



PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO



José Emmanuel Díaz Flores

Estudio y desarrollo de un escáner 3D de bajo costo.

Informe Proyecto de Título de Ingeniero Civil Electrónico



**Escuela de Ingeniería Eléctrica
Facultad de Ingeniería**

Valparaíso, 08 de octubre de 2018



Estudio y desarrollo de un escáner 3D de bajo costo

José Emmanuel Díaz Flores

Informe Final para optar al título de Ingeniero Civil Electrónico,
aprobada por la comisión de la
Escuela de Ingeniería Eléctrica de la
Facultad de Ingeniería de la
Pontificia Universidad Católica de Valparaíso
conformada por

Sr. Francisco Alonso Villalobos
Profesor Guía

Sr. David Velasco López
Segundo Revisor

Sr. Sebastián Fingerhuth Massmann
Secretario Académico

Valparaíso, 08 de octubre de 2018

Agradecimientos

A dios y mi familia, que siempre fueron mi sustento por seguir luchando cada día en la universidad, acompañándome y brindándome todo para poder finalizar esta hermosa carrera universitaria de 6 años de estudio, permitiéndome llegar a lugares donde nunca imaginé.

Además de mis amigos que han sido un fuerte apoyo desde un inicio guiándome por el buen camino para llegar a esta instancia.

Finalmente, a la persona especial que llegó a brindarme apoyo en el último tramo de mi carrera, confiando siempre en mí, siempre apoyándome y motivándome en todo sentido, así logré realizar dos labores que creía imposible, trabajar en mi área de carrera y llegar al tramo final de ésta.

Valparaíso, 8 de octubre de 2018

José Díaz.

Resumen

Se elabora un análisis detallado acerca de la historia que envuelve a los escáneres 3D, exhibiendo el contexto donde surgen los primeros modelos, hasta como fueron mejorando las técnicas de escaneado llegando a los modelos actuales conocidos. Además, se realiza un estudio del estado del arte, donde se investiga los diversos métodos existentes para la obtención de modelos en 3D, revisando en profundidad la documentación que existe en la red.

Por otra parte, se procede a señalar los diversos escáneres 3D de código libre que existen, en donde se comenta el por qué es mejor emplear un escáner de código libre y se muestran los equipos disponibles que poseen código libre, señalando tanto los costos aproximados, como sus fortalezas y debilidades.

A continuación, se analiza cada componente que conforman las soluciones anteriormente mencionadas, exhibiendo sus fortalezas y dando un análisis en base a esto, como los costos asociados a utilizar cada plataforma (como base), es importante señalar que tanto los componentes electrónicos como los mecánicos son fundamentales a la hora de tener un buen rendimiento en los modelos que se quieren obtener.

Una vez que se identifican los componentes mecánicos y electrónicos con los que se trabajará, se procede a elaborar los primeros modelos (prototipos) del diseño del escáner, para ello se trabaja con el software 123D Desing de Autodesk, el cual es una potente herramienta a la hora trabajar con ella y a su vez, posee una interfaz bastante amigable. Dentro del proceso de diseño, se realizan pruebas con el fin de verificar lo proyectado teóricamente con el modelo físico, en donde se proceden a realizar cambios en favor de mejorar la robustez de este proyecto.

Consiguientemente, se procede a detallar el proceso de calibración del driver, el cual es importante debido a que controla al motor, que está inserto en la base que moverá al objeto en cuestión, por esta razón se detalla paso a paso los procedimientos para lograr buenos resultados con este motor, lo cual brinda el siguiente paso que es integración software (en este caso: Horus) con los componentes electrónicos, donde se detalla la importancia de cada opción presente del software y a su vez, detallando como se realiza la afiliación software-electrónica.

Finalmente, se muestra los primeros modelos y como se fueron mejorando a través de las opciones que brinda el software, exhibiendo la importancia de las opciones seleccionadas que condujeron a mejoras en los diseños obtenidos.

Palabras claves: Escáner 3D, Código abierto, Calibración, Software Horus, Triangulación.

Abstract

They approached the 3D analysis, exhibiting the context in which the first models, even how improving the scanning techniques reaching the current known models. In addition, a study of the state of the art is carried out, where the diverse existing methods for the obtaining of 3D models are investigated, reviewing in depth the documentation that exists in the network.

On the other hand, we proceed to point out the obstacles open source 3D scanners, where it is said that it is better to use a free code and that the available equipment is available, such as its strengths and weaknesses.

Next, each component that makes up the aforementioned solutions is analyzed, showing their strengths and giving an analysis based on this, as the costs associated with using each platform (as a base), it is important to note that both the electronic and mechanical components They are fundamental when it comes to having a good performance in the models that you want to obtain.

Once the mechanical and electronic components that will be worked on are identified, the first models (prototypes) of the scanner design will be elaborated, for which we will work with Autodesk's 123D Desing software, which is a powerful tool to the time to work with it and in turn, has a fairly friendly interface. Within the design process, tests are carried out in order to verify what is projected theoretically with the physical model, where changes are made in favor of improving the robustness of this project.

