INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo terminal

“Escáner de objetos físicos para su visualización en 3D”

2013 – A019

Que para cumplir con la opción de titulación en la carrera de:

“Ingeniería en Sistemas Computacionales”

Presentan

Andrade Chávez Ángel Benjamín
Ventura Cruz Eduardo

Directores

M. en C Roberto Eswart Zagal Flores M. en C José David Ortega Pacheco

Mayo 2014

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

SUBDIRECCIÓN ACADÉMICA

No. de registro: 2013-A019 Serie: Amarilla Mayo del 2014

Documento técnico

"Escáner de objetos físicos para su visualización en 3D"

Autores

Andrade Chávez Ángel Benjamín[1] **Ventura Cruz Eduardo**[2]

Directores

M. en C. Roberto Eswart Zagal Flores

M. en C. José David Ortega Pacheco

RESUMEN

En este trabajo se presenta la documentación técnica del Trabajo Terminal 2013 - A019 titulado "Escáner de objetos físicos para su visualización en 3D", cuyo objetivo es crear un sistema de escaneo capaz de, obtener a partir de un objeto de nuestro entorno, un modelo 3D del mismo.

Palabras clave –Sensor Kinect, Digitalización de objetos, Modelado 3D

[1] E-mail: abachavez@gmail.com
[2] E-mail: laloevc@gmail.com

ESCUELA SUPERIOR DE CÓMPUTO

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE FORMACIÓN INTEGRAL E INSTITUCIONAL

COMISIÓN ACADÉMICA DE TRABAJOS TERMINALES

México, D.F. a 20 de Mayo de 2014

DR. FLAVIO ARTURO SÁNCHEZ GARFIAS

PRESIDENTE DE LA COMISIÓN ACADÉMICA

DE TRABAJOS TERMINALES

P R E S E N T E

Por medio del presente, informamos que los alumnos que integran el TRABAJO TERMINAL 2013-A019 titulado "Escáner de objetos físicos para su visualización en 3D" concluyeron satisfactoriamente su trabajo.

El empastado del Reporte Técnico Final y el Disco Compacto (CD) fueron revisados ampliamente por sus servidores y corregidos, cubriendo el alcance y el objetivo planteados en el protocolo original y de acuerdo a los requisitos establecidos por la Comisión que Usted preside.

ATENTAMENTE

M. en C. Roberto Eswart Zagal Flores

M. en C. José David Ortega Pacheco

Advertencia

"*Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan*."

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000 Extensión 52000.

Agradecimientos

A mis padres, por darme la vida y el apoyo incondicional en todo momento a pesar de las adversidades.

A mi padre, por ser el sostén de la familia ya que sin su esfuerzo del día a día no se hubiera logrado esta meta tan importante. Por ser mi ejemplo a seguir, por enseñarme a que todo se puede realizar sin importar cuán difícil sea.

A mi madre, me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento ya que sin su ayuda, nada de esto hubiera sido posible, gracias a su cariño, a sus atenciones y cuidados, a su continua motivación y en especial a enseñarme que no existe objetivo que no se pueda cumplir.

A mis hermanos, por escucharme, por apoyarme y guiarme en la toma de decisiones, por ser un ejemplo a seguir, ya que gracias a sus consejos he realizado este logro.

Quisiera hacer extensivo mi agradecimiento a mis compañeros, gracias por acompañarme en el cumplimiento de esta meta, sin olvidar mencionar a mi compañero Ángel Benjamín Andrade Chávez por su amistad y colaboración en este proyecto tan importante de nuestras vidas.

A mis maestros, que compartieron sus conocimientos para convertirme en un profesionista, por su tiempo y dedicación.

Eduardo Ventura Cruz

La presente tesis, sin duda es la muestra de incontables desvelos y mucho esfuerzo, no solo míos, si no de mi familia y mis compañeros, que con sus palabras y consejos me ayudaron a completar la misma.

A mi madre, quien ha sido y será un apoyo fundamental en mi vida. Quien siempre está dispuesta a platicar y debatir un sinfín de ideas. Por la confianza que se ha ganado de mi persona. Quien con su amor y su guía me ha formado como lo que soy.

A mi padre, quien ha sido un gran apoyo en mi vida. Quien me ha motivado a buscar la casi perfección en los trabajos realizados. Quien siempre encuentra la manera menos común de darte las palabras más sabias. Por su amor y su apoyo.

A mis hermanos, quienes han compartido experiencias sin igual conmigo. Quienes me han enseñado y me han dado sus palabras cuando más las he necesitado. Por sentirse orgullosos de lo que soy, lo que he logrado y lograré.

A mis compañeros, en especial a mi compañero y amigo Eduardo Ventura Cruz, quien con su colaboración en este proyecto se logró su culminación. Quien siempre tuvo la paciencia de saber cómo guiar el trascurso de las cosas.

A mis maestros, quienes por medio de reglamentos y recompensas me enseñaron la disciplina. Quienes me forjaron como profesionista con sus conocimientos y experiencias.

Ángel Benjamín Andrade Chávez

INDICE

Índice de figuras.

1 Introducción.

1.1 Introducción.

Hoy en día los avances tecnológicos son cada vez más continuos, y por ende las necesidades del hombre en cuanto a su vida cotidiana son proporcionales a dichos cambios. Uno de estos avances es la constante necesidad de querer introducir objetos de la vida cotidiana en un sistema de cómputo de manera digitalizada. La obtención de una visualización digitalizada de objetos se remonta a los años 60, donde el Dr. Hanratty por la incapacidad de diseñar piezas de manera eficiente comenzó a ver la posibilidad de hacer sus “dibujos” de una manera más precisa, y que pudiera compartirlos, dando así origen al Diseño Asistido por Computadora (CAD, por sus siglas en ingles), que básicamente eran los mismos dibujos pero vistos de manera digital.[1]

Las necesidades fueron creciendo y muchas veces los diseños no eran interpretados como el autor lo había previsto, por lo tanto, surge la exigencia de hallar la manera de digitalizar los objetos de tal modo que se pudiera mantener el diseño original en todo el proceso. Una necesidad que también se comenzó a trabajar fue el poder portar el objeto digitalizado para posteriormente construir de manera física, sin embargo, no fue hasta el año de 1981 cuando este requisito fue cubierto con una calidad aceptable, cuando sale a la venta el primer programa capaz de desarrollar dibujos sólidos de objetos.

La tecnología avanzó y con ella las necesidades, es por eso que a pesar de que en aquel entonces el DAC (Diseño asistido por computadora) o CAD solucionó la necesidad básica de modelado, hoy en día ya no es suficiente solo con estos programas. Las personas no especializadas necesitaron obtener por su cuenta modelos 3D, sin tener que poseer los conocimientos técnicos que nos ayudan a ocupar dichas aplicaciones.

1.2 Justificación.

En la actualidad, la necesidad de mostrar un modelado en tres dimensiones de las piezas o las partes especificas en una industria, se vuelven cada vez más comunes, es por este motivo que las compañías líderes en este proceso ofrecen día a día soluciones a cada una de estas necesidades.

Basándonos en las premisas anteriores, se propone desarrollar un trabajo capaz de cubrir estas necesidades, con la intención de brindar al mercado una solución a las insuficiencias tecnológicas a un menor precio del que ofrecen las compañías líderes del mercado, y aprovechar las nuevas investigaciones que

se están haciendo sobre la tecnología que se propone como fase medular del proyecto, intentando optimizar el proceso mismo.

Una herramienta que se encuentra aún en desarrollo y constante innovación que está revolucionando el mercado tecnológico es el sensor kinectel cual será utilizado para el desarrollo de este proyecto. Este sensor a diferencia de una cámara tradicional cuenta con diferentes características y componentes el cual nos permite explotar diferentes campos por todas las características de este dispositivo [2]. A continuación enlistamos algunas de ellas.

- Lentes con sensores de profundidad
- Micrófono
- Sensor de inclinación
- Amplio campo visual (57 grados horizontal y 43 vertical)
- Librerías Open Source de desarrollo

Cabe mencionar, que este trabajo se presenta debido a que nos pareció de interés este tema, porque no solo ofrece un buen camino para lograr el objetivo que se plantea, si no también que nos pareció una idea innovadora en base a las tecnologías que se manejan; la creación de proyectos de de este estilo con estas herramientas, nos demuestra que se puede innovar sin la necesidad de la alta tecnología.

Entre las ventajas que se encontraron al proyecto se encuentra el precio, el cual, en comparación de su competencia directa es completamente accesible, por otro lado, una desventaja que se presenta por el momento, es el tipo de objetos físicos que podrá escanear el prototipo, debido a que estará limitado por las características de los dispositivos que empleemos, sin embargo siempre quedará la posibilidad de extender la funcionalidad a objetos más grandes. Otra ventaja a nuestra propuesta es la posible portabilidad del trabajo, debido a que ocupa menos piezas y más pequeñas que la mayoría.

1.3 Objetivos.

1.3.1 Objetivo General.

Desarrollar un sistema capaz de escanear objetos físicos para su visualización en tres dimensiones.

1.3.2 Objetivos Particulares.

- Conocer el estado actual de los sistemas sobre la visualización de objetos para analizar sus características principales.
- Analizar los sistemas actuales para obtener un diagnostico.
- Desarrollar un sistema de escaneo de objetos para obtener un modelo 3D.
- Desarrollar un sistema de escaneo de objetos para su visualización.
- Comprobar la validación del sistema a través de su experimentación

2 Estado del arte.

El desarrollo de herramientas capaces de permitirnos la digitalización de objetos ha tenido un desarrollo histórico importante, a pesar de que se creía que las necesidades sobre dichas visualizaciones y diseño de los objetos estaban cubiertas por completo. Por ejemplo, surgen necesidades donde el manejo de objetos en 3 dimensiones es primordial.Se han empleado diversos mecanismos, algoritmos o sistemas para lograr una visualización en tres dimensiones de un objeto cualquiera, no obstante estos pueden ser costosos, complejos, difíciles de manejar o requerir un hardware especializado y caro. Un ejemplo claro de su aplicación seria en las empresas automotrices, las cuales ocupan este tipo de visualización con la finalidad de mejorar sus diseños.

En las tablas siguientes mostraremos los trabajos, artículos, etc. relacionados con el presente trabajo. Con la finalidad de mostrar que es un tema de interés y una necesidad constante.

2.1 Artículos.

Se puede apreciar en la Tabla 2.1, algunos artículos relacionados con el trabajo que se presenta. Hay que tener especial atención con el hecho de que la información relacionada con el tema se encuentra con mayor facilidad en ingles.

Tabla 2.1 Estado del Arte (Artículos)

Nombre del Artículo.	Descripción
3-D Reconstruction Using the Kinect Sensor and ItsApplication to a Visualization System [3].	Sistema de reconstrucción de una escena 3D,se reconstruye una escena 3-D con respecto al punto de vista del usuario para dar un efecto de perspectiva real, obteniendo profundidad y color del conjunto de datos del sensor de Kinect.
3D with Kinect [4].	Estudio de las capacidades del kinect en mediciones 3D y las compatibilidades con SFM y MultiviewStereo para la creación de imágenes 3D.
A Study in 3D-Reconstruction Using Kinect Sensor [5]	Descripción de las características del kinect, para la utilización de estas para el modelado 3D, mostrando la recreación de escenarios para su posterior visualización.

2.2 Comerciales.

Como se aprecia en la Tabla 2.2, existen algunas aplicaciones que hacen un escaneo de objetos, sin embargo muchos de ellos requieren materiales especializados para dicho propósito, y es necesario obtener una licencia de uso que suele ser muy costosa.

Tabla 2.2 Estado del Arte (Comerciales)

Nombre del Sistema	Objetivo	Características
Autodesk 123D catch [6]	Generar modelos en 3D a partir de una serie de imágenes	-.A partir de 20 o 40 fotos se puede obtener un modelo 3D. -Disponible vía web, PC, iphone o Ipad -Creación de videos, para su exportación.
Autocad[7]	Diseña y modela objetos en 3D	-Permite diseñar y crear modelos en 2D o 3D
Visual Nastran	Herramienta para el ingeniero mecánico para la creación y simulación de ensambles.	-Permite modelar piezas mecánicas. -Realiza las animaciones del comportamiento de las piezas diseñadas. -Visualiza el movimiento real de los diseños de manera virtual.
iREMS[8]	Herramienta que permite realizar diseños arquitectónicos.	-Diseña, crea y realiza desde pequeñas casas hasta ciudades e enteras. -Permite la manipulación de interiores a través de transparencias en objetos.
OpticScan[9]	Hardware capaz de escanear objetos para obtener su representación en 3D.	-Utilizado para pequeñas esculturas y en el modelado de piezas de autos. -Hardware para aplicaciones concretas.
Metra Scan 3D[10]	Sofisticado hardware capaz de realizar escaneos de manera precisa y automática dinámicamente.	-Realiza escaneos de todos los componentes en un ensamble, no importando el lugar donde se encuentre. -Escaneo preciso no importando las condiciones adversas.

2.3 Tesis.

A continuación en la Tabla 2.3,semuestran algunas tesis recientes que trabajan con la misma temática de lo que se plantea en este trabajo. Mediante estas tesis obtuvimos antecedentes recientes con las cuales podemos decir que, que es posible obtener el trabajo que se propone.

Tabla 2.3 Estado del arte (Tesis)

Tipo de tesis	Localidad	Nombre	Breve Descripción.	Fecha de publicación
Licenciatura[11]	Vienna University of Technology.Vienna, Europa	3D Reconstruction with theKinect-Camera	Usando el sensor de profundidad y el sensor RGB de la cámara Kinect, este trabajo obtiene un modelo 3D de algún objeto a su alrededor.	18 / Febrero / 2013
Licenciatura[12]	Oulu University. Finlandia	3D CONTENT CAPTURING AND RECONSTRUCTION USING MICROSOFT KINECT DEPTH CAMERA	Usando el sensor Kinect, este trabajo es capaz de hacer un escaneo de los 360 grados de un objeto.	Primavera, 2012
Licenciatura[13]		A Qualitative Analysis of TwoAutomated Registration Algorithms Ina Real World Scenario Using PointClouds from the Kinect	Usando el sensor de profundidad y la cámara infrarroja, este trabajo obtiene una nube de puntos del objeto propuesto haanálisis	26 / Junio / 2011
Licenciatura	México Distrito Federal, Escuela Superior de Computo	Escáner de objetos físicos para su visualización en 3D	Usando el sensor Kinect se obtiene mediante la nube de puntos un modelo en 3D del objeto de interés, a través de una visión de todas sus caras.	

3 Marco Teórico.

3.1 Kinect

La historia de Kinect comienza mucho antes de que el dispositivo en sí fuera concebido. Kinect tiene sus orígenes en ideas y visiones de interfaces de usuario basadas en el gesto y la voz. Todo comenzó con el éxito de la película MinorityReport, en 2002 fue uno de los incentivos que ayudo con la idea futurista de una interfaz de usuario espacial que junto a la intensa rivalidad y competencia entre las consolas de video juegos hizo posible la creación del Kinect.

Bill Buxton investigador de Microsoft, ha estado hablando en los últimos años de algo que él llama *Long Nose of Innovation,* de Chris Anderson, en este proceso se describen las décadas de tiempo de incubación necesario para producir una "revolucionaria" tecnología, aparentemente de la nada. El ejemplo clásico es la invención y el perfeccionamiento de un dispositivo para la revolución de la interfaz gráfica de usuario (GUI, por sus siglas en inglés): el *mouse*. El primer prototipo de *mouse* fue construido por Douglas Engelbart y Bill English, después, en el Instituto de Investigación de Stanford en 1963, donde le dieron al dispositivo el nombre de *murine*. Bill English mejoro el concepto cuando lo llevo a Xerox PARC en 1973 junto Jack Hawley, añadió la famosa pelota para el diseño del *mouse.* Durante este mismo período de tiempo, Telefunken en Alemania estaba desarrollando de manera independientemente su propio dispositivo *rollerballmouse* llamado TelefunkenRollkugel. Para 1982, el primer ratón comercial comenzó a encontrar su camino hacia el mercado. Logitech comenzó a vender uno por $299 dólares. Fue en este período en alguna parte que Steve Jobs visitó Xerox PARC y vio que el ratón trabajaba con la interfaz de WIMP (por sus siglas en inglés, ventanas, iconos, menús, punteros). Algún tiempo después, Jobs invitó a Bill Gates para ver el *mouse* basado en la interfaz GUI en la que estaba trabajando. Apple lanzó *Lisa* en 1983 con un *mouse* y, después, Macintosh contaba ya con un *mouse* en 1984. Microsoft anunció su sistema operativo de Windows poco después de la liberación de *Lisa* y comenzó a vender Windows 1.0 en 1985. No fue hasta 1995, con el lanzamiento del sistema de operativo de Microsoft Windows 95, que el *mouse* se convirtió en omnipresente. *The Long Nose of Innovation* describe el lapso de tiempo que requieren los dispositivos como el ratón para pasar de la invención a la indispensabilidad. [2]

A 30 años *The Long Nose of Innovation* puede ser utilizado para conocer la historia del Kinect. A partir de finales de los 70, a mitad de camino en la trayectoria del desarrollo del mouse, Chris Schmandt en el Grupo de Arquitectura de Máquinas (*Architecture Machine Group)* del Instituto

Tecnológico de Massachusetts (MIT, por sus siglas en inglés) inició un proyecto de investigación llamado *Put-That-There*, basado en una idea de Richard Bolt, que combina voz y el reconocimiento de gestos como vectores de entrada para una interfaz gráfica. *Put-That-There*se encontraba en una instalación de aproximadamente cinco por tres metros con una gran pantalla de proyección contra una pared. El usuario se sentaba en una silla de vinilo a unos dos metros por delante de la pantalla y tenía un cubo magnético oculto para la entrada espacial, así como un micrófono montado en la cabeza. Con estas entradas, y un poco de lógica de análisis del habla rudimentaria alrededor de pronombres como "eso" y "ahí", el usuario puede crear y mover las formas básicas alrededor de la pantalla. Bolt en su artículo de 1980 menciona algunos detalles del proyecto "*Put-That-There*: La voz y el gesto en la interfaz gráfica", que con el tiempo el micrófono montado en la cabeza deben ser reemplazados con un micrófono direccional y las versiones posteriores de *Put-That-There* permitirían a los usuarios dirigirse a través del Caribe y a los edificios coloniales en el mapa de Boston [3].

Otro proyecto de investigación del MIT de 1993 realizado por David Koonz, KristinnThorrison y CarltonSparrell-y de nuevo dirigido por Bolt-llamado *TheIconicSystem* refinó el concepto *Put-That-There* que trabajaba con el habla y los gestos, así como una tercera modalidad de entrada: rastreo visual(*eye-tracking)*. Además, en lugar de proyectar en un espacio de dos dimensiones, la interfaz gráfica era tridimensional generada por una computadora. En lugar de los cubos magnéticos utilizados para*Put-That-There*, *TheIconicSystem* incluía guantes especiales para facilitar el seguimiento de la persona.

En 1999 John Underkoffler también con el MIT y con Mark Lucente unos pocos años antes de la holografía, fue invitado a trabajar en un nuevo proyecto de Stephen Spielberg llamado *TheMinorityReport*. Underkoffler eventualmente se convirtió en el Asesor de Ciencia y Tecnología en la película y, con Alex McDowell, diseñador de producción de la película, creo la interfaz de usuario que utiliza Tom Cruise en la película. Algunos de los conceptos de diseño de la interfaz de usuario de *MinorityReport*, terminaron en otro proyecto de Underkoffler llamado G-Speak.Pero porque mencionar esta película, antes de empezar con esto debemos mencionar, a la ciencia ficción, ejemplo de esto es la franquicia de StarTrek, es que no ellos simplemente predecían el futuro. Cuando entras por primera vez a través de puertas automáticas en una tienda de conveniencia local, como no pensar que esto estaba basado en las puertas corredizas de la USS Enterprise.[14]

Así de esta manera TheMinorityReport impulsó el diseño y la adopción del sistema de reconocimiento de gestos de Kinect, StarTrek se puede decir que impulso las capacidades de reconocimiento de voz de Kinect. En las entrevistas con los empleados y ejecutivos de Microsoft, hay repetidas

referencias a hacer que el Kinect funcione como la computadora de StarTrek. Hay un sentido en esas entrevistas que si la parte de reconocimiento de voz del dispositivo no se habría resuelto, el sensor de Kinect no habría sido el dispositivo que cada miembro del equipo quería. [14]

3.1.1 Microsoft Project

En el mundo de los videojuegos, Nintendo lanzó *gautlet* en el TokyoGame Show en 2005 junto a la presentación de la consola Wii. La consola fue acompañada por un nuevo dispositivo de juego llamado*Wii Remoto*. Al igual que los cubos magnéticos del proyecto original *Put-That-There*, el mando de Wii puede detectar movimiento en tres dimensiones. Además, el mando a distancia contiene un sensor óptico que detecta donde está señalando. También es alimentado por batería, eliminando los cables largos para la consola común a otras plataformas.Después del lanzamiento de la Wii en 2006, Peter Moore, entonces jefe de la división Xbox de Microsoft, trabajo en un proyecto que pudiera competir con el Wii. Fue también en esta época que Alex Kipman, director de un equipo de incubación dentro de la división de Xbox, se reunió con los fundadores de PrimeSense en el 2006. Microsoft creo dos equipos que competían para crear el proyecto capaz de acabar con el Wii: uno que trabajaba con la tecnología PrimeSense y los otros que trabajan con tecnología desarrollada por una empresa llamada 3DV. Aunque el objetivo inicial era dar a conocer algo en el 2007, ningún equipo parecía tener nada suficientemente pulido a tiempo para la exposición, tiempo después Peter Moore dejo el proyecto para ir a trabajar ElectronicArts esto indicaba que no se tenía nada claro para trabajar [14].

