



Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Escáner 3D para la recreación de objetos físicos

Tesis presentada para obtener el grado de Licenciada en
Ingeniería en Ciencias de la Computación

Presenta

Violeta López Campos

Asesoras

Dra. Guillermina Sánchez Román

Mtra. Lilia Mantilla Narvárez

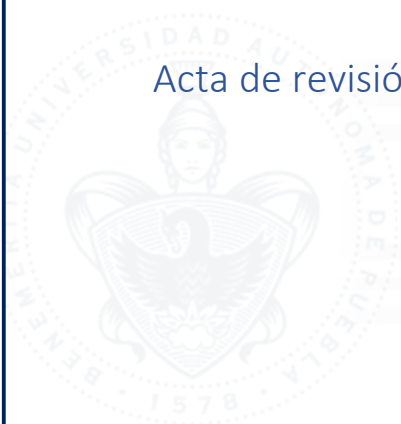
Puebla, Pue. Abril de 2023



Escáner 3D para la recreación de objetos físicos

Título

Acta de revisión de tesis



Oficio de aprobación en el tema de tesis



Facultad de Ciencias
de la Computación
Secretaría Académica

BUAP

secretariatacademica.fcc@correo.buap.mx

III Firmas

Firma del Alumno:

Firma del Asesor 1:

Firma del Asesor 2:

IV Dictamen de Comisión Revisora.

APROBADO A REVISIÓN RECHAZADO

Nombre: Roberto Contreras Juárez

Firma:

Observaciones y Recomendaciones:

Dedicatoria

Dedico esta tesis a mis padres, que son lo más importante que tengo en esta vida.

A mis hermanas, que quiero y amo con todo mi corazón.

Y a mis abuelos, que son parte fundamental en mi vida.

¡Muchas Gracias!

Agradecimientos

Agradezco a Dios por guiarme en mi camino y ayudarme a salir adelante.

A mi familia que siempre me ha apoyado incondicionalmente y que gracias a ellos estoy presentando este proyecto.

Agradezco también a una persona muy especial para mí por todo el apoyo emocional que me brindó para motivarme a realizar este proyecto.

Agradezco a mis profesoras por todo su apoyo para la realización de este proyecto.

Agradezco a todas las personas por el apoyo moral que me brindaron, amigos, maestros, incluyendo a mis compañeros de trabajo.

Índice general

<u>ÍNDICE GENERAL</u>	<u>7</u>
<u>ÍNDICE DE TABLAS Y FIGURAS</u>	<u>9</u>
<u>LISTA DE ABREVIATURAS Y SÍMBOLOS.....</u>	<u>12</u>
<u>INTRODUCCIÓN</u>	<u>13</u>
<u>CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA.....</u>	<u>14</u>
ANTECEDENTES	15
DESCRIPCIÓN DEL PROBLEMA.....	16
OBJETIVO GENERAL	17
OBJETIVOS ESPECÍFICOS.....	17
METODOLOGÍA	17
RESULTADOS ESPERADOS	18
ALCANCES.....	18
LIMITACIONES.....	18
JUSTIFICACIÓN	19
<u>CAPÍTULO II ESTADO DEL ARTE</u>	<u>20</u>
ESTADO DEL ARTE	21
<u>CAPÍTULO III MARCO TEÓRICO.....</u>	<u>27</u>
MARCO TEÓRICO	28
NUBE DE PUNTOS	28
MALLA 3D.....	28
SENSOR SHARP GP2Y0A21	29
MOTOR NEMA 17	31
OPEN3D	31
ARDUINO UNO.....	32
SPIDER PYTHON.....	32
CORELDRAW.....	33
EAGLE.....	34
ESTRUCTURA.....	34
PLACA GIRATORIA METÁLICA.....	35
PLATO GIRATORIO DE MADERA.....	35
<u>CAPÍTULO IV DISEÑO Y DESARROLLO DEL SISTEMA.....</u>	<u>36</u>
DISEÑO Y DESARROLLO DEL SISTEMA.....	37

MECÁNICA.....	37
ELECTRÓNICA.....	42
SOFTWARE	48

CAPÍTULO V PRUEBAS Y RESULTADOS 73

PRUEBAS Y RESULTADOS.....	74
PRUEBA 1.....	74
PRUEBA 2.....	79
PRUEBA 3.....	81
ESCANEADO DE PIEZA ARQUEOLÓGICA 1.....	85
ESCANEADO DE PIEZA ARQUEOLÓGICA 2.....	90
ESCANEADO DE PIEZA ARQUEOLÓGICA 3.....	93
ESCANEADO ARTESANÍA 1.....	94
ESCANEADO ARTESANÍA 2.....	97
ESCANEADO ARTESANÍA 3.....	100
ESCANEADO ARTESANÍA 4.....	103
ESCANEADO ARTESANÍA 5.....	106

CAPÍTULO VI CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS 109

CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS	110
CONCLUSIÓN.....	110
RECOMENDACIONES.....	110
TRABAJOS FUTUROS.....	110

CITAS Y REFERENCIAS BIBLIOGRÁFICAS 111

Índice de tablas y figuras

Ilustración 1.- Escaneado 3D para el control de calidad en la fabricación aditiva. Fuente: Soluciones de Metrología Creaform. (2021).	21
Ilustración 2 Diseño e implementación de prototipo de escáner 3D. Fuente: Redicces Quintanilla,R. (2017).	22
Ilustración 3 Este escáner 3D con una Raspberry Pi es de lo mejor que puedes construir. Fuente: El Output Santamaria, P. (2020).....	23
Ilustración 4 Pr#41 Escáner 3D con sensor IR casero Introducción Video. Fuente: ELECTRONOBS en Español. (2019).....	23
Ilustración 5 MakerBot Digitizer Desktop 3D Scanner Introduction Video. Fuente: MakerBot. (2013).	24
Ilustración 6 Ciclop 3D Escáner Láser.Video. Fuente: José Luis Rozúa Lucena. (2019).....	24
Ilustración 7 Cajete de Barro, nube de puntos	28
Ilustración 8 Malla en 3D del cajete de barro	29
Ilustración 9 Sensor Sharp.....	29
Ilustración 10 Método de triangulación para el sensor	30
Ilustración 11 Plato Giratorio	30
Ilustración 12 Motor Nema 17 Modelo 17HS4401	31
Ilustración 13 Logo OPEN3D.....	31
Ilustración 14 Tarjeta Arduino Uno	32
Ilustración 15 Ventana Principal de Anaconda Navigator.....	33
Ilustración 16 Software CorelDraw	33
Ilustración 17 Software Eagle.....	34
Ilustración 18 Estructura de Madera. Elaboración propia.	34
Ilustración 19 Placa Giratoria metálica Yosoo.....	35
Ilustración 20 Plato de madera	35
Ilustración 21 Boceto del diseño de la estructura. Elaboración propia.	37
Ilustración 22 Estructura renderizada en 3dmax .Elaboración propia.....	37
Ilustración 23 Estructura interna renderizada en 3dmax. Elaboración propia.	38
Ilustración 24 Escuadra de combinación.....	38
Ilustración 25 Diseño de la base sujetadora del sensor en CorelDRAW. Elaboración propia.....	39
Ilustración 26 Instalación de la base del sensor. Elaboración propia.	39
Ilustración 27 Diseño del plato giratorio con los orificios para su instalación en CorelDRAW. Elaboración propia.	40
Ilustración 28 Instalación del plato giratorio. Elaboración propia.	40
Ilustración 29 Sensor Posicionado a la altura del plato giratorio. Elaboración propia.	41
Ilustración 30 Pruebas del motor Nema 17	42
Ilustración 31 Pruebas en protoboard	42
Ilustración 32 Esquema de la tarjeta para el escáner. Elaboración propia.....	43
Ilustración 33 Board de la tarjeta para el escáner. Elaboración propia.....	43
Ilustración 34 Circuito revelado en el bastidor	44
Ilustración 35 Circuito Impreso con tinta uv en baquelita. Elaboración propia.	44
Ilustración 36 Placa lavada después del proceso de revelado de cobre con ácido férrico.....	45
Ilustración 37 Tarjeta Finalizada. Elaboración propia.	45
Ilustración 38 Colocación de componentes. Elaboración propia.....	46
Ilustración 39 Tarjeta sin la adaptación de sensor. Elaboración propia.	46

Ilustración 40 Tarjeta Final. Elaboración propia.	47
Ilustración 41 Cable creado para el sensor, diferencia de longitudes. Elaboración propia.....	47
Ilustración 42 Esquema del tiempo de ejecución	48
Ilustración 43 Representación del uso de la función millis.....	49
Ilustración 44 Programa de pruebas para el sensor y lecturas del sensor	52
Ilustración 45 Programa de pruebas para el giro.....	53
Ilustración 46 Programa de pruebas para el eje Z	55
Ilustración 47 Distancia leída del sensor al objeto.....	56
Ilustración 48 Distancia del punto medio del objeto al contorno del objeto	56
Ilustración 49 Programa en Arduino para la captura del contorno del objeto.....	57
Ilustración 50 Punto con Angulo de 45°	57
Ilustración 51 Obtención del P (x, y) con trigonometría	58
Ilustración 52 Triangulo rectángulo	58
Ilustración 53 Triángulo rectángulo, obtención del P (x, y)	59
Ilustración 54 Programa en Arduino para el eje x y el eje y.....	60
Ilustración 55 Creación del documento txt en Python y conexión de Python con Arduino	61
Ilustración 56 Error en Spyder, cuando se está ejecutando la consola de Arduino.....	61
Ilustración 57 Error en Arduino, cuando se está ejecutando la consola de Python	61
Ilustración 58 Función en spyder para la lectura de la consola y escritura en el archivo txt	62
Ilustración 59 Asignación para reemplazar los saltos de línea.....	62
Ilustración 60 Asignación para crear la matriz numpy.....	62
Ilustración 61 Impresión en consola de una Matriz[n,3]	63
Ilustración 62 Reasignación para la matriz en formato x,y,z	63
Ilustración 63 Ejemplo de uso de la librería Open3D.....	63
Ilustración 64 Objeto a escanear.....	64
Ilustración 65 Resultado de la nube de puntos del objeto escaneado.	64
Ilustración 66 Interfaz gráfica. Secuencia: 1) Asignación de datos; 2) Obtención de datos; 3) Visualización de mallas; y, 4) Finalización del escáner.....	65
Ilustración 67 Interfaz de la ventana principal.....	66
Ilustración 68 Interfaz para la opción de Partes	66
Ilustración 69 Ventana secundaria para la visualización de la captura de datos del sensor	67
Ilustración 70 Tercera ventana para la visualización de la nube de puntos	68
Ilustración 71 Cuarta ventana de visualización de malla y nube de puntos juntos.	68
Ilustración 72 Ventana de visualización para la malla creada por el algoritmo Ball-Pivoting	69
Ilustración 73 Ventana de visualización para la malla creada por el algoritmo Alpha Shapes... ..	70
Ilustración 74 Ventana de visualización para la malla Ball Pivoting filtrada.....	70
Ilustración 75 Ventana de visualización para la malla Alpha Shapes filtrada	71
Ilustración 76 Cuadro de diálogo que indica que ha finalizado el escáner	71
Ilustración 77 Visualización final del cuadro de dialogo	72
Ilustración 78 Dulcero de Halloween de calabaza	74
Ilustración 79 Valores asignados en la ventana principal para el escaneo	75
Ilustración 80 Nube de puntos de la calabaza escaneada	75
Ilustración 81 Nube de puntos y malla.....	76
Ilustración 82 Malla de la calabaza creada por el algoritmo Ball Pivoting.....	76
Ilustración 83 Malla de la calabaza creada por el algoritmo Alpha Shapes.....	77
Ilustración 84 Malla Ball Pivoting filtrada	77
Ilustración 85 Malla de Alpha Shapes filtrada.....	78
Ilustración 86 Alcancía de yeso	79

Ilustración 87 Nube de puntos distorsionada por el color negro	79
Ilustración 88 Nube de puntos distorsionada, visualización del lateral de la alcancía	80
Ilustración 89 Malla distorsionada creada a partir de la nube de puntos obtenida.....	80
Ilustración 90 Canasta de bejuco	81
Ilustración 91 Valores asignados en la ventana principal para el escaneo de la canasta	81
Ilustración 92 Nube de puntos obtenida de la canasta de bejuco.....	82
Ilustración 93 Malla y nube de puntos de la canasta de bejuco	83
Ilustración 94 Malla creada por Alpha Shapes.....	83
Ilustración 95 Filtrado de malla Alpha Shapes con Alpha=1	84
Ilustración 96 Pieza arqueológica parte de un jarrón	85
Ilustración 97 Valores asignados en la ventana principal para el escaneo de la pieza arqueológica	85
Ilustración 98 Nube de puntos de la pieza arqueológica (Parte del jarrón)	86
Ilustración 99 Malla y nube de puntos de la pieza arqueológica.....	86
Ilustración 100 Malla generada por el algoritmo Ball Pivoting.....	87
Ilustración 101 Malla generada por el algoritmo Alpha Shapes	87
Ilustración 102 Malla Ball Pivoting Filtrada.....	88
Ilustración 103 Malla Alpha Shapes Filtrada	88
Ilustración 104 Escaneo Pieza arqueológica 1 Alpha=0.515	89
Ilustración 105 Escaneo Pieza arqueológica 1 Alpha=0.513	89
Ilustración 106 Pieza arqueológica circular.....	90
Ilustración 107 Nube de puntos Pieza arqueológica circular.....	90
Ilustración 108 Malla y nube de puntos. Pieza arqueológica circular.....	91
Ilustración 109 Malla Ball Pivoting y Malla Filtrada de la pieza arqueológica	91
Ilustración 110 Malla Alpha Shapes y Malla Filtrada. Pieza arqueológica circular	92
Ilustración 111 Pieza arqueológica. Pie.....	93
Ilustración 112 Nube de puntos. Pie	93
Ilustración 113 Maceta.....	94
Ilustración 114 Pantalla de asignación para el escaneo.....	94
Ilustración 115 Nube de Puntos Maceta.....	95
Ilustración 116 Malla Ball Pivoting y malla filtrada. Maceta.....	95
Ilustración 117 Malla Alpha Shapes y Malla Filtrada. Maceta	96
Ilustración 118 Gato para colorear	97
Ilustración 119 Pantalla de asignación para el escaneo. Gato.....	97
Ilustración 120 Nube de puntos. Gato	98
Ilustración 121 Malla y nube de puntos. Gato	98
Ilustración 122 Malla Ball Pivoting y Malla filtrada. Gato	99
Ilustración 123 Malla Alpha Shapes y Malla filtrada. Gato	99
Ilustración 124 Adorno para decoración.....	100
Ilustración 125 Pantalla de asignación para el escaneo. Adorno.....	100
Ilustración 126 Nube de puntos. Adorno.....	101
Ilustración 127 Nube de puntos y Malla. Adorno	101
Ilustración 128 Malla Ball Pivoting y malla filtrada. Adorno.....	102
Ilustración 129 Malla Alpha Shapes y malla filtrada. Adorno	102
Ilustración 130 Cantarito de barro	103
Ilustración 131 Pantalla de asignación del escaneo. Cantarito de barro.....	103
Ilustración 132 Nube de puntos. Cantarito de barro.	104
Ilustración 133 Malla Ball Pivoting y malla filtrada. Cantarito de barro.	104

Ilustración 134 Malla Alpha Shapes y malla filtrada. Cantarito de barro	105
Ilustración 135 Tortillero	106
Ilustración 136 Pantalla de asignación para el escaneo. Tortillero.....	106
Ilustración 137 Nube de puntos. Tortillero	107
Ilustración 138 Malla y nube de puntos. Tortillero.....	107
Ilustración 139 Malla Ball Pivoting y malla filtrada. Tortillero	108
Ilustración 140 Malla Alpha Shapes y malla filtrada. Tortillero	108
Tabla 1 Análisis de escáneres	25
Tabla 2 Datos del eje Z	54

Lista de abreviaturas y símbolos

Bps: Bits per second

Eagle: Easily Applicable Graphical Layout Editor

INAH: Instituto Nacional de Antropología e Historia

IREC: Emisor de infrarrojos

PCB: Placa de circuito impreso

PSD: Detector sensible a la posición

UL: Milisegundos

UV: Ultravioleta

Introducción

En el municipio de Tlacotepec de Benito Juárez, se encuentra la comunidad de Santa María la Alta, en la cual sus habitantes son emprendedores natos que se dedican a realizar artesanías y siempre necesitan de nuevas tecnologías para realizar sus productos, así mismo para crear e implementar nuevos productos que les permitan generar sus propias ganancias.

En la comunidad de Santo Nombre, también perteneciente al municipio de Tlacotepec de Benito Juárez, se encuentra la zona arqueológica llamada “Los Teteles”, en donde se descubrieron piezas arqueológicas que fueron prestadas al INAH para su estudio y no fueron devueltas.

Estas son las razones por las que se propone realizar un escáner 3D, como una alternativa que permita a la comunidad de Santo Nombre escanear piezas arqueológicas para su estudio sin necesidad de llevárselas y, así, garantizar que las piezas se queden en el municipio y, a su vez, implementar esta tecnología para los habitantes de la comunidad de Santa María la Alta, quienes buscan innovar y crear nuevos productos.

De acuerdo con los acontecimientos expresados, el presente trabajo se divide en seis capítulos que se describen a continuación:

El Capítulo I se plantea el problema que presenta el municipio de Tlacotepec de Benito Juárez, se propone una solución, los objetivos que se quieren alcanzar y la metodología para llevar a cabo la realización del escáner 3D.

En el Capítulo II se presenta el estado del arte.

En el Capítulo III muestra la recopilación de información y consideraciones teóricas que sustentan la presente tesis.

En el Capítulo IV se define el diseño y desarrollo del sistema, dividido en tres secciones: 1) Mecánica donde se describe el diseño, la movilidad del escáner y la construcción del escáner; 2) Electrónica donde se describe el funcionamiento de los componentes electrónicos, la creación de la tarjeta para la movilidad del escáner y el funcionamiento de este, el software donde se crea el programa para el funcionamiento del escáner ; y, 3) El programa para el procesamiento de los datos capturados y la interfaz gráfica para la interacción del escáner con el usuario.

En el Capítulo V se realizan las pruebas y se hace la interpretación de los resultados obtenidos del escáner 3D.

En el Capítulo VI se plantean las conclusiones a partir de los resultados obtenidos y su interpretación, las recomendaciones y los trabajos futuros.

Por último, se muestran las referencias.



Capítulo I

Planteamiento del problema

Antecedentes

Los sistemas de modelado 3D que actualmente utiliza la empresa Lincoln Electric permiten la realización de inspecciones automatizadas superficiales de piezas, reduciendo tiempos de inspección y aumentando, por ende, la producción en el cual eligen el escaneo 3D. A diferencia de los métodos convencionales de medición, permiten obtener un modelo de la superficie 3D completo de la impresión, y luego, pueden compararlo con su modelo CAD original. Existen diversas maneras de obtener modelos 3D, como lo son los programas profesionales los cuales son utilizados para crear o diseñar modelos 3D, pero estos necesitan tiempo y noción del programa, otro ejemplo para obtener modelos 3D sería mediante el escaneo láser del objeto a analizar. Las diferentes implementaciones del escáner 3D permiten convertir un objeto real en uno virtual para generar una base de datos, analizarlo, estudiarlo, interpretarlo o representarlo; de igual forma, para modificarlo o reproducirlo dependiendo cual sea su principal función. Los escáneres profesionales actualmente tienen un precio muy elevado y no todas las personas tienen las mismas posibilidades de contar con tecnología de esa magnitud para su uso personal. En la zona rural, como lo es el municipio de Tlacotepec de Benito Juárez, Puebla, que cuenta con la producción de artesanías en la junta auxiliar de Santa María La Alta y la “Zona arqueológica de Tételes de Santo Nombre” (Museo de Sitio de Tételes de Santo Nombre, s. f.) surge la necesidad de crear un sistema similar de bajo costo, que permita a las personas de la zona aprovechar esta tecnología para su uso múltiple. El uso de escáneres 3D, es utilizado en otros lugares para la creación de modelos 3D, como figuras, partes mecánicas, envases, entre otros para comprobar y manejar el control de calidad en sus negocios, también para la creación de modelos 3D para animación, algunos para modificarlos o reproducirlos para su venta individual en donde convierten objetos cotidianos en modelos 3D. Otra de las implementaciones que tendría el escáner 3D sería para escanear piezas arqueológicas para poder digitalizarlas, reconstruirlas y estudiarlas, ya que día a día las personas de la zona encuentran estas piezas en sus parcelas, demostrando así las múltiples aplicaciones que tendría el escáner 3D.

Descripción del problema

En el municipio de Tlacotepec de Benito Juárez Puebla, en el año 2008, fue descubierta una zona arqueológica en los ejidos de Tételes de Santo Nombre, en donde a cargo del arqueólogo del INAH, Blas Castellón Huerta realizaba, en conjunto con la comunidad, trabajos de limpieza y restauración de la zona, la cual fue abierta al público el 25 de enero de 2017 e inaugurando su museo en el mismo año.

Durante los trabajos de investigación, exploración, limpieza y reconstrucción, se fueron descubriendo aproximadamente mil piezas arqueológicas de las cuales los habitantes de la comunidad de Santo Nombre exigen al Instituto Nacional de Antropología e Historia (INAH) el regreso de las mismas, ya que los habitantes especifican que el arqueólogo encargado de la reconstrucción fue quien se las llevó para así estudiarlas y conservarlas con el fin de ser devueltas cuando el museo abriera sus puertas en donde las exhibirían, sin embargo, esto nunca sucedió.

“Me hubiera gustado que concientizaran a las personas que ayudaban en la excavación y exploración de las piezas arqueológicas de como extraerlas, que al menos tuvieran una charla de la importancia de las piezas de las cuales se va reconstruyendo la historia, también me gustaría que no quedara impune el caso del extravío de las piezas arqueológicas de las cuales cuando se trabajaba en conjunto con la comunidad estas fueron resguardadas por el instituto para su estudio, al igual que la comunidad por falta de información, extraían las piezas inadecuadamente, las vendían o se quedaban con ellas”-(Habitante de la comunidad de Santo Nombre, comunicación personal, 1 de diciembre de 2022).

En las primeras actividades económicas de Santa María la Alta, se realizaban artesanías como son los capotes, petates, plumeros, tortilleros todos realizados y creados a partir de palma, con el paso del tiempo, la comunidad fue creciendo y desarrollándose al grado de convertirse en productores, emprendedores y creadores de pequeñas y medianas empresas. Actualmente, se dedican a fabricar vitrinas, anaqueles, góndolas, fabricación de maniqués, mallas, jaulas entre otros productos, lo que lo ha convertido en un lugar muy competitivo, dando como consecuencia la producción y creación de nuevos productos para competir en el mercado.

“La competencia es demasiado dura por eso se busca crear nuevos productos e innovar otros, si la competencia ve que pusiste un negocio, llegan los que tienen más capital y te ponen ellos uno más barato y no te dejan avanzar” -(Habitante de la comunidad de Santa María, comunicación personal, 29 de diciembre de 2022).

Por estos motivos se contempló crear un sistema que brinde una solución a los problemas presentados.

Objetivo general

Diseñar un sistema (hardware y software) capaz de construir un modelo digital 3D con la librería open3D a partir de la medición de un objeto físico (real), obteniendo los datos con un Arduino uno, para procesarlos mediante una interfaz, y así, crear el modelo virtual para analizarlo, interpretarlo, modificarlo o reproducirlo.

Objetivos específicos


- Realizar un algoritmo para el desarrollo del modelo
- Estudiar y utilizar la librería open3D para el procesamiento de datos y la creación de modelos 3D
- Diseñar un escáner 3D utilizando la tarjeta Arduino Uno
- Realizar un modelo de escáner giratorio para la recopilación de datos
- Diseñar e implementar una interfaz para controlar el escáner 3D

Metodología

Para el desarrollo del siguiente proyecto se realizará con el método Kanban que es *“una filosofía centrada en la mejora continua, donde las tareas se extraen de una lista de acciones pendientes en un flujo de trabajo constante”* (Martins, 2022).