Consequently, we proceed to detail the calibration process of the driver, which is important because it controls the motor, which is inserted in the base that will move the object in question, for this reason it details step by step the procedures to achieve good results with this engine, which provides the next step that is software integration (in this case: Horus) with the electronic components, where the importance of each present option of the software is detailed and in turn, detailing how the affiliation is made software- electronics.

Finally, it shows the first models and how they were improved through the options offered by the software, showing the importance of the selected options that led to improvements in the designs obtained.

Key words: 3D Scanner, Open Source, Calibration, Horus Software, Triangulation.

Índice general

Introducción	1
1 Antecedentes del proyecto.....	4
1.1 Contextualización.....	4
1.1.1 ¿Qué es un escáner 3D del tipo láser?	4
1.1.2 Historia del escáner 3D	5
1.1.3 Aplicaciones de los escáner 3D	8
1.1.3.1 Aplicación Industrial y manufactura	8
1.1.3.2 Aplicación de patrimonio cultural	8
1.1.3.3 Aplicación médica.....	9
1.2 Problemática	10
1.3 Solución propuesta.....	10
1.4 Objetivos	10
1.5 Estado del arte.....	11
1.5.1 Escáner 3D por triangulación láser	12
1.5.2 Escáner 3D por luz estructurada	14
1.5.3 Escáner 3D por tiempo de vuelo	16
1.5.4 Comparativa de las diversas tecnologías.....	16
2 Escáneres 3D Open Source	19
2.1 Escáneres disponibles.....	19
2.1.1 Spinscan	19
2.1.2 Fabscan	20
2.1.3 Atlas 3D	21
2.1.4 Bq Ciclop	23
2.2 Estudio de los posibles componentes a utilizar	24
2.2.1 Arduino.....	24
2.2.1.1 Composición del arduino	25
2.2.2 Sensores	26
2.2.3 Shields	27
2.2.4 Raspberry Pi.....	28
2.2.4.1 Composición del Raspberry Pi	28
2.2.5 Sensores	32
2.2.6 Motor paso a paso (Stepper motor)	33
2.2.7 Motor servo	34

2.2.8 Láseres	36
2.3 Estudio de plataformas Arduino y Raspberry Pi	36
2.3.1 Raspberry Pi 3	36
2.3.2 Arduino R3	37
2.3.3 Tabla comparativa	39
2.3.4 Plataforma que desarrollará el proyecto	40
2.4 Shield BQ ZUM SCAN	40
2.5 Drivers A4988	41
2.6 Motor Nema 17	42
2.7 Láseres	43
2.8 Estudio de las cámaras	44
2.8.1 Logitech C270	44
2.8.2 Logitech C920	45
2.8.3 Cámara a elegir.....	46
2.9 Rodamientos	46
2.10 Estudio del software	47
2.10.1 Horus	47
3 Proceso de diseño	49
3.1 Propuesta de diseño	49
3.2 Cambios en el prototipo	52
4 Calibración, Horus y Diseños obtenidos.	54
4.1 Procedimiento para calibrar.....	54
4.2 Configuración de parámetros.	58
4.3 Ejecución del Software.....	60
4.4 Primeros modelos escaneados.....	66
Discusión y conclusiones.....	69
Bibliografía.....	72

Introducción

El escáner 3D se puede definir como un dispositivo que es capaz de analizar un modelo físico con el fin de obtener datos sobre su forma y posible apariencia, es decir, se adquiere la geometría del objeto en cuestión. Dichos datos son posteriormente analizados y procesados para precisar la posición, además del modelo 3D del objeto.

El desarrollo del escáner 3D se remonta hacia la última mitad del siglo 20, con el objetivo de recrear las superficies de los lugares y objetos. Este tipo de tecnología fue creada en los años sesenta, los primeros prototipos utilizaban luces, cámaras y a su vez proyectores para desarrollar dicha labor.

Usualmente, la utilización los primeros prototipos de escáneres 3D suponía una tarea algo compleja, debido a las limitaciones que poseían en si los equipos, es por ello que se empleaba mucho tiempo y esfuerzo para escanear algo con precisión. No es hasta 1985, cuando los escáneres mencionados anteriormente son reemplazados por nuevas tecnologías que incluían el uso del láser, luz blanca y sombreado para capturar una superficie.

Actualmente, se puede identificar claramente dos categorías de escáneres las cuales son: escáner de contacto y escáner sin contacto. El primero posee una pieza que transita sobre el objeto, donde ese elemento suele ser una punta de zafiro o acero duro, debido a esto es complicado trabajar con esta tecnología sobre un objeto frágil. Mientras que, en el caso del escáner sin contacto se basa en diferentes técnicas, que en términos generales se describe como el lanzamiento de una señal y esta es comparada con el retorno de la misma (para el caso de los activos), para el caso de los pasivos se estudia la radiación reflejada del ambiente.