Tras la salida de Moore, Don Matrick tomó las riendas, dirigiendo el equipo de Xbox. En 2008, revivió el proyecto de reconocimiento de vídeo secreto en torno a la tecnología de PrimeSense. Si bien la tecnología de 3DV nunca llegó a cubrir lo que se quería para el Kinect, Microsoft compró la compañía en 2009 por $ 35 millones. Al parecer, esto se hizo con el fin de defenderse de posibles disputas de patentes alrededor de Kinect. Alex Kipman, gerente de Microsoft desde el año 2001, se hizo director general de Incubación y puesto a cargo de la creación del nuevo dispositivo de *Project Natal* para incluir el reconocimiento de la profundidad, el seguimiento de movimiento, reconocimiento facial y reconocimiento de voz. El dispositivo de referencia creado por PrimeSense incluye una cámara RGB, un sensor de infrarrojos, y una fuente de luz infrarroja. Microsoft patento el diseño de PrimeSense y el diseño de chips PS1080, que procesan los datos de profundidad a 30 fotogramas por segundo. Es importante destacar que los datos de profundidad, se procesan de una forma innovadora que redujo drásticamente el precio del reconocimiento de la profundidad en comparación con el método vigente llamado *time of flight*, una técnica que controla el tiempo que tarda un rayo de luz para salir y luego volver

a el sensor. La solución de PrimeSense fue proyectar un patrón de puntos infrarrojos a través del cuarto y usar el tamaño y el espacio entre los puntos para formar un mapa de profundidad de pixeles de 320X240 analizado por el chip PS1080. El chip también alinea automáticamente la información para la cámara RGB y la cámara de infrarrojos, proporcionando datos RGBD a los demás sistemas. [14]

Microsoft añadió un conjunto de micrófonos de cuatro piezas de esta estructura básica, proporcionando efectivamente un micrófono para el reconocimiento de voz que sería eficaz en una gran sala. Microsoft ya ha tenido años de experiencia con el reconocimiento de voz, que ha estado disponible en los sistemas operativos desde Windows XP.Si bien los problemas que se tenían con respecto al hardware se resolvieron gracias a PrimeSense, todo lo que quedaba era dar al dispositivo una forma más pequeña, en cambio los retos de software parecían insuperables. En primer lugar, el sistema de reconocimiento de movimiento tuvo que ser creado en base al RGB y los flujos de datos procedentes de Profundidad del dispositivo. El equipo de Project Natal se dirigió a Microsoft Research(MSR) para ayudar a resolver estos problemas. MSR es una multimillonaria inversión anual por Microsoft. Las diversas ubicaciones MSR son típicamente dedicadas a la investigación pura en ciencias de la computación e ingeniería en lugar de tratar de llegar a nuevos productos. Debe haber parecido extraño, entonces, cuando el equipo se acercó a Xbox, no sólo para ayudarles a llegar a un producto, sino que lo hagan de acuerdo con los ritmos de un ciclo de producción muy corto.[3]

El Kinect se introdujo al mercado de los video juegos el 4 de noviembre del 2010, rompiendo record de ventas y siendo aceptado por los usuarios de una buena manera, era simplemente lo que la gente quería, si jugaban con él todos los días o sólo por unas pocas horas. Era un pedazo de futuro que podrían tener en sus hogares.

El 17 de junio de 2011, Microsoft finalmente lanzó la beta del SDK de Kinect para el público bajo una licencia no comercial, después de demostrar que durante varias semanas en eventos como MIX. Según lo prometido, incluía los algoritmos de reconocimiento de movimientos corporales (*skeletonrecognition)* que hacen una primera pose innecesaria, así como la tecnología AEC y modelos acústicos necesarios para hacer el trabajo de reconocimiento de voz de Kinect en una gran sala. Todos los desarrolladores ahora tienen acceso a las mismas herramientas de Microsoft que utiliza internamente para el desarrollo de aplicaciones de Kinect. [14]

3.1.1.1 Aplicaciones

Kinect surgió como la idea de divertir a la gente y competir con las grandes marcas de video juegos, de esta forma entro en el mercado sin saber que le

esperaría, en el área de los video juegos tuvo una enorme aceptación pero lo que nadie se imaginaba era la cantidad de proyectos y aplicaciones que tenía escondido este dispositivo, gracias a las características y flexibilidad que tiene ha sido un dispositivo aceptado por las comunidades dedicadas a la investigación y al desarrollo de nuevas tecnologías, a continuación se mencionan algunas áreas donde el kinect ha tenido un impacto importante: [15]

- Video Juegos
- Mercadeo.
- Medicina
- Computación
- Realidad Aumentada
- Recreación de escenarios

Actualmente kinect se encuentra en un momento cumbre ya que la cantidad de aplicaciones que tiene es impresionante, la aceptación en el campo de la computación ha sido enorme y dado esto Microsoft se ha dado a la tarea de crear un área para el desarrollo exclusivo de proyectos orientados al uso de kinect.

3.1.2 Descripción

Kinect, figura 3.1, cuenta con una cámara RGB, un sensor de profundidad, un micrófono *multi-array* y un procesador personalizado que ejecuta el software patentado, que proporciona captura de movimiento de todo el cuerpo en 3D, reconocimiento facial y capacidades de reconocimiento de voz. El micrófono de Kinect permite llevar a cabo la localización de la fuente acústica y la supresión de ruido ambiental, permitiendo participar en el chat de Xbox Live sin utilizar auriculares.

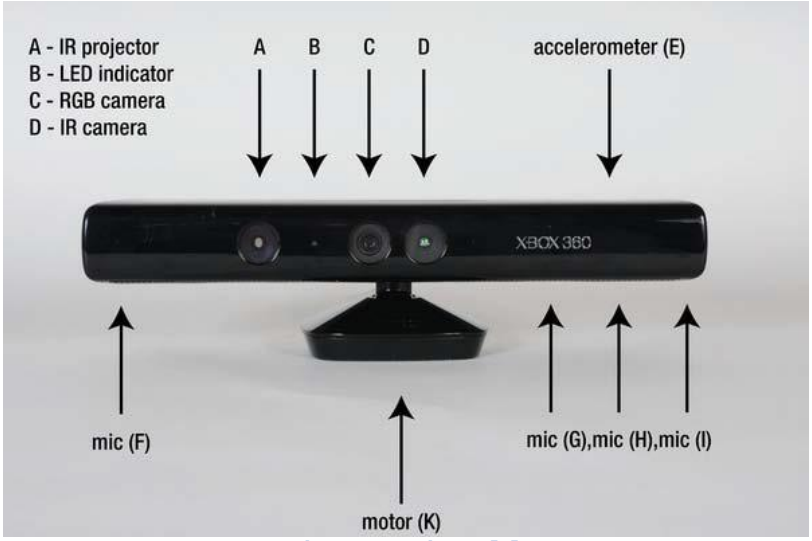


Figura 3.1 Kinect [2]

Las cámaras tienen dos resoluciones (320×240 y 640×480 de alto color) y envían datos con una frecuencia de actualización de 30 fps.[2]

3.1.2.1 Sensores

- Lentes de color y sensación de profundidad.
- Micrófono multi-arreglo.
- Ajuste de sensor con su motor de inclinación.
- Totalmente compatible con las consolas existentes de Xbox 360.

3.1.2.2 Campo de visión

- Campo de visión horizontal: 57 grados.
- Campo de visión vertical: 43 grados.
- Rango de inclinación física: ± 27 grados.
- Rango de profundidad del sensor: 1,2 - 3,5 metros.

3.1.2.3 Data Streams (Flujo de datos)

- 320 × 240 a 16 bits de profundidad a 30fps.
- 640 × 480 32-bit de color a 30fps.
- Audio de 16-bit a 16 kHz.

3.1.2.4 Sistema de Seguimiento

- Rastrea hasta 6 personas.
- Rastrea 20 articulaciones por persona.

3.1.2.5 Sistema de audio

- Sistema de cancelación de eco que aumenta la entrada de voz.
- Reconocimiento de voz múltiple.

3.1.3 Requerimientos para el uso de Kinect [14]

A continuación se enlistan las características físicas y lógicas que requiere el kinect para funcionar en una PC.

3.1.3.1 Hardware

- Computadora con procesador a 2.66 GHz o mayor.
- Tarjeta grafica 512 MB (recomendado 1 GB).
- Windows 7 o mayor.
- Al menos 2 GB RAM (recomendado 4 GB).
- Kinect para Xbox 360.
- Adaptador USB de Kinect.

3.1.3.2 Software

- LibreriasOpenNi para kinect.
- Microsoft .NET Framework 4
- SDK kinectpara Windows

3.2 Escaneo/Reconstrucción 3D

El **Escaneo**, proveniente de la palabra *escanear*, según el diccionario de la Real Academia Española, es pasar por el *escáner*, donde este último a su vez se define como, *un dispositivo que explora un espacio o imagen, y los traduce en señales eléctricas para su procesamiento*[16].

En sentido Homólogo, el **Escaneo 3D** es el *proceso de la captura digital de la silueta o la forma de un objeto con equipo para dicho propósito*[16]. Cabe mencionar que este tipo de Escaneo tiene muchas aplicaciones, desde documentación de un entorno hasta ingeniería inversa entre otras.

Por otro lado, la **Reconstrucción,** el diccionario de la RAE lo define como *acción y efecto de reconstruir,* donde reconstruir se define como, *volver a construir*[16].

De manera similar la **Reconstrucción 3D** podríamos definirla como *volver a construir un objeto a partir de otro*[16].

En este sentido y como conclusión, podríamos decir y afirmar que para una Reconstrucción 3D de manera virtual (que es lo que interesa en este estudio), necesita preferente de un Escaneo 3D.

3.3 Nube de Puntos (Cloud Points)

Una nube de puntos es un conjunto de vértices en un sistema de coordenadas tridimensional. Estos vértices se identifican habitualmente como coordenadas X, Y, y Z y son representaciones de la superficie externa de un objeto.

Las nubes de puntos se crean habitualmente con un láser escáner tridimensional. Este instrumento mide de forma automática un gran número de puntos en la superficie de un objeto. La nube de puntos representa el conjunto de puntos que ha medido el dispositivo.

Una nube de puntos es un conjunto de puntos 3D no conectados. Se les llama "nubes", porque cuando visualiza, la falta de conectividad entre puntos hace que se vean como si estuvieran flotando en el espacio. El tipo más simple de la nube de puntos contiene sólo información de posición, pero cada punto también puede contener otras propiedades (por ejemplo, el color y la orientación normal). Las nubes de puntos se pueden utilizar como un modelo mundial, como en la navegación del robot para evitar la colisión con puntos por encima del plano de tierra, o se pueden usar como un paso intermedio antes de reconstrucción de la superficie para producir un modelo de malla.

Una nube de puntos es la forma natural de representar la información obtenida de la Kinect, ya que cada píxel de la imagen de fondo puede ser transformado en un punto 3-D, pero no sabemos si se conecta a cualquier otro punto. Vamos a utilizar la Point Cloud Library (PCL) para almacenar los puntos [17].

3.4 Mesh

Malla (Mesh por uso preferente en Ingles). *MeshModels* son una forma compacta de representar los datos en 3-D y son la representación de la opción para los juegos de computadora y la animación 3-D. Ellos modelan la superficie de un objeto utilizando polígonos (generalmente triángulos). Las áreas planas requieren sólo unos pocos polígonos, mientras que las áreas complejas pueden utilizar muchos polígonos para representar la superficie con gran detalle. Ellos son el equivalente 3-D de imágenes vectoriales en 2-D.

MeshModels son una forma muy flexible y eficiente de representar los datos en 3-D, pero son difíciles de construir. Artista 3-D selecciona las posiciones de los

vértices y las orientaciones normales y determina la conectividad de la superficie. Sin embargo, desde el Kinect, sólo tenemos la posición 3-D y el color. Nosotros no tenemos ninguna información acerca de la orientación de la superficie o de la conexión de los píxeles en 3-D. Por lo tanto, es necesaria una representación más simple para empezar.

Existen varios algoritmos que pueden extraer de un mesh de triángulos de una nube de puntos con normalidad. La principal dificultad consiste en determinar qué vértices deben conectarse. La distancia simple entre los vértices no es un buen criterio, porque la escala de la nube de puntos y su densidad puede variar. Conexión de los vecinos más cercanos es una mejor manera, pero se debe tener cuidado para evitar las intersecciones del mismo en la superficie. [17]

Por lo tanto, tras un estudio profundo sobre lo que se pretende lograr, se propone como algoritmo para lograr el modelado 3D, el algoritmo Mesh, que por sus características, resulta óptimo para esta tarea.

3.5 Micro controlador

Se le llama microcontrolador a un circuito integrado que es capaz de tener memoria, unidad de operaciones y puertos programables. Donde un circuito integrado se define como una pequeña placa que conjunta varios componentes electrónicos interconectados para una tarea específica.

Hoy en día, un micro controlador, es un recurso muy común, debido a que:

- Están embebidos dentro de algunos dispositivos, donde pueden controlar acciones de los productos, es por esto que se les puede conocer como controlador embebido.
- Son dedicados a una sola tarea y ejecuta un específico programa.
- Casi siempre son de bajo consumo de energía.
- Son baratos, en comparación de otros dispositivos electrónicos que podrían realizar las mismas tareas.
- Son robustos de alguna forma, según la tarea que desarrollen.
- La programación de los sistemas dentro del micro controlador, siempre resulta ser una tarea bastante sencilla y cómoda.

Como es de imaginarse, existe un micro controlador para cada necesidad que se necesita resolver, así como muchas empresas encargadas de su desarrollo.

A continuación en la Tabla 3.1, se realiza la comparación de algunostiposde marcas de micro controladores, donde se muestran características de algunos modelos siendo estos los más económicos y completos.

Tabla 3.1 Micro controladores

Marca	Positivo	Negativo
ARM	- Bajo consumo - Muchos periféricos - Gran velocidad de reloj (60 – 150 MHz)	- Bastante complejo
Texas Instruments	- Bajo consumo - Portabilidad de código	- Se basan en la arquitectura de von Neuman
PIC	- Muy baratos - Software de desarrollo gratis - Gran Diversidad	- Programación complicada - No hay compatibilidad
ATMEL	- Bajo consumo - Arquitectura RISC Facilidad de programación	- No es muy rápido

La Tabla 3.2, que se muestra a continuación, muestra un listado de microcontroladoresque son económicos y completos en cuanto a la aplicación que se le planea dar.

Tabla 3.2 Micro controladores ATMEL [26]

Micro controlador	Voltaje entrada	ClockSpeed	Analog Inputs	Flash Space	Programming Interface
AT91SAM3X8E	5-12V	84MHz	12	512Kb	USB native
ATmega32U4	7-12V	16MHz	12	32Kb	USB native
ATmega328*	7-12V	16MHz	6	32Kb	USB via ATMega16U2
ATmega328	7-15V	16MHz	6	32Kb	USB via FTDI, USB via ATMega8U2, USB via FTDI, Bluetooth serial, FTDI-compatible Header
ATmega2560	7-12V	16MHz	16	256Kb	USB via ATMega16U2

ATmega1280	7-12V	16MHz	16	128Kb	USB via FTDI
ATmega2560	3.3-12V	8MHz	16	256Kb	FTDI-Compatible Header
ATmega2560	5-12V	16MHz	16	256Kb	FTDI-Compatible Header
ATmega328	7-9V	16MHz	6	32Kb	Serial Header
ATmega328	3.35 - 12V	8MHz	6	32Kb	FTDI-Compatible Header
ATmega328	5 - 12V	16MHz	6	32Kb	FTDI-Compatible Header
ATmega328P	3.35 - 12V	8MHz	8	32Kb	FTDI-Compatible Header or Wirelessly via XBee[1]
ATmega2560	3.3-12V	8MHz	16	256Kb	FTDI-Compatible Header
ATmega32U4	5 - 12V	16MHz	4	32Kb	Native USB
ATmega32U4	3.35 - 2V	8MHz	4	32Kb	Native USB
ATmega328	2.7-5.5V	8MHz	6	32Kb	FTDI-Compatible Header
ATmega8535	4.5 – 5.5V	16MHz		8Kb	

4 Factibilidad

Para el desarrollo de este proyecto se tomaron en cuenta cuatro tipos de Factibilidad.

- Factibilidad Económica
- Factibilidad Operacional
- Factibilidad Técnica

Se tomarán en cuenta los aspectos antes mencionados, ya que al utilizar el Sensor Kinect nos encontramos ante una nueva rama de la tecnología que llego al mercado aproximadamente hace 2 años y aún se encuentra en desarrollo.

El sensor Kinect actualmente es un área de investigación de la cual se pueden obtener resultados impresionantes si se conoce y tienen los conocimientos necesarios para trabajar con esta herramienta.

4.1 Factibilidad Económica

Consiste en un análisis de costos de los recursos que se requieren para la implementación del proyecto, comparando el valor económico de otros proyectos con el producto final obtenido.

A continuación se muestra la Tabla 4.1, con los precios aproximados de los recursos que son necesarios para la realización del trabajo propuesto.

Tabla 4.1 Precios

Recurso	Costo($)
Sensor Kinect	2500
Micro controlador	95
Software de desarrollo	Libre
Servo Motor	170 aprox.
	Total 2765

Por otro lado, en la Tabla 4.2, se muestran algunasaplicaciones y dispositivos capaces de realizar una tarea similar que tienen un precio bastante elevado, siendo estos los que se encuentran en el mercado actualmente.

Tabla 4.2 Comparativa de precios

Proyecto	Costo(dólares)
Autodesk123 (Aplicaciones varias)	199 (membrecía de un año de uso)
Metrascan series	20000 aprox.
NextEngine 3DScanner	2995

Esto son solo algunos ejemplos de proyectos que se han desarrollado y están en el mercado actualmente, el obtener cualquier de estas aplicaciones requiere de una inversión monetaria importante.

Dada la comparación de costos, resulta evidente que el desarrollo de este proyecto es factible por los bajos costos que generan el desarrollo y la implementación.

4.2 Factibilidad Operacional.

La Factibilidad Operacional radica en que este nuevo sistema se use para lo que fue planteado, sin que existan ambigüedades en su uso.Se consideraron cuatro aspectos para determinar si el proyecto es factible Operacionalmente:

- No debe de existir complejidad para hacer uso del sistema.
- Este nuevo sistema puede hacer que los usuarios se resistan a él como consecuencia de una técnica de trabajo.
- Un sistema nuevo puede introducir cambios demasiado rápidos que no permita adaptarse a él y aceptarlo.
- La probabilidad de obsolescencia en el sistema.

Para que el proyecto sea factible operacionalmente mencionaremos las razones por las cuales el sistema cubrirá estos aspectos:

- El sistema no será difícil de manipular ya que la interfaz de usuario será sencilla e intuitiva (ver pantallas), y si existiera algún problema al utilizar el sistema ya sea técnico o de usuario se resolvería con el manual correspondiente.
- Dado que esta tecnología se encuentra en desarrollo, un sistema nuevo en este ámbito no posee un predecesor y sería un producto que introduce un concepto nuevo al mercado utilizando estos recursos.

Ya que el Kinect tuvo una aceptación importante en el mercado, ha ganado confianza y credibilidad para la creación de nuevas aplicaciones.

4.3 Factibilidad Técnica.

En esta sección se evalúa si se cuenta con el equipo y software necesarios y que estos tengan las capacidades para el desarrollo del sistema, se consideran las técnicas y conocimientos que tenga el equipo de trabajo para el diseño, implementación y mantenimiento del sistema propuesto.

Para el desarrollo del proyecto los siguientes materiales son indispensables para su desarrollo:

- Sensor kinect
- Servomotor
- Microcontrolador
- SDK de Microsoft

Así como conocimiento de las siguientes tecnologías:

- Lenguaje de programación Java.
- Lenguaje C para la programación del microcontrolador

Para la construcción del modelo

Dado que contamos con los materiales antes mencionados, y experiencia con el desarrollo de aplicaciones con el lenguaje java podemos decir que el proyecto es factible técnicamente y es posible realizar lo que anteriormente se expone.

5 Análisis.

5.1 Descripción del proceso.

La figura 5.1, nos muestra lo que proponemos como proceso básico de escaneo, el cual especifica que el usuario necesita colocar un objeto en la base de nuestro dispositivo, después deberá iniciar el escaneo, lo que nos llevará a una confirmación en el sistema, la cual verificará si el sensor kinect está conectado, en caso afirmativo el sensor comenzará el escaneo, recogerá los datos y pasará esa información al sistema la cual procesara, ordenara y la presentará de vuelta al usuario, como un modelo en tres dimensiones. Este tipo de diagramas se conoce como diagrama de proceso.