La cual se dividirá en cuatro secciones con subtarear:

- ⇒ Mecánica: Consistirá en construir la estructura mecánica del escáner.
 - Investigar cuáles son los elementos que se requieren para integrar el escáner y tomar las medidas de estos para un posterior diseño mecánico.
 - Revisar el mejor modelo de fabricación para una mejor calidad y estética aceptable del escáner.
 - Diseñar la estructura mecánica del escáner para construirlo con una impresora 3D, máquina CNC o artesanalmente.
- ⇒ Electrónica: Se realizarán los circuitos necesarios para el funcionamiento de la estructura mecánica del escáner.
 - Diseñar el circuito electrónico que permita la movilidad del escáner y la recolección de datos.
 - Diseñar una tarjeta PCB en caso de ser necesario con el software Eagle.
 - Realizar las pruebas correspondientes de los componentes electrónicos para revisar el funcionamiento de estos y la implementación que debe llevar en el escáner.
 - Realizar las pruebas correspondientes para el correcto funcionamiento del circuito eléctrico y realizar las correcciones necesarias.
- ⇒ Software: Se realizarán los algoritmos correspondientes para la captura y el análisis de los datos que generarán los modelos 3D para analizarlo, interpretarlo, modificarlo o reproducirlo.

- 
- Realizar el software para la movilidad motriz del escáner 3D
 - Diseñar el algoritmo capaz de crear el modelo 3D con la librería open 3D para el procesamiento de datos.
 - Realizar las pruebas correspondientes para el correcto funcionamiento del algoritmo.
 - Modificar el algoritmo en caso de ser necesario.
 - Diseñar una interfaz amigable para el usuario con la finalidad de controlar el funcionamiento del escáner.
- ⇒ Ejecución y Resultados: Realizar las pruebas necesarias al escáner para determinar su correcto funcionamiento para la creación de modelos 3D.

Resultados Esperados

Se espera un adecuado funcionamiento del escáner 3D para la creación de modelos 3D ya sea para analizarlo, interpretarlo, modificarlo o reproducirlo con una interfaz visible y amigable para el usuario.

Alcances

El desarrollo de este proyecto tiene como finalidad recrear las piezas arqueológicas que las personas de la comunidad de Santo Nombre, Palmillas Tlacotepec de Benito Juárez Puebla encuentran día a día en sus parcelas para formar su propia base de datos y estudiarlas, al igual que recrear los modelos 3D de los objetos de artesanías de la junta auxiliar de Santa María la Alta de Tlacotepec de Benito Juárez, Puebla, para su comercialización.

Limitaciones

En el desarrollo del prototipo se presentaron las siguientes limitaciones:

Los proveedores de piezas electrónicas no ingresaban a la comunidad de Tlacotepec de Benito Juárez, Puebla, por seguridad.

Para la creación de la infraestructura del prototipo, se contempló una plataforma circular, pero al ser muy grande necesitaba tener puntos de apoyo para cargar objetos pesados.

Para la creación de la tarjeta electrónica, se estaba utilizando el método de planchado para circuitos impresos, la cual no se adhería bien a la placa de cobre por motivos de la impresión del tóner que se estaba usando. Se resolvió con el método de serigrafía para circuitos impresos.

Para el plato giratorio se agregó una base metálica, que le permitiera sujetar o bien sostener el objeto a escanear para que le diera mejor estabilidad.

Justificación

El escáner 3D será implementado para emprender, crear e innovar nuevos productos, que permitan los pobladores de la comunidad de Santa María la Alta generar ingresos para los emprendedores que van iniciando sus proyectos personales, ya que la competencia es muy fuerte y siempre se busca seguir innovando para salir adelante.

Así mismo, de acuerdo con los acontecimientos sucedidos en el municipio de Tlacotepec de Benito Juárez, Puebla, con el robo y extravío de las piezas arqueológicas de la zona de Teteles de Santo Nombre se propuso crear un escáner 3D que le permita a la comunidad obtener los modelos, para así generar su propia base de datos, en la cual pueden escanear las piezas arqueológicas encontradas día a día y, así, poder estudiarlas, reconstruirlas sin necesidad de llevárselas, ya que los habitantes de la zona ya no permiten el ingreso de personas externas especializadas que trabajen en las piezas encontradas, conforme a los testimonios recolectados especifican que han sufrido abusos de confianza.

“Optar por recurrir a la tecnología y aplicarla en las necesidades y problemáticas del día a día en la cual sería de gran ayuda, el poder tener un escáner que pueda escanear cualquier objeto y tener un archivo de sus características composición, medidas peso entre otras propiedades sería más fácil la visualización de datos sin tanta manipulación de objetos que quizás puedan ser peligroso o dañinos para la salud, o el uso de objetos muy frágiles, creo que un escáner tendría un gran campo de aplicaciones en diferentes áreas”- (Habitante de la comunidad Colonia Benito Juárez Tlacotepec, comunicación personal, 29 de diciembre de 2022).



Capítulo II

Estado del arte

Estado del arte

A lo largo de los años, se han implementado diversos métodos para obtener modelos 3D como lo es el modelado 3D que, mediante un software especial, podemos crear nuestros propios diseños o modelos 3D, o bien, tomar un objeto real y digitalizarlo con un escáner 3D, para así crear un modelo digital 3D.

Existen diferentes maneras de obtener modelos 3D y son implementados en múltiples funciones, como en la industria que requieren la comprobación de piezas exactas para el diseño y el control de calidad.

Un ejemplo de esta tecnología es la empresa Lincoln Electric Holdings, que es una multinacional estadounidense y fabricante mundial de productos de soldadura, equipos de corte por plasma, oxicorte y sistemas de soldadura robótica, donde se utilizan escáneres 3D para evaluar piezas metálicas grandes y únicas fabricadas con impresoras 3D con softwares especiales, con los cuales llevan su control de calidad en la recreación de piezas 3D mediante un escáner para realizar la comprobación sistemática del producto.

Lincoln Electric Additive Solutions es un proveedor de piezas grandes metálicas, donde optaron por elegir el escaneado 3D porque permite obtener toda la superficie completa de un modelo que varias veces son únicos. Esta tecnología ayuda a abarcar todo el modelo detallada y correctamente para luego así imprimirlo, al igual que verificar que el modelo este correcto. El escáner utilizado es el MetraSCAN 3D de Creaform con el que se pueden medir piezas de mayor magnitud en cuestión de minutos, es muy ligero y es muy complicado manipular las piezas, que incluso cuando ya se encuentran imprimiéndose es demasiada, caliente para tocarla.

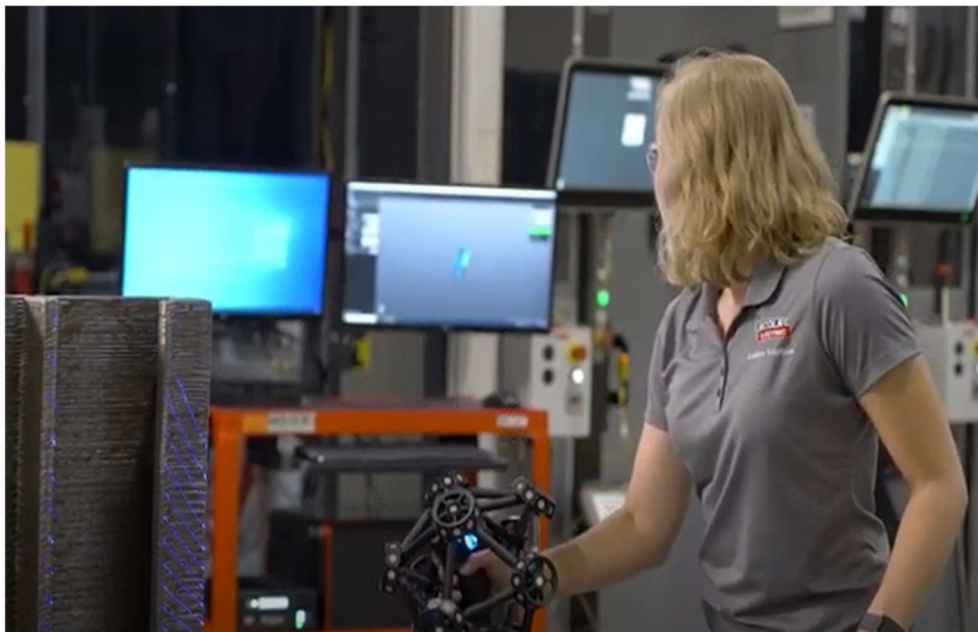


Ilustración 1.- Escaneado 3D para el control de calidad en la fabricación aditiva. Fuente: Soluciones de Metrología Creaform. (2021).

Algunos proyectos de investigación que implementaron una tecnología similar son los siguientes:

Quintana(2017) realizó un informe final de investigación con el nombre Diseño e Implementación de Prototipo de Escáner 3D en el cual se implementó un prototipo de escáner 3D con una base giratoria, que permite la extrapolación de puntos obtenidos que al unirlos crean un cuerpo geométrico del objeto escaneado y, para que se muestre la figura 3D, los datos son enviados a un software de diseño.



Ilustración 2 Diseño e implementación de prototipo de escáner 3D. Fuente: Redicces Quintanilla,R. (2017).

Santamaria(2020), en su trabajo “Este escáner 3D con una Raspberry Pi es de lo mejor que puedes construir”, se describe como un grupo de estudiantes realizan un escáner 3D mediante un láser para la creación de archivos .obj y su reproducción con impresoras 3D en donde se describe el uso del láser.

El láser ayuda a perfilar el objeto que luego se registra con el módulo de cámara para ya finalmente procesar la información y generarlo digitalmente, a partir de ese momento, con ese archivo .obj es con lo que podrás trabajar en aplicaciones 3D o usarlo para ser luego imprimirlos nuevamente a través de una impresora 3D. (Santamaria, 2020)



Ilustración 3 Este escáner 3D con una Raspberry Pi es de lo mejor que puedes construir. Fuente: El Output Santamaria, P. (2020).

El canal de YouTube, Electronoobs (2019), publica contenido de electrónica, impresión 3D, tutoriales de circuitos electrónicos básicos, impresoras 3D, luces, rayos, láseres con fines educativos de enseñanza y aprendizaje para crear tus propios modelos público un video en el cual implementa un escáner 3D de 13 cm con un sensor ir.

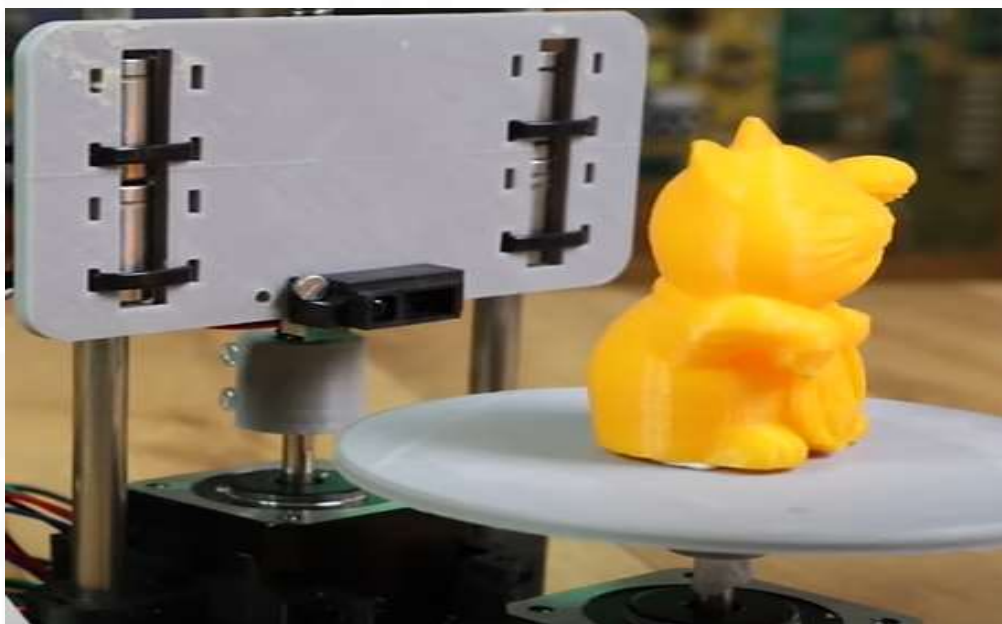


Ilustración 4 Pr#41 Escáner 3D con sensor IR casero| Introducción Video. Fuente: ELECTRONOBS en Español. (2019).

Un escáner laser profesional 3D es el MakerBot Digitizer, con un precio elevado de 1,400 dólares, lo cual hace difícil su adquisición y funciona de la siguiente manera: “se coloca el objeto en el plato giratorio, los láseres lo disparan y tu modelo físico se convierte en un diseño digital”.

(Pettis, 2013). MakerBot Digitizer se ejecuta con software especial de la empresa para crear modelos 3D.



Ilustración 5 MakerBot Digitizer Desktop 3D Scanner | Introduction Video. Fuente: MakerBot. (2013).

Otro escáner profesional es el escáner Láser 3D Ciclop, que implementa 2 láseres que produce en 2 nubes y una visión de 23 cm aproximadamente, a diferencia de los demás escáneres, este escáner se debe de calibrar antes de ser utilizado, la luminosidad también es un problema ya que si se encuentra con suficiente luz, no escanearía la figura adecuadamente.



Ilustración 6 Ciclop 3D Escáner Láser. Video. Fuente: José Luis Rozúa Lucena. (2019).

Cada uno de los escáneres tienen diferentes características, las cuales se analizaron y se interpretaron en la siguiente tabla:

Tabla 1 Análisis de escáneres

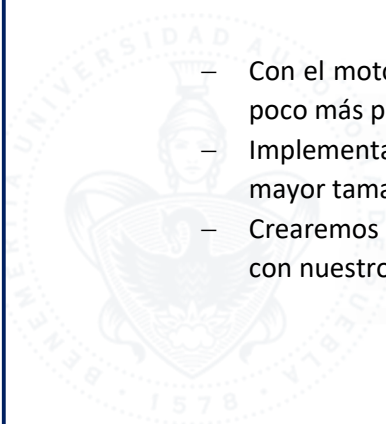
Nombre	Motores	Captura	Peso - Material	Altura eje Z	Cubierta
Diseño e implementación de prototipo de escáner 3D. Quintanilla, R. (2017)	Servos	Laser	Materiales ligeros	No especificado	No
Escáner 3D con RaspberryPi	No especificado	Laser	Materiales ligeros	No especificado	Si
Escáner 3D con sensor IR	Nema 17	Sensor IR	Materiales ligeros	13 cm	No
MakerBot Digitizer	No especificado	Laser	Materiales ligeros	20.03 cm	No
Ciclop	No especificado	Laser	Materiales ligeros -3 kilos	23 cm	No

De acuerdo con la literatura y el análisis de la Tabla 1, se infieren varios factores/aspectos se concluye que:

- a) Los motores utilizados para un escáner deben ser controlados con alta precisión. Con el motor Nema 17 podemos controlar la posición del objeto, su velocidad que está controlada por la frecuencia de pulsos y el movimiento de este, ya que el motor funciona a pasos podemos controlar el ángulo de rotación del objeto proporcional a los pulsos de entrada lo que permite una mejor recepción y lectura de los datos. Otra de las ventajas que tendría al utilizar motores Nema 17, es mejorar la carga del peso del objeto en el escáner ya que este al ser configurado adecuadamente posee un “torque de 4.0789 kg/cm (40 N.cm Min)” (MotionKing,2020) de acuerdo con la hoja de especificaciones.
- b) La captura de datos en la mayoría de los escáneres analizados utilizan una cámara y un láser que permite la recepción de datos, utilizando el método de triangulación para lograr su objetivo. El “sensor ir” integra un led infrarrojo y un receptor PSD que con el método de triangulación, de acuerdo con el ángulo obtenido, puede dar la distancia del objeto visualizado simplificando así la recepción de datos.
- c) Los escáneres presentados solo pueden escanear materiales ligeros, a excepción del escáner Ciclop que puede cargar 3 kilos, todos los escáneres analizados tienen un plato que solamente se encuentra sobrepuesto en el motor que están utilizando, esto representa un problema, ya que el motor está cargando todo el peso del objeto analizado. Si el motor tiene demasiado peso por ende el motor no va a girar, una base con estabilidad sujetaría más peso.
- d) La longitud en el eje z es fundamental en los escáneres 3D, ya que, a mayor longitud, permitirá escanear objetos o modelos de mayor tamaño, sin embargo, el plato giratorio debe de ser capaz de sostener objetos para su adecuado escaneó.
- e) La iluminación es uno de los factores más importantes, ya que con, esta se determinará si el escaneo se realizó correctamente. Los escáneres analizados en su mayoría no cuentan con una estructura que los cubra para que la recepción de los datos no se vea afectada por la luz solar. El escáner 3D con una Raspberry Pi se encuentra cubierto con una estructura que permite cerrarse y, así, evitan que la luz solar interfiera en los resultados de su escaneo, mientras los demás escáneres se ven afectados, ya que, entre más iluminación, les es imposible crear una figura adecuadamente.

Para la realización de nuestro escáner tomaremos en consideración los siguientes puntos:

- Usaremos el motor Nema 17 que es el motor de paso a paso para controlar con precisión la rotación del objeto.
- Implementaremos el escáner con el sensor ir que simplifica la recepción de los datos.



- Con el motor Nema 17, un plato y una base giratoria lograremos cargar materiales un poco más pesados que no rebasen los 4 kg.
- Implementaremos un poco más la longitud en el eje z para poder escanear objetos de mayor tamaño.
- Crearemos una estructura que no permita el paso de la luz solar para que no interfiera con nuestro sensor ir.



Capítulo III

Marco teórico

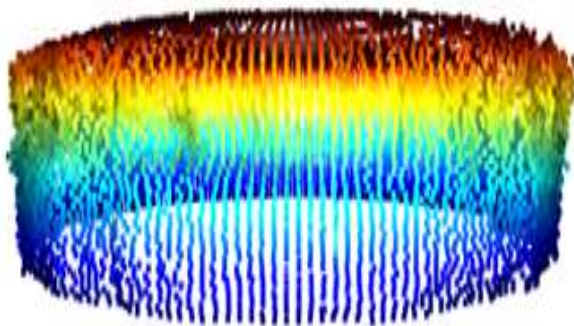
Marco teórico

“Un escáner 3D se trata de un dispositivo que tiene la capacidad de analizar un objeto o una escena para reunir datos sobre su forma y, ocasionalmente, su color” (Dynapro, 2021), para luego crear un modelo digital tridimensional del mismo. Para obtener un modelo tridimensional 3D se debe de crear una representación matemática del objeto en el cual el primer producto a obtener es una nube de puntos definida de la siguiente manera:

Nube de puntos

Es un conjunto de vértices o puntos tridimensionales con coordenadas x, y, z , es decir que cada punto consta de un sistema de coordenadas tridimensionales, los cuales crea la superficie de un objeto. *“Una nube de puntos es un sistema de representación de cualquier superficie a través de un conjunto de vértices (X,Y,Z) en un sistema tridimensional” (BIMnD, 2022)*

Cajete de barro



-Nube de puntos



-Original

Ilustración 7 Cajete de Barro, nube de puntos

A partir de la nube de puntos se puede generar la malla 3D o 3D Mesh.

Malla 3D

Es la construcción estructural de un modelo 3D, las cuales no contienen propiedades de masa, por lo que, se basan en un conjunto de vértices unidos por aristas y caras con la representación poligonal pueden crear una figura 3D.

“Un modelo de malla consta de vértices, aristas y caras que utilizan una representación poligonal” (AutoCAD, 2024)



Ilustración 8 Malla en 3D del cajete de barro

El escáner 3D debe contener las siguientes recomendaciones para el correcto proceso de la adquisición de datos: 1) Debe de estar en un lugar oscuro para el correcto funcionamiento del Sensor de distancia ir Sharp GP2Y0A21, recordando que funciona con la luz infrarroja y la luz solar puede interferir en el espectro infrarrojo ; 2) El objeto a escanear debe tener una distancia considerable que permita el correcto escaneo del objeto, en el rango más corto que permite el sensor le brinda mediciones de mayor resolución y la distancia de detección mínima baja hace que el sensor detecte objetos muy cercanos.

Sensor Sharp GP2Y0A21



Ilustración 9 Sensor Sharp

Es un sensor de medición de distancia de 10 cm a 80 cm, de tipo analógico, integrado por PSD (detector sensible a la posición) e IRED (Emisor de infrarrojos diodo). Su funcionamiento se basa en la triangulación, por lo que, la temperatura ambiente y el tiempo de ejecución no altera la

medición de los datos. "Un sensor SHARP es un sensor óptico capaz de medir la distancia entre él y un objeto, para esto el sensor con la ayuda de un emisor infrarrojo y un receptor miden la distancia usando triangulación" (Tutorial Sensor de Distancia SHARP, 2016)

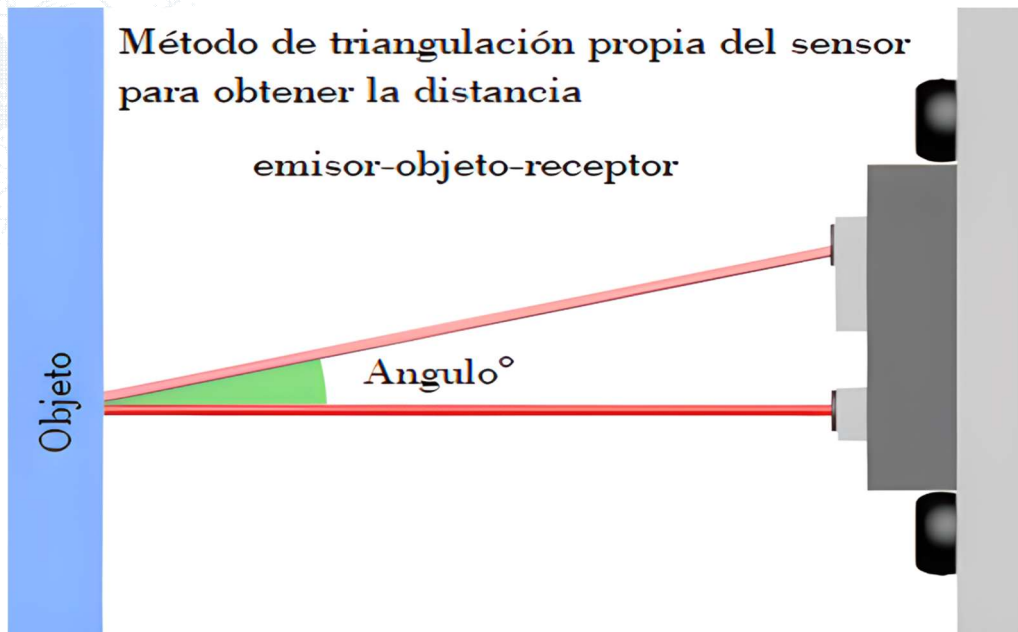


Ilustración 10 Método de triangulación para el sensor

Para obtener la nube de puntos de la figura a escanear, utilizaremos una base en la que el sensor este fijo, mientras el objeto gira 360°.

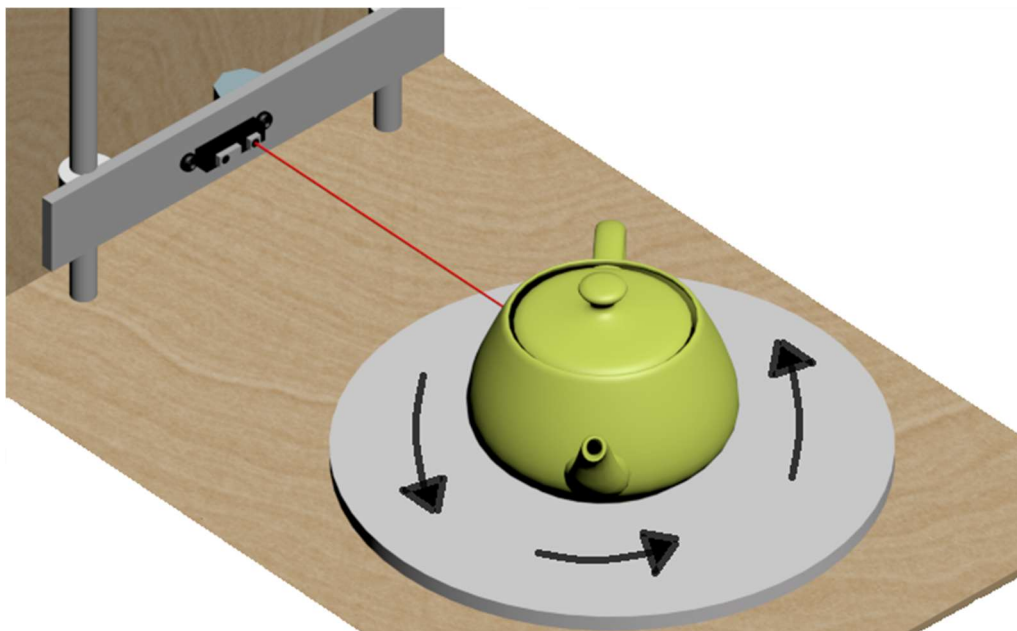


Ilustración 11 Plato Giratorio

Con la finalidad de que el objeto gire, se implementará una base giratoria que permita obtener todos los datos del objeto con más detalle desde cualquier ángulo, para lo que requeriremos de motores de paso que permitan controlar dichos ángulos.