A medida que ha transcurrido el tiempo, los escáneres 3D han tomado gran relevancia en todo ámbito, siendo un ejemplo el área de la industria donde se emplean con el fin de comparar lo proyectado con los modelos reales a la hora de fabricar productos, siendo una gran herramienta de retroalimentación, que permite tanto a los investigadores como los diseñadores encontrar las posibles falencias de lo que están buscando desarrollar. En otras áreas como la medicina, también están jugando papeles importantes, específicamente hablando en la sección forense, ya que es posible analizar cuerpos de personas fallecidas que necesitan un análisis en profundidad y la solución es el escáner 3D, ya que brinda una ventaja inmejorable para analizar detalles no perceptibles al ojo humano muchas veces.

Es importante señalar que el presente proyecto, posee la finalidad de obtener diseños 3D a un costo bajo empleando materiales que no sean difíciles de conseguir y a su vez, que posean grandes prestaciones resguardando la relación costo/ calidad, por lo tanto se elige emplear un software que sea de código abierto, utilizando como base el Horus que pertenece a la empresa BQ Ciclop, la cual busca que las personas desarrollen sus propios modelos y que éstos puedan ser escalables dada la gran interacción entre usuarios dentro de la red, los cuales van implementando mejoras al software, por esta razón BQ Ciclop resulta vital a la hora de desarrollar este proyecto.

En septiembre del 2013 un ingeniero industrial llamado Jesús Arroyo comienza a trabajar en una empresa española de nombre BQ, en dicho momento se estaba poniendo en marcha un nuevo departamento en dicha empresa “Departamento de Innovación y Robótica”. Es aquí donde surge el “Ciclop” uno de los primeros prototipos de escáner 3D que son fabricados en base a materiales de fácil adquisición, estos se basan específicamente en los materiales existentes en la comunidad RepRap.

Gracias al desarrollo del “Ciclop”, muchas empresas se han visto interesadas en el desarrollo de esta tecnología, la cual tiene el fin de apoyar a los diversos escáneres 3D existentes en la actualidad. El día de hoy, existen varios escáneres 3D de bajo costo basados en el “Ciclop”, dando ayuda al desarrollo de diversos softwares que han ido en beneficio directo de todas las personas interesadas a acceder en esta innovadora tecnología.

En la actualidad, existe una gran oferta de softwares de diseño para computador, dentro de estos se tienen los open source, gratuitos y pagados. En este proyecto, se elaborará el prototipo 3D con el software gratuito elaborado por la empresa AutoDesk. En específico, el software elegido es el 123D Desing, con el cual se podrá desarrollar el diseño completo del prototipo de escáner, para posteriormente imprimirlo. Seguidamente después de elaborar el primer diseño en 3D del escáner, se realiza un análisis en profundidad acerca del modelo, así se identificarán las posibles deficiencias, como también las posibles mejoras para así obtener destacadas prestaciones en el escáner.

Antes de obtener el diseño final con el cual se trabajará, se procede a imprimir cada una de las partes que conforma el prototipo de escáner, siendo importante verificar que la construcción del modelo esté acorde a los estándares proyectados inicialmente, específicamente en este caso, se realizaron aproximadamente 4 iteraciones, de las cuales en la práctica se encontraron detalles que requieren volver a diseñar e imprimir el prototipo, así se obtuvo el diseño final de acuerdo a lo proyectado en un inicio y con el fin de cumplir con los objetivos propuestos.

Durante la construcción del prototipo, se requiere ser meticuloso, con el fin de obtener el diseño acorde a lo proyectado inicialmente (teoría).

Quizás una de las partes más importantes y que presenta mayor dificultad a lo largo de este proyecto, el cual es el firmware Horus, el cual en términos generales es bastante completo, sirviendo tanto para el desarrollo de este proyecto, sino también para máquinas CNC, impresoras 3D, entre otros. Es importante destacar que la base de este firmware presta soporte para diversas plataformas de placas y cuenta con una gran cantidad de funcionalidades disponibles, por esta razón es importante verificar con cuales se trabajará y se optimizarán para obtener un modelo de escáner 3D con grandes prestaciones y que sea completamente funcional.

Por esta razón, este proceso fue uno de los más complejos, ya que requiere de gran cantidad de parámetros a probar con el fin de tener los resultados proyectados inicialmente, esto conlleva al uso del software Horus, ya que la unión del firmware con el software Arduino resulta ser lo importante a la hora de trabajar con el escáner.

Un punto a destacar, es que se deben realizar varias pruebas a lo largo del proyecto, todo esto en el contexto de lo señalado anteriormente, ya que el software Horus lee la información inserta dentro de la placa Arduino (donde se programa previamente) y esta interacción es la que permite controlar la electrónica que conforma al escáner, es decir, los láseres, motor y cámara.

Cuando se obtiene el código configurado, consigue otra parte importante, la configuración de los parámetros que brinda el software Horus, estos parámetros se hacen realmente necesarios de estudiar y comprender a cabalidad, ya que al comprenderlos se entiende el funcionamiento real del escáner y se puede ir obteniendo mejoras en la captura de los modelos a través del procesamiento de imágenes.

En términos generales, lo que busca este proyecto es ser un aporte para los diversos estudiantes que conforman la Escuela de Ingeniería Eléctrica de la PUCV, por esta misma razón se elabora un detalle minucioso acerca de los factores que influyen en el desarrollo de este proyecto, tales como calibración de motor, diseño de piezas, configuración del software Horus, programación y montaje.