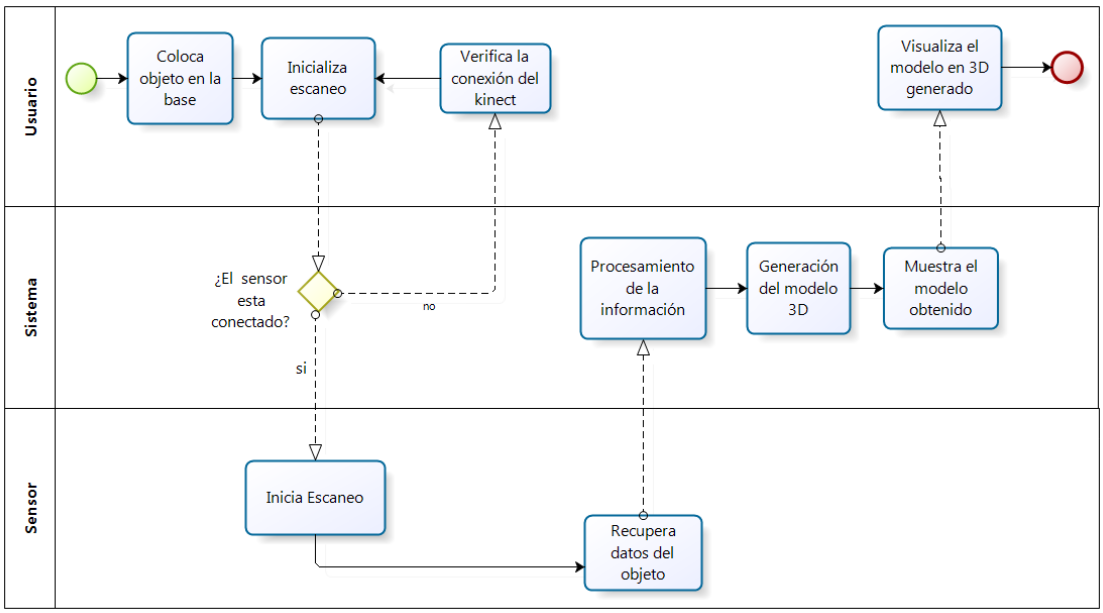


Figura 5.1 Proceso para obtener un modelo en 3D y su visualización

5.2 Especificación de requerimientos.

5.2.1 Requerimientos funcionales.

Para definir todo sistema, siempre es recomendable una buena especificación de requerimientos o de funcionalidades principales con las que debe de contar un sistema. En la Tabla 5.1, se enlistan los requerimientos funcionales, con la finalidad de especificar lo quetiene que realizar el sistema.

Tabla 5.1 Requerimientos funcionales

Id	Nombre	Descripción
RF1	Escaneo	El sistema deberá escanear objetos no más pesados de 2.7 kg ni más alto a 1.80m
RF2	Visualización	Una vez realizado el escaneo, se podrá visualizar el objeto obtenido.
RF3	Abrir	El sistema podrá abrir archivos previamente escaneados
RF4	Guardar	La aplicación podrá guardar o exportar el objeto escaneado
RF5	Adquisición de datos del Escaneo 3D	Manipulará del sensor para la obtención de los datos del objeto.
RF6	Construcción del modelo 3D	El sistema procesará los datos obtenidos y modelará estos.
RF7	Gestión de conexión del Escáner	El sistema verificará si el escáner está conectado y bien calibrado

5.2.2 Requerimientos no funcionales.

Por otro lado, la descripción de los requerimientos no funcionales, engloba todas aquellas características que son deseables en nuestro sistema pero no son parte de la funcionalidad como tal, con tal efecto, la Tabla 5.2, nos muestra todas aquellas características que son esenciales y que no tienen interacción directa con la funcionalidad del sistema.

Tabla 5.2 Requerimientos no funcionales

Id	Nombre	Descripción
RNF1	Accesibilidad	El sistema podrá ser usado por aquellas personas con un conocimiento básico en informática, exceptuando a las personas con debilidad visual.
RFN2|	Usabilidad	El uso del software será intuitivo y no se necesita de capacitación para hacer uso de él.
RNF3	Interfaz	La aplicación contará con una interfaz cómoda y de fácil entendimiento.
RNF4	Operatividad	El sistema entregará los resultados de la misma forma, tras el escaneo de cada objeto.
RNF5	Desempeño	El desempeño del sistema será proporcional al tamaño del objeto analizado, y de la calidad con la que se presente.

5.3 Diagrama de casos de uso.

La Figura 5.2, es un diagrama de casos de uso, el cual estáencargado de describir como es la interacción entre las funcionalidades del sistema, así como definir en que puede derivar cada acción del sistema o que necesita para funcionar, por ejemplo, el presente diagrama nos dice que para escanear es necesaria una adquisición de datos por medio del escáner, el cual a su vez tiene que estar gestionado y validar que se encuentre conectado el sensor. Por otro lado un escaneo necesita de la construcción de un modelo y este se puede derivar en la visualización, y a su vez en la apertura de un escaneo previo.

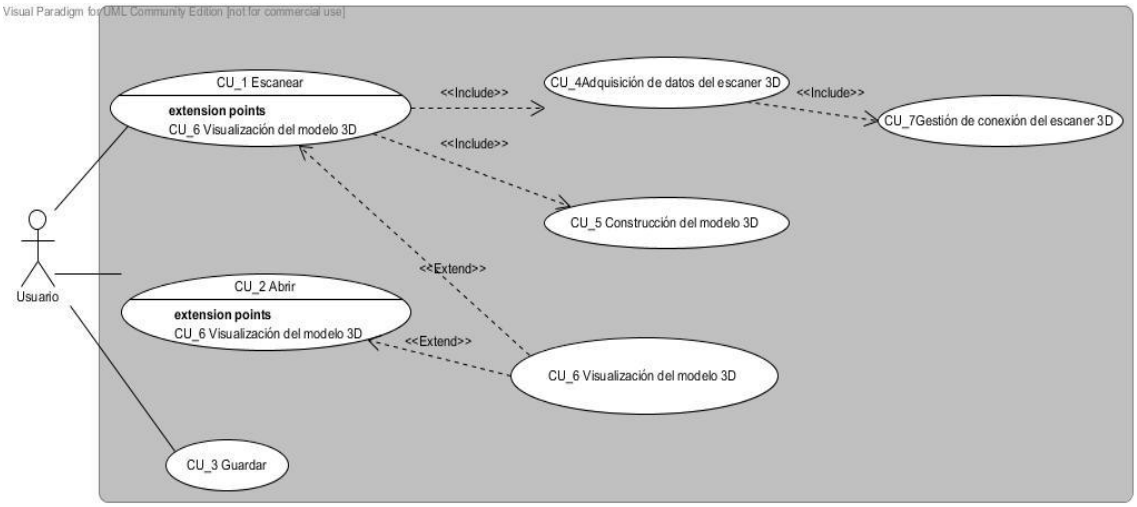


Figura 5.2 Diagrama de casos de uso

5.4 Especificación de casos de uso.

5.4.1 CU_1 Escanear.

Descripción completa
Este caso de uso el usuario podrá dar inicio a la etapa de escaneo del objeto. La tabla 5.3, describe de manera específica el caso de uso CU_1.Se recomienda su lectura debido a que nos ayudará entender mejor el funcionamiento de esta acción.

Tabla 5.3 CU_1 Escanear

Caso de Uso:	CU_1 Escanear
Versión:	1.0
Actor:	Usuario
Propósito:	Inicializar el escaneo del objeto físico deseado.
Entradas:	Objeto físico.
Salidas:	Visualización del objeto.
Precondiciones:	El sensor kinect debe estar calibrado.
Postcondiciones:	El usuario podrá visualizar el objeto escaneado
Tipo:	Primario

Trayectorias del CU

Trayectoria Principal

1. El sistema muestra la pantalla P1
2. Presiona el botón “Comenzar” en la pantalla P1.[Trayectoria Alternativa A]
3. Muestra la ventana M1 de confirmación.
4. Da clic en el botón “Si” de la ventana M1.[Trayectoria Alternativa B]
5. Inicia el escaneo.

Fin de caso de uso.

Trayectoria Alternativa A.
Condición. Si el sensor no se encuentra conectado.

1. Muestra el mensaje de error M4.

2.  Regresa a la pantalla P1.

Fin de caso de uso.

Trayectoria Alternativa B.
Condición. Si el usuario presionó el botón “no”.

1.  Regresa a la pantalla P1.

Fin del caso de uso.

5.4.2 CU_2 Abrir

Descripción completa
En este caso de uso el usuario podrá elegir el archivo previamente escaneado para visualizarlo. La tabla 5.4, describe de manera específica el caso de uso CU_2.Se recomienda su lectura debido a que nos ayudará entender mejor el funcionamiento de esta acción.

Tabla 5.4 CU_2 Abrir

Caso de Uso:	CU_2 Abrir
Versión:	1.0
Actor:	Usuario
Propósito:	Visualizar un objeto previamente escaneado.
Entradas:	Archivo de escaneo del objeto.
Salidas:	Visualización del objeto.
Precondiciones:	Debe existir un objeto previamente escaneado.
Postcondiciones:	El usuario podrá visualizar el objeto elegido.
Tipo:	Primario

Trayectorias del CU

Trayectoria Principal

1.  El sistema muestra la pantalla P1
2.  Presiona el botón “Abrir” en la pantalla P1.
3.  Muestra la pantalla P3.[Trayectoria Alternativa A]
4.  Selecciona el archivo a visualizar.

5.  Da clic en el botón “Abrir” de la ventana P3.[Trayectoria Alternativa B.]
6.  Muestra el objeto seleccionado en la pantalla P1.

Fin de caso de uso.

Trayectoria Alternativa A
Condición. Si el usuario presiono el botón “Cancelar”.

1.  Regresa a la pantalla P1.

Fin de caso de uso.

Trayectoria Alternativa B.
Condición. Si el archivo seleccionado no es compatible.

1.  Muestra el mensaje de error M5.

Fin de la trayectoria.

5.4.3 CU_3 Guardar

Descripción completa
Este caso de uso el usuario podrá guardar el objeto analizado para su posterior visualización. La tabla 5.5, describe de manera específica el caso de uso CU_3.Se recomienda su lectura debido a que nos ayudará entender mejor el funcionamiento de esta acción.

Tabla 5.5 CU_3 Guardar

Caso de Uso:	CU_3 Guardar
Versión:	1.0
Actor:	Usuario
Propósito:	Guardar una visualización de un objeto escaneado
Entradas:	Modelo en 3D de objeto ya escaneado
Salidas:	Archivo de visualización del objeto.
Precondiciones:	Modelo en 3D de objeto ya escaneado y en visualización
Postcondiciones:	El usuario tendrá el archivo que poseerá los datos del objeto ya modelado para su posterior visualización.
Tipo:	Primario

Trayectorias del CU

Trayectoria Principal

1. El sistema muestra la pantalla P1, el modelo ya escaneado.
2. Presiona el botón “Guardar” en la pantalla P1.[Trayectoria Alternativa A]
3. Muestra la ventana P4. [Trayectoria alternativa B]
4. Selecciona la ruta y asigna nombre para guardar.
5. Da clic en el botón “Guardar” de la pantalla P4.
6. Guarda el archivo.
7. Muestra el mensaje M7.
8. Regresa a la pantalla P1.

Fin de caso de uso.

Trayectoria Alternativa A.
Condición: No existo escaneo previo.

1. Muestra el mensaje M6

Fin de caso de uso.

Trayectoria Alternativa B.
Condición. Si el usuario presiono el botón “Cancelar”.

1. Regresa a la pantalla P1.

Fin de caso de uso

5.4.4 CU_4 Adquisición de datos del Escáner

Descripción completa
En este caso de uso se recopilará la información obtenida por el sensor.
La tabla 5.6, describe de manera específica el caso de uso CU_4.

Tabla 5.6 CU_4 Adquisición de datos del escáner

Caso de Uso:	CU_4 Adquisición de datos del Escáner.
Versión:	1.0
Actor:	Sistema
Propósito:	Obtener la información recopilada por el sensor..
Entradas:	Objeto a escanear.
Salidas:	Datos para procesar.
Precondiciones:	CU_1 Escanear.
Postcondiciones:	El sistema tendrá los datos necesarios para generar el modelo.
Tipo:	Secundario.

Trayectorias del CU

Trayectoria principal

1. Inicia el proceso de escaneo.
2. Recopila la información que el escaneo proporciona en tiempo real.
3. Envía la información a la etapa de construcción del modelo.

Fin del caso de uso.

5.4.5 CU_5 Construcción del modelo 3D

Descripción completa
El sistema construirá el modelo en 3D del objeto ya escaneado.
Véase la tabla 5.7, que describe de manera específica el funcionamiento del caso de uso CU_5.

Tabla 5.7 CU_5 Construcción del modelo 3D

Caso de Uso:	CU_5 Construcción del modelo 3D
Versión:	1.0

Actor:	Sistema
Propósito:	Generar el modelo 3D del objeto previamente escaneado
Entradas:	Datos del Escáner 3D
Salidas:	Modelo 3D del objeto de estudio
Precondiciones:	CU_4 Adquisición de datos del Escáner.
Postcondiciones:	Se tendrá el modelo en 3D del objeto en cuestión.
Tipo:	Secundario

Trayectorias del CU

Trayectoria Principal

1. El sistema toma los datos del escaneo previo y mediante el algoritmo seleccionado se crea el modelo.

Fin de caso de uso.

5.4.6 CU_6 Visualización del modelo 3D

Descripción completa
El sistema mostrará el modelo obtenido en la construcción del modelo 3D
Sobre el caso de uso visualización del modelo 3D, la Tabla 5.8, muestra la información completa para su correcto entendimiento.

Tabla 5.8 CU_6 Visualización del modelo 3D

Caso de Uso:	CU_6 Visualización del modelo 3D
Versión:	1.0
Actor:	Sistema
Propósito:	Mostrar el modelo 3D obtenido.
Entradas:	Modelo 3D construido.
Salidas:	Visualización del modelo 3D ya construido.
Precondiciones:	CU_5 Construcción del modelo 3D
Postcondiciones:	Se apreciará el modelo en 3D del objeto analizado
Tipo:	Secundario

Trayectorias del CU

Trayectoria Principal

1. El sistema desplegará el modelo 3D construido previamente. [Trayectoria Alternativa A].

Fin de caso de uso.

Trayectoria Alternativa A.

1. Interactúa con el caso de uso CU_2.

Fin del caso de uso.

5.4.7 CU_7 Gestión de conexión del Escáner 3D

Descripción completa
El sistema podrá probar que el sensor se encuentre en las condiciones correctas.
La siguiente tabla, Tabla 5.9, especifica información importante para entender el presente caso de uso.

Tabla 5.9 CU_7 Gestión de conexión del escáner 3D

Caso de Uso:	CU_7 Gestión de conexión del Escáner 3D
Versión:	1.0
Actor:	Sistema
Propósito:	Verificar que el sensor se encuentre en condiciones adecuadas para realizar el escaneo.
Entradas:	Objeto de prueba..
Salidas:	Mensaje de verificación.
Precondiciones:	El sensor debe estar conectado.
Postcondiciones:	El sensor estará listo para realizar escaneos.
Tipo:	Primario

Trayectorias del CU

Trayectoria Principal

1. Presiona el botón “Configuración” en la pantalla P1.
2. Muestra la pantalla P2.
3. Presiona el botón “Calibrar” de la pantalla P2[Trayectoria Alternativa A].
4. Inicia la calibración del sensor, muestra el mensaje M10.
5. Fija el panel de calibración en un rango visible para el sensor hasta que el sistema lo indique [Trayectoria Alternativa B].
6. Muestra el mensaje M8 de éxito.
7. Presiona el botón “Testear” en la pantalla P2. [Trayectoria Alternativa A].

8.  Muestra la pantalla P5 con la imagen que capta el sensor.

Fin del caso de uso.

Trayectoria Alternativa A

Condición. Si el sensor no se encuentra conectado.

1.  Muestra el mensaje M4.
2.  Muestra la pantalla P2.

Fin de la trayectoria.

Trayectoria Alternativa B

Condición. Si la calibración no se realizó de manera adecuada.

1.  Muestra el mensaje M9.
2.  Muestra la pantalla P2.

Fin de la trayectoria.

5.5 Interfaces.

5.5.1 Pantallas

La Figura 5.3, nos da la perspectiva de cómo será la pantalla principal de la aplicación, siendo una interfaz, sencilla y concreta para los objetivos que se persiguen.

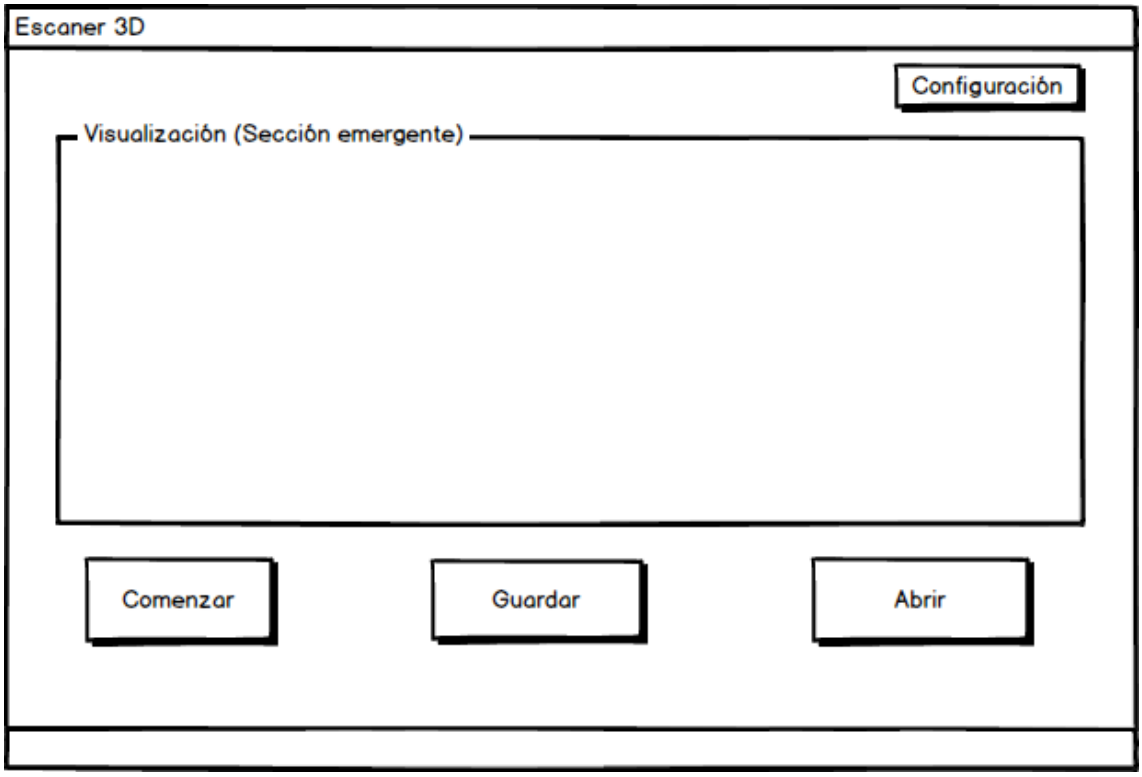


Figura 5.3 Pantalla inicio (P1)

Por otro lado, la Figura 5.4, nos muestra la pantalla con la que se han de configurar la conexión del sensor, siendo el primer botón la calibración del sensor para un correcto funcionamiento y el segundo para probar que el kinectesté conectado correctamente.

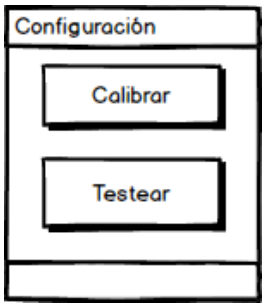


Figura 5.4 Pantalla Configuración (P2)

La Figura 5.5, y la Figura 5.6, nos muestran lo que podrían ser dos paneles para buscar un archivo para abrir o para buscar la ruta idónea para guardar nuestro modelo, respectivamente.