Motor Nema 17



Ilustración 12 Motor Nema 17 Modelo 17HS4401

El motor Nema 17 es un motor de paso a paso, de tipo bipolar, con un ángulo de paso 1.8° , el cual necesita de 200 pasos para dar una vuelta de 360° . De acuerdo con su hoja de especificaciones, el motor Nema 17 posee un “torque de 4.0789 kg/cm (40 N.cm Min)” (MotionKing,2020). Son empleados para controlar la posición de un objeto.

Open3D

[Docs](#) » [Open3D: A Modern Library for 3D Data Processing](#)



Ilustración 13 Logo OPEN3D

Open3D es una librería para el procesamiento de datos 3D, en la cual se pueden realizar estructuras de datos, reconstrucciones de escenas, alineación de superficies, visualización 3D entre otras funcionalidades. Fue presentado por investigadores de Intel Labs, *“Open3D es una biblioteca de código abierto que admite el desarrollo rápido de software que trata con datos 3D”* (Zhou et al., 2018).

Arduino Uno

Es una placa de desarrollo de código abierto que está compuesto por un microcontrolador programable, *“basado en el microcontrolador ATmega328P”* (Aguayo, 2019).

La placa de Arduino uno, contiene una serie de pines de puertos de entrada y salida, que reciben señales digitales y analógicas y pueden conectarse con otros circuitos. El software utilizado es Arduino, en él se crean los programas dependiendo las necesidades que se requieran, después este archivo se programa en el controlador y finalmente se prueba en el circuito.



Ilustración 14 Tarjeta Arduino Uno

Spider Python

Es un entorno de desarrollo integrado para escribir scripts en Python de código abierto. Diseñado para el análisis de datos masivos, procesamiento de grandes volúmenes de información, ciencia de datos, entre otros. *“Spyder es un potente entorno científico escrito en Python”* (Welcome to Spyder’s Documentation — Spyder 5 Documentation, n.d.)

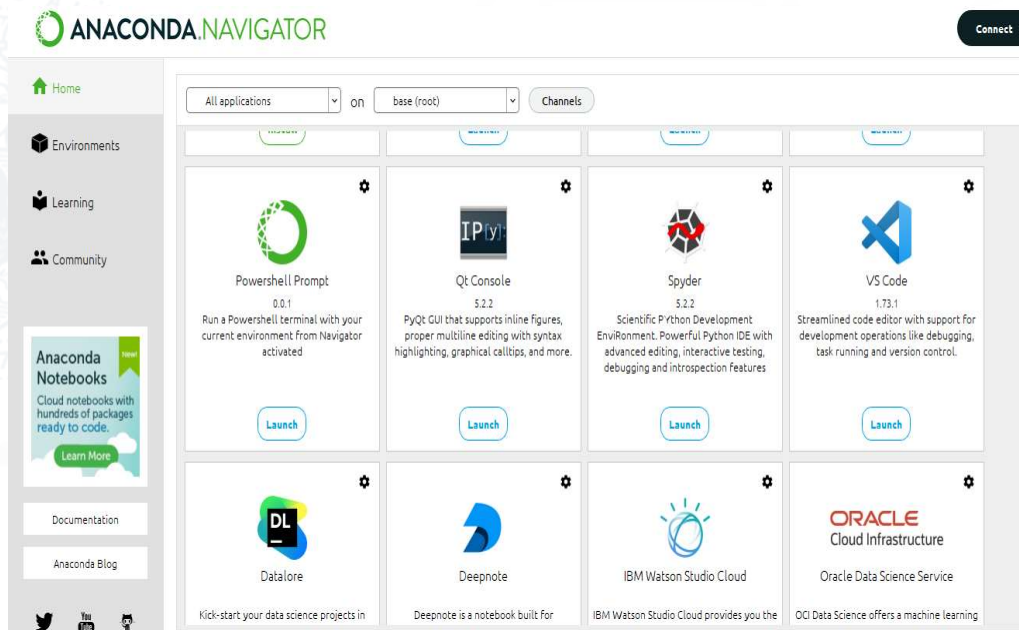


Ilustración 15 Ventana Principal de Anaconda Navigator

CorelDraw

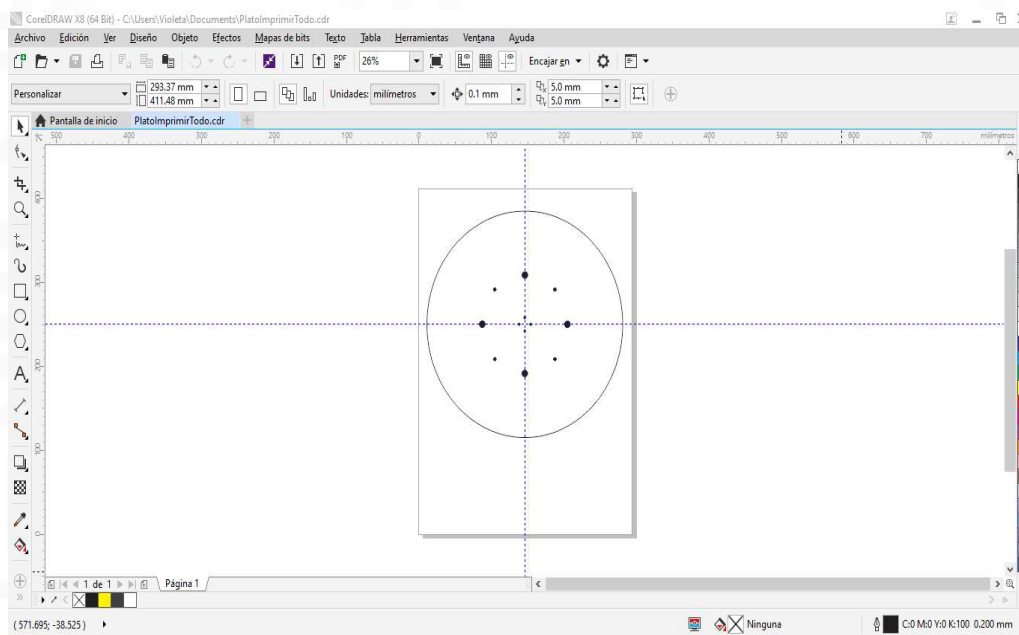


Ilustración 16 Software CorelDraw

CorelDRAW es un sofisticado software de diseño gráfico o aplicación que “permite a los usuarios crear diseños profesionales con ilustraciones vectoriales, diseño, edición de fotos, tipografía, herramientas de plantillas” (CorelDRAW, 2023). Entre otras aplicaciones, CorelDRAW permite realizar archivos que son enviados a las máquinas láser para realizar grabado o cortes en diferentes superficies.

Eagle

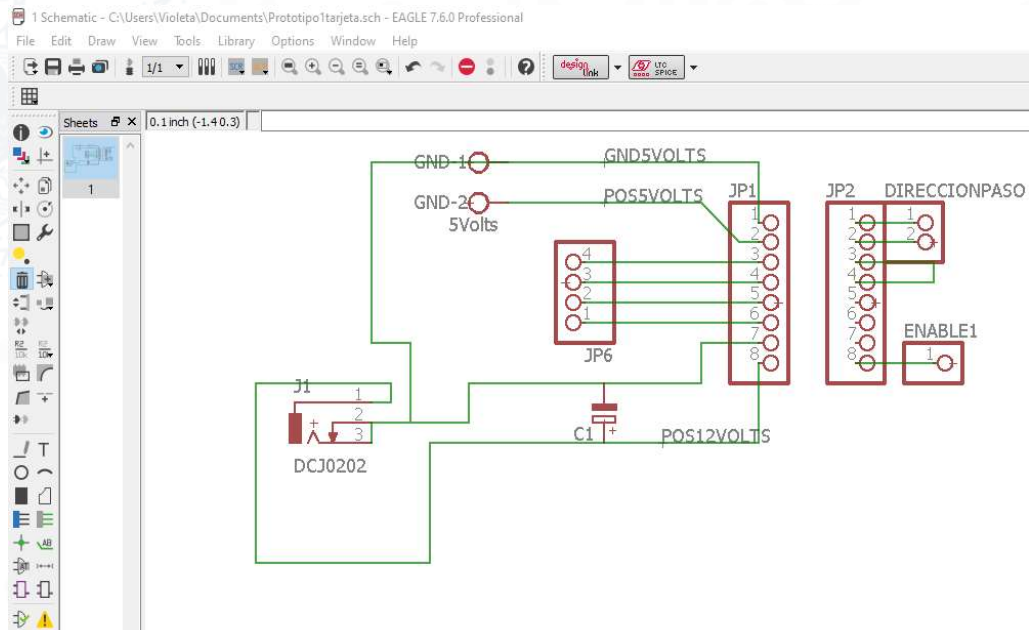


Ilustración 17 Software Eagle

“EAGLE es un software de automatización de diseño electrónico” (EAGLE | PCB Design and Electrical Schematic Software | Autodesk, 2023)

Es un software que permite a los usuarios diseñar esquemáticos electrónicos, placas de circuito impreso (PCB), provee de una gran cantidad de bibliotecas de componentes electrónicos en la red.

Estructura

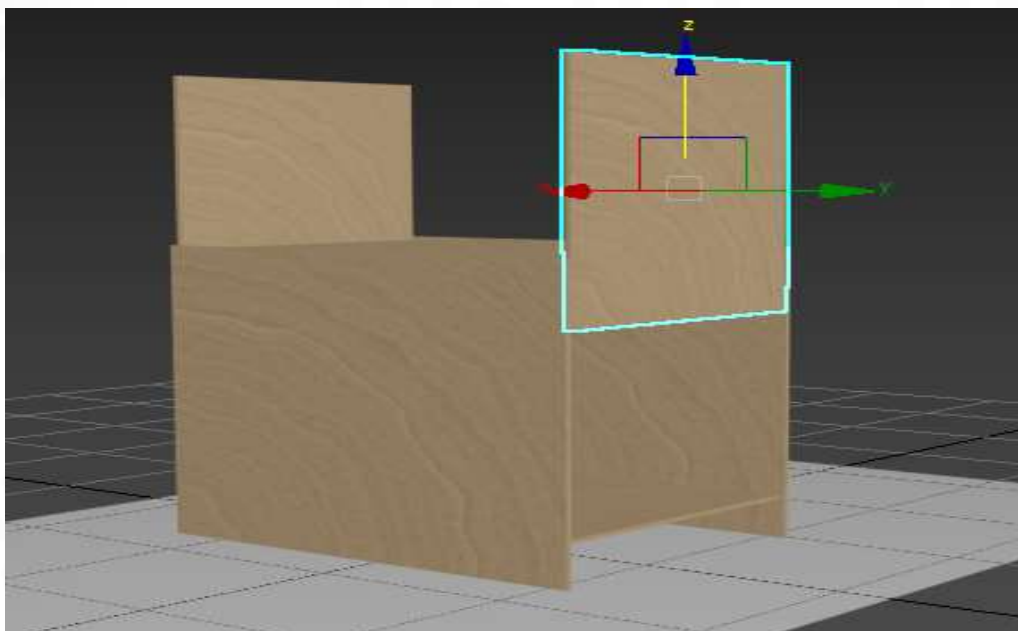


Ilustración 18 Estructura de Madera. Elaboración propia.

La función de la caja servirá para sostener el circuito, así mismo, para no permitir que ningún haz de luz interfiera con las lecturas del sensor y para poder escanear piezas de gran tamaño.

Placa giratoria metálica



Ilustración 19 Placa Giratoria metálica Yosoo

Permite una mejor estabilidad y soporte con una rotación de 360°, contiene rodamientos pre-lubricados, que permiten que se mueva de manera efectiva y soporta 50kg.

Plato giratorio de madera

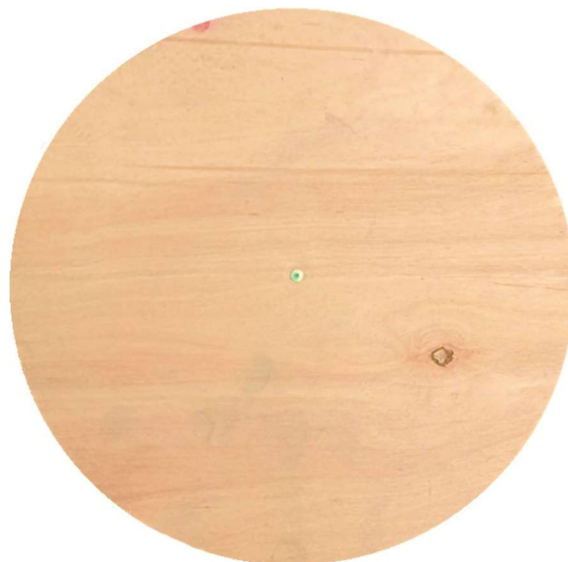


Ilustración 20 Plato de madera

La base giratoria permitirá ser un soporte para cargar o sostener los objetos a escanear. El plato giratorio debe tener o contar con los orificios que permitan atornillar la placa giratoria.



Capítulo IV

Diseño y desarrollo del sistema

Diseño y desarrollo del sistema

Mecánica

Al iniciar la construcción del escáner para la comunidad de Teteles de Santo Nombre y Santa María, se esperaba que fuera resistente y también que se adaptara a las necesidades que requería la comunidad.

Se eligió una estructura cuadrada para poder escanear elementos grandes con las siguientes medidas:

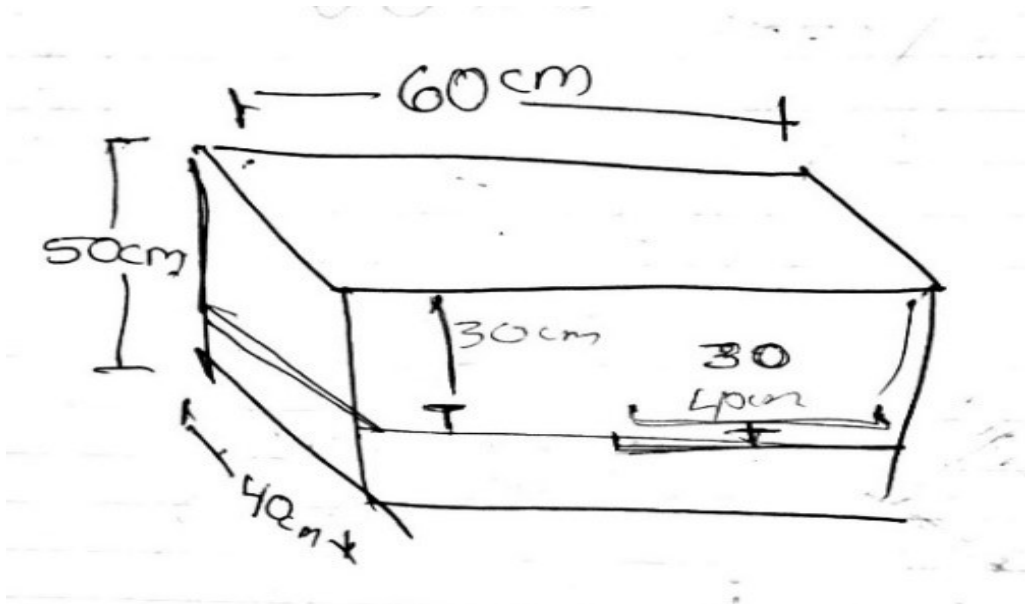


Ilustración 21 Boceto del diseño de la estructura. Elaboración propia.

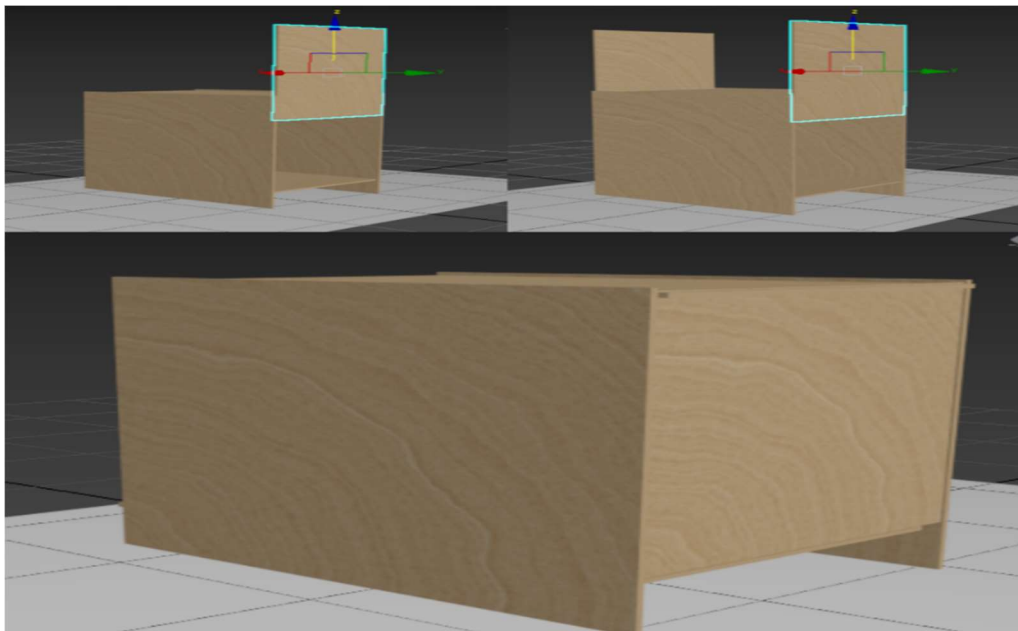


Ilustración 22 Estructura renderizada en 3dmax. Elaboración propia.

Se creó la primera estructura para sostener el escáner, la cual consta de una tuerca t8, que con ayuda de una varilla enroscada y un acoplador para el motor Nema 17, subirá los milímetros establecidos y 2 bloques deslizantes lineales que ayudarán a subir y bajar el sensor con el motor Nema 17.

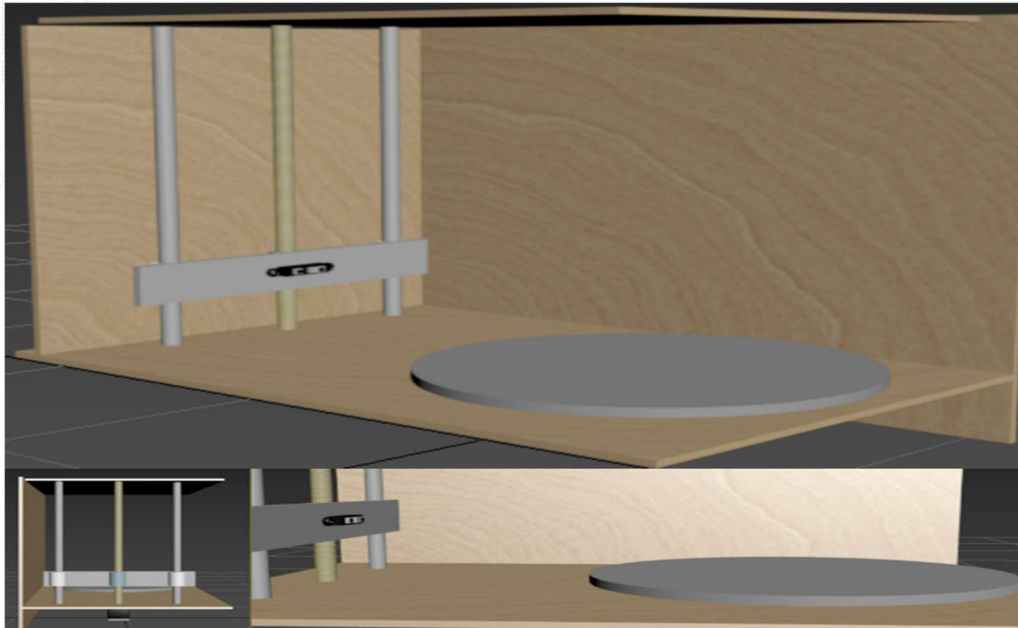


Ilustración 23 Estructura interna renderizada en 3dmax. Elaboración propia.

Se usa una escuadra de combinación para medir y calibrar el ángulo a 90°.



Ilustración 24 Escuadra de combinación

Se necesitan 2 soportes para los ejes guías para que sujete las varillas.

La base que sostenía el sensor se diseñó con el software CorelDRAW x8 con las medidas de la tuerca t8, y los orificios para los bloques deslizantes.

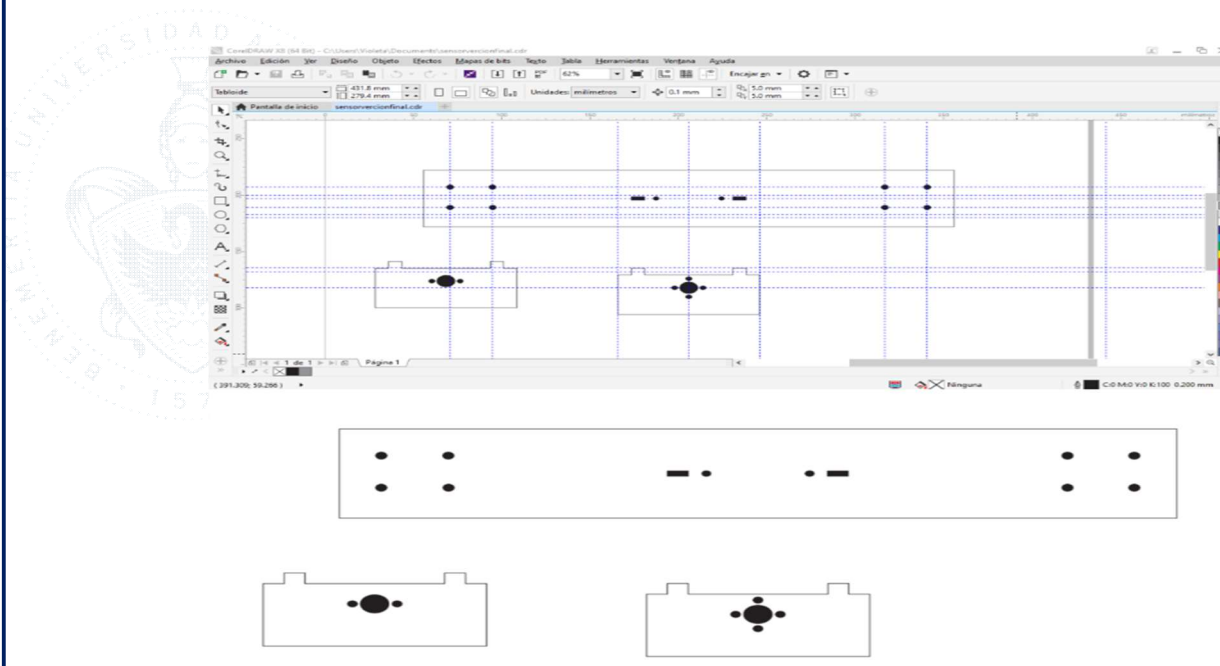


Ilustración 25 Diseño de la base sujetadora del sensor en CorelDRAW. Elaboración propia.

El material elegido para la base fue acrílico de 3mm, que es más ligero y se cortó con cortadora laser para tener mejor precisión.



Ilustración 26 Instalación de la base del sensor. Elaboración propia.

Se coloca una placa giratoria sobre la base metálica para apoyar el plato giratorio con ella y, a su vez, cargar el objeto para que gire 360° con el motor. La base giratoria se atornilló a la base del contenedor del escáner.

Para el motor Nema 17, se le coloca un acoplamiento de brida ajustable al motor Nema 17, que también permite atornillarse con el plato giratorio.

El plato giratorio se diseña con el software CorelDRAW x8, en el cual se contemplan los orificios del acoplamiento de brida y los orificios de la base metálica.

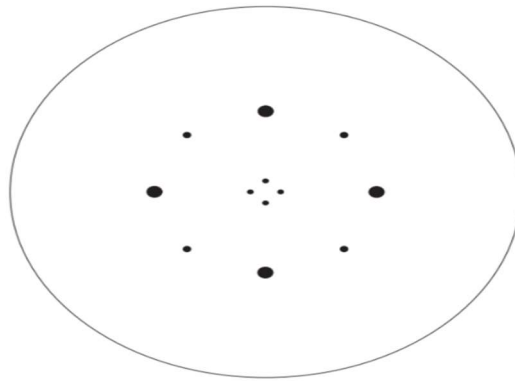
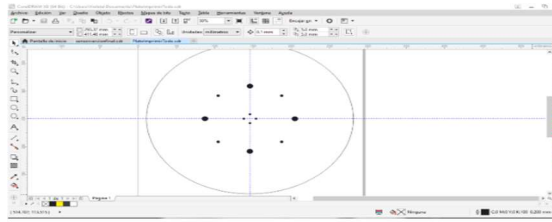


Ilustración 27 Diseño del plato giratorio con los orificios para su instalación en CorelDRAW. Elaboración propia.