Por esta razón, el anexo inserto dentro de este informe, resulta vital para cualquier persona que quiera insertarse dentro de este mundo, ya que brinda una completa guía respecto a el montaje, configuración e instalación del software con los cuales se pensó elaborar este proyecto.

1 Antecedentes del proyecto

Una vez establecidos los objetivos del proyecto, se requiere investigar y documentar las diversas tecnologías existentes hasta este momento, como se han abordado y a su vez, exhibir tanto la utilidad como proyectos similares al que se está proponiendo en esta ocasión.

1.1 Contextualización

1.1.1 ¿Qué es un escáner 3D del tipo láser?

Antes de explicar que es un escáner 3D tipo láser, es importante destacar el término “escaneo láser”, ya que este puede hacer referencia dos cosas: En una primera instancia podría hacer referencia a la deflexión controlada de rayos láser, ya sea invisible como visible [2]. En ese sentido, los rayos láser pueden ser empleados para el procesamiento de materiales, escáneres de código de barra, láser oftalmológico, grabado láser, entre otros.

Por otro lado, se podría asignar el término “escaneo láser” a algo más específico que la definición anterior y hacer referencia a la dirección controlada de los rayos láser, sucesivamente se realiza una evaluación de la distancia de cada dirección de señalización. Usualmente dicho método, se le denomina “Exploración de objetos en 3D o escaneo láser 3D”, así se puede capturar rápidamente las formas de los objetos, paisajes y edificios.

El general, se podría especificar al escáner 3D tipo láser como una tecnología sin contacto y no destructiva, la cual captura el aspecto de los objetos utilizando una luz láser para posteriormente almacenarlo digitalmente. Un escáner 3D del tipo láser concibe una “nube de puntos” de datos la cual proviene de la superficie del objeto analizado.

Dicho de otra forma, el escaneo 3D tipo láser permite capturar la forma y el tamaño exacto de un objeto físico para así llevarlo al mundo computacional como una representación tridimensional digital. Además, este tipo de escáner permite determinar finos detalles y captar la figura de forma libre, para así engendrar rápidamente una nube de puntos, la cual es de gran precisión.

El escáner 3D tipo láser es muy útil a la hora de realizar inspecciones en las superficies contorneadas, mediciones y geometrías complejas que solicitan una gran cantidad de datos para una descripción precisa, en donde probablemente un escáner de contacto no sería práctico utilizarlo.

1.1.2 Historia del escáner 3D

Hacia la última mitad del siglo XX se comienza a desarrollar el escáner 3D, con el objetivo de recrear las superficies de objetos y lugares. Este tipo de tecnología es principalmente útil en el área de diseño y en la investigación. Los primeros prototipos de escáner 3D fueron desarrollados hacia los años sesenta, donde éstos empleaban luces, cámaras y proyectores con el fin de realizar el escaneo [1]. Como era de esperar, en los primeros prototipos se encontraron muchas limitaciones de equipo, los cuales influían directamente en el tiempo y esfuerzo para poder realizar un escaneo de precisión de un objeto.

Ya en 1985, es cuando es reemplazada la tecnología anteriormente mencionada, por escáneres que empleaban luz blanca, láseres y sombras para poder capturar una superficie. En paralelo a esto, en los años ochenta, fue desarrollada una sonda de contacto por la industria de fabricación de herramientas. Esta tenía por objetivo crear un modelo preciso, sin embargo, su labor la realizaba de manera lenta.

Debido a esto, los expertos se dedican a trabajar con la tecnología óptica. El utilizar la luz, resultó ser un recurso mucho más eficiente que la sonda física, ya que el primero era expedito a la hora de trabajar y además no comprometía a los objetos frágiles.

En ese contexto, se encontraban disponibles tres tipos de tecnologías ópticas, las cuáles eran:

- Por sensor de punto: Este tipo de tecnología era semejante a tener una sonda física, donde se utilizaba solamente un punto de referencia, el cual era repetido muchas veces, tal como se muestra en la figura 1-1. Cabe mencionar, que este tipo de tecnología fue la más lenta, porque se necesitaba bastante movimiento físico en el sensor.

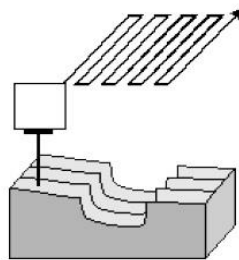


Figura 1-1: Diagrama de tecnología óptico por sensor de punto [1].

- Por sensor de área: Esta fue una tecnología que poseía una gran dificultad técnica, ya que se utilizaba una gran área (representada por la pirámide) para realizar el escaneo, tal como muestra la siguiente figura 1-2.

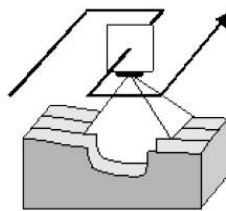


Figura 1-2: Diagrama de tecnología óptico por sensor de área [1].