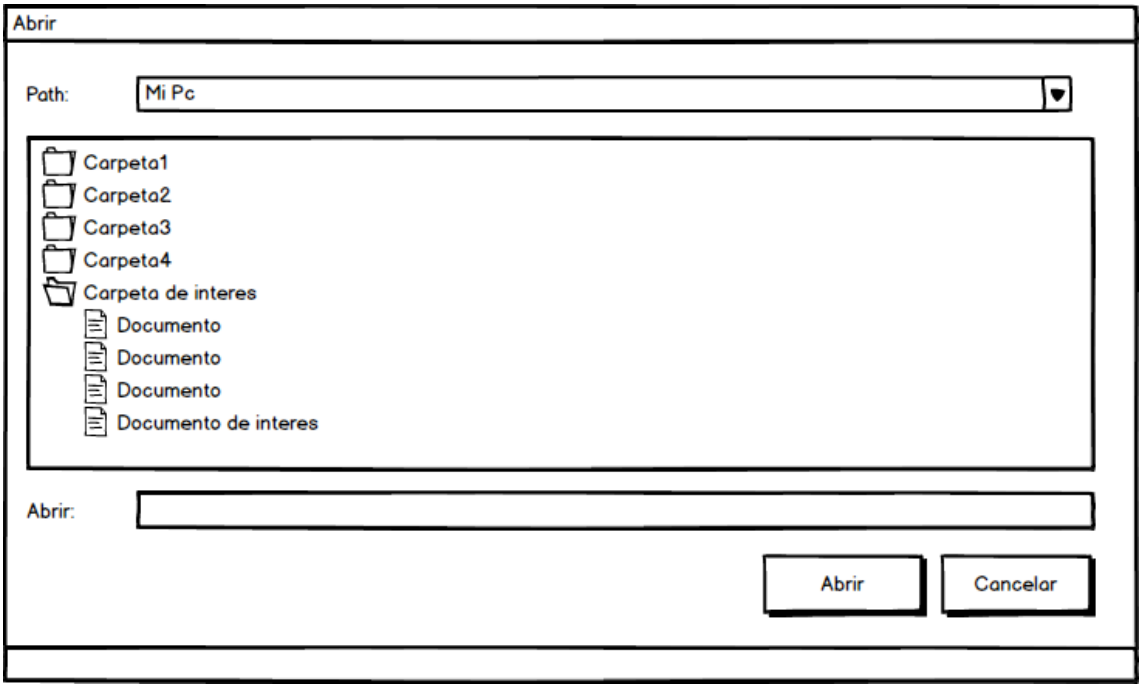


Figura 5.5 Pantalla Abrir (P3)

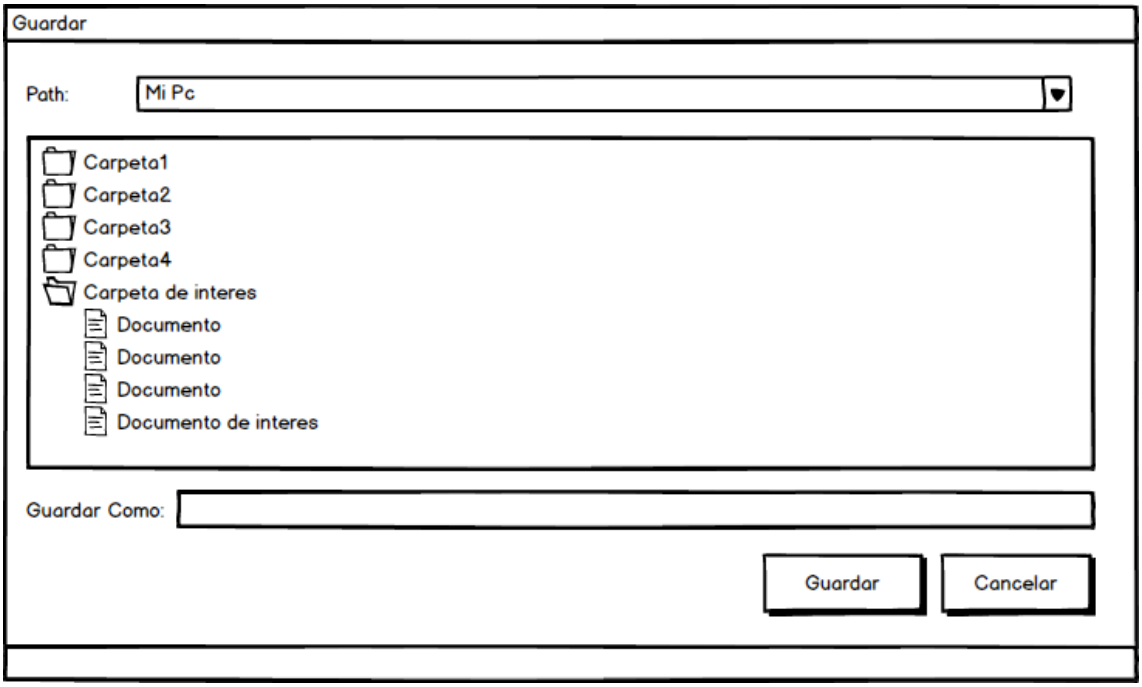


Figura 5.6 Pantalla Guardar (P4)

Por último, la Figura 5.7, nos muestra la pantalla con la que se hará la prueba (testing, por término técnico y de uso común): La zona de la cruz muestra el

espacio donde se desplegara la imagen que se recibe del dispositivo de escaneo.

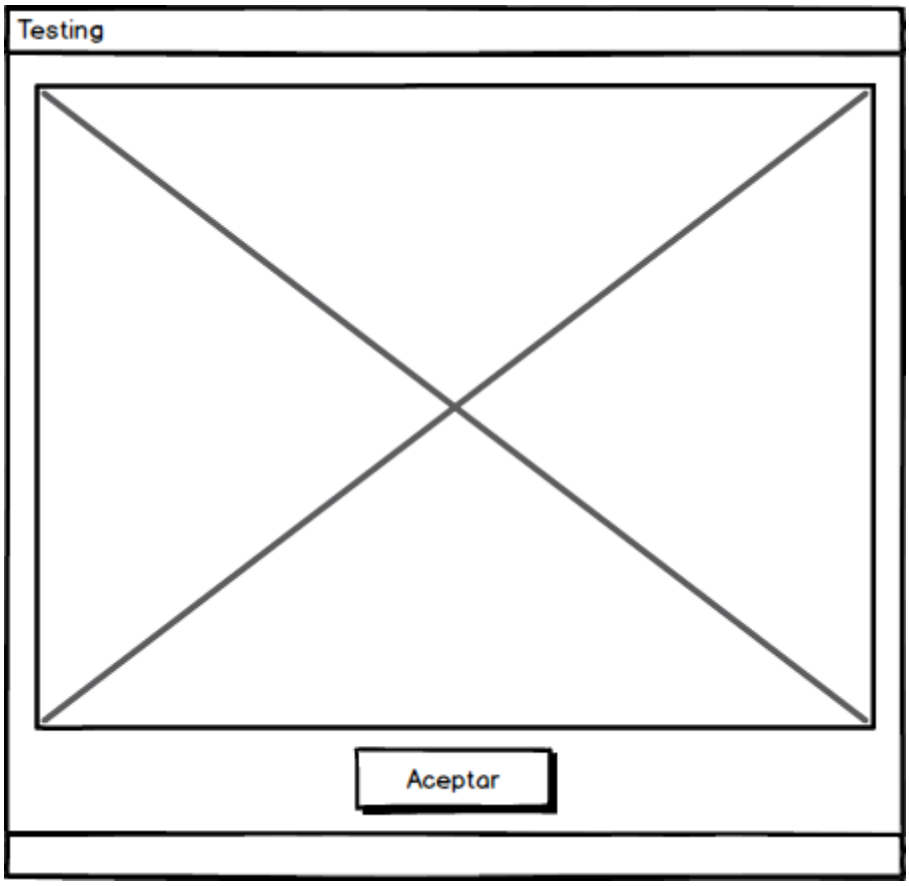


Figura 5.7 Pantalla prueba (P5)

5.5.2 Mensajes

Siempre es bueno tener un mensaje de confirmación para cualquier tarea, es por eso que la Figura 5.8, la Figura 5.9, y la Figura 5.10, nos muestran dichos mensajes para las acciones de comenzar, cancelar y salir.

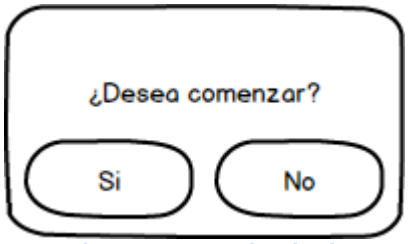


Figura5.1Mensaje 1(M1)

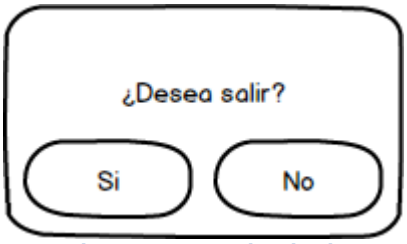


Figura5.2Mensaje 2(M2)

Así también, los descuidos son muy comunes, es por eso que la Figura5.11, Figura 5.12, y la Figura 5.13, nos muestra cómo avisaríamos al usuario de algún error que se esté cometiendo al usar la aplicación.

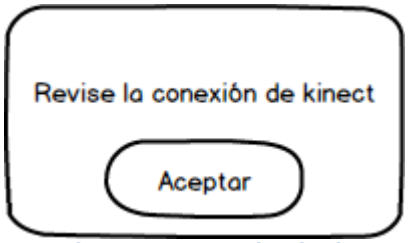


Figura 5.8 Mensaje 4(M4)

Figura 5.9 Mensaje 5(M5)

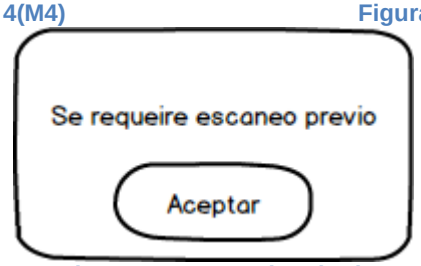


Figura 5.10 Mensaje 6 (M6)

También es una buena práctica, avisar cuando una tarea ya se realizo o cuando la operación que se estaba llevando a cabo ha concluido de manera exitosa, es por eso que la Figura 5.14, la Figura 5.15, y la Figura 5.16, muestran avisos sobre la tarea que se cumplió satisfactoriamente.

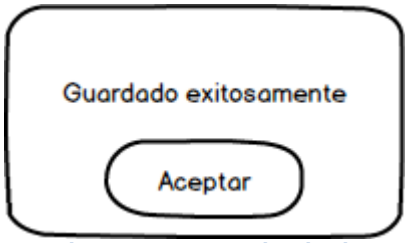


Figura 5.11 Mensaje 7(M7)

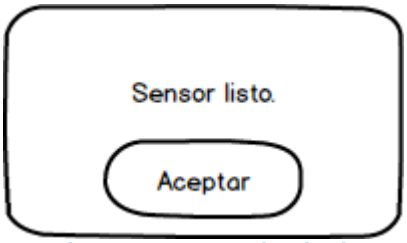


Figura 5.12 Mensaje 8(M8)

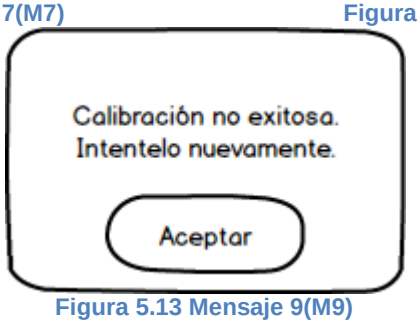


Figura 5.13 Mensaje 9(M9)

Y siempre es necesaria alguna notificación de instrucciones si es que se requiere que el usuario realice alguna tarea extra, es por eso que la Figura5.17, indica al usuario la tarea que tiene que realizar.

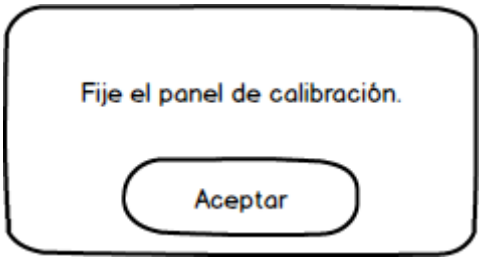


Figura 5.14 Mensaje 10(M10)

6 Diseño.

6.1 Diseño arquitectónico.

6.1.1 Arquitectura lógica.

Mediante el siguiente diagrama se describe la arquitectura lógica de la aplicación. La Figura6.1, nos muestra cómo interactúan todos los módulos del sistema, así como el uso de las interfaces para la transferencia de información. En este sentido, bajo este esquema podemos ver que el modulo Escanear, se relaciona con la adquisición de datos del escáner 3D mediante una interfaz de obtención de la información. De la misma manera se pueden aplicar para todos los módulos e interfaces aquí descritos.

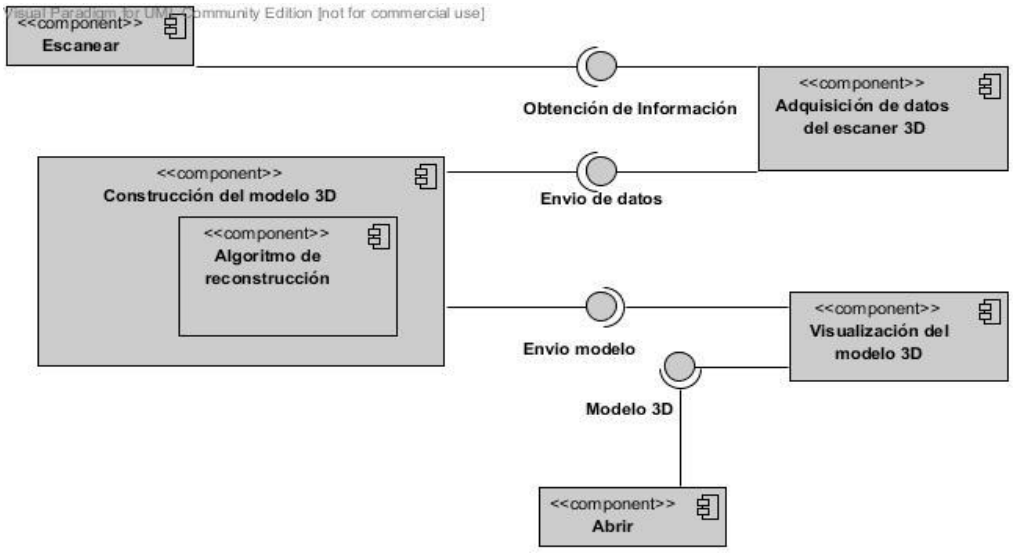


Figura 6.1 Arquitectura lógica (Diagrama de componentes)

Donde:

- **Escáner.-** Es el módulo donde se indicará al sensor en qué momento empezar la medición y en qué momento detenerse.
- **Adquisición de losdatos del escáner 3D.-** En esta parte se manejara el escáner de tal manera que se procese la información que deseamos de la manera que se espera.
- **Procesamiento de la información.-** Utilizando un componente interno, como lo es el algoritmo Mesh, se construirá el objeto virtualmente.
- **Visualización del modelo 3D.-** Módulo encargado de presentar al usuario la construcción realizada con el componente anterior.
- **Abrir.-**Componente que realizara la apertura de los archivos con el objeto ya construido.

6.1.2 Arquitectura física.

Dividiremos la explicación de la arquitectura física del sistema en dos esquemas. La Figura 6.2muestra el acomodo general de los componentes del sistema. Cabe mencionar que se marca una parte que será la que se explique a mayor detalle en la segunda sección de la arquitectura física.

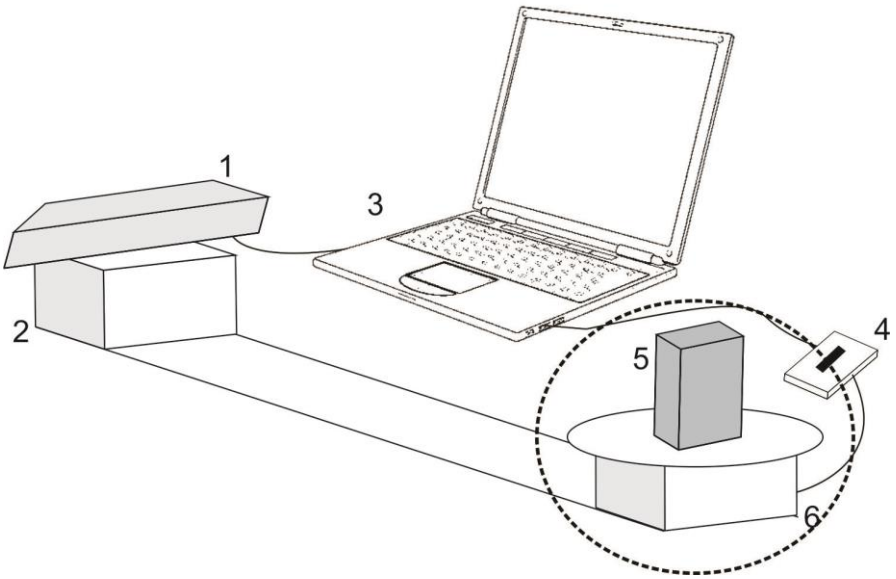


Figura 6.2 Arquitectura Física (a)

1) Escáner.- El escáner será el sensor Kinect.
2) Base para Escáner.- Este componente consiste básicamente en una plataforma donde el sensor kinect se encontrará.
3) Computadora Personal.- Será aquí donde se controlará el dispositivo de giro, se procesará toda la información visual recolectada por el escáner y donde se podrá visualizar.
4) Micro controlador.- Que llevará el control de un servomotor para manipular el dispositivo de giro, que se comunicará con la computadora personal vía USB (Bus de Serie Universal, USB por sus siglas en ingles)
5) Objeto.- Objeto de estudio.
6) Dispositivo de giro (Base del Objeto).- Esta etapa consiste en un dispositivo controlado desde la PC (Computadora Personal, PC por sus siglas en ingles) a través de un micro controlador que sea capaz de hacer girar el objeto sobre su propio eje.

Por otro lado, la Figura 6.3, muestra con mayor detalle la sección seleccionada en el apartado anterior, donde se despliega por componentes para su mejor apreciación.

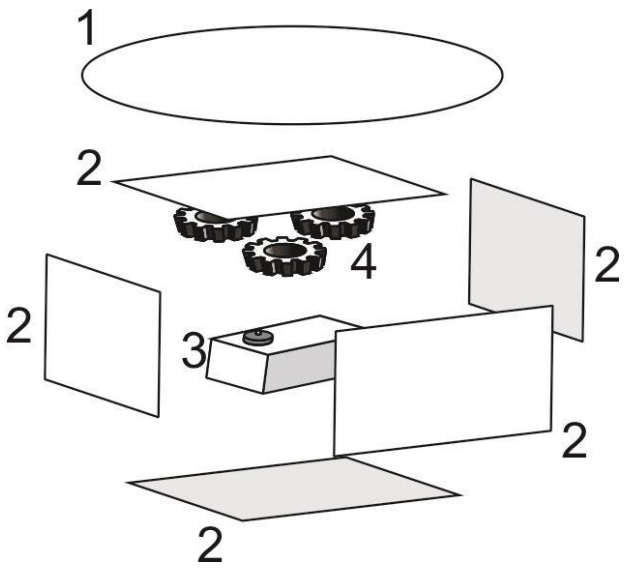


Figura 6.3 Arquitectura Física (b)

1) Plataforma.- Plataforma donde se colocara el objeto de interés, será la que gire.
2) Parte de carcasa.- Estructura que protegerá el funcionamiento interno.
3) Servo motor.- Dispositivo encargado del giro.
4) Engranes.- Facilitaran la tarea de mover la plataforma.

6.2 Diagrama de actividades.

La Figura 6.4, muestra que para lograr un modelado 3D se necesita primero iniciar el sistema, verificar si se encuentra conectado el sensor, de ser afirmativo, verificará si está calibrado el sensor, y de esta manera el usuario podrá comenzar el escaneo, el cual hará que el sensor obtenga la información del objeto en cuestión y la mandará al sistema, quien procesará esta información y mediante un algoritmo seleccionado, construirá el modelo, y lo presentará al usuario, donde decidirá si lo desea guardar o desechar.Este tipo de diagramas se le conoce como diagramas de actividades.