El ensamble final queda con una altura de 14 mm.

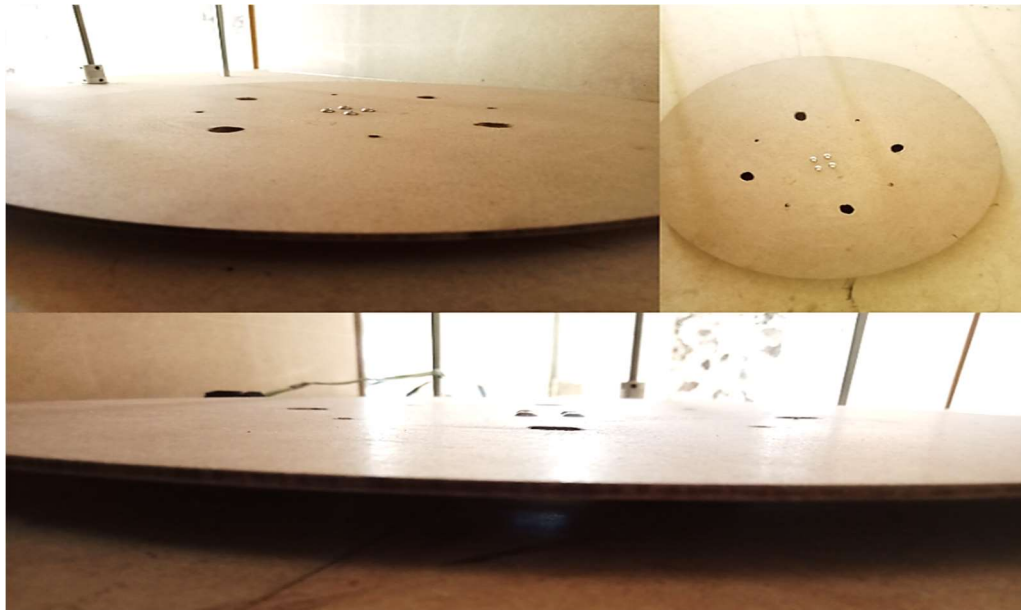


Ilustración 28 Instalación del plato giratorio. Elaboración propia.

A la misma altura se posiciona el sensor ir en la base creada para él.

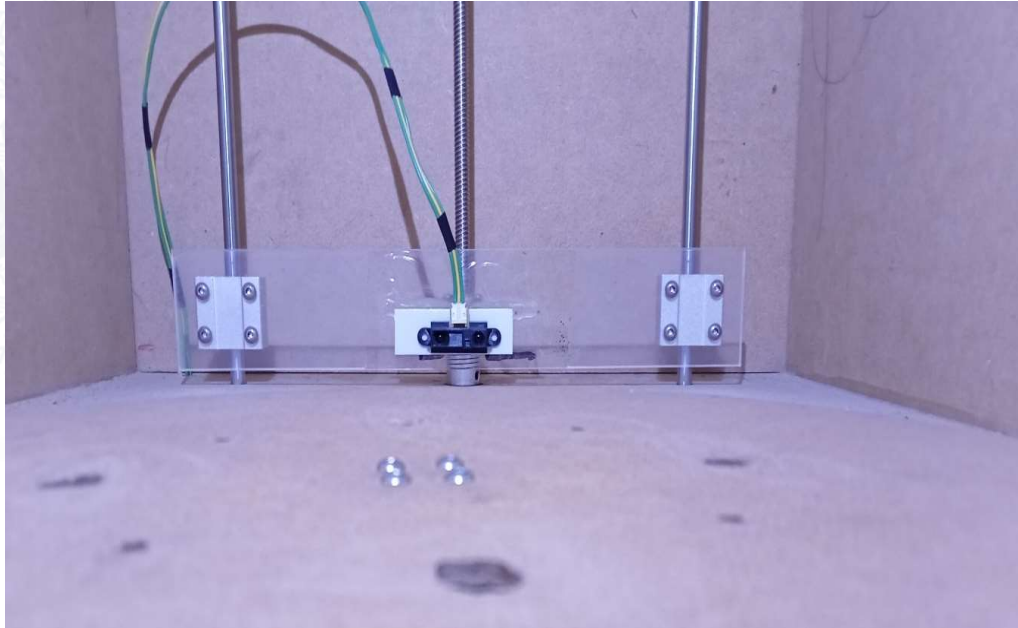


Ilustración 29 Sensor Posicionado a la altura del plato giratorio. Elaboración propia.

Para el funcionamiento del sensor, se sustituyó su cable original y se creó otro con más longitud, para que pudiera subir y bajar adecuadamente.

Electrónica

Para el funcionamiento del escáner, se contempló que fuera programado con la tarjeta Arduino y se realizó un diagrama esquemático del funcionamiento del escáner 3D, para ello:

Se investiga el funcionamiento de cada uno de los dispositivos electrónicos que lo conformarán y se hacen las pruebas correspondientes por separado.

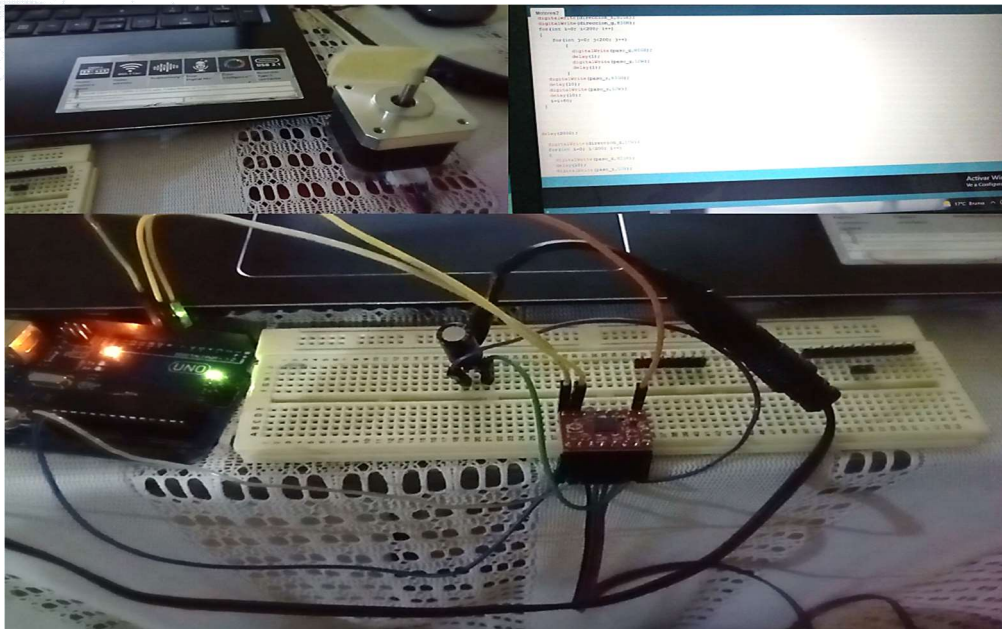


Ilustración 30 Pruebas del motor Nema 17

Se realiza un diagrama esquemático de las conexiones, para el funcionamiento del escáner 3D y se hacen pruebas con una *protoboard*.

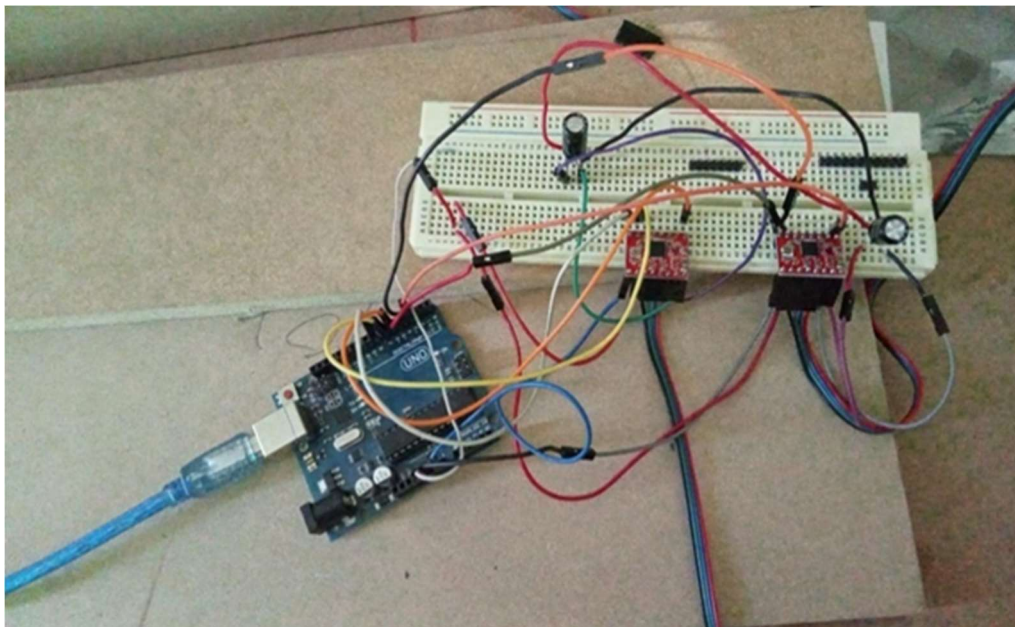


Ilustración 31 Pruebas en protoboard

En las pruebas, se estudia el funcionamiento de ambos motores, y las lecturas del sensor.

- Se expone a la luz ultravioleta sin que le dé luz exterior
- Se lava el bastidor y se revela el circuito

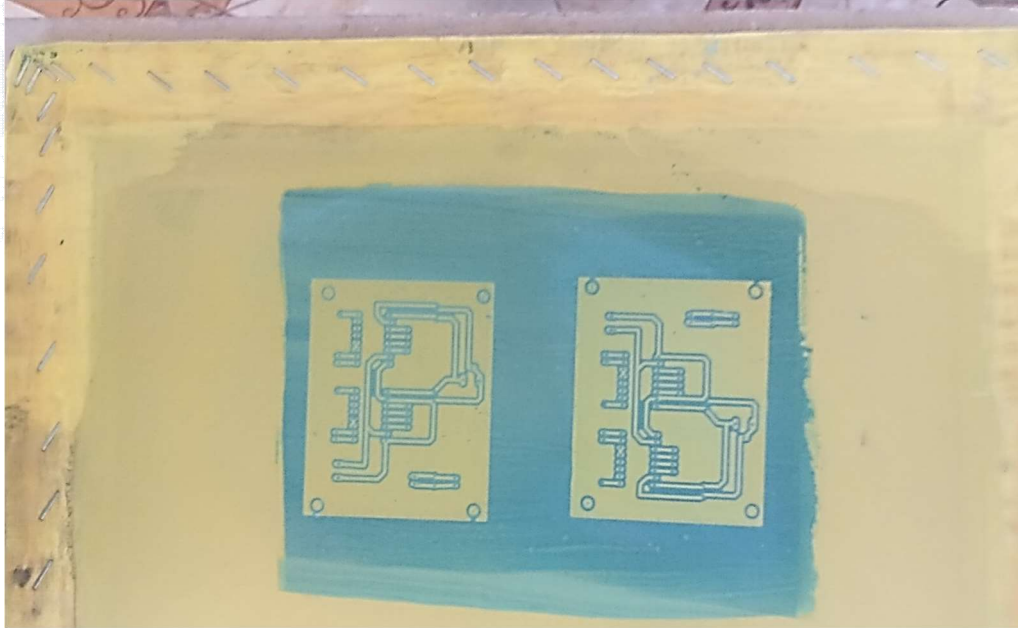


Ilustración 34 Circuito revelado en el bastidor

- Se coloca el bastidor sobre la baquelita
- Se le aplica la tinta uv y se imprime el circuito del bastidor
- Dejamos secar

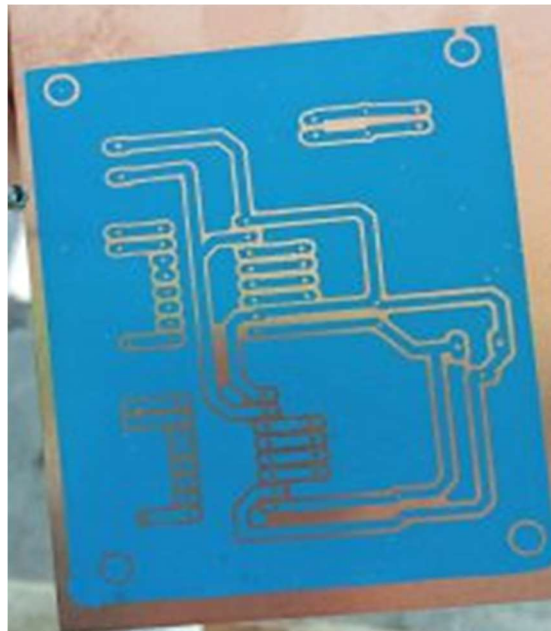


Ilustración 35 Circuito Impreso con tinta uv en baquelita. Elaboración propia.

- Coloca la baquelita impresa en cloruro férrico

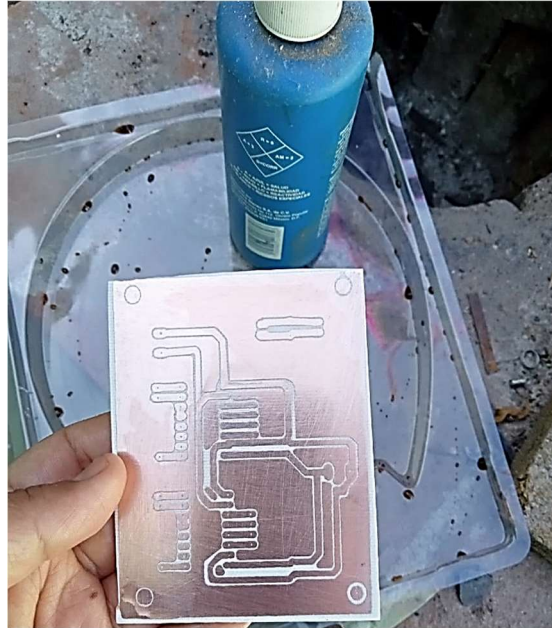


Ilustración 36 Placa lavada después del proceso de revelado de cobre con ácido férrico.

- Finalmente obtenemos la tarjeta que utiliza nuestro escáner 3D

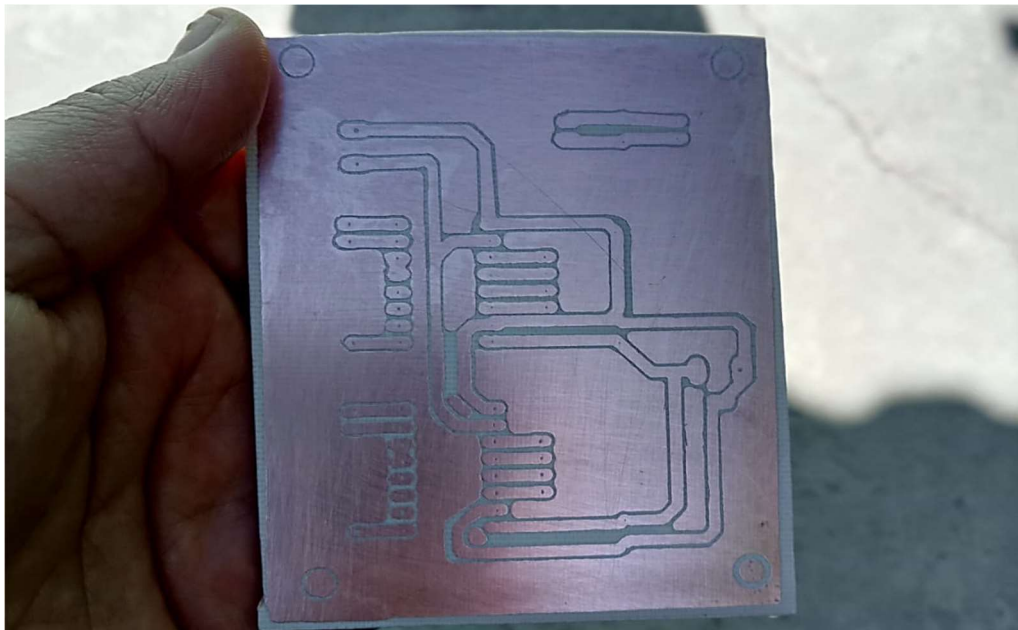


Ilustración 37 Tarjeta Finalizada. Elaboración propia.

Una vez terminada la tarjeta, procedemos a colocar todos los componentes electrónicos.

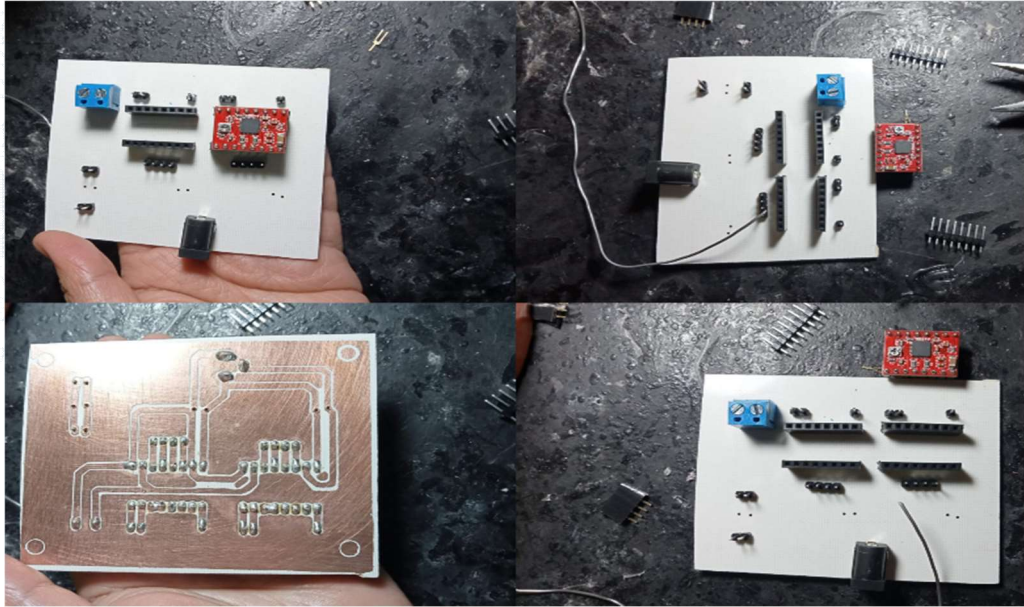


Ilustración 38 Colocación de componentes. Elaboración propia.

Tarjeta aún sin la adaptación para el sensor



Ilustración 39 Tarjeta sin la adaptación de sensor. Elaboración propia.

Tarjeta final con adaptación

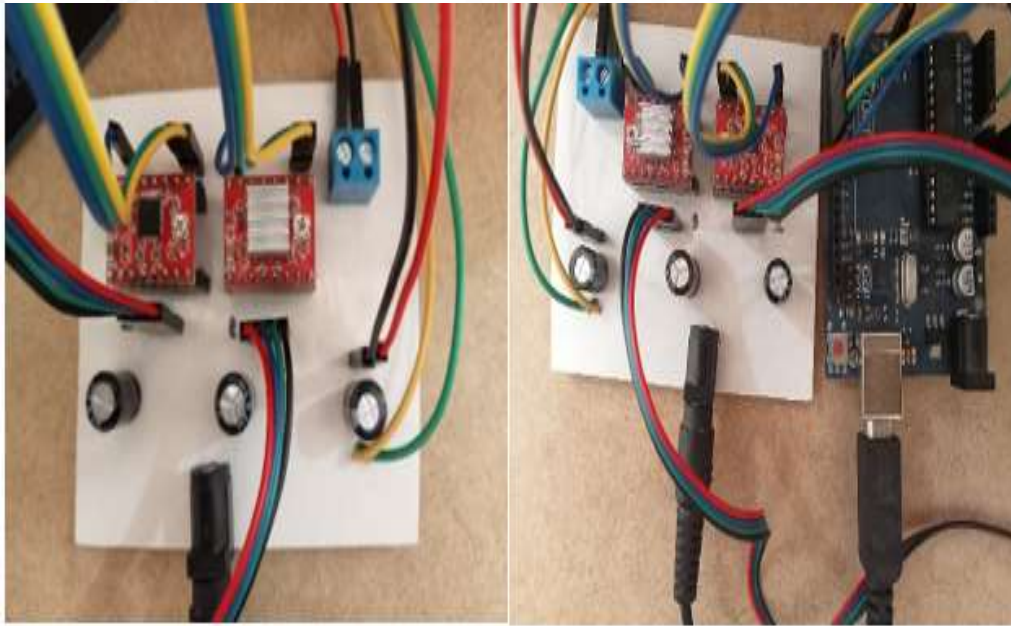


Ilustración 40 Tarjeta Final. Elaboración propia.

Para el funcionamiento del sensor, se hace la instalación que consiste en conectar las corrientes con un capacitor y se realiza un cable nuevo de mayor longitud, que permita la movilidad del sensor, ya que el original es muy pequeño. Este cable nuevo está soldado directamente a la tarjeta electrónica y su cable de datos está conectado al Arduino para obtener las lecturas del sensor.

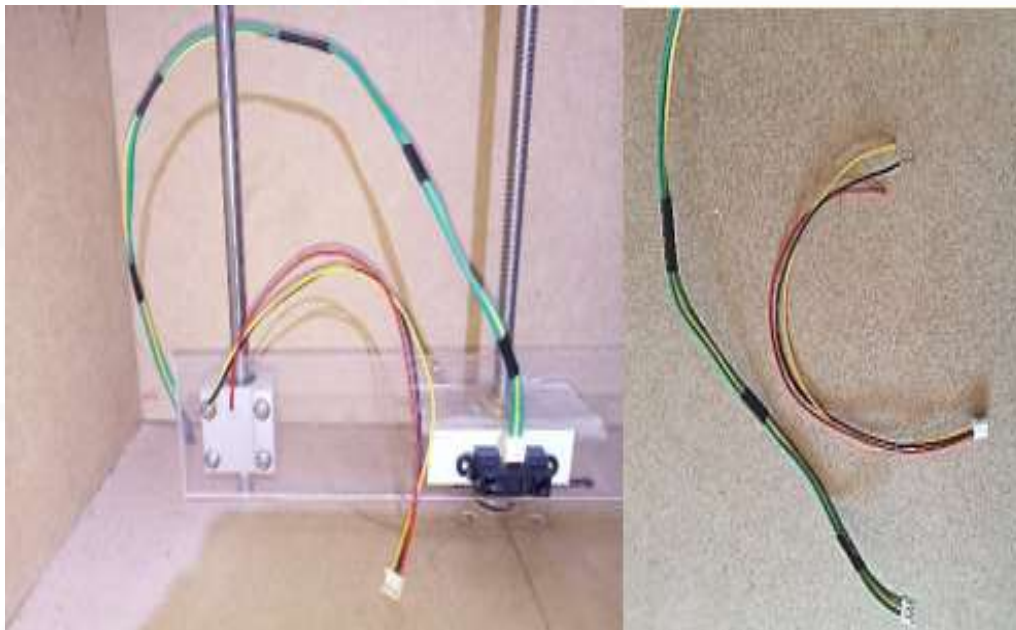


Ilustración 41 Cable creado para el sensor, diferencia de longitudes. Elaboración propia.

Se realizan las pruebas correspondientes para el uso del escáner.

Software

Para realizar la funcionalidad del escáner, se hace la simulación de multitasking en Arduino, utilizando la función *millis* que permite ejecutar varios componentes de manera simultánea, ya que la función *delay* al ser ejecutada, detiene completamente el procesador y no permite que alguna otra tarea pueda ser ejecutada, hasta que se termine el tiempo de retardo que se ha determinado en la función.

La función *millis* toma el tiempo desde que se inició el programa y permite cronometrar el tiempo, sin detener las operaciones de los demás eventos.