- Por sensor de línea (banda): Esta última tecnología, resultó ser más rápida que la por sensor de punto, debido a que utiliza una banda de puntos para recorrer el objeto que se está escaneando. Al ser una banda de puntos, se vuelve más exacta a la hora de comparar con las otras tecnologías mencionadas anteriormente.

En la siguiente figura 1-3, se presenta la forma en que trabajaba esta tecnología:

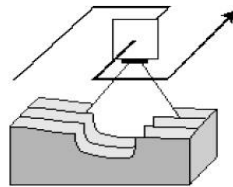


Figura 1-3: Diagrama de tecnología óptica por sensor de línea [1].

Como era de esperar, el tercer tipo de tecnología fue el que más próspero, pero junto con ello surgió la problemática del software a utilizar. El sensor al realizar el escaneo en tres dimensiones, se obtienen diversas posiciones del objeto. Aquí es donde existía el mayor problema, ya que el software debía ser capaz de eliminar los datos duplicados, unir los diferentes datos que se obtenían del escaneado.

La tecnología por sensor de línea o banda obtiene una de sus primeras aplicaciones importantes, en los años ochenta para capturar modelos de los seres humanos, con el fin de utilizarlos en la animación. Dicha empresa fue “Cyberware Laboratories of Los Angeles” y nombró su dispositivo como “Head Scanner” [1], el dispositivo es mostrado en la siguiente figura 1-4.



Figura 1-4: Primeros modelos de escáner por sensor de línea (Head Scanner) [1].

Estos prototipos fueron realmente útiles y dieron inicio al surgimiento de nuevos modelos, como el escáner de cuerpo completo en 3D. Este último, nace a mediados de los noventa y captura digitalmente la forma humana utilizando una tecnología conocida como “Fotogrametría de superficie digital (digital surface photogrammetry)”. Este prototipo de escáner trabaja proyectando un patrón de luz aleatoria sobre la persona, con el fin de capturar los contornos de la superficie y la información de color con diversas cámaras digitales ubicadas en distintos ángulos, siendo estas sincronizadas con precisión. Este modelo se puede observar en la siguiente figura 1-5.



Figura 1-5: Escáner de cuerpo completo empleando la tecnología de Fotogrametría [1].

Debido a los altos costos económicos que significa fabricar un escáner 3D, surge a través de la empresa “Immersion and Faro Technologies” el primer escáner 3D de bajo costo, el cual era operado manualmente para digitalizar. Éstos eran capaces de producir modelos complejos, pero su desventaja radicaba en la lentitud para realizar un modelo detallado y no podía digitalizar las superficies de color. Este tipo de escáner 3D es mostrado en la figura 1.6.



Figura 1-6: Modelo de escáner de bajo costo [1].

Ya en 1996, los escáneres 3D toman las tecnologías anteriormente mencionadas (brazo accionado manualmente y escaneo por línea o banda) y al juntarlos en “ModelMaker” crean un sistema muy rápido y a la vez flexible, convirtiéndose en el primer sistema de captura de realidad del mundo. Realizando modelos complejos y texturas con color. Un ejemplo de ésto, se muestra en la siguiente figura 1.7.

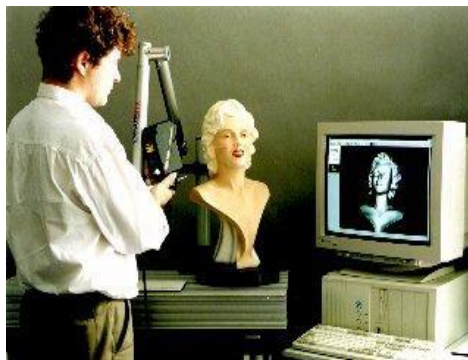


Figura 1-7: Escáner 3D operado manualmente con tecnología por sensor de línea [1].

1.1.3 Aplicaciones de los escáner 3D

1.1.3.1 Aplicación Industrial y manufactura

Antiguamente, desarrollar diseños de un determinado producto, realizar mediciones de objetos de una geometría compleja, entre otras, solía tomar desde días a semanas en completar dichas tareas, sin embargo, en la actualidad el uso de escáneres 3D ha influido para realizar de forma expedita todas estas tareas.

En la actualidad, las empresas pueden utilizar este tipo de tecnología para capturar desde pequeñas piezas hasta otras de tamaños grandes como una turbina y todo esto con una gran precisión. Además, los modelos 3D pueden ser exportados a diversos programas (digitalizados) para poder realizar mediciones, modificar o mejorar los diseños, integración con nuevos sistemas de producción, entre otros.

En la figura 1-8, se muestra un ejemplo de manufactura, donde se tiene un diseño ideal por parte del fabricante, posteriormente es fabricado y luego se realiza un escaneado para poder comparar los posibles defectos de fabricación del producto [3].

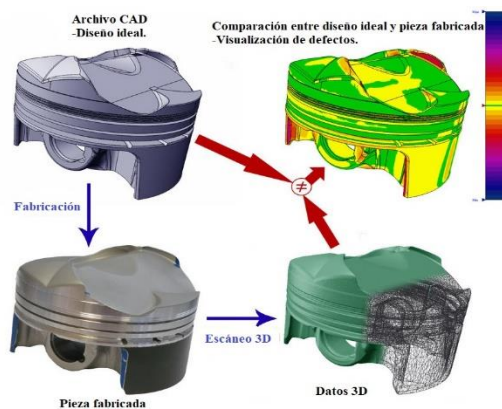


Figura 1-8: Aplicación de escáneres 3D en la manufactura de un producto [3].