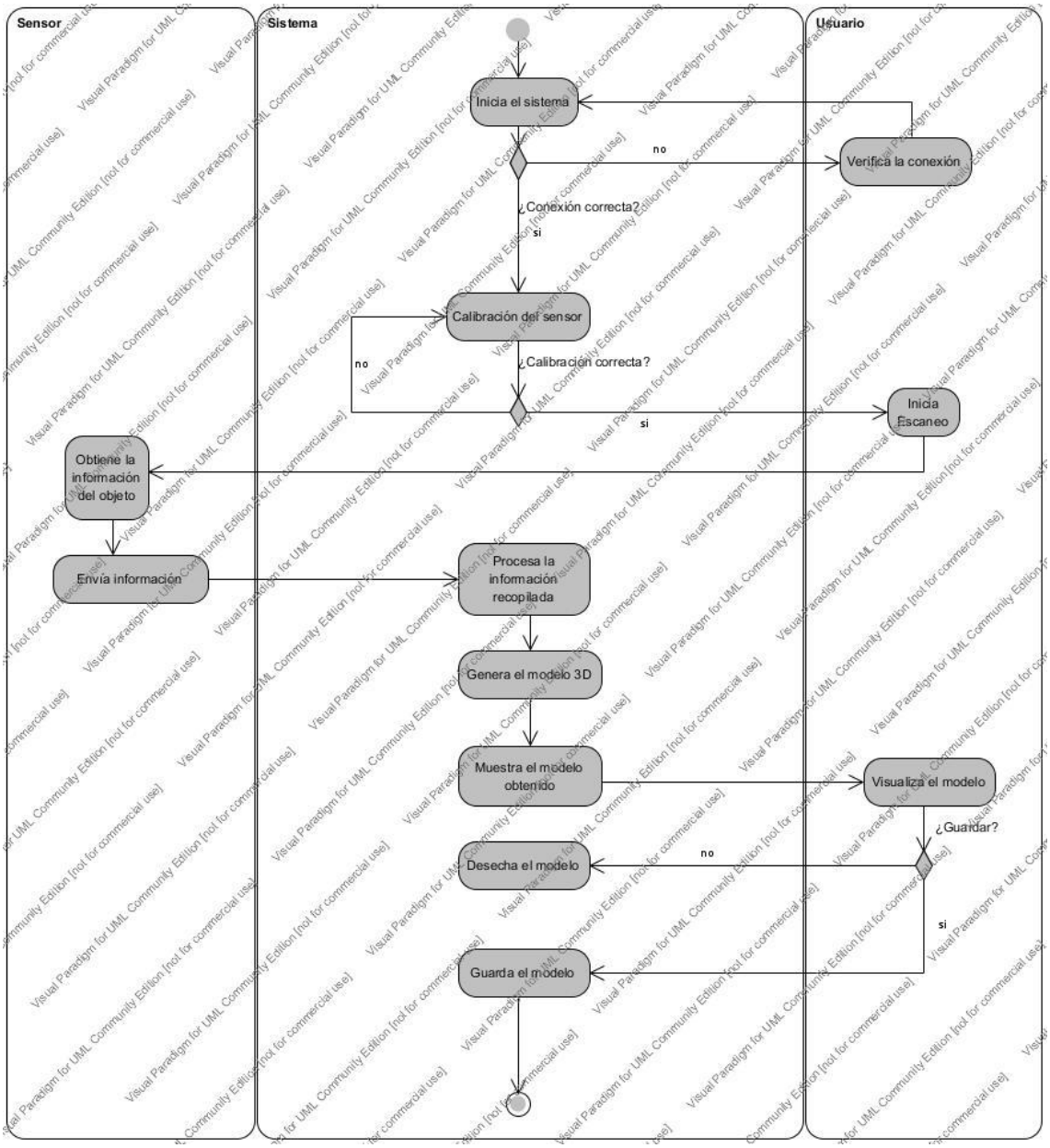


Figura 6.4 Diagrama de actividades Escanear

Por otro lado, el diagrama de actividades que se muestra en la Figura 6.5, explica el proceso que se tendría que llevar a cabo para abrir un archivo de un escaneo previo. En este diagrama se muestra que una vez iniciado el sistema, el usuario podrá abrir un archivo, donde le pedirán que elija el archivo de interés, una vez elegido el archivo, el sistema validará si es el tipo de archivo correcto, de ser así procederá a mostrar el modelo para la visualización por el usuario.

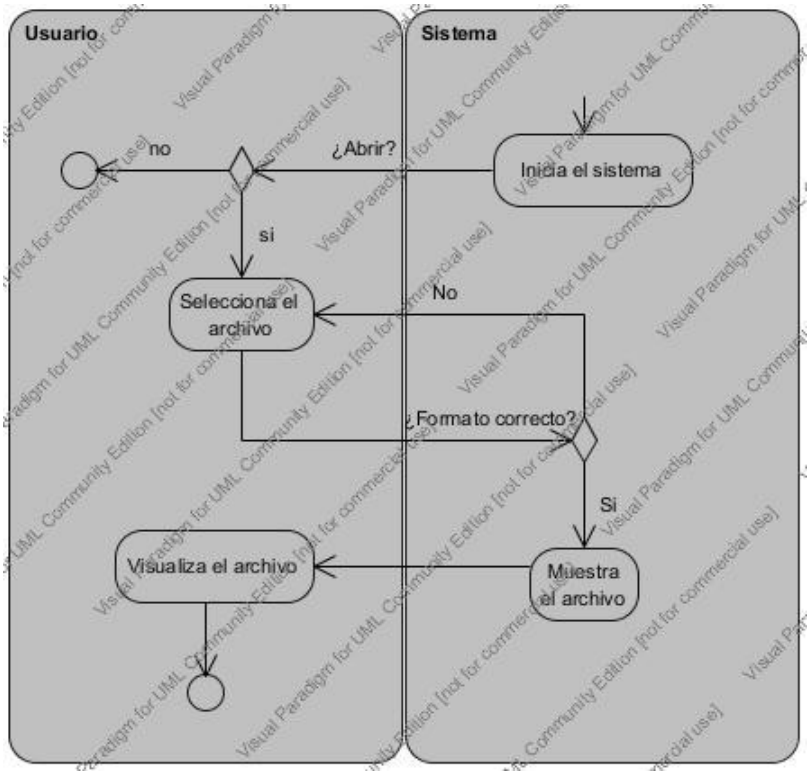


Figura 6.5 Diagrama de actividades abrir

6.3 Diagrama de clases.

En la Figura 6.6, se muestra un diagrama de clases, el cual explica como interaccionan las clases y los paquetes entre sí. En esta figura, se explica que el paquete, que contiene las clases Conexión Sensor, Escanear y la reconstrucción, llamado lógica de negocios, hará uso del paquete SDK (entorno de desarrollo propio de kinect). La GUI, usara el paquete de lógica de negocios, donde GUI contiene clases como Escanear y configuración.

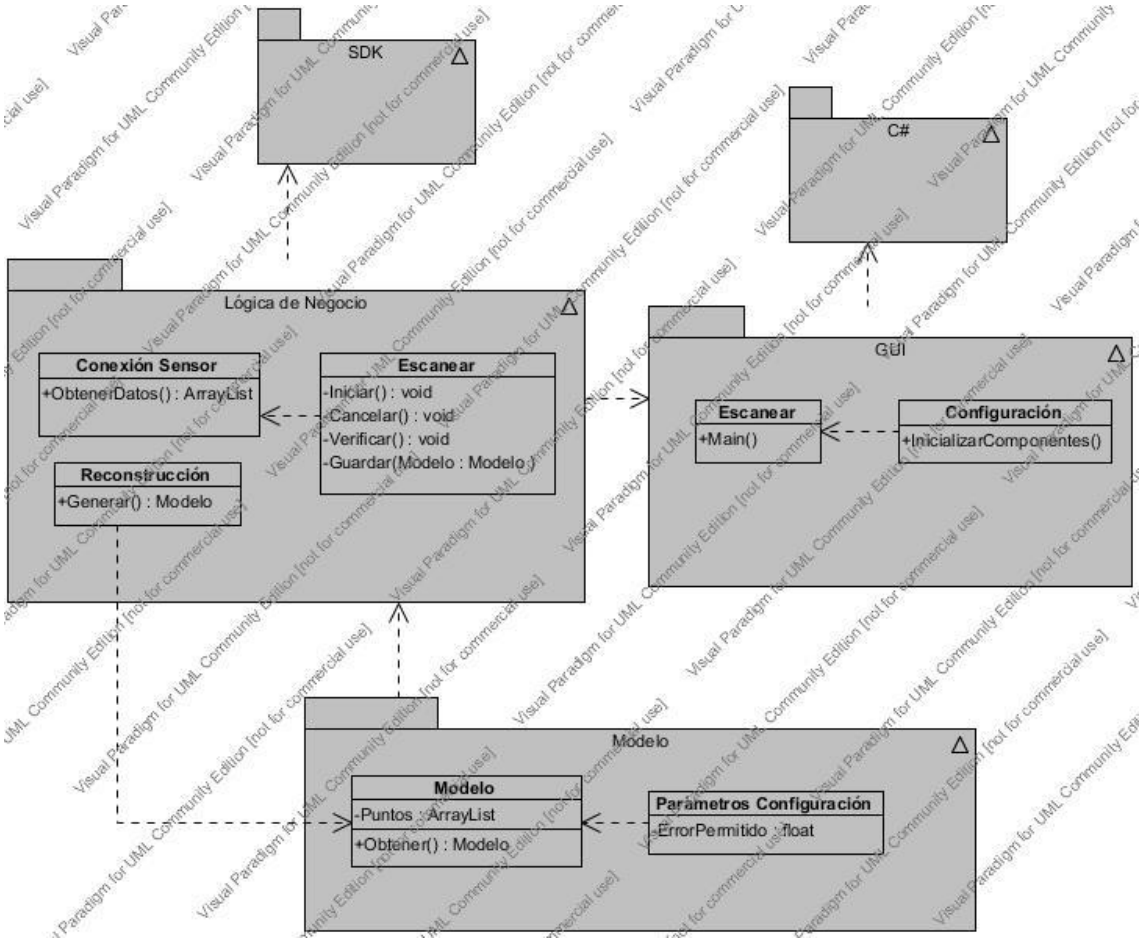


Figura 6.6 Diagrama de clases

6.4 Diagrama de secuencias.

6.4.1 Escanear.

La interacción entre clases, para la tarea de Escanear, esta mostrada en la Figura6.7, que se presenta a continuación. La cual nos explica que el usuario seleccionará comenzar en la interfaz grafica de Usuario, donde dará inicio el sistema para escanear, el cual validará si realmente se desea comenzar, donde después de aceptar, el sensor comenzara a obtener los puntos, que serán procesados y la clase Reconstrucción generará el modelo, que se regresará a la GUI donde el usuario podrá visualizarlo.

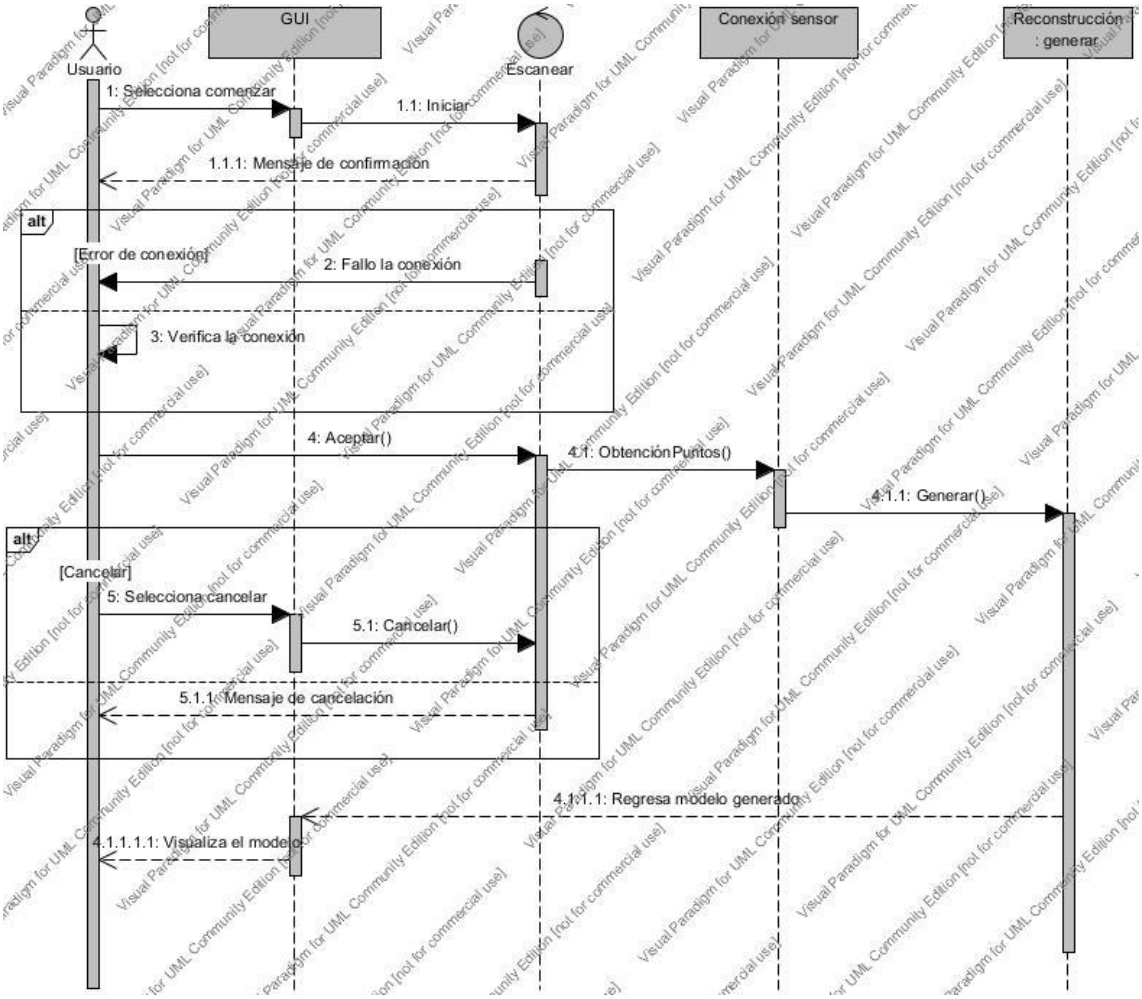


Figura 6.7 Diagrama de secuencias Escanear

6.4.2 Abrir

Así también, la interacción entre clases, para la tarea de Abrir, esta mostrada en la Figura 6.8, donde se muestra que el usuario seleccionará la acción de abrir en la GUI, la que desplegara a éste un panel para seleccionar la ubicación del archivo. Una vez seleccionada la ruta y el archivo, se abrirá mediante la clase de escanear, donde se obtendrá el modelo por medio del objeto Modelo, que lo devolverá a la GUI, donde lo visualizará el usuario.

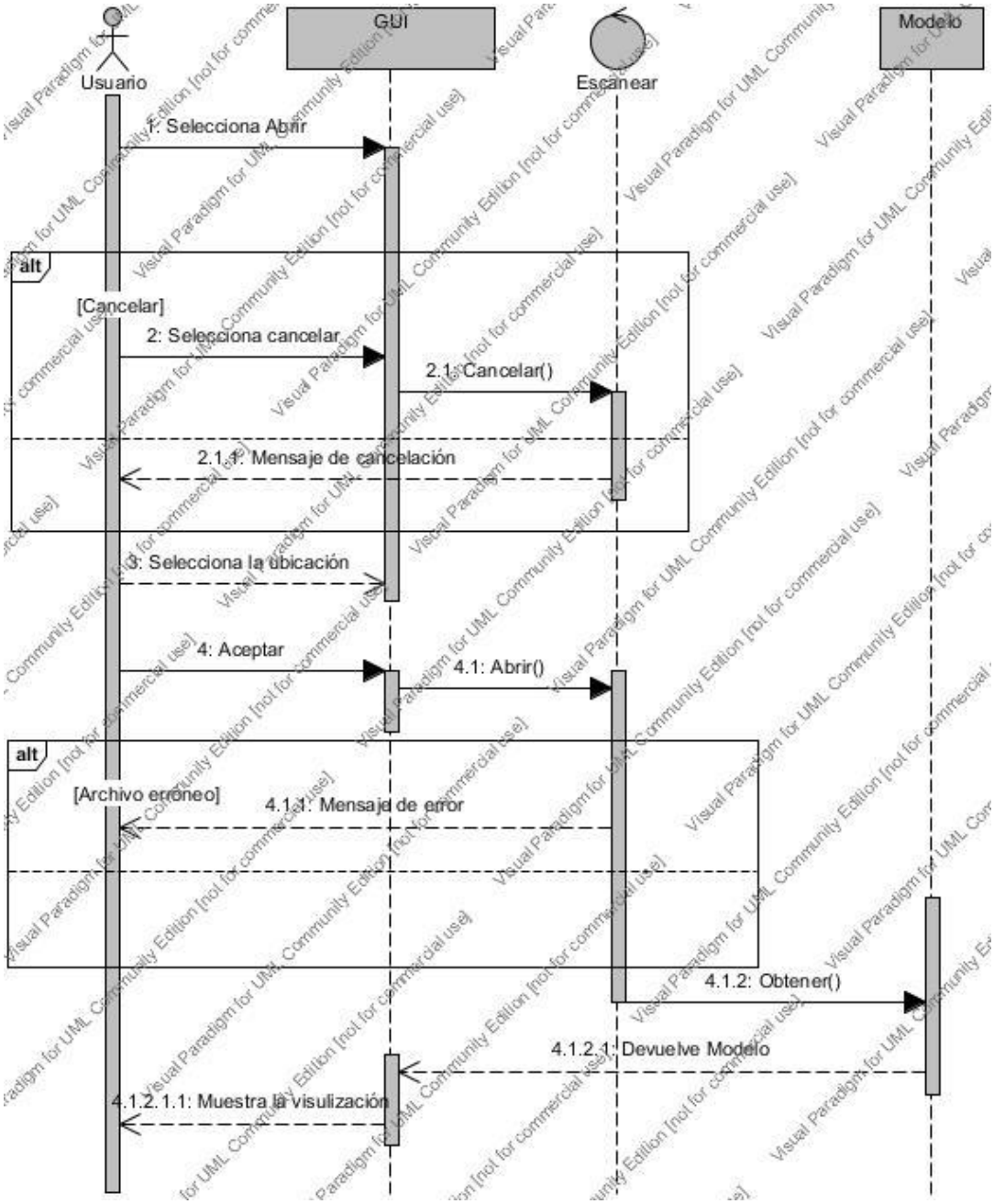


Figura 6.8 Diagrama de secuencias Abrir

6.4.3 Guardar

Por último en la Figura 6.9, nos explica que para guardar un modelo, se hará uso de la GUI y de la clase escanea, donde el usuario pedirá guardar el archivo y seleccionara la ruta en cuestión, una vez aceptada la ruta se guardará y se mostrará una confirmación del proceso.

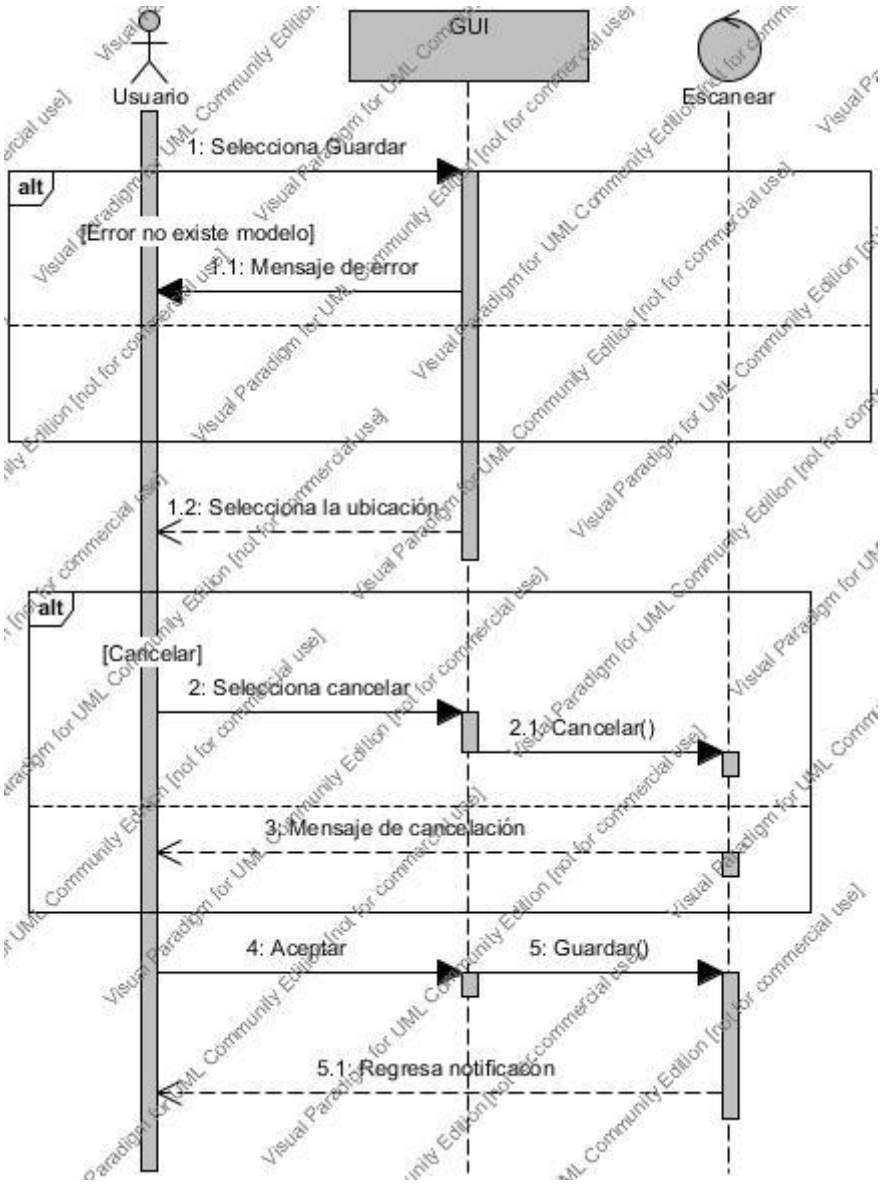


Figura 6.9 Diagrama de secuencias Guardar

6.5 Descripción del prototipo.

En base a los análisis que se ha hecho sobre el proyecto y dado el diseño que se ha llevado a cabo del mismo, podemos decir que para la construcción del prototipo, necesitaremos de los siguientes materiales:

- 2 Discos giratorios que, servirán para que la plataforma gire libremente.
- 2 cajas de metal que serán el soporte de la plataforma de giro y del sensor Kinect.
- Una hoja de acrílico, con la que haremos gran parte del dispositivo físico.
- Un potenciómetro.
- Un servo motor “Parallax Estándar Servo”, para el movimiento de la plataforma giratoria.
- Soporte para el servo.
- Engranes.
- Protectores metálicos.