El funcionamiento del tiempo para el escáner se dividió en tres eventos de acuerdo con el esquema:

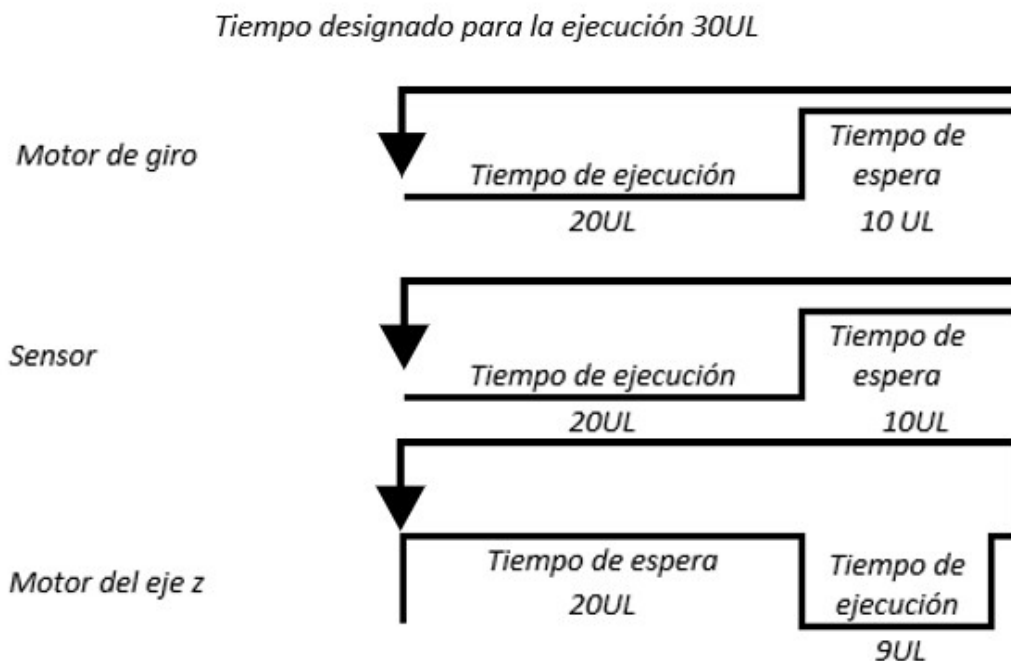


Ilustración 42 Esquema del tiempo de ejecución

La ejecución de cada evento se hace por separado en el ciclo establecido del Arduino. Los tiempos deben coincidir para un correcto funcionamiento del escáner, ya que ahora no dependen de los tiempos de los demás eventos para ejecutarse.

- ▶ El motor de giro se ejecutará por 20UL = 20 milisegundos y esperará 10UL para que el motor de eje z se ejecute.
- ▶ En conjunto, el sensor irá capturando los datos de distancia cada vez que el motor de giro dé un paso. Cuando el motor de giro y el sensor terminen su ejecución, ambos esperarán 10UL.
- ▶ El motor del eje z esperará 20UL y se ejecutará solo 9UL, para dejar 1UL de espera y que los tres eventos comiencen al mismo tiempo.
- ▶ Para que no existan un desfase de tiempo, se implementaron condiciones que ayudarán a mejorar su funcionamiento.

- ▶ Concluidos los eventos, se irá incrementando el tiempo de forma periódica, cada vez más adelante del tiempo del inicio del programa, hasta terminar el escaneo completo.

Al iniciar la ejecución del programa del escáner, se tiene un retardo de 7 segundos, tiempo usado para revisar que todo esté en orden, una vez ejecutado este evento se captura el tiempo actual de la ejecución con la siguiente asignación:

```
unsigned long actual = millis();
```

Para cronometrar el tiempo de forma periódica se realizó la siguiente función:

- Si $actual > evento_motor1$
 - $evento_motor1 = evento_motor1 + intervalo$
- Fin Si

donde:

$evento_motor1$ es una variable, que irá incrementando el tiempo y el intervalo es el tiempo definido para el incremento, en este caso de 30 milisegundos. La función *millis* se mueve sobre una línea de tiempo, cuando es ejecutada devuelve el valor en el que se encuentra y el evento será ejecutado cuando *millis* pase por esa posición.

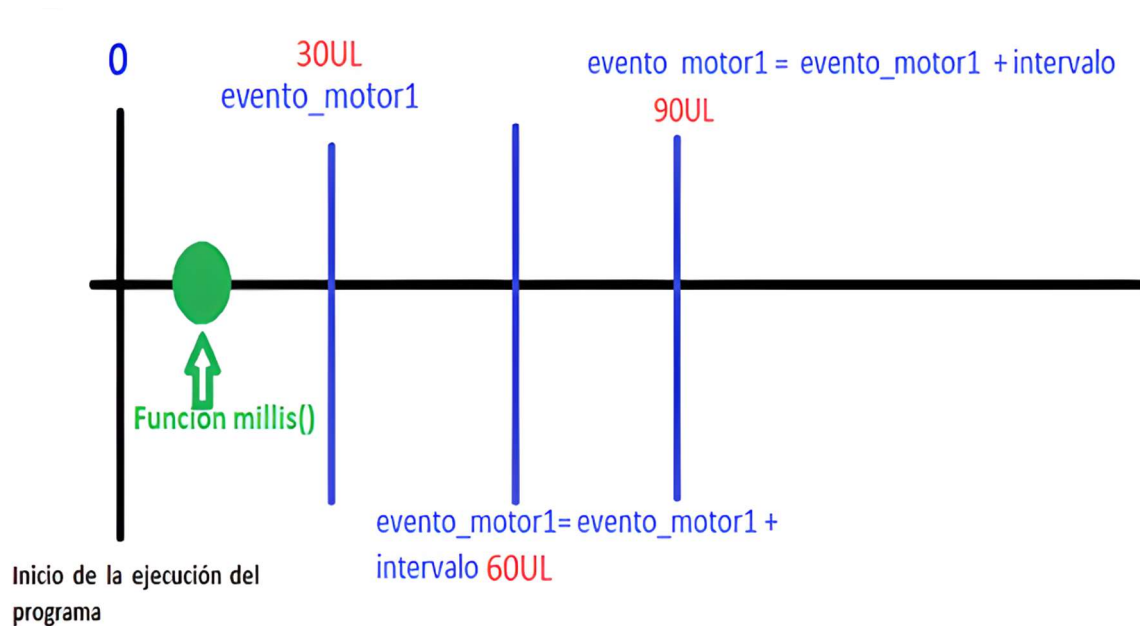


Ilustración 43 Representación del uso de la función millis

Desde el principio de la adquisición del sensor, se empezaron a hacer pruebas de funcionamiento. Para cumplir con el objetivo, requerimos que el sensor obtenga datos en forma flotante o bien en punto decimal.

Se realizó un pequeño programa, para que el sensor obtenga los datos decimales, cuyo pseudocódigo consiste en:

- Inicio
 - Leer la entrada analógica 100 veces (declarada para el sensor)
 - Obtener el promedio
 - Almacenar el promedio en una variable
 - Obtener la distancia en cm con la formula $\text{Distancia_Cm} = 17569.7 * [\text{promedio}]^{-1.2062}$
 - Imprimir en pantalla la distancia en cm
- Fin

Nota: Los valores 17569.7 y -1.2062 pueden cambiar, de acuerdo con la calibración que se le dé al sensor.

Estos valores son los más usados:

“Formula de distancia en cm” (Tutorial Sensor De Distancia SHARP, 2016)

$$L = ax^b$$

donde se obtienen los valores X_1 y X_2 , que son las lecturas del sensor(tensión), L_1 y L_2 son las distancias del sensor al objeto medido manualmente. Por ejemplo, podemos decir entonces que cuando la tensión de $X_1 = 350$, la medida manual es de $L_1 = 15$ cm.

“Matriz” (Tutorial Sensor De Distancia SHARP, 2016)

$$\begin{array}{ll} X_1 = 350 & L_1 = 15 \\ X_2 = 129 & L_2 = 50 \end{array}$$

Sustituimos por los valores en cada caso, para formar la ecuación lineal:

$$15 = a350^b$$

$$50 = a129^b$$

Despejamos la incógnita a de ambas ecuaciones:

Aplicamos el método de igualación que consiste en despejar la misma incógnita de las ecuaciones dadas y posteriormente hacer la igualación de ambas incógnitas.

Caso 1

$$L = ax^b$$

$$15 = a(350)^b$$

$$\frac{15}{350^b} = a$$

Caso 2

$$L = ax^b$$

$$50 = a(129)^b$$

$$\frac{50}{129^b} = a$$

Por el método de igualación tenemos $a = a$:

$$\frac{15}{350^b} = \frac{50}{129^b}$$

Resolvemos:

$$\frac{15}{350^b} = \frac{50}{129^b} \Rightarrow \frac{15}{50} = \frac{350^b}{129^b} \Rightarrow \frac{3}{10} = \left(\frac{350}{129}\right)^b$$

Aplicando $a^x = b$ es igual a $x = \log_a b$ por propiedad logarítmica

$$\left(\frac{350}{129}\right)^b = \frac{3}{10} \Rightarrow b = \log_{\left(\frac{350}{129}\right)}\left(\frac{3}{10}\right) \Rightarrow b = -1.20623$$

Sustituyendo el valor de "b" en cualquiera de las ecuaciones dadas para obtener el valor de "a"

$$\frac{15}{350^b} = a \Rightarrow \frac{15}{350^{-1.2062}} = a \Rightarrow a = 17568.96324$$

Quedando como final:

$$L = 17568.96324x^{-1.206223}$$

Es importante revisar siempre la calibración del sensor para obtener buenos resultados en caso de ser diferentes a las medidas dadas se deben de calibrar nuevamente.

Para la calibración del escáner 3D se ocuparon los siguientes valores:

$$X_1 = 323 \quad L_1 = 15 \text{ cm}$$

$$X_2 = 115 \quad L_2 = 42 \text{ cm}$$

Sustituimos por los valores en cada caso para formar la ecuación lineal:

$$15 = a323^b$$

$$42 = a115^b$$

Despejamos la incógnita a de ambas ecuaciones:

$$\frac{15}{323^b} = a$$

$$\frac{42}{115^b} = a$$

Por el método de igualación tenemos $a = a$:

$$\frac{15}{323^b} = \frac{42}{115^b}$$

Resolvemos

$$\frac{15}{323^b} = \frac{42}{115^b} \Rightarrow \frac{15}{42} = \frac{323^b}{115^b} \Rightarrow \frac{15}{42} = \left(\frac{323}{115}\right)^b$$

Aplicando $a^x = b$ es igual a $x = \log_a b$ por propiedad logarítmica

$$\left(\frac{323}{115}\right)^b = \frac{15}{42} \Rightarrow b = \log_{\left(\frac{323}{115}\right)}\left(\frac{15}{42}\right) \Rightarrow b = -0.9969974658$$

Sustituyendo el valor de "b" en cualquiera de las ecuaciones dadas para obtener el valor de "a"

$$\frac{15}{323^b} = a \Rightarrow \frac{15}{323^{-0.9969}} = a \Rightarrow a = 4758.995055$$

Quedando como final:

$$L = 4758.9x^{-0.9969}$$

Programa de pruebas en Arduino con los valores más comunes:

```

PruebaSensorSharp

void loop() {
  // Leemos la entrada analógica 0 :
  float distancia = promedio(100);
  Serial.print(distancia);
  Serial.print(" ");

  float distancia_cm = 17569.7*pow(distancia, -1.2062);
  Serial.print(distancia_cm);
  Serial.print("cm ");
  Serial.println("");
  delay(5000);
}

double promedio(int n)
{
  long suma=0;
  for(int i=0;i<n;i++)
  {
    suma=suma+analogRead(A0);
    delay(2);
  }
  return (suma/n);
}
  
```

COM3

```

251.00 22.40cm
251.00 22.40cm
249.00 22.62cm
250.00 22.51cm
251.00 22.40cm
250.00 22.51cm
251.00 22.40cm
251.00 22.40cm
250.00 22.51cm
249.00 22.62cm
250.00 22.51cm
250.00 22.51cm
251.00 22.40cm
251.00 22.40cm
250.00 22.51cm
  
```

Autoscroll Mostrar mapa

Ilustración 44 Programa de pruebas para el sensor y lecturas del sensor

Para la implementación del escáner 3D se necesitan 2 motores Nema 17.

El primero para el giro de 360°, que debe hacer la pieza para poder capturar todos los puntos en ella y, el segundo, para el eje z que subirá y bajará los milímetros o centímetros establecidos dependiendo los pasos determinados. El eje z incrementará su valor cada vez que acabe la rotación del motor de giro.

El motor Nema 17, para el giro, debe dar un paso y capturar la lectura del sensor y dar otro paso, así sucesivamente hasta llegar a los 200 pasos, en ese momento nuestro objeto habrá dado los 360° y siempre se le pasará el valor de z en el que se encuentra. La función de prueba para el motor de giro está determinada de la siguiente manera:

- Para (i=0, i<200, i incremento de 1)
 - Declaramos paso en HIGH
 - Esperamos 10 milisegundos
 - distanciaCm_Total(z);
 - Declaramos paso en LOW
 - Esperamos 10 milisegundos
- Fin Para

Programa en Arduino para el giro:



```
Motor_sha3 Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

Motor_sha3

void loop() {

  digitalWrite(direccion_g,HIGH); //declaro la direccion del motor

  for(int i=0; i<giroC; i++) //Giro 200 pasos = 360° cada paso es equivalente a 1.8°
  {
    digitalWrite(paso_g,HIGH);
    delay(10);
    distanciaCm_Total(z);
    digitalWrite(paso_g,LOW);
    delay(10);
  }

  delay(2000);
```

Ilustración 45 Programa de pruebas para el giro

Para la variable z, que determinará la altura conforme avance el motor, está establecida de la siguiente manera:

Donde z empieza en 0cm y se maneja un tornillo T8 = 8 milímetros

1 vuelta del motor Nema 17 consta de 200 pasos = 8 milímetros

La función para el eje z entonces consiste en la siguiente fórmula:

$$\frac{200 \text{ pasos}}{0.8 \text{ milímetros}} = \frac{1 \text{ paso}}{x}$$

$$x = \frac{0.8 * 1}{200}$$

$$x = 0.004 \text{ milímetros}$$

1 paso = 0.004 milímetros

Tabla del eje z

Tabla 2 Datos del eje Z

Pasos mínimos del motor Nema 17	Medida en eje z
23	0.092 mm
28	0.112 mm
31	0.124 mm
35	0.14 mm
38	0.152 mm
39	0.156 mm
42	0.168 mm
43	0.172 mm
44	0.176 mm
46	0.184 mm
48	0.192 mm
50	0.2 mm
52	0.208 mm
54	0.216 mm
56	0.224 mm
57	0.228 mm
59	0.236 mm
60	0.24 mm

Pasos mínimos del motor Nema 17	Medida en eje z
62	0.248 mm
63	0.252 mm
67	0.268 mm
68	0.272 mm
71	0.284 mm
73	0.292 mm
74	0.296 mm
76	0.304 mm
80	0.32 mm
83	0.332 mm
84	0.336 mm
85	0.34 mm
90	0.36 mm
100	0.4 mm
125	0.5 mm
200	0.8 mm
225	0.9 mm
250	1 cm

La variable z irá incrementando, de acuerdo con los pasos que el usuario establezca.

Donde:

0.004 milímetros = 1 paso del motor Nema 17

z = variable que irá incrementando la altura

Navance= número de pasos que queremos que avance el motor que equivale a una altura en mm

Ecuación resultante:

$$z = z + (0.004 * Navance)$$

Ejemplo:

Navance = 74 pasos

Sustituimos en la ecuación:

$$z = z + (0.004 * Navance)$$

$$z = z + (0.004 * 74)$$

$$z = z + 0.296 \text{ mm}$$

La función del motor del eje z y la asignación de la variable z, está determinada de la siguiente manera:

Se declara el número de pasos para que avance del motor con forme a la altura que se establezca:

- Para (i=0, i< número de pasos para que avance, i incremento de 1)
 - Declaramos paso en HIGH
 - Esperamos 1 milisegundos
 - Declaramos paso en LOW
 - Esperamos 1 milisegundos
- Fin Para
- Incrementa z

Programa en Arduino para el eje z y la asignación para la variable:

The image shows a screenshot of the Arduino IDE interface. At the top, the title bar reads "Motor_sha3 Arduino 1.8.19". Below the title bar is a menu bar with "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". A toolbar with icons for save, undo, redo, and other functions is visible. The main editor area shows the following code:

```
digitalWrite(direccion_z, HIGH);

for(int i=0; i<Navance; i++)
{
  digitalWrite(paso_z, HIGH);
  delay(1);
  digitalWrite(paso_z, LOW);
  delay(1);
}

delay(2000);

// para el incremento en cada paso
// numero de pasos, por cada paso se suma 0.004 milímetros
// z=z+0.004; que es el valor de cada paso; 1 paso de 1.8° es igual a 0.004 milímetros

z=z+(0.004*Navance);
```

Ilustración 46 Programa de pruebas para el eje Z

Para calcular las coordenadas x , y , z se creó la siguiente función que consiste en obtener la distancia del objeto en cm



Ilustración 47 Distancia leída del sensor al objeto

Obtenemos la distancia del objeto restando la distancia que existe entre el sensor y el centro del plato giratorio menos la distancia en cm obtenida del sensor.

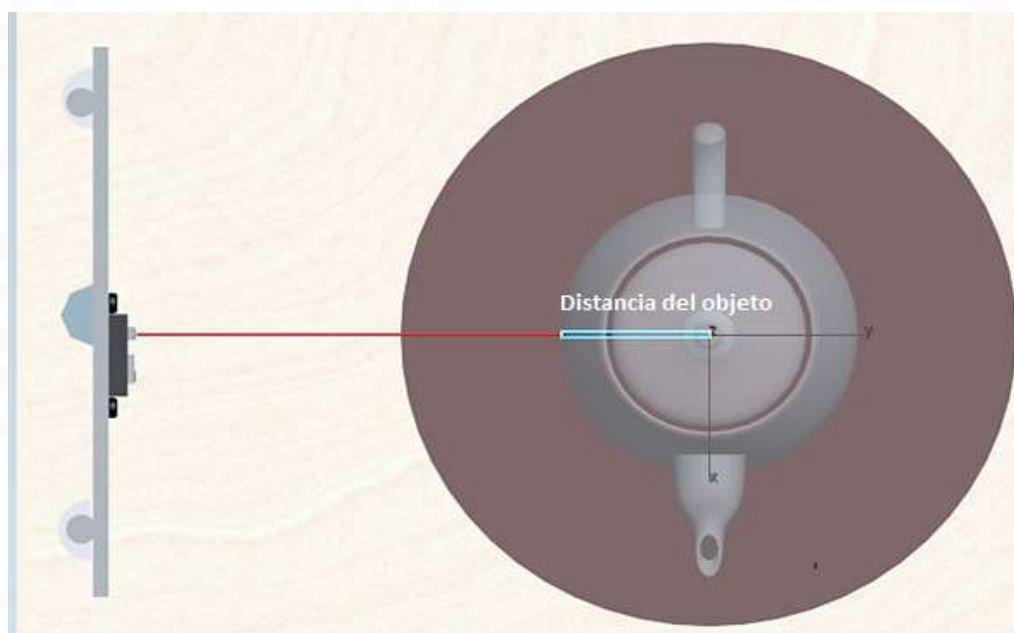


Ilustración 48 Distancia del punto medio del objeto al contorno del objeto

Programa en Arduino de la distancia del objeto

```
long suma=0; // variable de suma
for(int i=0;i<muestras;i++)// contamos lo que va leyendo nuestro sensor
{
    suma=suma+analogRead(A0); //sumamos lo de nuestro sensor
    delay(2);
}
float distancia = suma/muestras; //Sacamos el promedio
float distancia_cm = 17569.7*pow(distancia, -1.2062);
totalDistancia=puntoMedio-distancia_cm;
```

Ilustración 49 Programa en Arduino para la captura del contorno del objeto

El motor da un paso, se almacena la distancia leída y se forma un ángulo de 1.8° , así sucesivamente hasta formar los 360° que representan 200 pasos. El ángulo puede cambiar de acuerdo con el número de pasos que queremos que se mueva el plato giratorio, recordando que cada paso es de 1.8° , si queremos que se mueva cada 54° debemos definir el número de pasos como 30.

En este caso es paso a paso de 1.8° para no perder información.

Analizaremos el de 45°

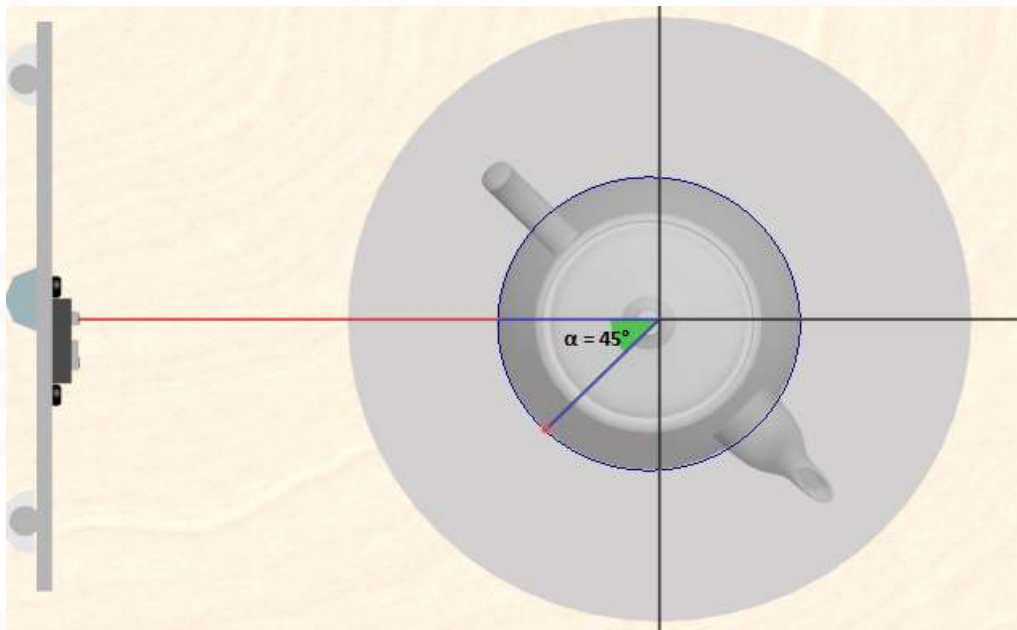


Ilustración 50 Punto con Angulo de 45°

Obtenemos el punto mediante las funciones trigonométricas

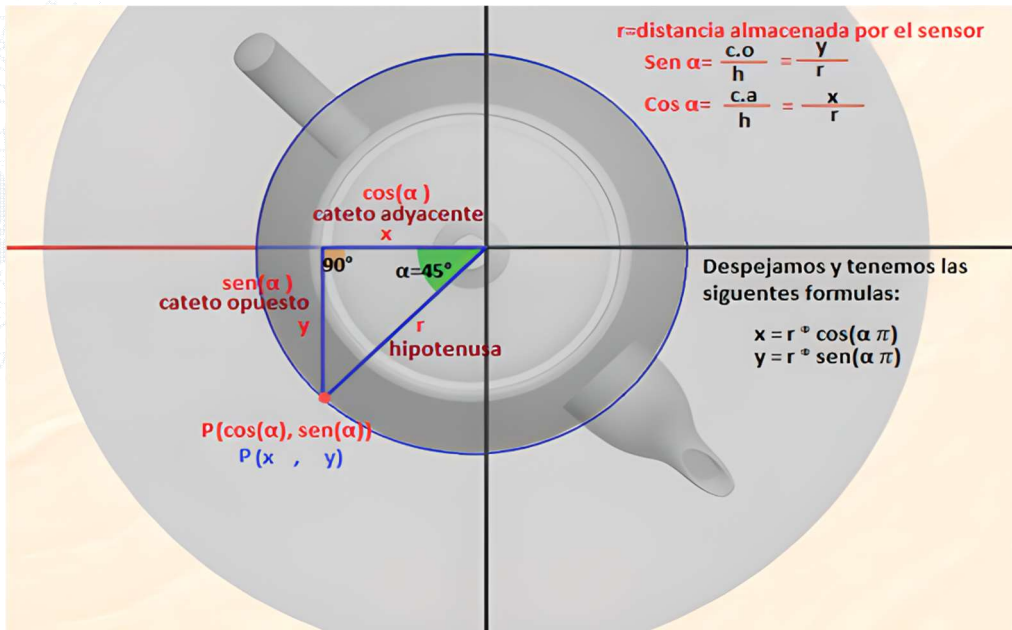


Ilustración 51 Obtención del $P(x, y)$ con trigonometría

En cada paso, se almacena la distancia leída y se forma un ángulo. En este momento es donde se aplican las funciones trigonométricas.

