y construyó un escáner digital 3D con la utilización de una webcam, un motor a pasos, un PIC con su respectivo circuito de control y potencia y el software Matlab versión de 32 bits, dicho sistema está diseñado para objetos que deben estar entre 5cm como mínimo y 13cm como máximo en lo que se refiere al alto y al ancho, el peso debe estar entre 100gr como mínimo y 500 gr como máximo del objeto, el objeto no debe tener muchos detalles ya que no se pueden generar como son en el original y simplemente salen los rasgos.
- Se creó un entorno gráfico para la visualización de las fotos obtenidas por la webcam y a su vez datos adicionales como por ejemplo cuántas imágenes se van a capturar, el número de pasos que da el motor, el número de las imágenes que ya van capturadas.
- Se creó un programa para encontrar las coordenadas de un punto en el espacio y guardar la información en un nuevo archivo con un cierto tipo de extensión, en este caso .obj y .txt.
- Con Matlab se creó el archivo en extensión .obj en el cual se puso la información necesaria para poder tener un mallado de todas las imágenes y así visualizar el objeto en tres dimensiones.
- El software SolidWorks permitió crear un sólido a partir de una nube de puntos de una manera fácil, ya que primero se creó una cierta cantidad de curvas, luego una superficie para rellenar las curvas y al final se lo convirtió en sólido y finalmente se añadió detalles.

- Con el archivo .txt creado se tuvo un mejor resultado gracias a que la nube de puntos se la pudo manipular de mejor manera en cuanto a creación de curvas, superficies y sólidos con el software SolidWorks a diferencia del archivo en forma de malla que resulto más complicado su manipulación.
- Con el escáner 3D Noomeo se determinó que puede capturar mejor los detalles en cuanto al escáner creado ya que para reconstruir la imagen utiliza más de 1'000,000 de puntos en el espacio a diferencia del otro escáner que en el mismo objeto que se escaneó solo dio alrededor de 10,000 puntos por eso motivo no tiene una buena resolución en cuanto a detalles.
- Este proyecto nos enseñó el principio básico de funcionamiento de un escáner tridimensional con un software de cálculo matemático, en este caso es Matlab.
- El tiempo que se demora en escanear un objeto está alrededor de 10 minutos, un valor alto debido a la velocidad de comunicación que existe entre el Matlab y el pic, ya que se necesita de ciertos tiempos de espera tanto en el programa de Matlab como el programa del pic para que pueda leer los datos de envío y recepción.
- La correcta extracción del perfil de cada imagen se dio gracias al corte que se efectuó antes de generar las coordenadas, el mismo que tiene como restricción cortarlo como máximo hasta la mitad de la imagen reducida ya que solo se necesita el perfil del lado izquierdo del objeto, otro parámetro que ayudó es la utilización de un foco cerca del objeto ya que es grande y a la vez la luz es igual desde la base hasta la parte más alta.
- El sistema empleado es flexible ya que el operador puede cambiar la programación a cualquier instante e ir haciendo las pruebas necesarias hasta conseguir la correcta programación para la aplicación que desee realizar.

- El entorno gráfico creado para la manipulación del mismo es muy sencillo de utilizarlo, permitiendo que el operador no tenga ningún tipo de contratiempo al momento de ejecutarlo.
- Para que el escáner funcione correctamente es necesario tener instalado Matlab versión de 32 bits ya que los archivos para la comunicación son para esta versión.
- Con el proceso de creación de curvas, superficies y de sólido a partir de la nube de puntos el escáner no tiene problemas en cuanto se refiere a dificultad de escanear un objeto que tenga muchos detalles, ya que en el proceso de edición el operador puede modificar paso a paso el objeto hasta obtener un resultado similar al original o a su vez añadir nuevos detalles.

4.2. Recomendaciones

- Para mejorar la resolución del escaneado en cuanto a detalles se refiere se utilizar otro motor que posea un menor número de grados por paso para poder obtener más imágenes y más unidas, o a su vez utilizar un circuito capaz de convertir los pasos del motor en micro pasos y así lograr un menor ángulo de giro.
- Para tener un mejor resultado de escaneado los objetos deben estar pintados de color blanco sin brillo, ya que si el objeto tiene diferentes colores se debe modificar la programación para que los detecte y que de esta manera no se pierdan píxeles al momento de realizar el cambio a escala de grises y por ende que no se distorsione el resultado.
- Antes de empezar a escanear debemos hacer una vista previa en Matlab para ubicar a nuestro objeto en la mitad de la base y a la vez mover la cámara hasta tener la visualización completa del objeto.