Con la finalidad de especificar qué clase de objetos serán soportados por elarmazón haremos algunas aclaraciones, en base a algunos materiales previamente mencionados. El peso soportado dependerá directamente de la capacidad de arrastre del servo motor, mientras que la altura máxima del objeto de estudio lo determinara el sensor kinect.

En base a las características del servo motor que se está empleando solo podremos soportar objetos de peso menor a 2.7 kg.

Ahora bien, para determinar la altura máxima del objeto de estudio, empleamos la “ley de senos”, tomando como referencia los ángulos de visión del sensor y proponiendo alejar el Kinect del objeto de interés 2 m, con esto obtenemos como resultado una altura no mayor a 1.8m.

Entonces, se considerará objeto factible de modelado aquel que cumpla con:

- Un peso no mayor a 2.7kg.
- Una altura no mayor a 1.80 m
- Se recomienda evitar objetos metálicos, debido al peso.

7 Diseño (Final)

7.1 Arquitectura física.

7.1.1 Especificación de la base de giro.

7.1.1.1 Dimensiones y geometría

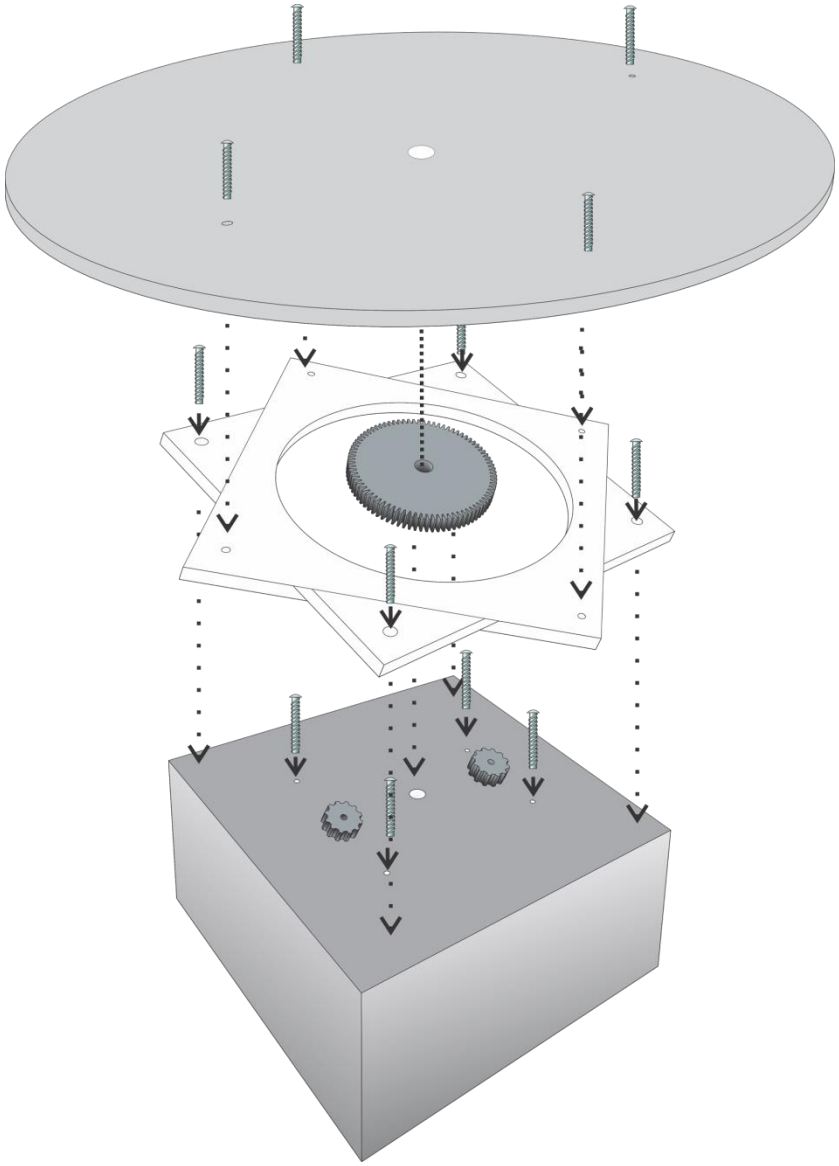


Figura 7.1 Diagrama General de la base.

En la imagen (Figura 7.1) se muestra la estructura de una carcasa de metal de 20 cm. por 20 cm. donde van a estar concentrados todos los componentes. En el centro, de este armazón metálico, se hizo una perforación de 0.9cm de diámetro aproximadamente; la perforación central servirá para que pueda entrar parte del engrane de mayor diámetro, para que se ajuste con el sistema de engranes que se explicará posteriormente.

Así también, a una distancia de 5 cm. a cada lado, se presentan dos perforaciones: una de 1cm. de diámetro (aprox.), y otra de 0.3 cm. (aprox.) al otro lado, estas perforaciones serán para el potenciómetro y el servomotor (este último tendrá la perforación de mayor tamaño de los costado). Cabe decir que se cuentan con la presencia de algunas perforaciones extra que nos ayudaran a mantener cada uno de los componentes en su lugar. En la figura siguiente se aprecia mejor lo que se ha descrito, mostrando la posición de cada una de las perforaciones y reiterando las separaciones de estas, sin mostrar las perforaciones adicionales que se le harán para la localización de los demás componentes.

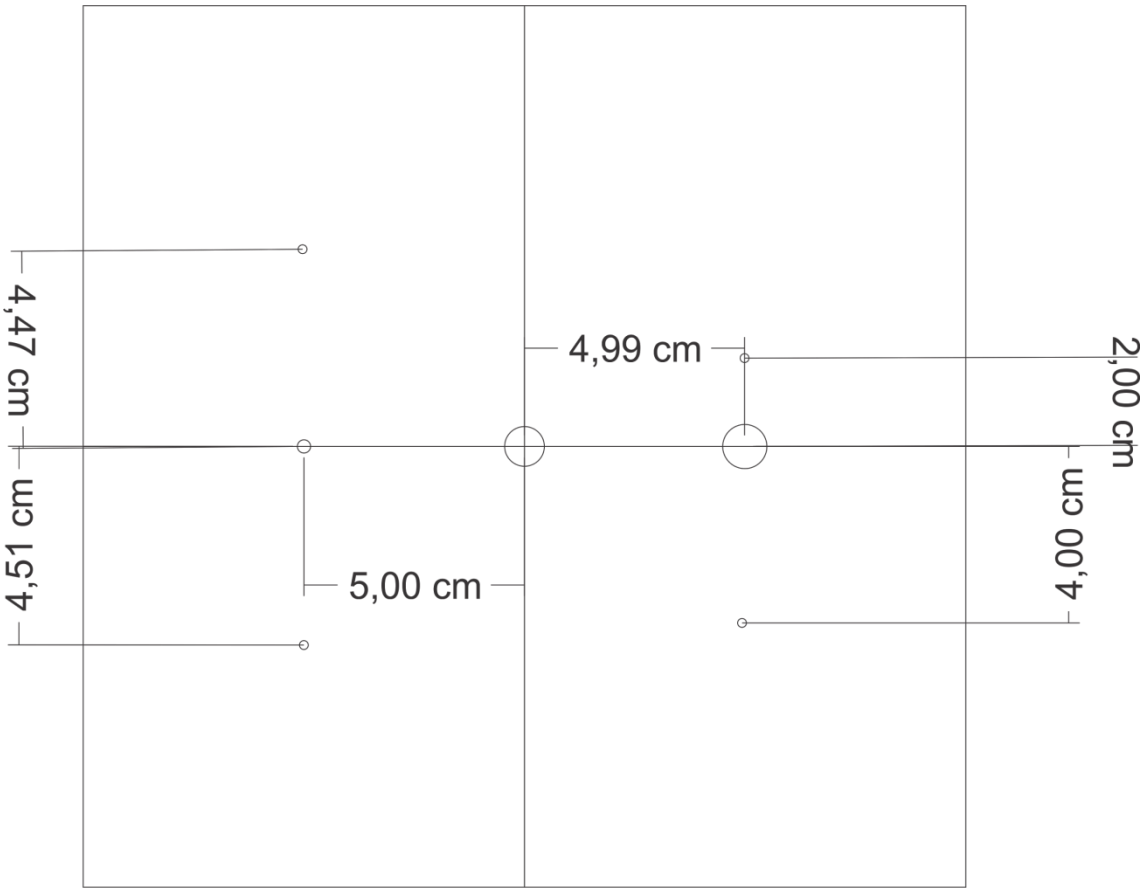


Figura 7.2 Vista superior del armazón metálico

Una vez explicada la base metálica, es preciso explicar los componentes que van a reposar sobre esta. En la figura 7.2, se aprecia al lado derecho una perforación a 5 cm de la central, es en esta donde saldrá la flecha del servomotor, que es la que hará girar el sistema.

Cabe destacar, que el dispositivo final que se utiliza para hacer girar el sistema es un servomotor de marca “Futaba” modelo “s3003”, que a pesar de ser un servomotor estándar de media vuelta, se ajustó a nuestras necesidades al modificar su funcionamiento, eliminando los “topes” que coinciden con el

potenciómetro interno que controla el giro, de esta manera el servomotor será capaz de girar 360 grados.

En la figura siguiente, (figura 7.3) se muestra el diagrama de las medidas y algunas vistas del servomotor previamente mencionado, con el fin de generar una imagen mental de las dimensiones de todos los componentes y que se vislumbre el tamaño del prototipo final. Es preciso decir que, la imagen (que fue recuperada de la documentación oficial), presenta las medidas en milímetros.

Figura 7.3 Servomotor

En cuanto al sistema de engranes se refiere, se requiere tener un sistema de engranes que tuviese una relación en cuanto a sus dientes de 1 a 7, esto nos ayudará a conseguir la velocidad necesaria y la trasmisión del movimiento justa para que el objeto de interés sea capaz de dar una vuelta completa a la velocidad de procesamiento de la imagen. A continuación se muestra un diagrama del sistema de engranes que se propone para la implementación (Figura 7.4).

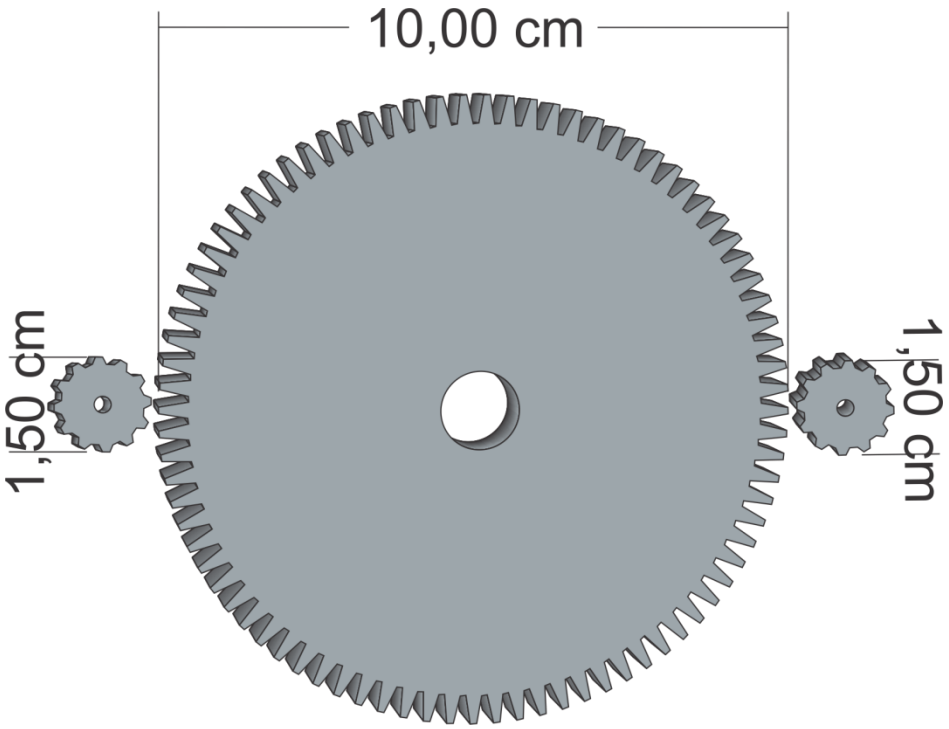


Figura 7.4 Sistema de engranes de transmisión de movimiento.

7.1.1.2 Construcción

La construcción fue una parte que llevo, como es de esperarse, más tiempo que la de conseguir todos los componentes del sistema. La lista de los componentes, con algunas especificaciones y recomendaciones, se presenta en la Tabla 7.1. Por otro lado, en la Figura 7.5, se ven cada uno de estos, de manera más tangible. De izquierda a derecha y de arriba a abajo tenemos, la base giratoria, las bases de metal, para el mecanismo y para el kinect respectivamente, tres engranes[3] para el mecanismo, un potenciómetro adaptado para ser añadido a la base de metal, el servomotor, plato de madera, tornillos y tuercas y algunos cables.

[3] Se tuvieron que modificar para su ajuste a nuestros intereses. Después se explican dichas modificaciones

Figura 7.5 Componentes de base giratoria

.

Tabla 7.1 Descripción y costo de los componentes

Componente	Descripción	Costo
Engranes.	Un juego de engranes que podemos encontrar en cualquier tienda de robótica. Marca VEX	$282.00
Discos Giratorios	Ayudarán a girar el objeto de estudio. Se encuentran en cualquier ferretería.	$85.00
Círculo de plástico/madera/acrilico	Es aquí donde reposará el objeto de estudio, se recomienda que sea de 39 o 30 cm de diametro.	$50.00
Potenciómetro multivuelta	Se recomienda uno de 10k a 10 vueltas.	$261.00
Servomotor	Parallax (Futaba S3003)	$222.19
Armazón metálico	Se recomienda sea de metal.	$300.00

El potenciómetro nos va a ayudar a controlar el servomotor, debido a las modificaciones internas que se le hicieron a este, donde se eliminaron los bordes internos de la flecha del servomotor, por lo que se pierde el control sobre la posición exacta del giro (ya que lo bordes hacían girar el potenciómetro interno del mismo servo, lo cual le indicaba a su sistema de control cuando parar o en qué posición se encontraba), ahora lo único que haremos será reemplazar el control interno por uno externo utilizando el potenciómetro multivuelta que nos va a permitir obtener girar los 360 grados.

El primer paso a llevar a cabo fue la de la perforación de la base metálica que contendría todos los componentes, en base al diagrama que se propuso en el apartado de diseño, poniendo especial cuidado en no exceder o limitar demasiado las medidas propuestas, ya que esto podría resultar en un mal funcionamiento (o nulo) del mecanismo. Tal como se muestra en la figura 7.6, donde incluso se puede ver el servomotor colocado el lugar designado para este, a la espera de que le sea colocado el engrane en la flecha, y que se empiece a añadir todos los demás componentes.

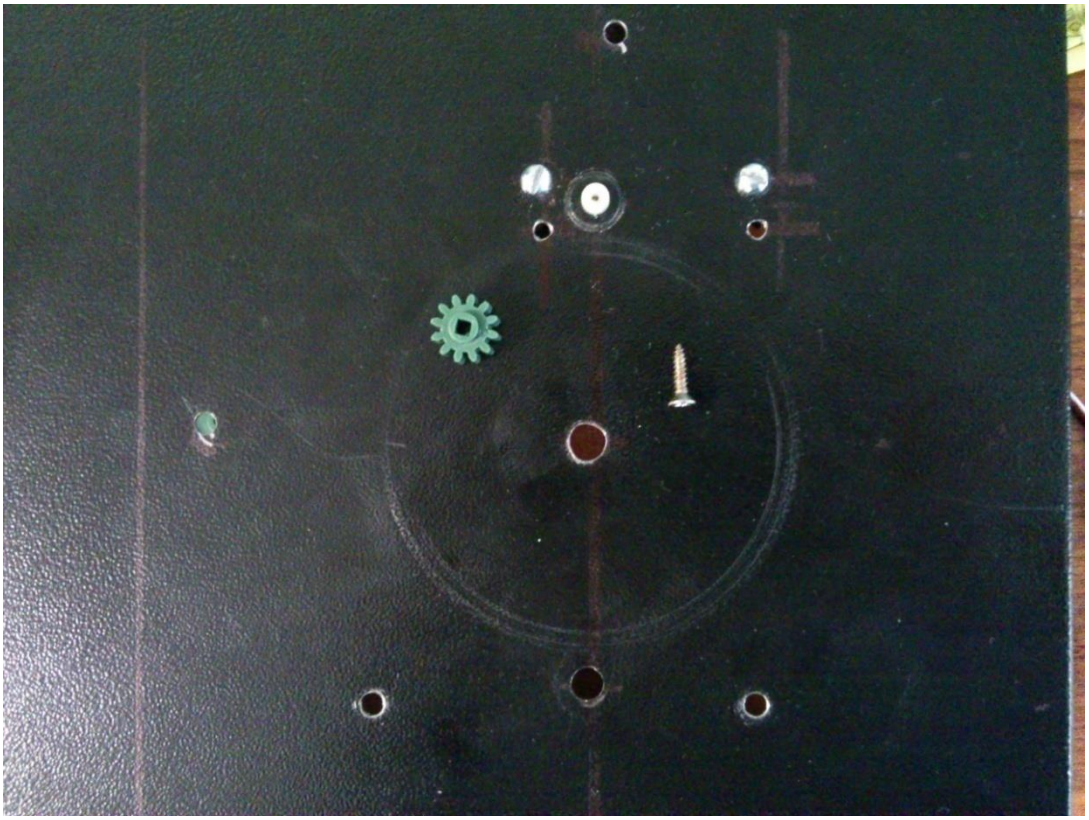


Figura 7.6 Base perforada y servomotor en posición.

En el otro extremo de la perforación central, tal como se había diseñado, es donde ajustará el potenciómetro, con el anexo metálico que le ayudará a añadirse a la base metálica.

Después del acople de los componentes al armazón, se procedió a ajustar los engranes, para que se pudiera adaptar de mejor manera a los componentes. Dichas modificaciones en los engranes, básicamente consistieron en el desgaste de la pieza desde el centro, ampliando la cavidad central de estos.

Una vez modificados los engranes, y acoplados al servomotor y al potenciómetro (los cuales ya se encontraban colocados en la base metálica) se obtiene una estructura como la que se muestra en la Figura 7.7, donde se aprecia cómo es que el sistema se va conformando.

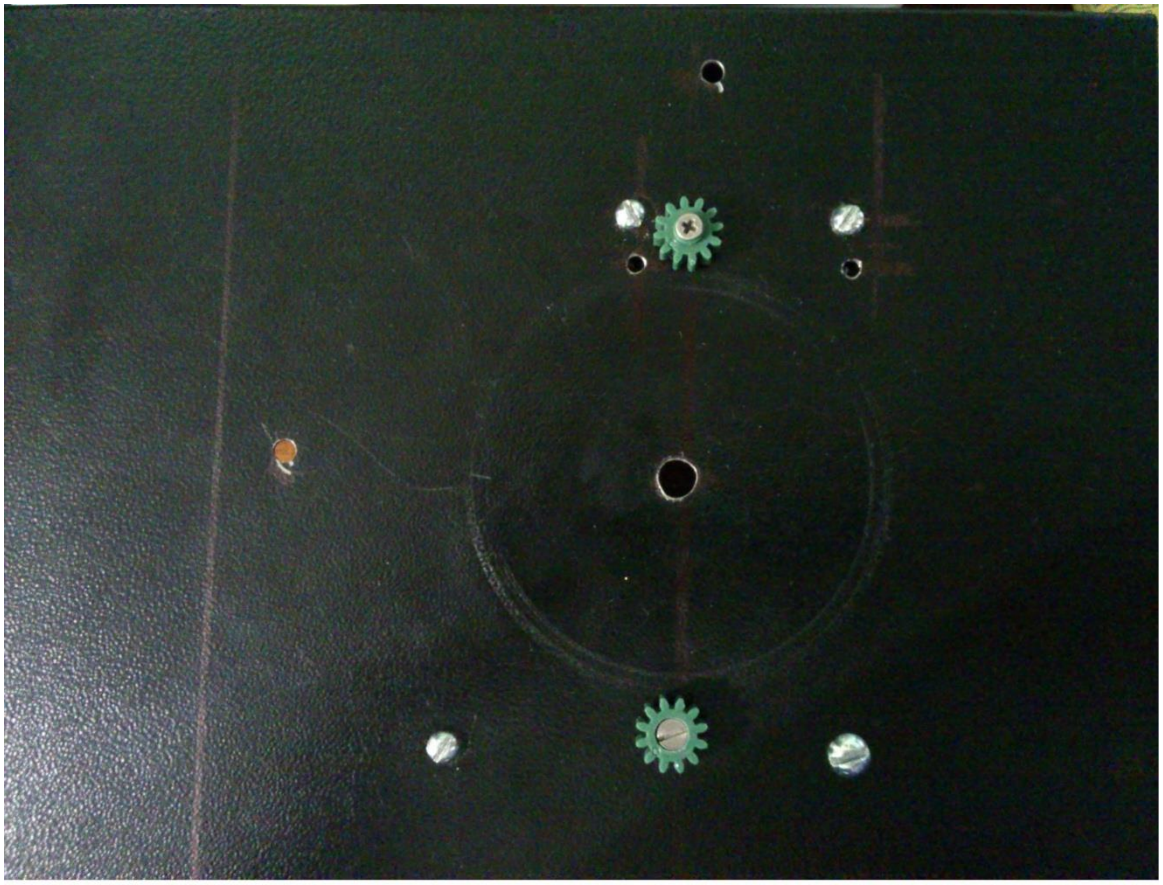


Figura 7.7 Servomotor, potenciómetro y engranes.