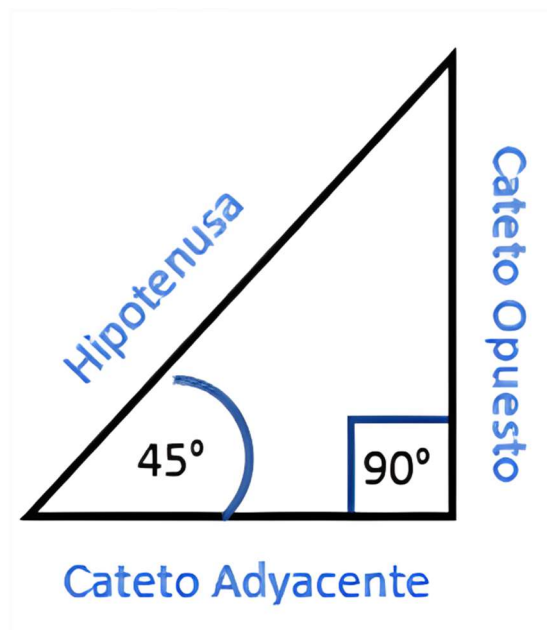


Ilustración 52 Triángulo rectángulo



$$\sin \alpha = \frac{C.O}{H}$$

$$\cos \alpha = \frac{C.A}{H}$$

En donde utilizamos seno y coseno porque solo conocemos:

r = distancia almacenada

α = ángulo

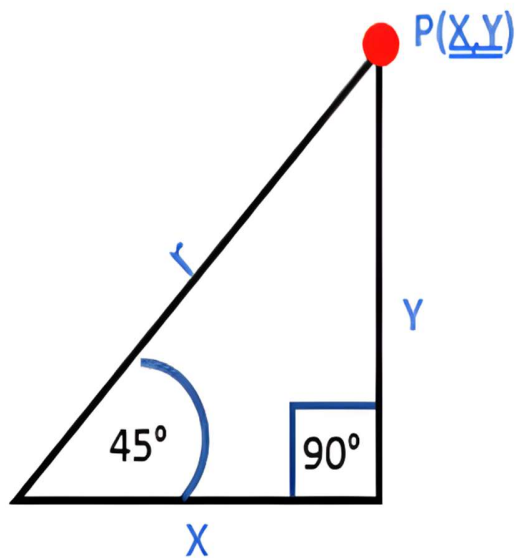


Ilustración 53 Triángulo rectángulo, obtención del $P(x, y)$

$$\sin \alpha = \frac{y}{r}$$

$$y = r * \sin \alpha$$

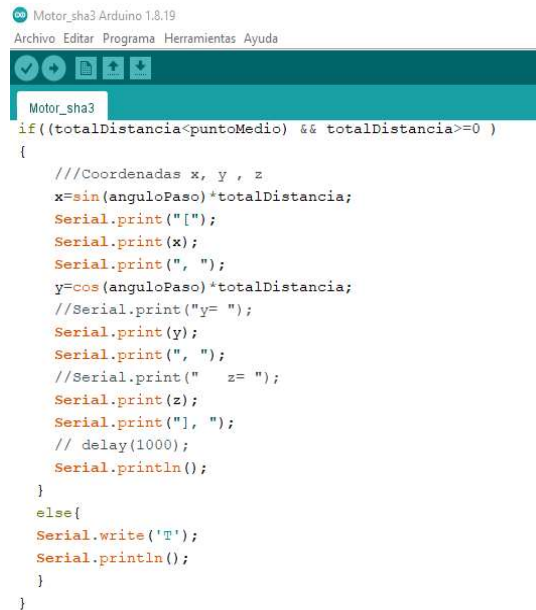
$$\cos \alpha = \frac{x}{r}$$

$$x = r * \cos \alpha$$

Por lo tanto, el punto P corresponde a: $P(\cos \alpha, \sin \alpha) = P(x, y)$.

Cada ciclo del motor del eje z se irá incrementando la variable z y cada paso que haga el motor de giro, preguntará cuál es el valor de z, conformando así el punto P(x,y,z) que almacenamos en un archivo para formar la nube de puntos correspondiente a la figura que se encontrará escaneando.

Programa en Arduino de las coordenadas x, y, z



```
Motor_sha3 Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
Motor_sha3
if((totalDistancia<puntoMedio) && totalDistancia>=0 )
{
  //Coordenadas x, y , z
  x=sin(anguloPaso)*totalDistancia;
  Serial.print("[");
  Serial.print(x);
  Serial.print(", ");
  y=cos(anguloPaso)*totalDistancia;
  //Serial.print("y= ");
  Serial.print(y);
  Serial.print(", ");
  //Serial.print(" z= ");
  Serial.print(z);
  Serial.print("], ");
  // delay(1000);
  Serial.println();
}
else{
  Serial.write('\n');
  Serial.println();
}
}
```

Ilustración 54 Programa en Arduino para el eje x y el eje y

Implementación en Python

Se realizó un programa en Python en Spyder, para que los datos almacenados se procesen y, así, generar la nube de puntos.

Se utilizó la distribución anaconda para los lenguajes de programación en Python, ya que es utilizada para el procesamiento de grandes volúmenes de información, ciencias de datos, entre otros.

Para generar los modelos 3D en este caso, la nube de puntos utiliza la librería open3D para el procesamiento de datos. Al enviar las coordenadas x, y, z en la consola de Arduino, se creó un programa capaz de obtener los datos de la consola y guardarlos en un archivo. Cuando el objeto termine de escanear, tendrá que leer nuevamente el archivo y generar una matriz que se interpretará con open3D para la creación de la nube de puntos.

Se creó un archivo de tipo .txt, para recolectar todos los datos que arroja el escáner 3D y se implementó la conexión serial del puerto "COM3" a una velocidad de transmisión estándar de 9600 bps.

Función en Spyder para crear el archivo y la conexión en Arduino.

Asignación en Python:

```
Nubedepuntos= open('C:\\Users\\Violeta\\Documents\\Nubedepuntos.txt', 'w+')  
arduino = serial.Serial('COM3', 9600)  
time.sleep(1)
```

Ilustración 55 Creación del documento txt en Python y conexión de Python con Arduino

Para la conexión del puerto COM3 con Spyder, se debe contemplar que la consola de software Arduino no se encuentre abierta ya que, el programa en Spyder abre la consola para ir capturando todos los datos en ella, de lo contrario mandará un error.

Error en Spyder

```
SerialException: could not open port 'COM3': FileNotFoundError(2, 'El sistema no puede encontrar el archivo especificado.', None, 2)
```

Ilustración 56 Error en Spyder, cuando se está ejecutando la consola de Arduino

Error en Arduino

```
Tarjeta en COM3 no disponible  
El Sketch usa 5422 bytes (16%) del espacio de almacenamiento de pro  
Las variables Globales usan 242 bytes (11%) de la memoria dinámica,  
Tarjeta en COM3 no disponible
```

Ilustración 57 Error en Arduino, cuando se está ejecutando la consola de Python

Una vez completa la conexión satisfactoriamente, leemos los datos de la consola y los almacenamos en un archivo *.txt*, hasta que Arduino indique que terminó el escaneo:

- Leer Cadena del puerto serial con decodificación en ASCII
- Mientras no sea Cadena == 'T'
 - Escribir en archivo Cadena
 - Leer Cadena del puerto serial con decodificación en ASCII
- Fin Mientras

Función en Spyder

```
Cadena=arduino.readline().decode('ascii')
while not Cadena == 'T\r\n':#b'Termine\r\n'
    pass
Nubedepuntos.write(Cadena)
print(Cadena)
Cadena=arduino.readline().decode('ascii')
```

Ilustración 58 Función en spyder para la lectura de la consola y escritura en el archivo txt

Cuando se finaliza el escaneo, se cierra el puerto serial 'COM3' y el archivo .txt

Abrimos el archivo generado, leemos y revisamos que todos los datos se encuentren en el formato correcto, sin saltos de línea, ni espacios para obtener en todo el mismo número de filas y columnas. Así mismo se creó una función para eliminar estos espacios.

Función en Spyder para eliminar saltos de línea.

```
with open('C:\\Users\\Violeta\\Documents\\Nubedepuntos.txt','r') as f:
    datos=f.read().replace(' \n\n',';')
```

Ilustración 59 Asignación para reemplazar los saltos de línea

Leemos el archivo y se guardan todas las cadenas leídas en una variable llamada datos y reemplazamos " , \n \n " por ";" para así eliminar todos los espacios vacíos y saltos de línea.

Se asignan los datos leídos a una variable en donde se convierten las coordenadas a una matriz de [n,3] con la biblioteca de Python Numpy, que permite crear y modificar matrices.

Asignación en Spyder

```
m = np.matrix(datos)
```

Ilustración 60 Asignación para crear la matriz numpy

Donde la variable *m* es una matriz de [n,3].

Ejemplo de matriz[n,3]:



```
[[ 5.18 -1.21  0. ]
 [-2.35 -4.77  0. ]
 [-4.11  3.38  0. ]
 ...
 [-5.64  6.5   6.81]
 [ 7.84  4.14  6.81]
 [ 2.23 -8.49  6.81]]
Matriz de dimension: (14766, 3)
```

Ilustración 61 Impresión en consola de una Matriz[n,3]

Una vez asignada la matriz, se utiliza la función “*reshape()*” que se implementa para cambiar la forma de la matriz, en este caso para redimensionar los datos *x*, *y*, *z* sin cambiarlos para no generar ningún error.

Asignación de la función *reshape()* en Spyder

```
xyz=np.reshape(np.ravel(m), m.shape)
```

Ilustración 62 Reasignación para la matriz en formato *x,y,z*

Se instancia la Clase *PointCloud* de la librería de *Open3D* y se determina que se usará para crear la una nube de puntos, asignamos nuestra matriz de coordenadas *x*, *y*, *z* a la función *Vector3dVector()* para convertir los puntos de la matriz al formato de *Open3D* para su lectura, definimos el color para la nube de puntos y visualizamos.

Asignación en Spyder

```
print("Matriz para la nube de puntos...")
print(xyz)
#visualización del espacio de la nube de puntos y la instanciamos
pcd = o3d.geometry.PointCloud() #instanciamos la libreria para crear las formas 3D
pcd.points = o3d.utility.Vector3dVector(xyz)#Convertimos los puntos de la matriz al formato de Open3D
pcd.paint_uniform_color([0, 0, 0])
print("Visualizando...")
o3d.visualization.draw([pcd])
o3d.visualization.draw_geometries([pcd])
```

Ilustración 63 Ejemplo de uso de la librería *Open3D*

Una vez ejecutado el programa tenemos como resultado la siguiente visualización de la figura escaneada.



Ilustración 64 Objeto a escanear

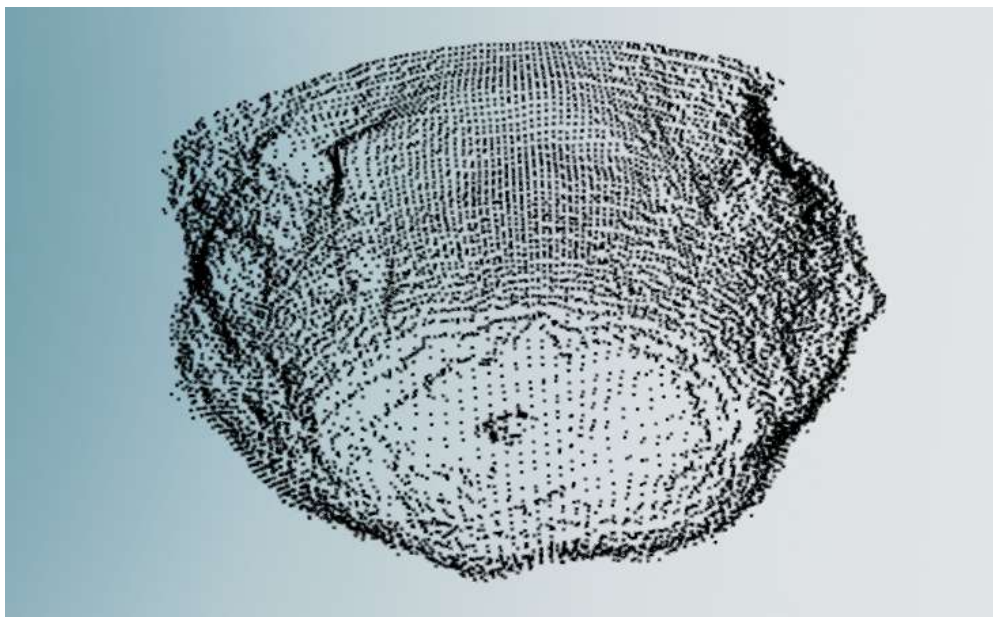


Ilustración 65 Resultado de la nube de puntos del objeto escaneado.

Interfaz gráfica del escáner

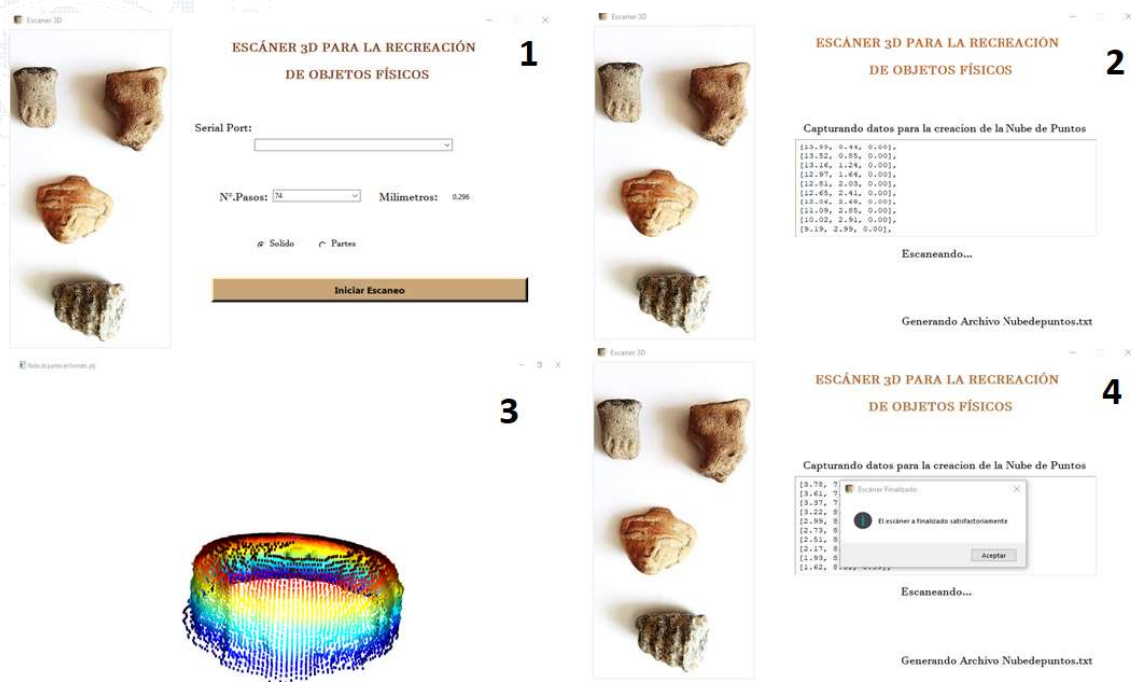


Ilustración 66 Interfaz gráfica. Secuencia: 1) Asignación de datos; 2) Obtención de datos; 3) Visualización de mallas; y, 4) Finalización del escáner.

La ventana principal del escáner 3D, como se muestra en la ilustración 67, cuenta con 2 listas desplegables con las que el usuario puede interactuar. La primera es para la etiqueta del *Serial Port* en la cual se alistan todos los puertos “COM” disponibles en los cuales puede obtener acceso nuestro Arduino.

Para la etiqueta N°. Pasos, se implementó una lista que contiene la numeración de pasos que puede tener el eje z y, delante de la lista desplegable, se encuentra la etiqueta que representa los milímetros de avance del número de pasos que el usuario seleccionó. Por default, el número de pasos seleccionados es 74.

También, se encuentran 2 *radio buttons*, con las etiquetas sólido y partes. Cuando se habla de escanear piezas o figuras completas, el usuario debe seleccionar la etiqueta sólido. De igual manera cuando seleccionamos la etiqueta partes se refiere a escanear alguna pieza incompleta en la cual puede a ver o tener huecos en donde nuestro sensor no detectará ningún punto.



ESCÁNER 3D PARA LA RECREACIÓN DE OBJETOS FÍSICOS

Serial Port:

Nº.Pasos: Milímetros: 0.296

Solido Partes

Iniciar Escaneo

Ilustración 67 Interfaz de la ventana principal

Cuando se selecciona la etiqueta partes, se muestra la siguiente ventana, en la cual se observa una etiqueta llamada "Altura" y una lista desplegable, en la que el usuario tendrá que definir la altura del objeto a escanear. Por default, la etiqueta tiene como altura 10 cm.



ESCÁNER 3D PARA LA RECREACIÓN DE OBJETOS FÍSICOS

Serial Port:

Nº.Pasos: Milímetros: 0.296

Solido Partes Altura: cm

Iniciar Escaneo

Ilustración 68 Interfaz para la opción de Partes

La segunda ventana que se muestra es principalmente visual, donde el usuario puede observar los datos de la nube de puntos que se van obteniendo del escáner y esperar hasta que la figura termine

de escanearse. Una vez realizado este paso, se obtienen las coordenadas que en conjunto generarán la nube de puntos.



Ilustración 69 Ventana secundaria para la visualización de la captura de datos del sensor

La tercera ventana que se despliega es la visualización de las coordenadas obtenidas, convirtiendo los puntos de la matriz calculada en formato de Open3D para su lectura.

La nube de puntos que se muestra es obtenida por el escáner, y transformada en formato polígono con extensión *.ply*

El archivo que genera se llama "Nubedepuntos.ply", este archivo puede ser procesado o modificado por otros programas de procesamiento 3D.



Nube de puntos en formato .ply

- □ X

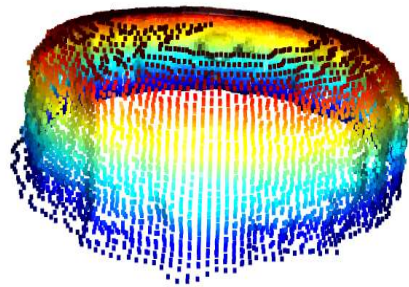


Ilustración 70 Tercera ventana para la visualización de la nube de puntos

Una cuarta ventana, muestra el objeto escaneado, tipo polígono, con una malla generada por el algoritmo *Ball-Pivoting* y la nube de puntos obtenida por el escáner. El usuario puede visualizar como debe formarse la malla tocando todos los puntos de la nube de puntos.

Malla con ruido y Nube de puntos

- □ X

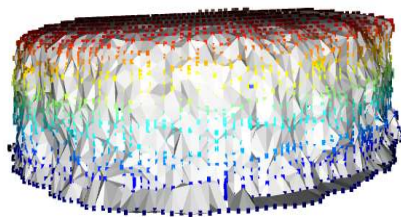


Ilustración 71 Cuarta ventana de visualización de malla y nube de puntos juntos.

Existen varios algoritmos que permiten construir la malla de un objeto a partir de la nube de puntos. Para la quinta y sexta ventana de visualizaciones, se seleccionaron dos algoritmos de reconstrucción implementados en la librería de Open3D en el cual, lo primero que se visualiza es una malla creada por el algoritmo *Ball-Pivoting* y, en el segundo, una malla creada por el algoritmo *Alpha Shapes*.

Logo of Universidad Autónoma de Baja California Sur (UABC) in the background.

Algoritmo *Ball-Pivoting*: consiste en obtener el promedio de la nube de puntos, de un punto a otro y se asigna a nuestro radio, este debe ser mayor que el promedio ya que cada que toque tres puntos, formará un triángulo, así sucesivamente hasta formar la malla.

Algoritmo *Alpha Shapes*: se genera con la triangulación de Delaunay que consiste en conectar los puntos creando triángulos y se asigna un valor para alpha. El valor alpha determina si el triángulo se muestra en la malla o no, mientras el valor de alfa sea mayor se van incluyendo más triángulos de los cuales la malla puede tener cambios en su topografía, como lo son tapar agujeros o también crearlos. El valor de ejemplo que utiliza para la reconstrucción de superficies es de 1.5 en alfa.

Las mallas generadas tienen ruido, por lo que se necesita hacer una filtración para tener mejores resultados. En las ventanas quinta y sexta, se pueden ver los detalles que contienen las mallas y se crean los archivos “Nubedepuntos_MeshBP.ply” y “Nubedepuntos_MeshAS.ply”, con este formato se puede procesar en otros programas de procesamiento 3D.

Malla con ruido creada con Ball pivoting

— □ ×

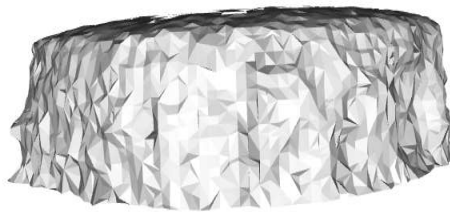


Ilustración 72 Ventana de visualización para la malla creada por el algoritmo Ball-Pivoting



Malla con ruido creada con Alpha shapes

- @ X



Ilustración 73 Ventana de visualización para la malla creada por el algoritmo Alpha Shapes

Para la séptima y octava ventana, se implementa el método de filtrado para suavizar la malla de triángulos con promedios de puntos vecinos. Primero mostramos la malla filtrada generada por el “algoritmo Ball-Pivoting” (*Surface Reconstruction — Open3D 0.17.0 Documentation, 2018*) y, después, la malla filtrada por el “algoritmo Alpha Shapes” (*Surface Reconstruction — Open3D 0.17.0 Documentation, 2018*).

Malla Ball pivoting Filtrada

- @ X



Ilustración 74 Ventana de visualización para la malla Ball Pivoting filtrada



Malla Alpha shapes Filtrada

- □ X

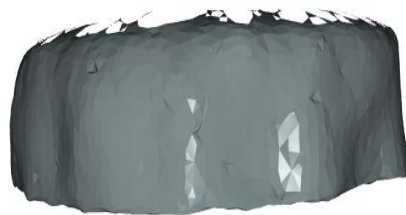


Ilustración 75 Ventana de visualización para la malla Alpha Shapes filtrada

Al finalizar se muestra un mensaje de finalizado y se cierra el programa, todos los archivos quedarán guardados en la carpeta de "Documentos", con sus nombres correspondientes, para que el usuario pueda utilizarlos.

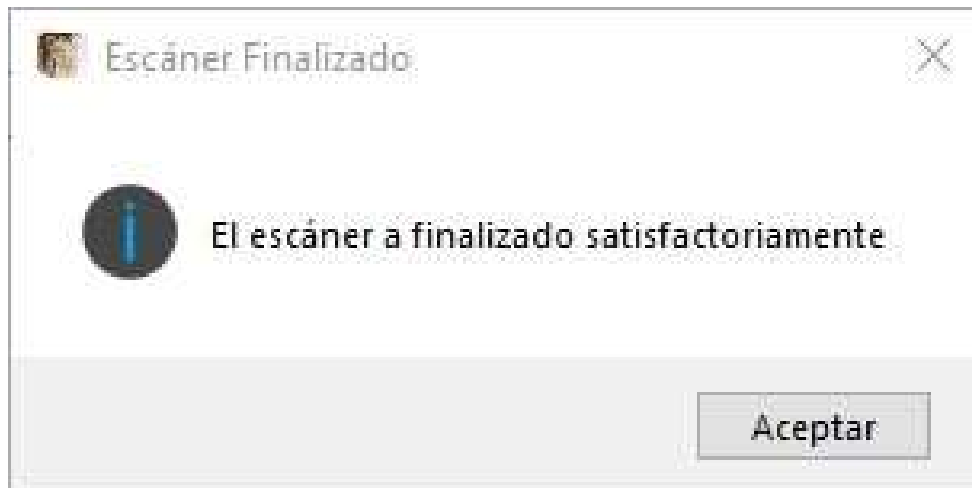
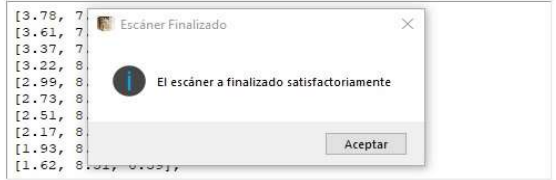


Ilustración 76 Cuadro de diálogo que indica que ha finalizado el escáner



ESCÁNER 3D PARA LA RECREACIÓN DE OBJETOS FÍSICOS

Capturando datos para la creación de la Nube de Puntos



Escaneando...

Generando Archivo Nubedepuntos.txt

Ilustración 77 Visualización final del cuadro de dialogo



Capítulo V

Pruebas y resultados

Pruebas y resultados

Los primeros objetos para escanear fueron objetos cotidianos que se seleccionaron al azar para realizar las primeras pruebas. Se determinó que el paso más pequeño tiene mejor recepción de los datos ya que, la distancia de un punto a otro es más corto lo que permite capturar más a detalle el objeto escaneado. Si el objeto escaneado no tiene mucho detalle, el punto medio será suficiente para capturar todo el objeto, mientras menor sea el número de paso, más tiempo llevará escanear el objeto, pero si el paso es mayor, el objeto estará escaneado en unos minutos.