- Siempre que conectemos nuestra placa de control del motor a la PC vía USB comprobar en el administrador de dispositivos si le reconoció sino para proceder a cargar el archivo .inf y volver a conectar el dispositivo.
- Verificar los códigos en los programas tanto de Matlab como del PIC para que la comunicación no se vea afectada y los datos lleguen sin ningún problema de pérdida o en exceso.
- Durante el proceso de captura del número de imágenes seleccionado podemos levantar la tapa sin ningún contratiempo e ir verificando que el motor esté girando y que el objeto no se haya caído.
- No cerrar nuestra pantalla principal en la PC hasta que se despliegue los mensajes de Capturas Terminadas, Filtro de Canny Terminado y Generador de Coordenadas Terminado.
- Comprobar que los cables de las fuentes no se vayan a unir debido a que pueden producir que los elementos electrónicos se dañen o en el peor de los casos se lleguen a quemar.
- Siempre que se realice algún cambio en la programación poner algún símbolo o palabras que indique dentro del código las líneas que se han modificado.
- Tener en cuenta de todos los mensajes que salgan durante el proceso de ingreso de datos ya que tienen ciertas limitaciones.
- Para que el programa se desarrolle con normalidad se debe guardar todos los archivos en una misma carpeta tanto los .m y .fig que se crean en Matlab así como los necesarios para que exista la comunicación entre el PIC y Matlab y las figuras que se desee que aparezcan como fondo en las interfaces gráficas.
- En caso de no saber de qué manera funciona todo el sistema se debe ir al manual de funcionamiento para que todo el proceso salga de una manera correcta y obtener los mejores resultados.

BIBLIOGRAFÍA

- Amos, G. (2006). *Matlab una introducción con ejemplos prácticos* (2° ed.). Barcelona: Reverté, S.A.
- Báez López, D., & Cervantes Villagómez, O. (2012). *MATLAB con aplicaciones a la Ingeniería, Física y Finanzas* (2° ed.). México, D.F.: Alfaomega Grupo Editor, S.A. de C.V.
- Lajara, J., & Pelegrí, J. (2007). *LabVIEW: Entorno gráfico de programación* (1° ed.). México, D.F.: Alfaomega Grupo Editor, S.A. de C.V.
- Lázaro, A., & Del Rio Fernández, J. (2005). *LabVIEW 7.1 Programación gráfica para el control de instrumentación*. Madrid: Paraninfo, S.A.

NETGRAFÍA

- Agfra. (s.f.). *Escáner 3D*. Recuperado el 5 de Noviembre de 2014, de <http://www.agfra.es/proyectos/hola-mundo/>
- Barragán Guerrero, D. O. (s.f.). *Manual de interfaz gráfica de usuario en Matlab*. Recuperado el 28 de Diciembre de 2012, de Escuela Superior Politécnica del Litoral: https://www.dspace.espol.edu.ec/bitstream/123456789/10740/11/MATLAB_GUIDE.pdf
- Carpente Recouso, A., & Albela Pérez, D. (18 de Mayo de 2012). *Visión en 3D*. Recuperado el 5 de Noviembre de 2014, de Scribd: <https://es.scribd.com/doc/97305139/Vision-en-3D>
- CINPI. (19 de Octubre de 2011). *OptiNum Escáner 3D - Digitalización3D, Inspección e Ingeniería Inversa*. Recuperado el 5 de Mayo de 2012, de <http://www.cinpi.es/blog/2011/10/optinum-escaner-3d-digitalizacion-3d-inspeccion-e-ingenieria-inversa/>
- Cortés Parejo, J. (Marzo de 2005). *Integración en 3D Studio Max de elementos generados con aplicaciones externas*. Recuperado el 22 de Mayo de 2013, de Universidad de Sevilla: http://personal.us.es/jcortes/Material/Material_archivos/Articulos%20PDF/3DStudio.pdf