1.1.3.2 Aplicación de patrimonio cultural

En la actualidad, el empleo de imágenes 3D ha crecido bastante en los museos y especialistas de arqueología. Diversos laboratorios de investigación han trabajado en el modelado estatuas, virtualización en 3D de sitios arqueológicos, entre otros. En general, el proceso de medición y elaboración es la ingeniería inversa al igual que en el ítem mencionado anteriormente. El procedimiento habitual es: adquisición, modelado y visualización para monitorear el patrimonio arqueológico.

El poder tener los modelos 3D, permite la reproducción virtual, las cuales pueden tener fines didácticos y de visualización. En la figura 1-9 se muestra como la empresa Artec colaboró con la Academia de Ciencias de Rusia, en la documentación de los restos mortales de un soldado. Todo esto, en el marco del aniversario de los 200 años de la batalla de Borodino (cuando Napoleón invade Rusia) [3].



Figura 1-9: Escaneo de campo de batalla de Borodino [3].



Figura 1-10: Modelo digitalizado de campo de batalla de Borodino [3].

1.1.3.3 Aplicación médica

El empleo de los escáneres 3D ha ido en aumento en el área de la medicina, entregando datos de la superficie con una alta resolución y con posibilidad de construir modelo del cuerpo muy precisos. Los diversos datos 3D son aprovechados en el diseño de cirugías plásticas, diseño de prótesis, ortodoncia, ortopedia, dermatología y cirugía.

Actualmente, los equipos 3D también pueden ser utilizados en la medicina forense, los cuales en conjunto con datos radiológicos proporcionan una completa información de lesiones. En la siguiente figura 1-11, se muestra una víctima golpeada por un tubo de metal en la cabeza en la primera imagen, la segunda imagen muestra la digitalización de dicha persona para un análisis más detallado, mientras que la tercera imagen exhibe la lesión a profundidad para un estudio aún más profundo, lo cual otorga una gran ventaja para el médico que está analizando el caso [3].



Figura 1-11: Aplicación de escáneres 3D en la medicina Forense [3].

1.2 Problemática

Si bien antiguamente el escáner 3D era utilizado para aplicaciones muy específicas y a su vez, su costo monetario era extremadamente alto, el día de hoy es posible encontrar diversas aplicaciones para un escáner 3D, como por ejemplo una integración con un sistema de impresión 3D.

Actualmente para poder imprimir un objeto en este tipo de impresoras, se hace necesario conocimiento tanto sobre el equipo (características y la forma en que se va a utilizar la impresora 3D), como también del manejo de diversos softwares para poder modelar el objeto que se desea imprimir, lo cual es una limitación para las personas que quieran introducirse en este nuevo concepto de impresión que se está masificando día a día a nivel mundial, pudiendo complicar bastante a los diversos usuarios que manejan pequeños negocios con impresoras de “bajo costo” si es que no se tiene el conocimiento para poder manipular los diversos softwares que existen para el diseño.

Al llevar esto a la realidad de la Escuela de Ingeniería Eléctrica, se posee una impresora 3D, a la cual no muchos alumnos pueden acceder, debido a la carencia de conocimiento a la hora del manejo de software de diseño. Y si se lleva esto al nivel de la Universidad, el problema es aún mayor, ya que existen impresoras para realizar proyectos de diversas carreras y ahí el conocimiento es mucho menor.

1.3 Solución propuesta

Se sugiere elaborar un escáner 3D de triangulación con código abierto de bajo costo, pero con una gran calidad de materiales, con el fin de facilitar el uso de las impresoras 3D que posee tanto la Escuela de Ingeniería Eléctrica como las que pertenecen a la Universidad.

El escáner será construido por una impresora 3D para la estructura base, con el objetivo de minimizar los costos de fabricación y a su vez de favorecer posibles modificaciones en el diseño propuesto inicialmente, para así poder mejorarlo.

Una de las ventajas de tener un escáner es acercarse a la comunidad universitaria al mundo 3D y a su vez, ofrecer la posibilidad de replicar estructuras de manera más rápida, ya que el usuario puede utilizar el tiempo ahorrado en el diseño en otra tarea.

1.4 Objetivos

El principal objetivo del proyecto es obtener un escáner de bajo costo, sin embargo, para la elaboración de éste se buscarán las mejores opciones tanto de material de construcción como el software a utilizar, con el fin de tener un buen rendimiento empleando un código abierto. Además, se pretende que el diseño posibilite a los usuarios de tener un fácil acceso a la reparación, montaje, obtener mejoras en un futuro y que sea de un uso simple.

Para llevar a cabo todo esto, es necesario definir objetivos más específicos los cuales son:

- Diseño y fabricación de piezas con factibilidad de modificación con el objetivo de mejorar las prestaciones.