Prueba 1



Ilustración 78 Dulcero de Halloween de calabaza

Calabaza con una altura 15.6 cm

El tiempo estimado para el escaneo de la calabaza con paso 28 fue de 15 min aproximadamente, la siguiente captura muestra los datos con los que se inició el escaneo.



ESCÁNER 3D PARA LA RECREACIÓN DE OBJETOS FÍSICOS

Serial Port:

COM3

Nº.Pasos: 28

Milímetros: 0.112

Solido Partes

Iniciar Escaneo

Ilustración 79 Valores asignados en la ventana principal para el escaneo

Los resultados fueron los siguientes:

Se obtuvo la nube de puntos que muestra la parte más baja en color azul hasta llegar a la parte más alta del objeto representado en color rojo. Al analizar la nube de puntos se determinó que se encuentra en modo espejo.

y

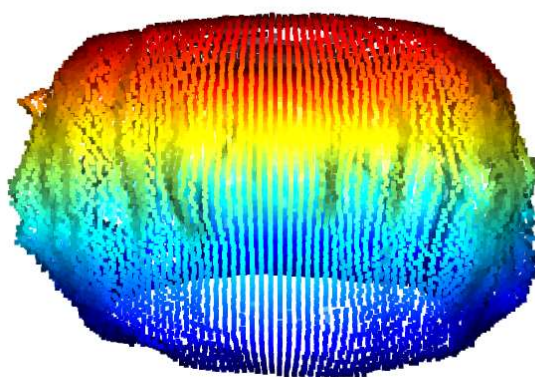


Ilustración 80 Nube de puntos de la calabaza escaneada

La ilustración 81 es una representación de una malla con ruido y la nube de puntos.

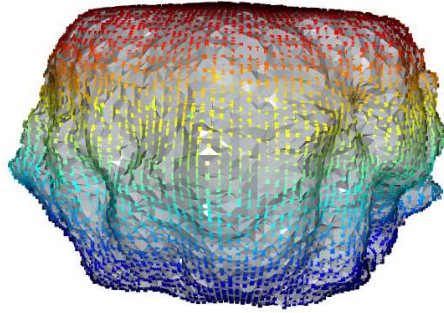


Ilustración 81 Nube de puntos y malla

Malla creada por el algoritmo Ball Pivoting:

Ball pivoting

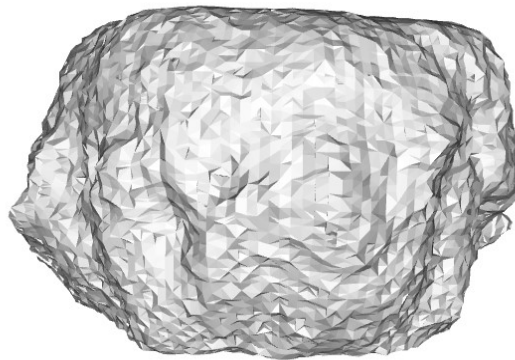


Ilustración 82 Malla de la calabaza creada por el algoritmo Ball Pivoting

Malla generada por el algoritmo Alpha Shapes:

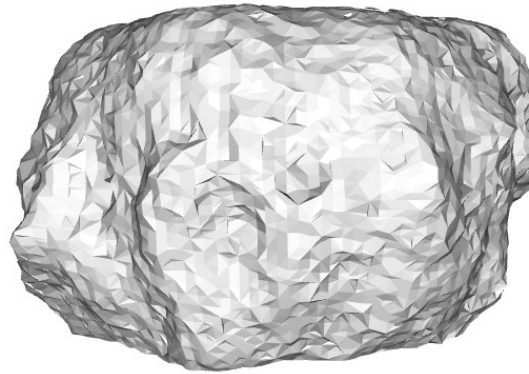


Ilustración 83 Malla de la calabaza creada por el algoritmo Alpha Shapes

Una vez obtenida la malla procedemos a disminuir el ruido de la malla triangular con un filtro para que este sea lo más exacto con la nube de puntos capturada.

La malla creada por Ball Pivoting con el filtrado se visualiza de la siguiente forma:

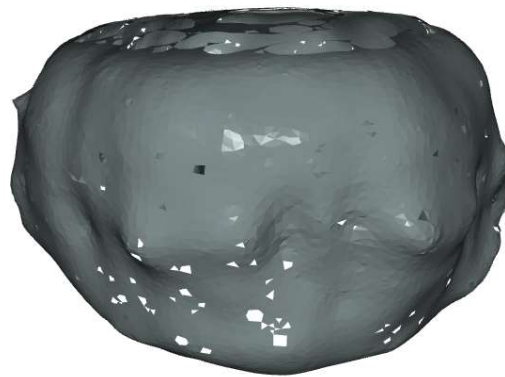


Ilustración 84 Malla Ball Pivoting filtrada

Podemos notar que el filtrado de malla se encuentra con espacios vacíos. El escáner 3D proporciona el archivo de esta malla si es que el usuario deseara modificarla con programas profesionales.

La malla creada por Alpha Shapes con el filtrado se visualiza de la siguiente forma:

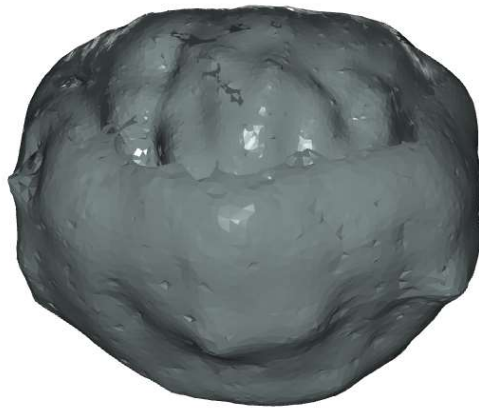


Ilustración 85 Malla de Alpha Shapes filtrada

Podemos notar que el filtrado de la malla creada por Alpha Shapes tiene mejores resultados para la reconstrucción de superficies de los objetos escaneados.



Prueba 2
Alcancía



Ilustración 86 Alcancía de yeso

El tiempo estimado para el escaneo de la alcancía con paso 28 fue de 7 min aproximadamente.

Se visualiza que la nube de puntos generada por el escáner tiene una mejor fidelidad con el objeto cotidiano escaneado, pero tiene una distorsión en la parte de arriba.

se puntos en formato .ply

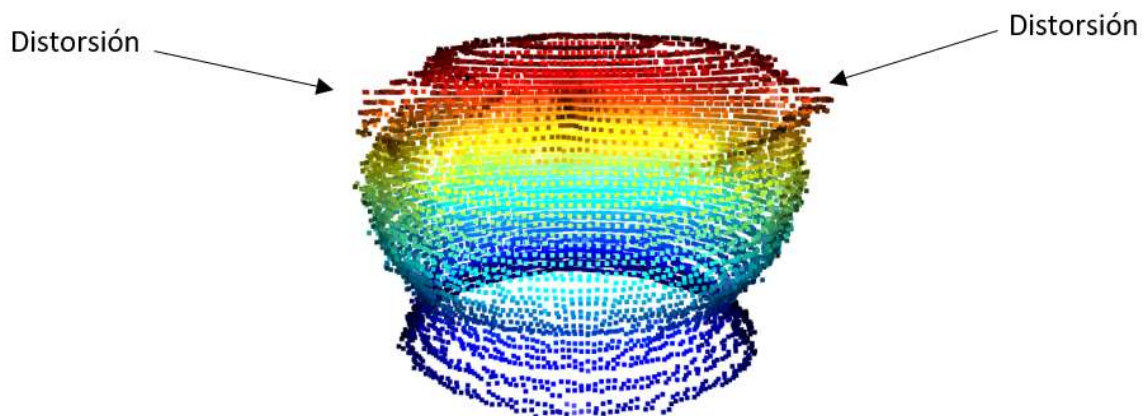


Ilustración 87 Nube de puntos distorsionada por el color negro

Cuando rotamos la figura escaneada, se nota que la distorsión solo se encuentra en la parte superior delantera de la figura.



Nube de puntos en formato .ply

Distorsión

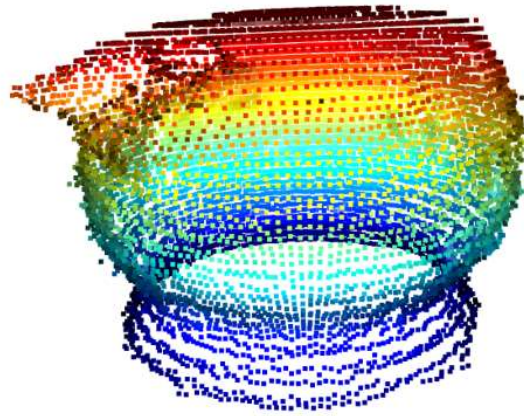


Ilustración 88 Nube de puntos distorsionada, visualización del lateral de la alcancía

Cuando se crea la malla, se nota que la parte trasera tiene una correcta fidelidad con el objeto escaneado.

Nube de puntos

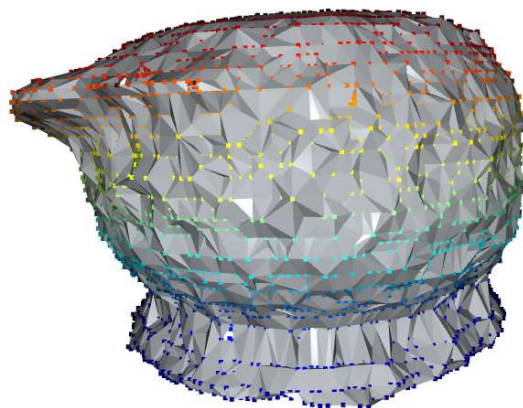
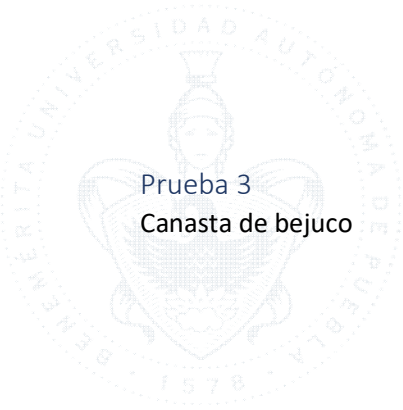


Ilustración 89 Malla distorsionada creada a partir de la nube de puntos obtenida

Con esta prueba, se determina que cuando el *sensor ir* escanea el color negro, genera distorsión y hace que la nube de puntos sea errónea a pesar de haber escaneado adecuadamente la figura.



Prueba 3
Canasta de bejuco



Ilustración 90 Canasta de bejuco

Los parámetros para el escaneo fueron de 74 para el número de pasos, este escaneo fue por partes, ya que el asa de la canasta genera un espacio vacío, con una altura de 22 cm.



Ilustración 91 Valores asignados en la ventana principal para el escaneo de la canasta

Los resultados fueron los siguientes:

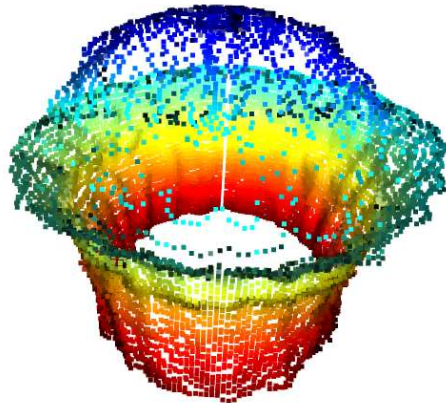
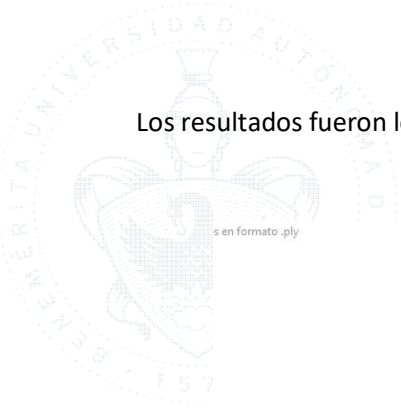


Ilustración 92 Nube de puntos obtenida de la canasta de bejuco

Se visualiza que antes de que empiece el asa, la nube de puntos generada es bastante fiel a la figura escaneada, sin embargo, cuando encontramos el espacio, que existe, entre el asa y el contorno de la canasta, se genera ruido y se crean puntos que tienen diferente distancia a los que se van obteniendo conforme al escaneado de la pieza, esto tiende a suceder, ya que el sensor solamente puede visualizar la parte superior o la primera capa que se encuentra en la figura escaneada. La solución para este problema sería eliminar todos los puntos que tienen diferentes distancias o bien apoyarnos con un software especial de diseño para así eliminarlo.

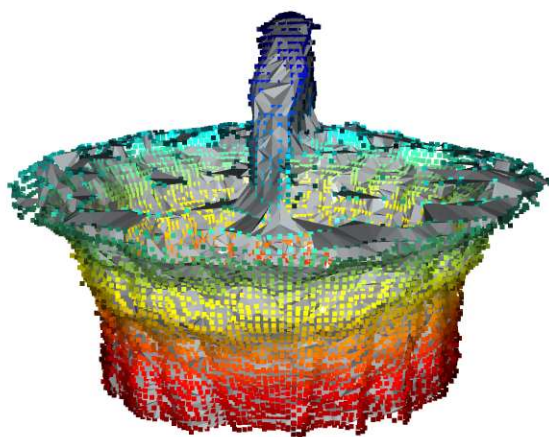


Ilustración 93 Malla y nube de puntos de la canasta de bejuco

En la ilustración 93, se visualiza la nube de puntos escaneada y la malla creada con el algoritmo Ball Pivoting, con lo que podemos apreciar los detalles que contiene nuestra canasta de bejuco y así mismo visualizar todos los puntos que debe de conectar nuestra malla.

con Alpha shapes



Ilustración 94 Malla creada por Alpha Shapes

La malla que entrega mejores resultados es la creada por Alpha Shapes.



Ilustración 95 Filtrado de malla Alpha Shapes con Alpha=1

Visualización final del filtrado del algoritmo Alpha Shapes para la canasta de bejuco, esta es un ejemplo de la reconstrucción para un objeto físico.

Es importante generar la nube de puntos adecuadamente, ya que con ella podemos crear o reconstruir las superficies para la recreación del objeto físico, una vez obtenida la nube de puntos se crea la reconstrucción con diferentes algoritmos, de los cuales debemos de elegir cuál es el que favorece a nuestra nube de puntos para una reconstrucción adecuada.



Escaneo de pieza arqueológica 1

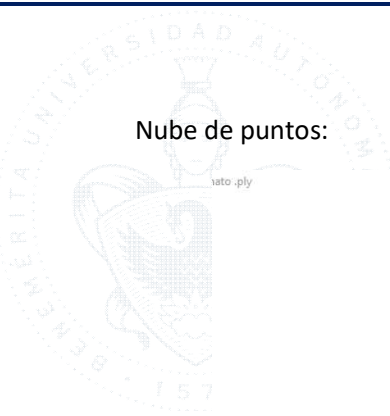


Ilustración 96 Pieza arqueológica parte de un jarrón

Pantalla principal con los datos del escaneo:



Ilustración 97 Valores asignados en la ventana principal para el escaneo de la pieza arqueológica



Nube de puntos:

lato.ply

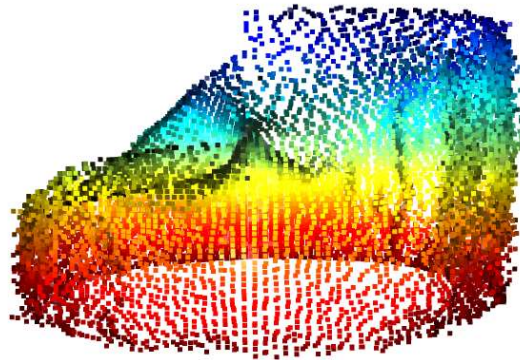


Ilustración 98 Nube de puntos de la pieza arqueológica (Parte del jarrón)

Nube de puntos malla generada por Ball Pivoting:

de puntos

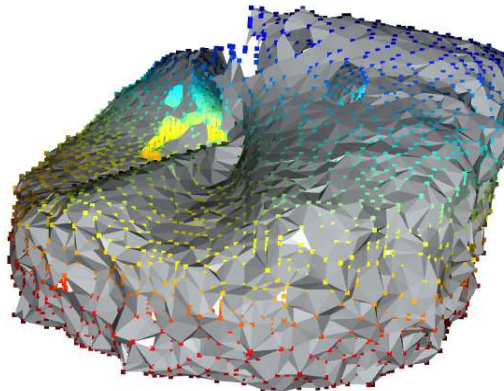


Ilustración 99 Malla y nube de puntos de la pieza arqueológica



Las ventanas de visualización para la reconstrucción de superficies con mallas son las siguientes:

creada con Ball pivoting

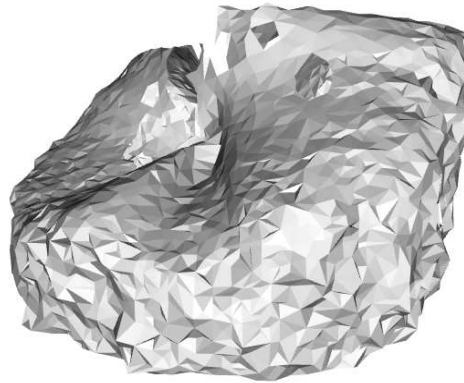


Ilustración 100 Malla generada por el algoritmo Ball Pivoting

Para el uso del algoritmo Alpha Shapes, el valor de $\text{Alpha}=1$ por default, ya que este es el valor especificado en el escáner.

creada con Alpha shapes

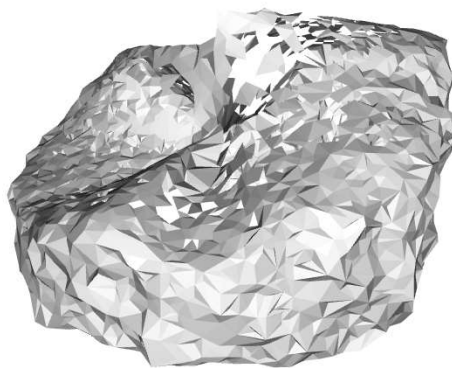


Ilustración 101 Malla generada por el algoritmo Alpha Shapes



Las mallas filtradas se muestran de la siguiente forma:

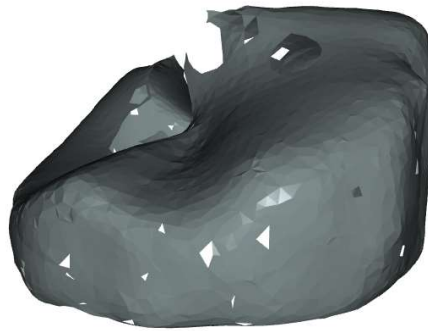


Ilustración 102 Malla Ball Pivoting Filtrada

Con el valor de Alpha=1 podemos obtener una malla filtrada de la siguiente forma, ya que se encuentra el Alpha por default:



Ilustración 103 Malla Alpha Shapes Filtrada

Para obtener mejores resultados en la malla, es necesario cambiar el valor del Alpha para que se pueda formar la figura adecuadamente.

Ejemplo con un valor de Alpha= 0.515 y Alpha=0.513



Ilustración 104 Escaneo Pieza arqueológica 1 Alpha=0.515

Alpha=0.513



Ilustración 105 Escaneo Pieza arqueológica 1 Alpha=0.513

Escaneo de pieza arqueológica 2



Ilustración 106 Pieza arqueológica circular

Se observa que la nube de puntos generada no es muy precisa, ya que la pieza es muy pequeña y las medidas del sensor empiezan a predecir la medida del punto medio.

La nube de puntos que se genera es la siguiente:

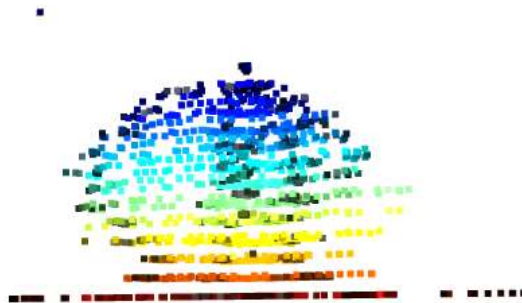


Ilustración 107 Nube de puntos Pieza arqueológica circular

Malla Ball Pivoting con nube de puntos:

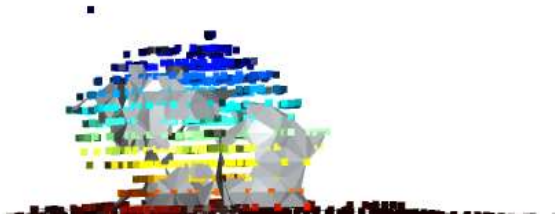


Ilustración 108 Malla y nube de puntos. Pieza arqueológica circular

Por el algoritmo Ball Pivoting, vemos que la malla que se genera no es la correcta.



Ilustración 109 Malla Ball Pivoting y Malla Filtrada de la pieza arqueológica

Por el algoritmo Alpha Shapes, tenemos una estimación más cercana de lo que es la pieza arqueológica escaneada.

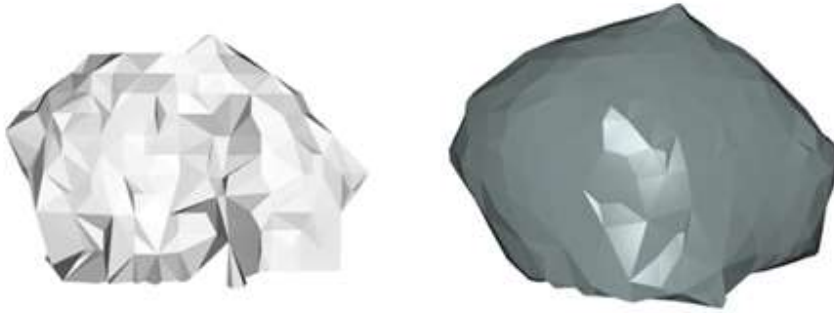


Ilustración 110 Malla Alpha Shapes y Malla Filtrada. Pieza arqueológica circular

Escaneo de pieza arqueológica 3



Ilustración 111 Pieza arqueológica. Pie

Nube de puntos:

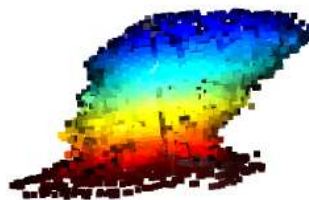


Ilustración 112 Nube de puntos. Pie

Se visualiza que la nube de puntos no es fiel a la pieza escaneada, ya que es muy pequeña y el sensor no puede dar una buena predicción de lo que es la pieza arqueológica adecuadamente.