- Desarrollo PIC. (6 de Abril de 2011). *Proyecto 1*. Recuperado el 8 de Julio de 2013, de <http://www.desarrollopic.com/tutoriales/proyecto-usb-con-pic-winxp/>
- Escuela Superior de Arte de Asturias. (28 de Mayo de 2008). *Escaneado 3D*. Recuperado el 5 de Noviembre de 2013, de Restauras: <http://blog.educastur.es/restauras/2008/05/28/escaneado-3d/>
- Fernández Asunción, O. E. (13 de Abril de 2009). *DETECCIÓN DE BORDES: algoritmo de canny*. Recuperado el 3 de Abril de 2013, de <http://oefa.blogspot.com/2009/04/deteccion-de-bordes-algoritmo-de-canny.html>
- Gordillo Erazo, L. E., & Yáñez Roca, J. L. (2009). *Aplicación de visión con LabVIEW para la detección de frascos con turbiedades*. Recuperado el 6 de Febrero de 2012, de Escuela Superior Politécnica del Litoral: <http://www.dspace.espol.edu.ec/bitstream/123456789/16157/1/Aplicacion%20de%20Vision%20con%20LabVIEW%20para%20la%20deteccion%20de%20frascos%20con%20turbiedades.pdf>
- MakerBot. (23 de Agosto de 2013). *Digitizer, el escáner 3D de MakerBot se pone a la venta por 1.400 dólares*. Recuperado el 8 de Octubre de 2013, de Imapas: <http://imapas.net/2013/08/23/digitizer-el-escaner-3d-de-makerbot-se-pone-la-venta-por-1-400-dolares/>
- Microchip Technology Inc. (2006). *PIC18F2455/2550/4455/4550 Data Sheet*. Recuperado el 24 de Junio de 2013, de Microchip: <http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>
- Narváez González, A. C. (Octubre de 2010). *Escáner 3D de bajo costo empleando Webcams*. Recuperado el 11 de Septiembre de 2011, de Universidad Central de Venezuela: <http://saber.ucv.ve/jspui/bitstream/123456789/7297/1/Teisis%20Celeste%20Narv%C3%A1ez.pdf>
- Rodríguez Salavarría, J. P. (Octubre de 2009). *Sistema de control de acceso basado en el perfil lateral de una persona utilizando visión artificial*. Recuperado el 23 de Enero de 2013, de Escuela Politécnica Nacional: <http://bibdigital.epn.edu.ec/bitstream/15000/1906/1/CD-2463.pdf>
- SANYO DENKI. (s.f.). *Stepping Motors catalogue*. Recuperado el 24 de Junio de 2013, de Transfer Multisort Elektronik:

<http://www.tme.eu/es/Document/48ac9f8512124ec9a791d3759c517a64/sanyo-motor.pdf>

Téllez Zamora, C., Cisneros Muñoz, D., Pedraza Ortega, J. C., Tovar Arriaga, S., Gorrostieta Hurtado, E., & Canchola Magdaleno, S. L. (Noviembre de 2011). *Sistema para digitalización y modelado 3D de objetos, mediante proyección laser utilizando hardware de arquitectura abierta*. Recuperado el 12 de Abril de 2012, de ResearchGate:

http://www.researchgate.net/publication/220006751_Sistema_para_digitalizacion_y_modelado_3D_de_objetos_mediante_proyeccion_laser_utilizando_hardware_de_arquitectura_abierta

Universidad de Valladolid. (s.f.). *Manual Básico de Matlab*. Recuperado el 23 de Enero de 2013, de Escuela de Ingenierías Industriales:

<http://www.eis.uva.es/~jossan/scyc/practicas/matlab.pdf>

Wikimedia. (s.f.). *Escáner 3D*. Recuperado el 15 de Septiembre de 2012, de Wikipedia: http://es.wikipedia.org/wiki/Esc%C3%A1ner_3D

ANEXOS

ÍNDICE DE ANEXOS

ANEXO A: Ficha técnica del motor paso a paso

ANEXO B: Datasheet del PIC 18f2550

ANEXO C: Programación del PIC 18f2550

ANEXO D: Cálculo del número de imágenes a capturar

ANEXO E: Programación en Matlab

ANEXO F: Manual de operación del escáner 3d

ANEXO G: Dimensiones del sistema y ubicación de los elementos

ANEXO H: Fotos del sistema completo

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INGENIERÍA MECATRÓNICA
CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el señor Molina Hidalgo Miguel Ángel bajo mi supervisión.

ING. FAUSTO ACUÑA
DIRECTOR

ING. EDDIE GALARZA
CODIRECTOR

ING. VICENTE HALLO
DIRECTOR CARRERA DE
INGENIERÍA MECATRÓNICA

DR. JARAMILLO FREDDY
SECRETARIO ACADÉMICO