Después de estos puntos, se procede al acoplamiento de la base giratoria, la cual ayudará a soportar el peso y de facilitar el giro del objeto que se desea estudiar. En la Figura 7.8, se puede apreciar la conformación previamente mencionada.

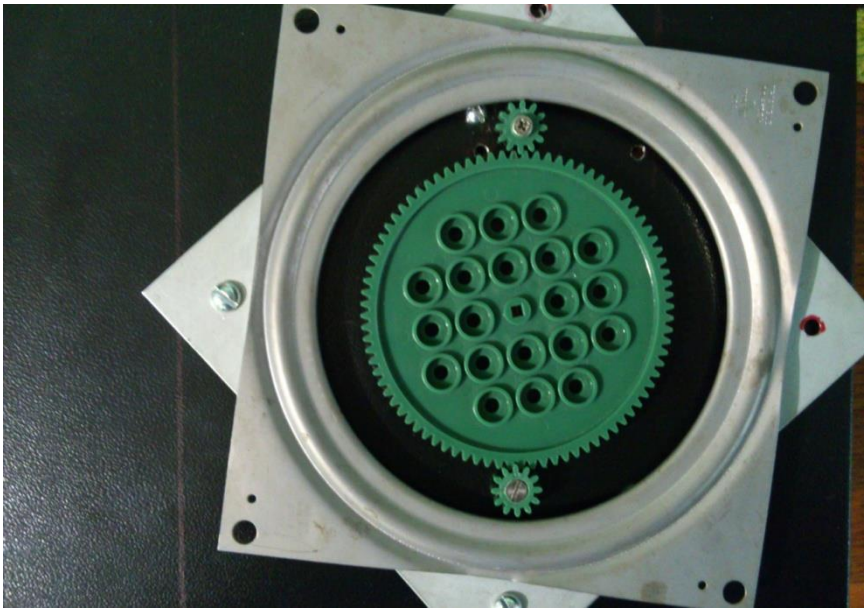


Figura 7.8 Servomotor, potenciómetro, engranes y base giratoria.

Para finalizar la construcción de nuestro dispositivo de giro, se ajustó el plato de madera a la base giratoria y a su vez, el engrane central (es decir, el de mayor tamaño) también se unió al plato de madera. Tal como se mostró en el diseño.

7.1.2 Especificaciones de la tarjeta arduino.

7.1.2.1 Programa y otras especificaciones.

Para el control y la comunicación entre el mecanismo de giro y la computadora, se ha ocupado la tarjeta arduino uno, la cual es una tarjeta basada en un micro controlador ATmega328, el cual tiene 14 pines de entrada y salida donde 6 de estas son entradas analógicas. Las bondades de esta tarjeta radican en que contiene todo lo que se necesita para soportar un funcionamiento íntegro del micro controlador.

Para los propósitos de este trabajo, se ocuparon básicamente una entrada analógica, una salida digital, los puertos de “tierra” y el puerto de salida de 5 volts; el servomotor se conectó a la entrada analógica para que reportara siempre el estado de giro en que se encontraba el servomotor, mientras que el servomotor fue conectado en el pin 6, que se había configurado como salida digital, para poderlo hacer girar ya fuese hacia un lado o hacia otro, según se le dijera con el potenciómetro. Así mismo, se configuró para esperar siempre una señal de inicio y el ángulo que tenía que cubrir según se le ordenara mediante la computadora. En apartados posteriores se propone explicar con mayor claridad el programa que se ha implementado para los propósitos de este trabajo.

7.1.3 Especificación de la comunicación (Base de giro – arduino PC)

A continuación, la figura 7.9 muestra, mediante un diagrama de componentes, el proceso de comunicación entre la base de giro el arduino y la computadora. Esta imagen intenta simplificar este proceso, proceso que se ha descrito de manera muy somera en los pasos previos.

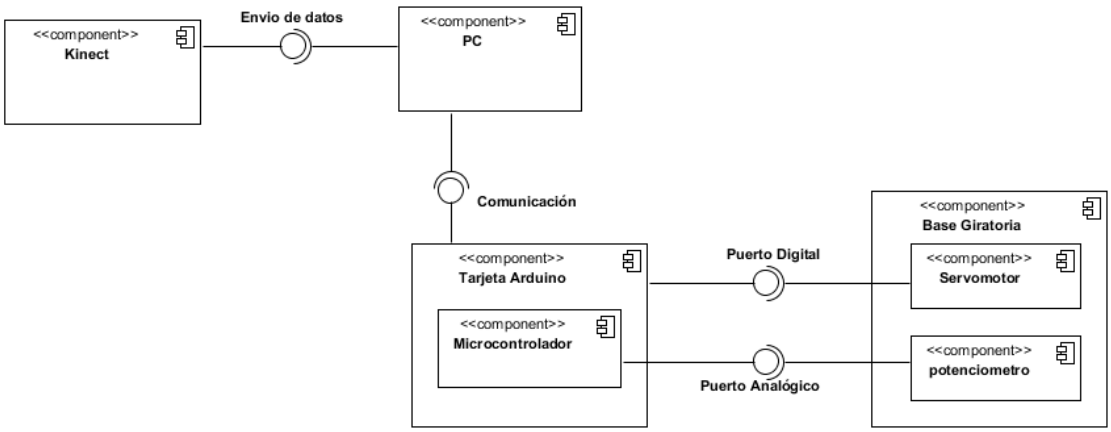


Figura 7.9 Diagrama de componentes de la comunicación.

Cabe mencionar que para llevar a cabo esta parte, se tuvo que implementar un programa (o llamado también sketch) sobre la plataforma arduino para controlar el sistema mecánico. Se mostrarán algunas figuras sobre los datos más importantes de este programa.

En la figura 7.10, podemos apreciar la función principal (y la más importante) del programa de control, su funcionamiento básicamente radica en que cada 20 milisegundos revisa si se encuentra en el rango de movimiento, de no ser así lo ajusta y espera la señal que se le envía desde la máquina para mover el servomotor, tantos grados como se desee, a la velocidad que se haya especificado (en los pulsos de adelante y atrás).

```
void loop() {
  unsigned long currentMillis = millis();
  if(currentMillis - milisAnteriores > intervalo) {
    milisAnteriores = currentMillis;
    anguloPot = analogRead(0); // Esta funcion, lee los datos que se leen de la entrada analogica (A0)
    Serial.println(anguloPot); // Imprime el valor que leyo del potenciometro
    if(anguloPot < posicionIni){ //verificar si esta en el intervalo de giro
      estado = 0;
      actualizar(servoPin,adelante);
      Serial.println("start");
    }
    if(anguloPot > posicionFin){
      estado = 0;
      actualizar(servoPin,atras);
      Serial.println("end");
    }
    leerSerial(); // Verifica si hay datos en el buffer
    if(estado==1){
      ir(anguloObjetivo);
    }
  }
}
```

Figura 7.10 Código principal

Por otro lado, segunda función en importancia en este código es la que se muestra en la figura 7.11, la cual estará siempre a la espera de la activación desde nuestro ordenador, el cual mandara una “S” y el “ángulo” en que se debe mover.

```
void leerSerial(){
  if (Serial.available() > 1) { // ¿Hay datos disponibles para leer?
    char trigger = Serial.read();
    if(trigger == 'S'){
      estado = 1;
      int newAngle = Serial.read();
      anguloObjetivo = (int)map(newAngle,0,255,posicionIni,posicionFin);
    }
  }
}
```

Figura 7.11 Función de lectura desde el ordenador.

Por último, la figura 7.12 muestra una tercera función de interés, la cual será la que mueva el servomotor, según el pulso (números declarados a principio del programa) que se le indique.

```
void actualizar (int pin, int pulse){
  digitalWrite(pin, HIGH);
  delayMicroseconds(pulse);
  digitalWrite(pin, LOW);
}
```

Figura 7.12 Función de actualización de la posición del servomotor

7.2 Arquitectura lógica

7.2.1 Diagrama de componentes

Mediante el siguiente diagrama, Figura 7.13, se describe la arquitectura lógica final que tiene la aplicación, los componentes se describen más adelante, detallando la funcionalidad de cada uno.

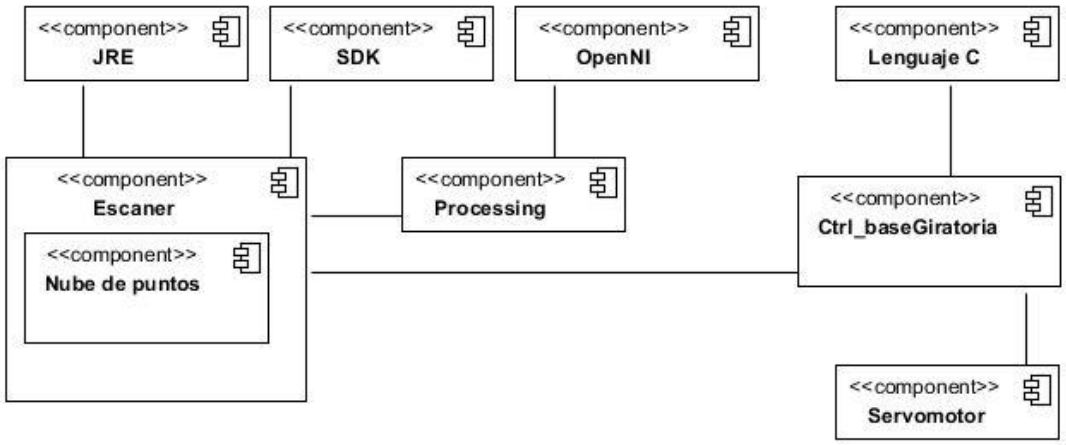


Figura 7.13 Diagrama de componentes

7.2.2 Adquisición de datos del escáner 3D.

Para la obtención de los datos del escáner se utilizaron las siguientes herramientas de software.

- Sensor kinect
- Kinect para windows SDK.
- Processing
- Librería SimpleOpenNi
- Java Development Kit

7.2.2.1 Kinect para windows SDK

Para poder utilizar el kinect en el equipo de cómputo es necesario instalar el paquete de desarrollo del sensor kinect para windows el cual nos permite tener acceso a la funcionalidad del dispositivo, en este caso nos proporciona los controladores necesarios para que nuestro equipo de computo pueda reconocer al sensor como un dispositivo del sistema, como se muestra en la Figura 7.14.

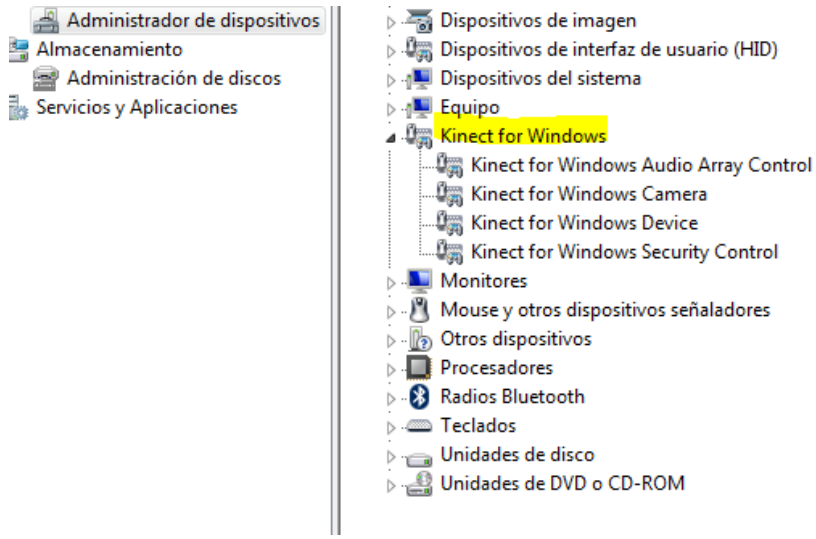


Figura 7.14 Configuraciones

En el caso de que no contemos con el paquete sdk de Microsoft para kinect instalado en nuestra computadora veremos que no se reconoce el dispositivo, tal como se muestra en la figura 7.15, dejando en claro que de esta manera no funcionará el sensor en nuestro equipo.

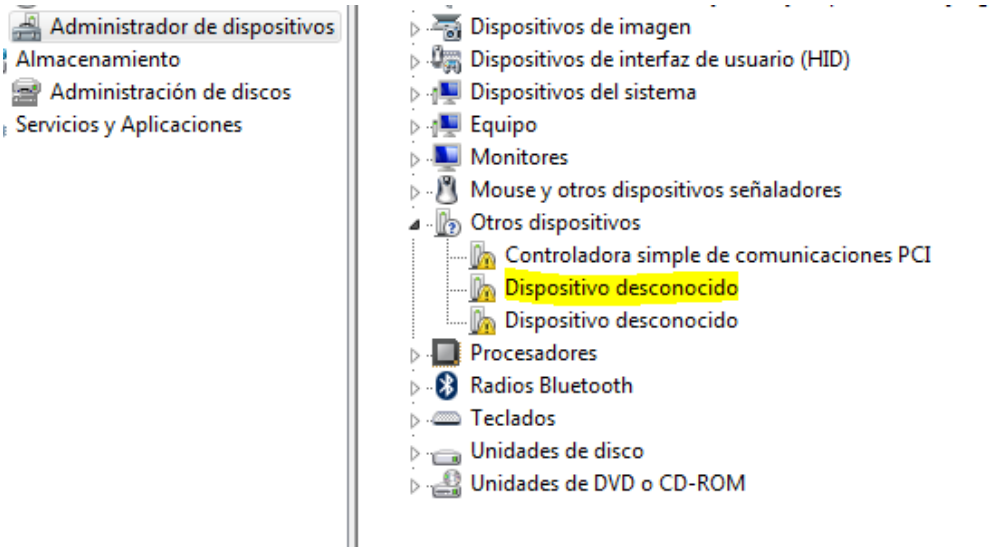


Figura 7.15 Configuraciones (b)

7.2.2.2 Processing

Es un lenguaje de programación y entorno de desarrollo de código abierto basado en Java, el cual gracias a la librería SimpleOpenNi nos va a dar acceso al manejo de la nube de puntos, algo que las librerías que se utilizan para desarrollar proyectos con C# no lo permiten.

Gracias a la librería SimpleOpenNi se implementó una función llamada “dibujarNubePuntos” (Figura 7.16) la cual nos va permitir ver la nube de puntos en tiempo real, el código se muestra a continuación.

```
public void dibujarNubePuntos(){
  int inicio;
  PVector puntosReales;
  for(int y=0;y<kinect.depthHeight();y+=pasos){
    for(int x=0;x<kinect.depthWidth();x+=pasos){
        inicio=x+y*kinect.depthWidth();
        puntosReales=kinect.depthMapRealWorld()[inicio];
        point(puntosReales.x,puntosReales.y,puntosReales.z);
    }
  }
}
```

Figura 7.16 FunciondibujaNube

La función “kinect.depthMapRealWorld” nos va a devolver un vector con tres coordenadas, donde estas coordenadas son la posición del punto que el kinect está obteniendo, una vez que tenemos las coordenadas, con la función point(), vamos a dibujar el punto en la pantalla para que el usuario tenga una vista de lo que el kinect está proyectando. En la figura 7.17 podemos observar una vista con algunos puntos dibujados, en este caso todos los puntos son blancos ya que no les asignamos color.

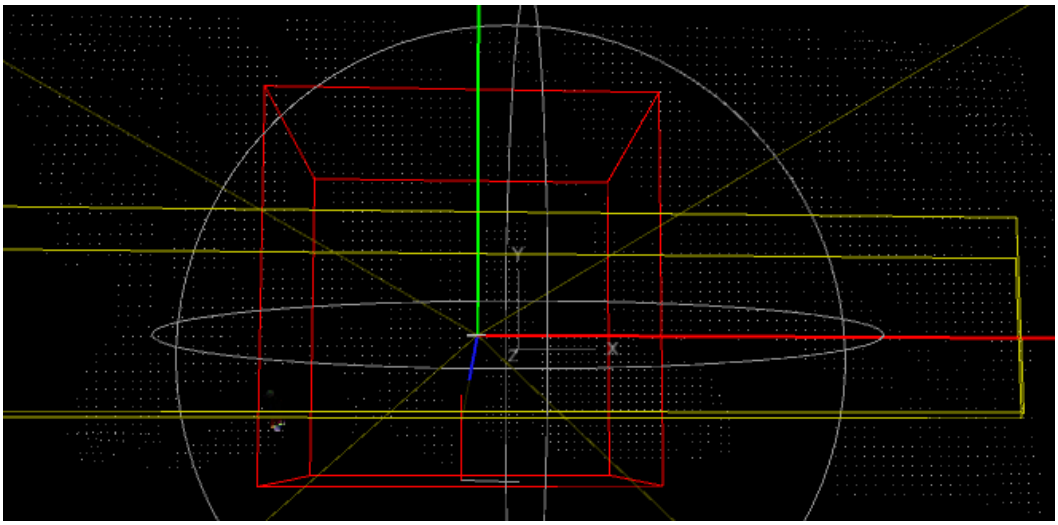


Figura 7.17 Vista desde kinect

En la Figura 7.18, podemos observar un cubo a caja que es la encargada de delimitar el espacio, todo lo que se encuentre fuera de esa caja no será

detectado por sensor. La caja la dibujamos con el siguiente fragmento de código.

```
public void dibujaCaja(){
  stroke(255,0,0);
  line(eje.x,eje.y,eje.z,eje.x,eje.y+100,eje.z);
  noFill();
  pushMatrix();//guardamos las coordenadas actuales
  translate(eje.x,eje.x+baseAltura+objetoAltura/2,eje.z);
  box(objetoAncho,objetoAltura,objetoAncho);
  popMatrix();//volvemos a la configuracion inicial
}
```

Figura 7.18 Método dibujaCaja

A continuación se muestra (Figura 7.19) la función encargada de aislar el objeto de estudio de acuerdo al tamaño de la caja y se encarga de obtener los colores de cada punto y los almacena para posteriormente asignarle el color al objeto.

```
public void actualizarObjeto(int escAncho,int paso){
  int inicio;
  PVector puntoReal;
  escaneoPuntos.clear();
  escaneoColores.clear();
  float angulo = map(Integer.valueOf(anguloBase),100,824,2*PI,0);
  pushMatrix();
  translate(eje.x,eje.y,eje.z);
  rotateY(angulo);
  line(0,0,100,0);
  popMatrix();
  int xMin =(int)(kinect.depthWidth()/2-escAncho/2);
  int xMax=(int)(kinect.depthWidth()/2+escAncho/2);
  for(int y=0;y < kinect.depthHeight();y += paso){
    for(int x=xMin; x<xMax; x+=paso){
      inicio = x+(y*kinect.depthWidth());
      puntoReal=kinect.depthMapRealWorld()[inicio];
      int puntoColor=kinect.rgbImage().pixels[inicio];
      //verificamos que los puntos se encuentren dentro del rango definido
      if(puntoReal.y<objetoAltura+baseAltura && puntoReal.y>baseAltura){//revisamos la altura(eje y)
        if(abs(puntoReal.x-eje.x)<objetoAncho/2){//revisamos el ancho(eje x)
          if(puntoReal.z < eje.z+objetoAncho/2 && puntoReal.z>eje.z-objetoAncho/2){//revisamos la profundida (eje z)
              PVector rPunto;
              puntoReal.z-=eje.z;
              puntoReal.x-=eje.x;
              rPunto= vecRotY(puntoReal,angulo); //
              escaneoPuntos.add(rPunto.get());
              escaneoColores.add(new PVector(red(puntoColor),green(puntoColor),blue(puntoColor)));
          }
        }
      }
    }
  }
}
```

Figura 7.19 Método actualizaObjeto

Ahora para asignarle un color a cada punto de la nube se recurre a la función “dibujaObjetos” la cual será la encargada de hacer que el objeto sea visible en pantalla.

A continuación se muestra (Figura 7.20) la función encargada de mostrar lo que se tiene del objeto durante el escaneo.