Escaneo Artesanía 1



Ilustración 113 Maceta

Escaner 3D

**ESCÁNER 3D PARA LA RECREACIÓN
DE OBJETOS FÍSICOS**

Serial Port:

Nº.Pasos: Milímetros:

Solido Partes

Iniciar Escaneo

Ilustración 114 Pantalla de asignación para el escaneo



Nube de puntos:

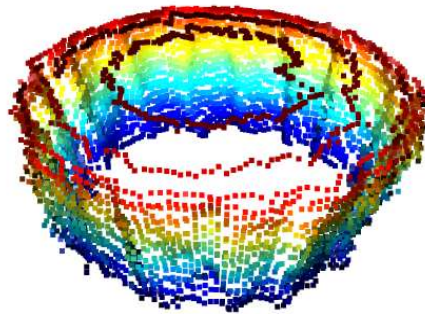


Ilustración 115 Nube de Puntos Maceta

Por el algoritmo Ball Pivoting:

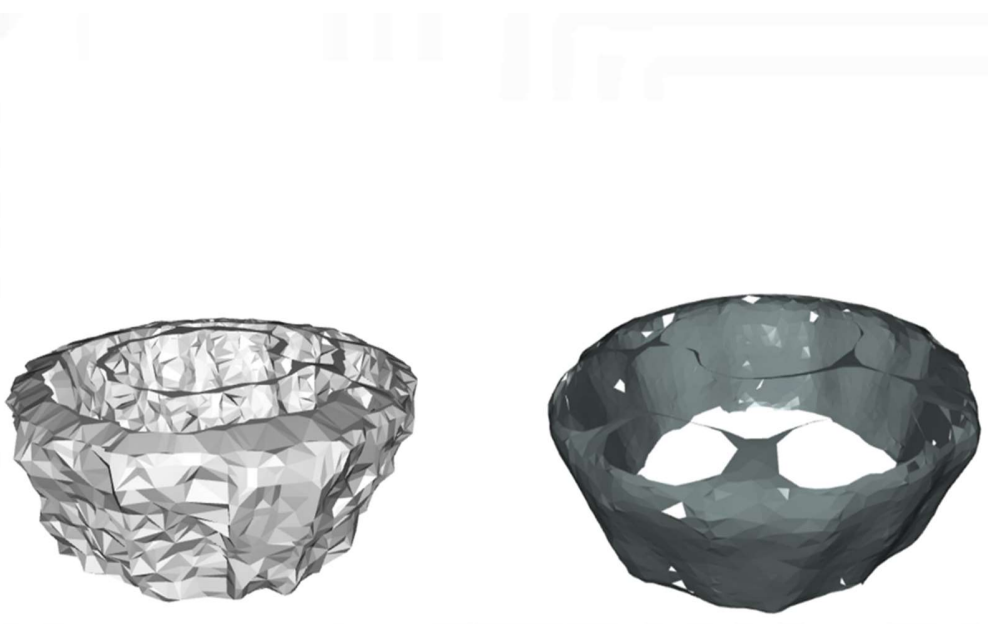


Ilustración 116 Malla Ball Pivoting y malla filtrada. Maceta

Por el algoritmo Alpha Shapes:



Ilustración 117 Malla Alpha Shapes y Malla Filtrada. Maceta

Escaneo Artesanía 2



Ilustración 118 Gato para colorear



Ilustración 119 Pantalla de asignación para el escaneo. Gato



Nube de puntos:

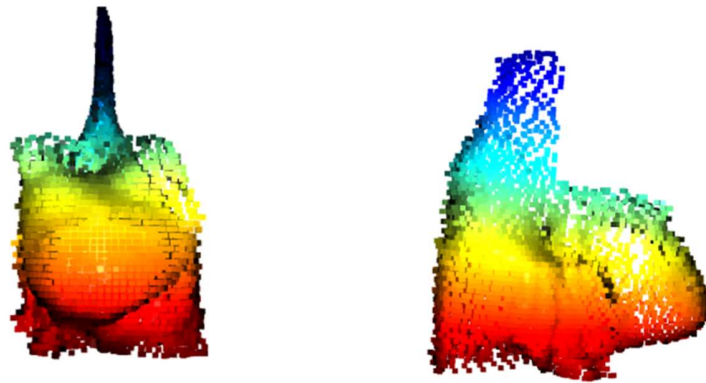


Ilustración 120 Nube de puntos. Gato

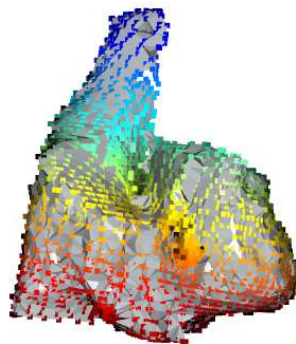


Ilustración 121 Malla y nube de puntos. Gato

Por el algoritmo Ball Pivoting tenemos los siguientes resultados:

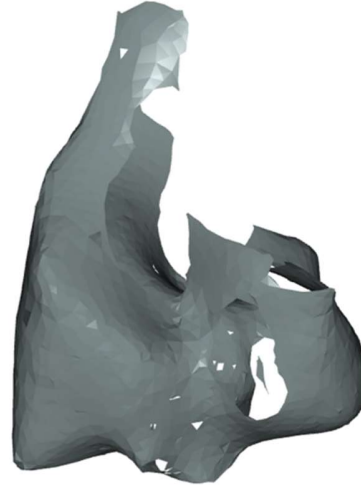


Ilustración 122 Malla Ball Pivoting y Malla filtrada. Gato

Por el algoritmo Alpha Shapes:

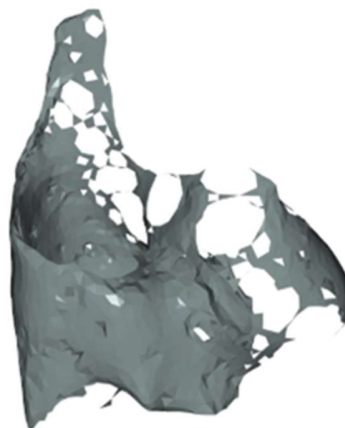


Ilustración 123 Malla Alpha Shapes y Malla filtrada. Gato

Se visualiza que tenemos mejores resultados con el Algoritmo Ball Pivoting, sin embargo, tenemos por default $\text{Alpha}=1$ y las mallas para este valor son las mostradas en la ilustración 121, cambiando el valor de Alpha podemos tener mejores resultados.



Escaneo Artesanía 3



Ilustración 124 Adorno para decoración



Ilustración 125 Pantalla de asignación para el escaneo. Adorno



Nube de puntos:



Ilustración 126 Nube de puntos. Adorno

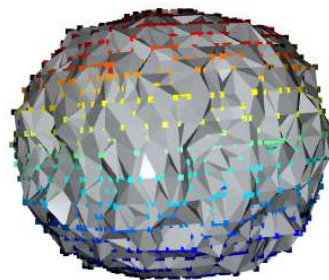


Ilustración 127 Nube de puntos y Malla. Adorno

Por el algoritmo de Ball Pivoting:

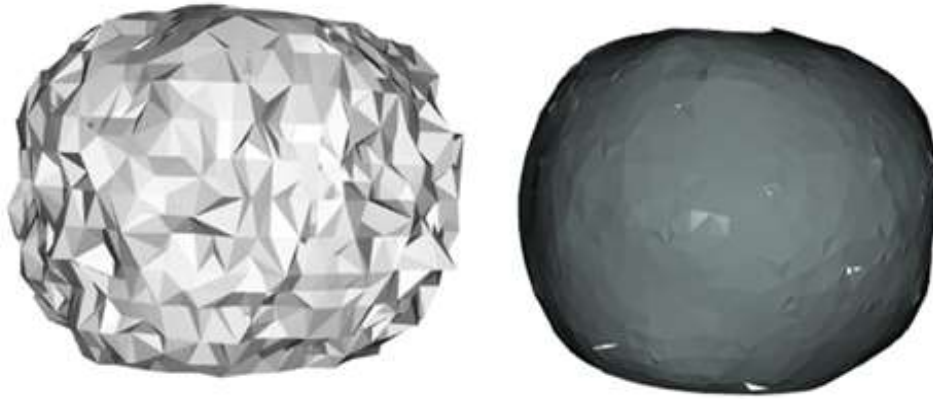


Ilustración 128 Malla Ball Pivoting y malla filtrada. Adorno

Por el algoritmo Alpha Shapes:



Ilustración 129 Malla Alpha Shapes y malla filtrada. Adorno



Escaneo Artesanía 4



Ilustración 130 Cantarito de barro

Escaner 3D

**ESCÁNER 3D PARA LA RECREACIÓN
DE OBJETOS FÍSICOS**

Serial Port:

Nº.Pasos: Milímetros: 0.296

Solido Partes

Iniciar Escaneo

Ilustración 131 Pantalla de asignación del escaneo. Cantarito de barro



Nube de puntos:



Ilustración 132 Nube de puntos. Cantarito de barro.

Por el algoritmo Ball Pivoting:

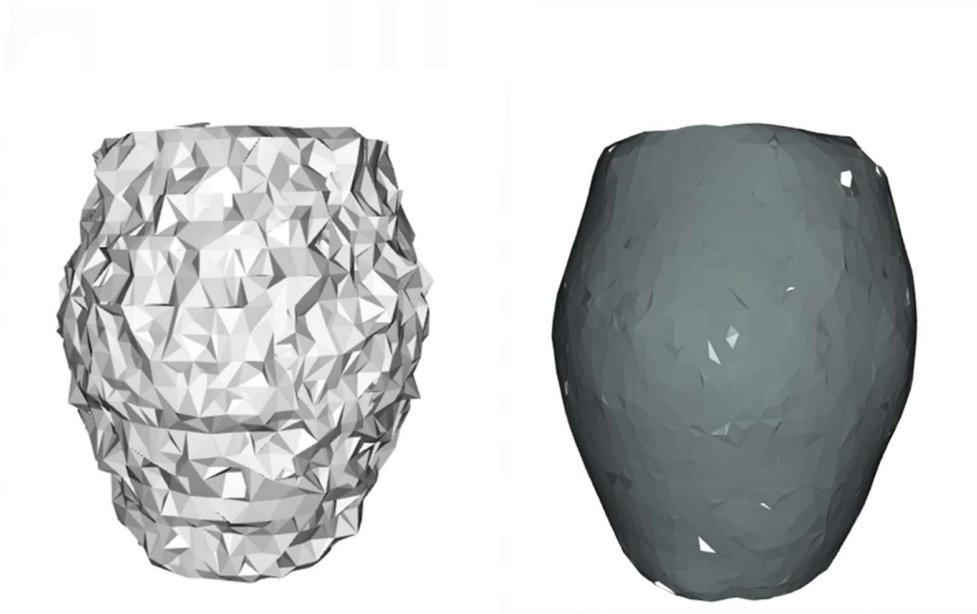


Ilustración 133 Malla Ball Pivoting y malla filtrada. Cantarito de barro.

Por el algoritmo Alpha Shapes:

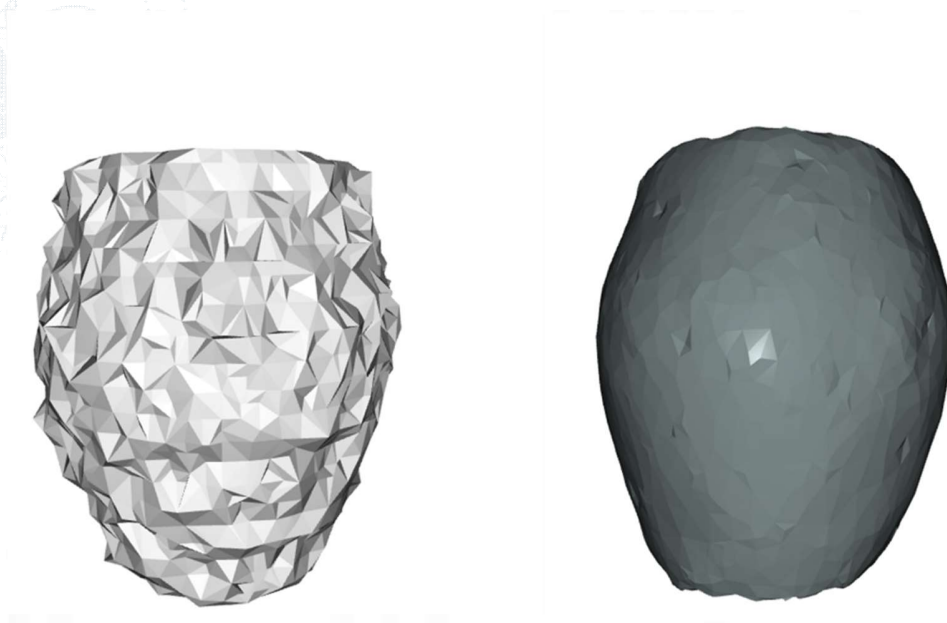


Ilustración 134 Malla Alpha Shapes y malla filtrada. Cantarito de barro.



Escaneo artesanía 5



Ilustración 135 Tortillero



Ilustración 136 Pantalla de asignación para el escaneo. Tortillero



Nube de puntos:

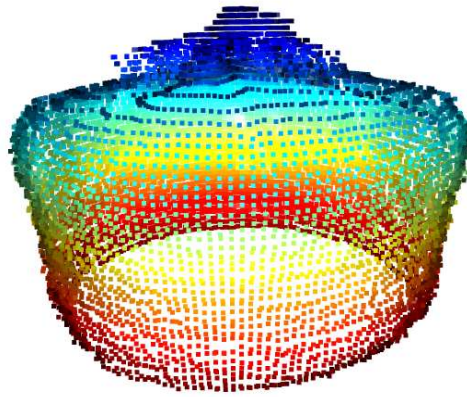


Ilustración 137 Nube de puntos. Tortillero

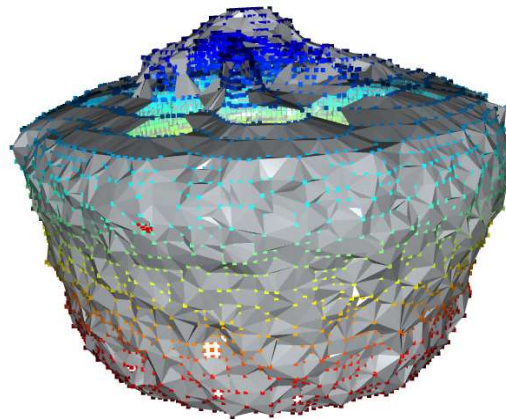


Ilustración 138 Malla y nube de puntos. Tortillero



Ilustración 139 Malla Ball Pivoting y malla filtrada. Tortillero

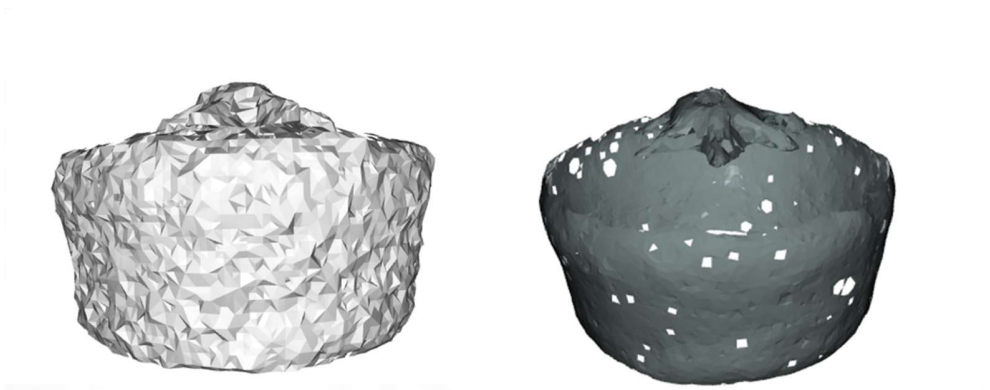


Ilustración 140 Malla Alpha Shapes y malla filtrada. Tortillero



Capítulo VI

Conclusiones, recomendaciones y trabajos futuros

Conclusiones, recomendaciones y trabajos futuros

Conclusión

Es importante que siempre la comunidad se interese por cuidar, perseverar y estudiar su patrimonio cultural, todos lo hacen de diferente manera, unos transmiten su conocimiento a otras personas para poder crear artesanías e ir innovando conforme pasa el tiempo, otros se preocupan por preservar la herencia cultura que las generaciones pasadas dejaron en la comunidad y otros investigan nuevas tecnologías para adaptarlas a la comunidad, y con ellas, resolver problemas reales que se presenten en ella. Como es el caso de las piezas arqueológicas extraviadas en Santo Nombre y la problemática de competencia en Santa María la Alta, pertenecientes al municipio de Tlacotepec de Benito Juárez por los cuales hubo la necesidad de crea un escáner 3D (hardware y software) capaz de construir un modelo digital 3D a partir de un objeto real para analizarlo, interpretarlo, modificarlo o reproducirlo, del cual se tiene los siguientes resultados:

El escáner 3D tiene mejores predicciones cuando se trata de escanear objetos de dimensiones mayores a 2cm de radio, ya que el sensor utilizado en las pruebas realizadas no detecta adecuadamente la figura o la forma de lo que se está escaneando. Entre más alejado el objeto se encuentre del punto medio, más exacto será.

Detecta piezas arqueológicas mayores a 2 cm, pero no detecta detalles exactos, ya que estos pueden llegar a ser muy diminutos, lo mismo sucede con el escaneo de artesanías, entre más grande este la pieza a escanear mejores serán los resultados obtenidos.

Una de las desventajas que se encontró al utilizar el *sensor ir* es que el espectro infrarrojo se pierde en los materiales como acrílico, vidrio, papel metalizado o cualquier materia que pueda absorber el espectro infrarrojo o en generar reflexión difusa, ya que estos materiales no son detectados por el sensor adecuadamente y generan datos erróneos.

Cuando el *sensor ir* detecta el color negro este distorsiona la nube de puntos en el área donde se encuentra y los colores que se parecen o son similares tienen el mismo efecto.

Recomendaciones

No usar distancias demasiado largas porque dificultan la recepción de datos.

Trabajos Futuros

Una mejora para el escáner 3D sería implementar otro sensor más profesional, que permita una mejor recepción de los datos para objetos pequeños e implementar una interfaz que pueda procesar imágenes y las convierta en nube de puntos para guardar información de forma y color del objeto escaneado, lo cual será una solución para capturar los datos de las piezas arqueológicas pequeñas y no perder información de sus detalles.

Citas y referencias bibliográficas

Bitfab. (2022). *Tipos de escaneado 3D a examen: los comparamos*. marzo 10, 2022, de Bitfab - Servicio de impresión 3D bajo demanda Sitio web: <https://bitfab.io/es/blog/tipos-escaneado-3d/>

Pedro Santamaría. (febrero 20, 2020).Este escaner 3D con una Raspberry Pi es de lo mejor que puedes construir. Abril 15, 2022, de El output Sitio web: <https://eloutput.com/productos/gadgets/escaner-3d-raspberry-pi/>

Carlos Abad García, C.A.G (2017). *Estudio de técnica de generación de modelos 3D a partir de secuencias de imágenes* (Ingeniería de Tecnologías de Telecomunicación). Escuela técnica superior de ingenieros industriales y de telecomunicación universidad de cantabria. <https://repositorio.unican.es/xmlui/bitstream/handle/10902/11974/396326.pdf?sequence=1>

Quintanilla Padilla, Ricardo Edgardo(2017) *Diseño e implementación de prototipo de escáner 3D s.v* Ricardo Edgardo Quintanilla Padilla. -- 1ª ed. -- Santa Tecla, La Libertad, El Salvador: ITCA Editores, 2017. ISBN: 978-99961-50-56-2 <http://www.redicces.org.sv/jspui/handle/10972/3052>

De Matías Bejarano, J. (2013). *Técnicas de fotogrametría y visión por computador para el modelado 3D de estructuras geomorfológicas dinámicas* [tesis doctoral, universidad de extremadura]. <https://robolab.unex.es/wp-content/papercite-data/pdf/de-tecnicas-vision.pdf>

Soluciones de Metrología Creaform. (2021). *Escaneado 3D para el control de calidad en la fabricación aditiva* [Archivo de Vídeo]. Youtube. <https://www.youtube.com/watch?v=5lHZrkbdly8>

MakerBot (2013). MakerBot Digitizer Desktop 3D Scanner | Introduction Video [Archivo de Vídeo]. Youtube. <https://www.youtube.com/watch?v=9GfnKKczec0&t=6s>

Chaudhary, P. (2023,21 febrero). Point Cloud to Mesh Conversion - A Rookie Way. [www.linkedin.com](https://www.linkedin.com/pulse/point-cloud-mesh-conversion-rookie-way-puru-chaudhary/). <https://www.linkedin.com/pulse/point-cloud-mesh-conversion-rookie-way-puru-chaudhary/>

MotionKing (China) Motor Industry (32020). 2 Phase Hybrid Stepper Motor17HS series-Size 42mm(1.8 degree) <http://www.datasheet.es/PDF/928661/17HS4401-pdf.html>

José Luis Rozúa Lucena. (2018, 22 octubre). *Ciclop 3D Escáner Láser* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=70LICigdX78>

Inky3D (s.f.) Scanner Laser 3D Kit Impresora Reprap Diy Ciclop Escanear. https://www.inky.mx/Scanner-Laser-3d-Kit-Impresora-Reprap-Diy-Ciclop-Escanear,938_1562954364

Calibration problems. (s. f.). Google Groups. <https://groups.google.com/g/ciclop-3d-scanner/c/DQFTVO102Hc?pli=1>

Ciclop dimensions(s.f.) <https://lh3.googleusercontent.com/-EVnddvWeOSQ/VrB1g8fKSdl/AAAAAAAAA8/6uP8yWpBlxs/s1600/Ciclop%2Bdimensions.PNG>

Introduction — *Open3D 0.17.0 documentation.* (2018).
<http://www.open3d.org/docs/release/introduction.html>

ABREN LA ZONA ARQUEOLÓGICA TETELES DE SANTO NOMBRE Y SU MUSEO DE SITIO. (2017, 26 Enero). NTCd. <https://ntcd.mx/nota-turismo-abren-la-zona-arqueologica-teteles-de-santo-nombre-y-su-museo-de-sitio201726133>

Museo de Sitio de Teteles de Santo Nombre. (2022, 3 octubre). Secretaría de Cultura/Sistema de Información Cultural. https://sic.cultura.gob.mx/ficha.php?table=museo&table_id=1902

Pirámides los Teteles de Santo Nombre El Valor de una Identidad | . . . (2016, 3 marzo). <https://www.facebook.com/tlacotepecpaginaoficial/videos/piramides-los-teteles-de-santo-nombre-el-valor-de-una-identidad/1544148942544773/>

González, M. G. M. (s. f.). *Zona Arqueológica y Pirámides de Tlacotepec de Benito Juárez, Puebla.* <http://www.tlaco.net/piramides-de-tlacotepec-teteles-santo-nombre.html>

Lomelí, D. V. (2020). ¿Sabías que Tlacotepec de Benito Juárez y Teotihuacán tienen algo en común? Aquí te cuento. <https://www.periodicocentral.mx/2020/rayas/opinion/puebla-vivela-bien/item/14004-sabias-que-tlacotepec-de-benito-juarez-y-teotihuacan-tienen-algo-en-comun-aqui-te-cuento>

En Tlacotepec piden al INAH el regreso de mil piezas arqueológicas | Municipios Puebla | Noticias del estado de Puebla. (2022, 28 abril). <https://municipiospuebla.mx/nota/2022-04-28/tlacotepec-de-benito-ju%C3%A1rez/en-tlacotepec-piden-al-inah-el-regreso-de-mil-piezas>

html.alldatasheet.com. (s. f.). GP2Y0A21YK0F Datasheet(1/9 Pages) SHARP | Distance Measuring Sensor Unit Measuring distance: 10 to 80 cm Analog output type. <https://html.alldatasheet.com/html-pdf/412635/SHARP/GP2Y0A21YK0F/155/1/GP2Y0A21YK0F.html>

ELECTRONOBS en Español. (2019, 31 enero). Pr#41 escáner 3D con sensor IR casero [Vídeo]. YouTube. <https://www.youtube.com/watch?v=JYm2Fbi6eAw>

Carmenate, J. G. (2021, 11 mayo). 🕒 Función millis y micros de Arduino. *Programar fácil con Arduino.* <https://programarfácil.com/blog/arduino-blog/millis-micros-arduino/>

BIMnD, E. (2022). Guía rápida para quitarte el miedo a trabajar con Nube de Puntos 3D. *BIMnD.* <https://www.bimnd.es/trabajar-con-nubes-de-puntos-miedo-fuera/>

Tutorial Sensor de Distancia SHARP. (2016). Naylamp Mechatronics - Perú. https://naylampmechatronics.com/blog/55_tutorial-sensor-de-distancia-sharp.html

Dynapro. (2021, 20 diciembre). ¿Cómo funciona un escáner 3D? - Dynapro 3D | Empresa dedicada al diseño digital e impresión 3D. Dynapro 3D | Empresa dedicada al diseño digital e impresión 3D. <https://dynapro3d.com/funciona-escaner-3d/>

AutoCAD 2024 Ayuda | Acerca de la creación de mallas 3D | Autodesk. (s. f.). AUTODESK AutoCAD 2024. Recuperado 2 de abril de 2023, de <https://knowledge.autodesk.com/es/guidref/ACD/2018/learn-explore/GUID-A6232957-5039-4AB7-8B1D-8FD0AD98F77B>

Zhou , Q.-Y., Park , J., & Koltun, V. (2018). Introduction — Open3D 0.17.0 documentation. Open3d. Recuperado 3 de marzo de 2023, de <http://www.open3d.org/docs/release/introduction.html>

Aguayo, P. (2019). Arduino UNO | Arduino.cl - Compra tu Arduino en Línea. *Arduino.cl - Compra Tu Arduino En Línea.* <https://arduino.cl/arduino-uno/>

Welcome to Spyder's Documentation — Spyder 5 documentation. (n.d.). <https://docs.spyder-ide.org/current/index.html>

CoreDRAW. (2023, February 23). GetApp. <https://www.getapp.com.mx/software/119883/coreldraw>

EAGLE | PCB Design And Electrical Schematic Software | Autodesk. (2023). <https://www.autodesk.com/products/eagle/overview?term=1-YEAR&tab=subscription>

Surface reconstruction — Open3D 0.17.0 documentation. (2018). http://www.open3d.org/docs/release/tutorial/geometry/surface_reconstruction.html

LibGuides: APA, 7ma edición - Citas y Referencias: Comunicaciones personales. (2021). <https://uprrp.libguides.com/c.php?g=985694&p=7256166>

Martins, J. (2022, October 10). ¿Qué es la metodología Kanban y cómo funciona? [2022] • Asana. Asana. <https://asana.com/es/resources/what-is-kanban>