- El diseño del escáner no emplee demasiadas piezas, ya que esto conllevará a un engorroso armado.
- Empleo de materiales de fácil acceso al usuario (en caso de reemplazo) y a su vez, que sean de un bajo costo, pero de buena calidad.
- Uso de software de código abierto, para así tener mayores posibilidades de mejoras de éste y además se pueda obtener un mejor rendimiento. Esto es debido, a que el código abierto promueve la colaboración entre los usuarios.

1.5 Estado del arte

En los últimos años, la demanda de imágenes ópticas en 3D se ha incrementado relevantemente, ésto ha llevado al desarrollo de diversos instrumentos que el día de hoy están disponibles comercialmente.

Usualmente, los sensores de imagen 3D operan proyectando (de forma activa) o adquiriendo (en forma pasiva) energía electromagnética hacia / desde un objeto, posteriormente se realiza un registro de la energía transmitida o reflejada. Como, por ejemplo, la tomografía computarizada (CT). Esta última emplea rayos X de alta energía y luego evalúa la radiación transmitida a través del objeto.

Sensores de reflexión son empleados para adquirir las formas. éstos pueden subdividirse en sensores ópticos y no ópticos. En la detección no óptica se tienen los sensores acústicos (ultrasónicos, sísmicos); electromagnéticos (infrarrojos, ultravioleta, radar de microondas, etc.), entre otros: Las tecnologías mencionadas anteriormente miden las distancias a los objetos, mediante la medición del tiempo requerido para que el pulso de sonido o energía de microonda rebote detrás del objeto.

Cabe mencionar, que existen en la actualidad una variedad amplia de tecnologías para obtener imágenes en 3D, no existiendo una única clasificación.

A continuación, se presenta la tabla 1-1 donde se muestra la clasificación de las principales tecnologías 3D en la actualidad:

Tabla 1-1: Clasificación de las diversas tecnologías 3D.

	Triangulación.	Tiempo de retardo.	Imágenes monoculares.	Pasivo.	Activo.	Directo.	Indirecto.	Rango.	Orientación de superficie.
Triangulación láser.	X				X	X		X	
Luz estructurada.	X					X		X	
Visión estereoscópica.	X			X		X		X	
Fotogrametría.	X			X		X		X	
Tiempo de vuelo.		X			X	X		X	
Interferometría.		X			X	X		X	

En la actualidad, existen diversas tecnologías 3D las cuales son basadas en la triangulación óptica, en el retardo de tiempo y en el uso de imágenes monoculares. A su vez, éstas se clasifican en métodos pasivos y activos. El método pasivo, la iluminación de la escena y reflectancia del objeto son empleadas para deducir información acerca de la forma. Para el caso del método activo, se utilizan fuentes lumínicas (adecuadas) como vector interno de información.

Para el caso de las técnicas directas, éstas hacen referencia a que dan un rango de datos, en otras palabras, éstas dan un conjunto de distancias entre la superficie desconocida (del objeto) y el sensor de alcance, mientras que las mediciones indirectas son deducidas de las imágenes monoculares y del conocimiento previo de las propiedades del objeto.

1.5.1 Escáner 3D por triangulación láser

Este tipo de escáner utilizan la luz láser con el fin de analizar el objeto. El funcionamiento de este tipo de escáner se basa en la incidencia de la luz láser sobre el objeto y, a su vez, el empleo de una cámara para indagar la posición del punto del láser. Al variar la distancia a la que el láser golpea a la superficie del objeto, dicho punto tendrá diversos lugares en el sensor de la cámara.

El nombre “triangulación” proviene de la ubicación entre el láser, la cámara y el punto láser, los cuales forman un “triángulo”. Una típica configuración de este sistema se muestra en la figura 1-12.

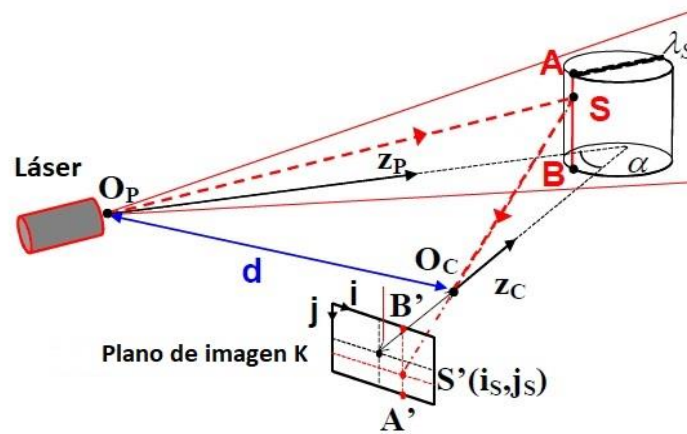


Figura 1-12: Diagrama general de la tecnología triangulación láser [4].

En este sistema, los puntos O_P y O_C son las respectivas salida y entrada de la fuente de láser y cámara. Para este caso, la distancia entre dichos puntos se denomina “d”, mientras que los ejes ópticos Z_P y Z_C correspondientes al láser y la cámara forman un ángulo α grados.