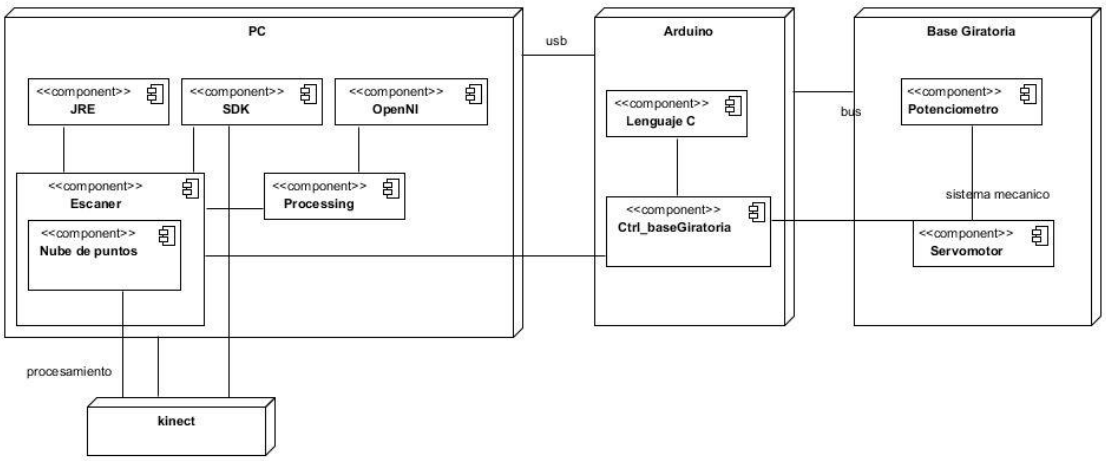
```
public void dibujaObjetos(){
  pushStyle();
  strokeWeight(2);
  for(int i =1; i<puntos.size();i++){
    stroke(colores.get(i).x,colores.get(i).y,colores.get(i).z);
    point(puntos.get(i).x,puntos.get(i).y,puntos.get(i).z+eje.z);
  }
  for(int i=1;i<escaneoPuntos.size();i++){
    stroke(escaneoColores.get(i).x,escaneoColores.get(i).y,escaneoColores.get(i).z);
    point(escaneoPuntos.get(i).x,escaneoPuntos.get(i).y,escaneoPuntos.get(i).z+eje.z);
  }
 }
```

Figura 7.20 Método dibujaObjetos

Más adelante en las pruebas unitarias se mostrará lo que se obtiene con estos métodos.

7.3 Diagrama de implementación

Una vez definidas las arquitecturas (física y lógica), se implemento el sistema de la siguiente manera, el cual se muestra en el diagrama de despliegue que se presenta a continuación (figura 7.21), donde se observan los componentes que están dentro de cada uno de los módulos realizados.



8 Implementación

Una vez terminados los módulos que se describieron anteriormente, procedemos a obtener el diseño final de la aplicación en cual se muestra en la Figura 8.1

Figura 8.1 Ventana principal

Donde el botón “Comenzar” desplegará una nueva ventana(para la obtención de los puntos), en la cual podremos ver la información que está obteniendo el kinect y recopilar los datos que el kinect envía al equipo de cómputo para su posterior tratamiento y visualización.

Por otra parte el botón “Probar” nos va a indicar si el kinect se encuentra conectado a nuestro equipo de cómputo, si el kinect se encuentra conectado y funcionando la aplicación nos mostrará algo similar a la Figura 8.2

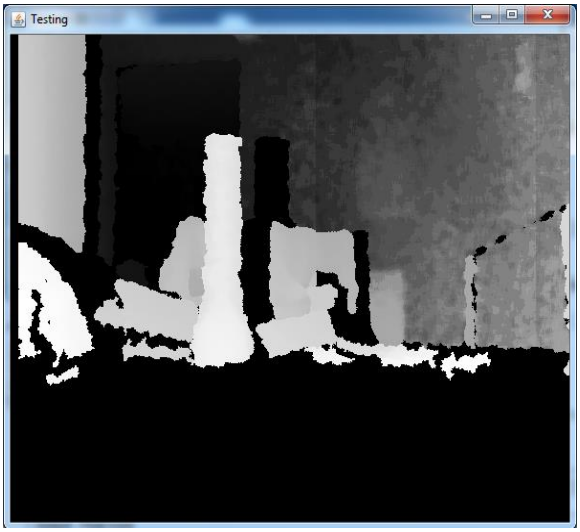


Figura 8.2 Ventana de prueba

En caso de que el sensor no se encuentre bien conectado o tenga algún error veremos la siguiente pantalla (Figura 8.3).

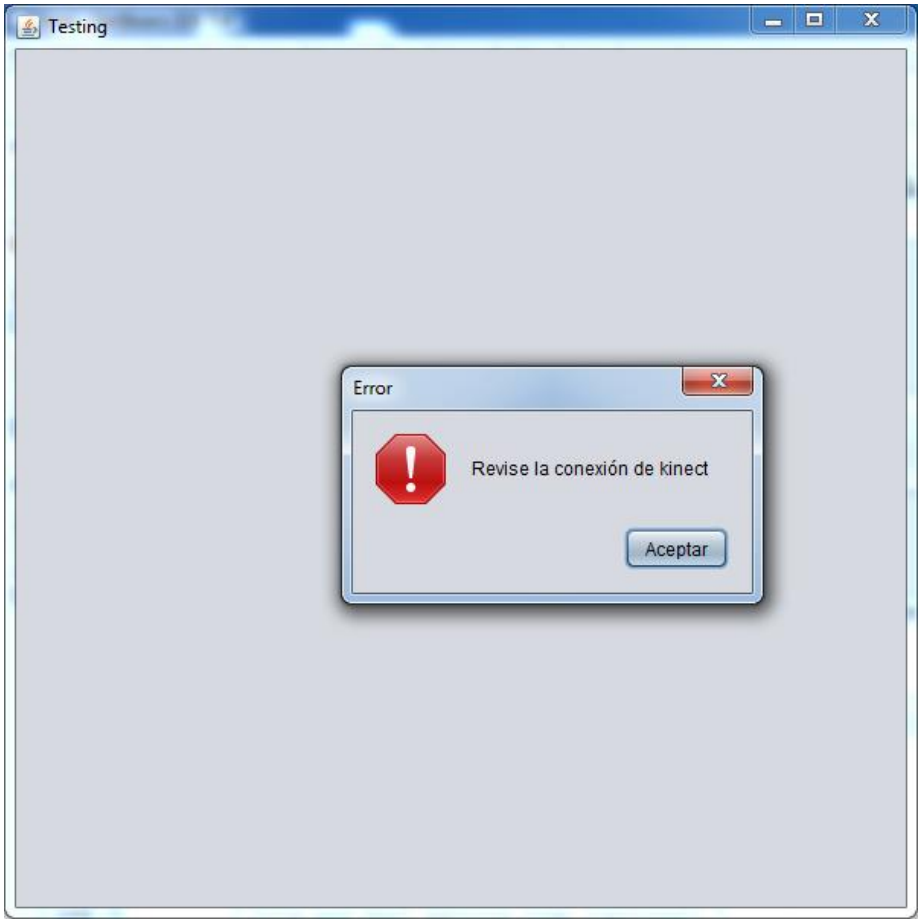


Figura 8.3 Pantalla de error

El botón “Generar Modelo” será el encargado de procesar la información que nos otorgó el sensor kinect, generando el archivo final del modelo 3D.

9 Pruebas

9.1 Descripción de las pruebas unitarias

- Componente base giratoria
- Entradas
- Salidas
- Componente Arduino
- Entradas
- Salidas

9.1.1 Componente Obtención de puntos

Entrada: el objeto de estudio que se encontrara sobre la base(en este caso será una lámpara), como la de la figura 9.1 que se muestra a continuación.

Figura 9.1 Objeto de prueba

Salidas:
La pantalla para la obtención de los puntos será la que se muestra (Figura 9.2).

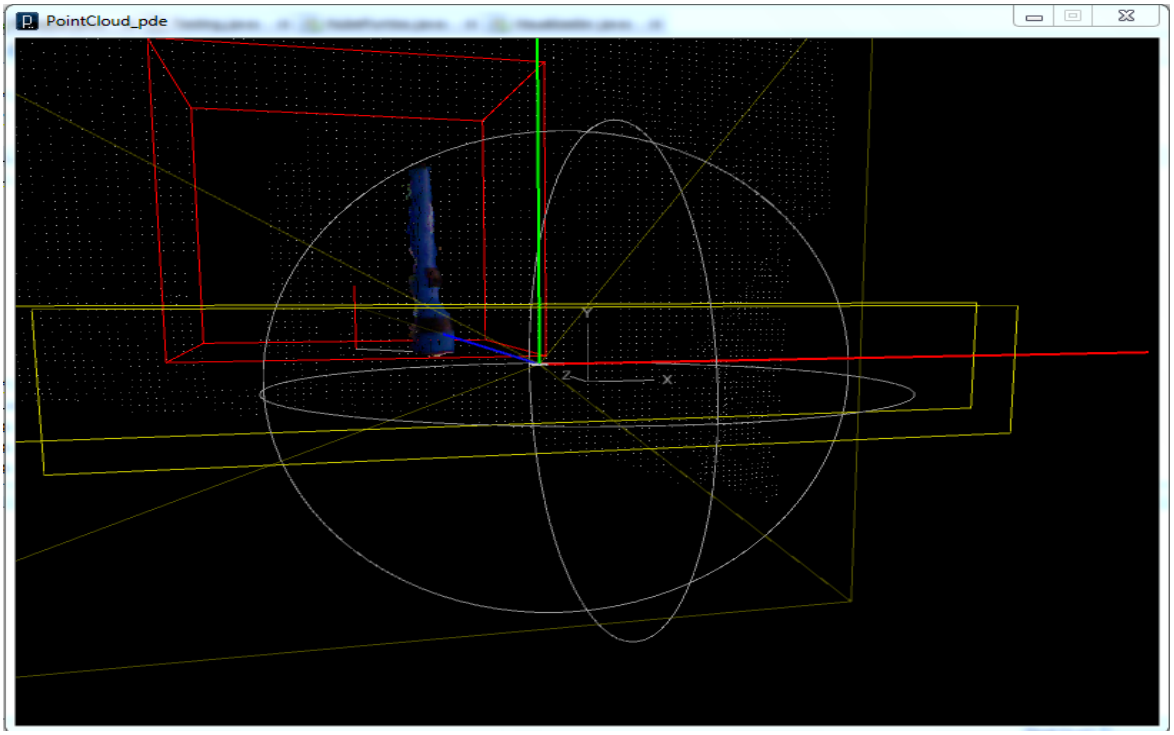


Figura 9.2 Salida de prueba

Y obtendremos como salida un archivo en formato “ply”, que se puede abrir desde cualquier aplicación que maneje formatos 3d.

9.2 Pruebas de Sistema

Las pruebas de sistema se iniciaron respetando las siguientes reglas:

- El objeto a escanear debería de pesar menos de 2 kg aprox.
- El objeto que se someterá al escaneo puede medir hasta 1.2 mts de altura, en este caso se probó con objetos que medían no más de 45 cm de altura (esto debido al poder de computo que se tiene).

Así también, una vez que se empezaron a hacer las pruebas, se encontró que para asegurar el correcto funcionamiento, se tenía que:

- Escanear objetos de peso menor a 1.5 kg
- La distancia promedio entre el objeto a escanear y el sensor kinect tiene que ser de 1.50 mts.

Una vez ajustados estos parámetros, se procedió a conjuntarlos para obtener el prototipo propuesto. En la Figura 9.3, se aprecia la implementación de la arquitectura física final del sistema.

Figura 9.3Pruebas de sistema

Dentro de la figura anterior, se puede apreciar los componentes acoplados para proceder con las pruebas finales.

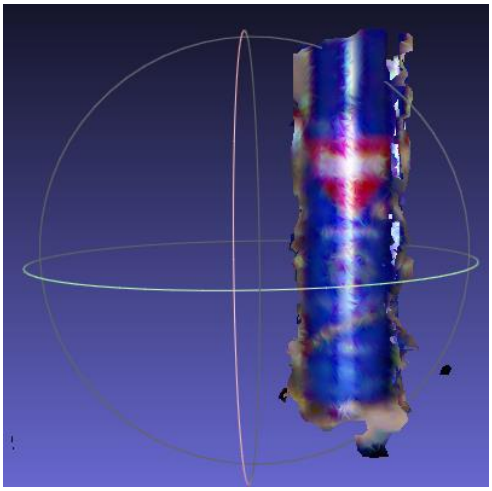
Prueba 1(Envase cilíndrico)

Características aproximadas

Altura	30 cm
Peso	250 gramos

ENTRADA

SALIDA

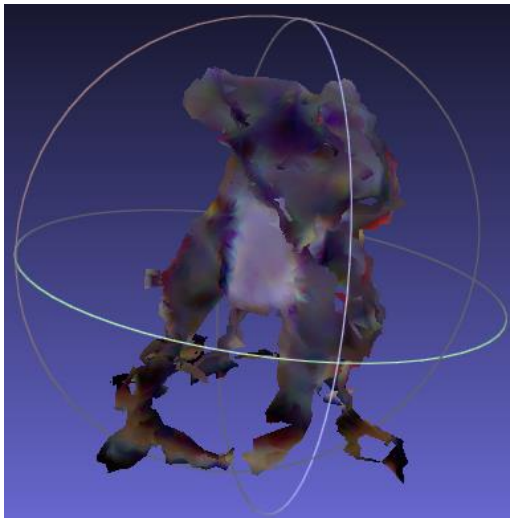


Prueba 2(Muñeco)

Características aproximadas

Altura	15 cm
Peso	200 gramos

Entrada **Salida**

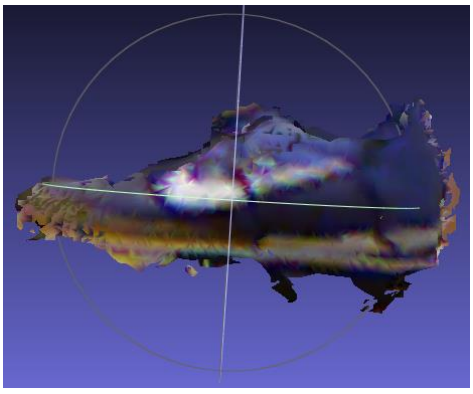


Prueba 3(Tenis)

Características aproximadas

Altura	10 cm
Peso	250 gramos

ENTRADA **SALIDA**

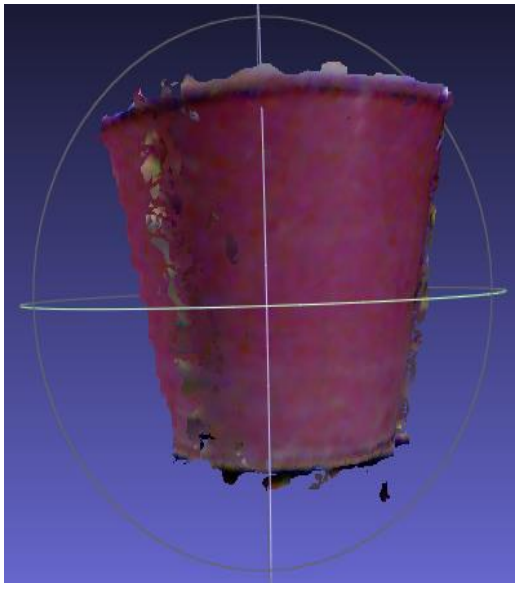


Prueba 4(Cubeta)

Características aproximadas

Altura	45 cm
Peso	350 gramos

Entrada **Salida**

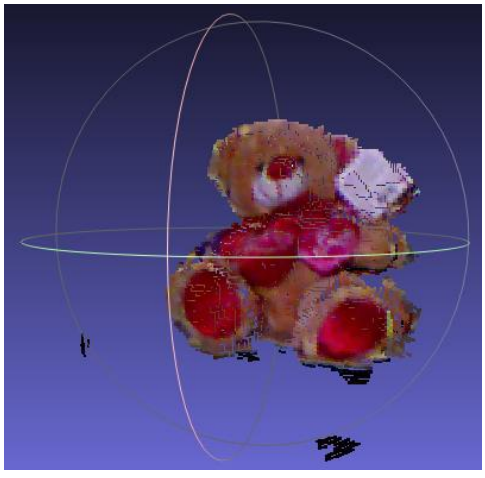


Prueba 5(Peluche)

Características aproximadas

Altura	35 cm
Peso	200 gramos

Entrada **Salida**



Después de haber analizado diversos objetos de prueba, se encontró que los objetos que poseen trasparencias son particularmente difíciles de “ver” por el sensor kinect, prueba que se respalda en la Figura 9.4 donde se escaneó un garrafón transparente, en la cual las partes donde el garrafón es transparente no lo capto.

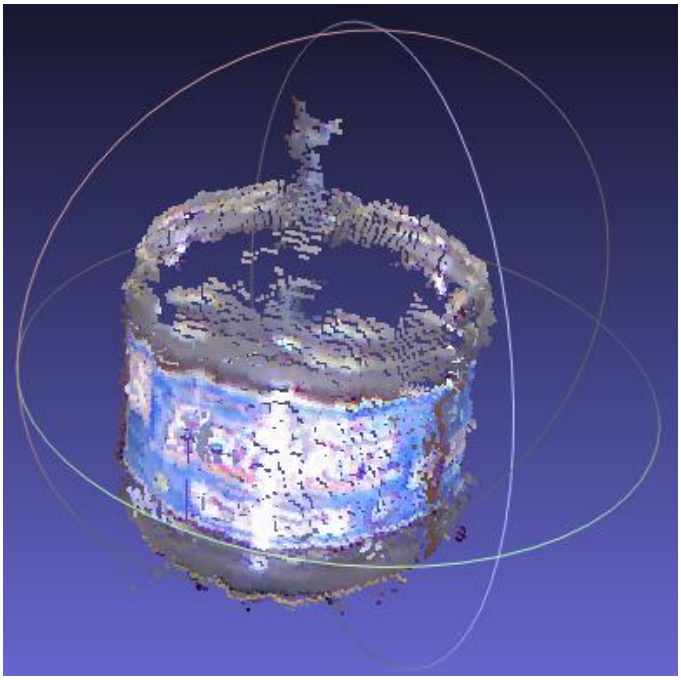


Figura 9.4 Escaneo de objetos transparentes

10 Referencias

[1] American Machinist, “The CAD/CAM Hall of Fame”, Septiembre 2013, [Online] Disponible: http://americanmachinist.com/cadcam-software/cadcam-hall-fame

[2] S.Kean, J. Hall y P.Perry. *Meet the Kinect*. ed. New York: apress,2011.

[3] Y. Lim, H. Lee, etal, “3-d Reconstruction Using the Kinec Sensor and Its Application to a Visualization System”, in *2012 IEEE Intrnational Conference on Systems, Man and Cybernetics*, October 2012.

[4] J. Smisek, M. Jancosek, etal, “3D with kinect”, Praga, 2011.

[5] W. Yi, W. Jin, etal, “A Study in 3D-Reconstruction Usin Kinect Sensor” China, 2012.

[6] [Online] Autodesk 123D Disponible: http://www.123dapp.com/catch

[7] [Online]Autodesk Disponible:http://mexico.autodesk.com/adsk/servlet/pc/index?id=14607888&siteID=1002155

[8] [Online]iremsDisponible:http://www.irems.co/

[9] [Online]OpticScan Disponible:http://www.shining3dscanner.com/en-us/product_opticscan.html

[10] [Online] Metra Scan 3D Disponible:http://www.creaform3d.com/lp/template1/metrascan/en/index.php

[11] KlemensHahrmann, “3D reconstructionwiththekinect-camara”, Tesis de licenciatura, Universidad Tecnologica de Viea, Viena, Europa, 18-02-2013.

[12] Miika Santana, “3D contentcapturin and reconstructionusing Microsoft kinectdepthcamara”, Tesis de licenciatura, Universidad de Oulu, oulu, Finlandia, primavera del 2012.

[13] Jacob Kjaer, “A qualitative analysis of two automated registration algorithms in a real world scenery using point clouds from kinect”, tesis de licenciatura, 26-06-2011

[14] J. Webb, J. Ashley. *Kinect Programming with the Microsoft Kinect SDK.* Ed. New York: apress 2012.

[15] [Online] Kinect for Developers Disponible: http://www.kinectfordevelopers.com/

[16] [Online] Diccionario Real Academia Española: http://rae.es/recursos/diccionarios/drae/

[17] J. Kramer, N. Burrus, F. Echtler, D. Herrera y M. Parker *Hacking the Kinect.* Ed. New York: apress,2012.

[18] [Online] Xbox Kinect Disponible: http://www.xbox.com/es-ES/Kinect

[19] [Online] Atmel Corporation. Disponible: http://www.atmel.com/

[20] [Online] Embedded magazine. Disponible: http://www.embedded.com/magazines/Embedded-Systems-Design-magazine-archive
[21] [Online] My Circuits 9. Disponible: http://www.mycircuits9.com/2013/04/Simple-Steps-Burn-Program-into-Microcontroller.html
[22] [Online] How Stuff Work, blog. Disponible: http://electronics.howstuffworks.com/microcontroller3.htm
[23] [Online] Encoder the newsletter of the Seattle Robotics Sociaty. Disponible: http://www.seattlerobotics.org/encoder/sep97/basics.html
[24] [Online] MikroElektronika magazine. Disponible: http://www.mikroe.com/chapters/view/64/chapter-1-introduction-to-microcontrollers/
[25] [Online] Arduino. Disponible: http://www.arduino.cc/
[26] Atmel, ATMega8535 Datasheet. Diciembre, 2003.