Cuando la fuente láser genera un haz de luz, este impacta al objeto en el punto S (en el caso de una triangulación con un solo punto). Luego, el haz disperso se representa en el punto S' del plano de imagen K.

La medida de la posición (i_s, j_s) del punto de imagen en el plano S' define la línea de visión $\overline{S'O_C}$ y mediante geometría simple, se produce la posición de S. La medición de la superficie es conseguida mediante el escaneado [4].

Es importante resaltar que, existe un compromiso entre el campo de visión, la incertidumbre y la medida de resolución, y los efectos de sombra son producidos por los grandes valores del ángulo α en el caso de una configuración convencional de triangulación.

Cuando el láser utilizado posee un lente cilíndrico, este expande el haz luminoso en una sola dirección. Es por ello, que se origina un plano de luz y, a su vez, se iluminan múltiples puntos del objeto al mismo tiempo. En la figura 1-12, el plano de luz es anotado como λ_S , además los puntos iluminados pertenecen a la intersección del plano y el objeto (desconocido), a los cuales se le denomina en la figura como la línea \overline{AB} .

Al medir la ubicación de todos los puntos de la imagen desde el punto A' al punto B' en el plano K es posible determinar la forma en 3D del objeto en relación de los puntos iluminados.

Este tipo de escáner posee una ventaja muy grande en cuanto a su precisión, y su parcial insensibilidad al estado de iluminación y textura de la superficie del objeto.

En cuanto a la utilización de este tipo de escáner, el uso es bastante amplio para el de punto único, el cual va desde el área industrial para la medición de distancias, diámetros, espesores, como para aplicación de control de calidad (en la superficie).

Algunos ejemplos de este tipo de escáner son:

- Vivid 910, Konica Minolta, Inc (mostrado en la figura 1-13, primera imagen).

- Sensores SmartRay series, SmartRay GmbH, Alemania (mostrado en la figura 1-13 segunda imagen).
- ShapeGrabber (ShapeGrabber Inc, CA, EE.UU).
- Operaciones “Pick and Place” con brazos robotizados.
- NextEngine (NextEngine, Inc, CA, EE.UU).
- TriAngles (ComTronics, Inc, EE.UU).



Figura 1-13: Ejemplos de escáneres 3D [29].

Finalmente, dada las ventajas que posee este tipo de escáner y que son mostradas anteriormente, éste será el modelo de escáner a desarrollar para este proyecto.

1.5.2 Escáner 3D por luz estructurada

Este tipo de escáner comparte el planteamiento de un escáner por triangulación. La diferencia entre ambos radica en que, en vez de escanear la superficie estos proyectan patrones de luz en el objeto, pudiendo ser el patrón unidimensional o bidimensional de luz incoherente. El concepto de luz incoherente hace referencia a que la luz es una superposición de dos o más ondas las cuales producen interferencias, de forma que existirán máximos o mínimos de intensidades.

Una vez que se proyecta el patrón se elabora un análisis de la deformación de este, con el fin de obtener información acerca del rango de cada punto visto simultáneamente. Dicho de otra forma, se estudia la deformación realizada por la geometría de la escena al patrón de luz. En la proyección de patrones, es posible rodar un patrón único, como así múltiples. Para ambos casos, el único plano mostrado en la figura 1-12 del ítem anterior es reemplazado por los planos geométricos mostrados en la figura 1-14.

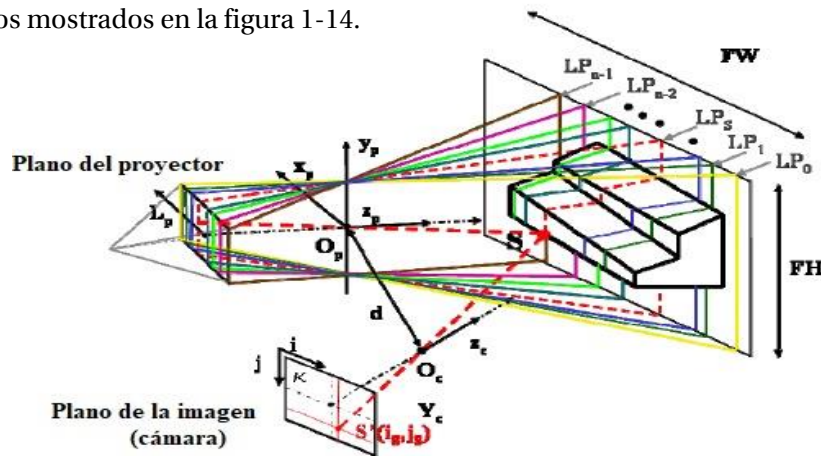


Figura 1-14: Diagrama general de la tecnología Luz estructurada [4].

En la figura anterior 1-14, es posible observar los diversos planos a lo largo de la coordenada L_p correspondientes al plano del proyector. La información del punto del objeto S , se consigue a través de la intersección de SS' y el plano $L_p = LP_S$.

En este tipo de escáner, es de suma importancia asignar los diferentes puntos del objeto a los diversos planos a lo largo de LP_n . Para esto, existe un gran número de tácticas de proyección para las proyecciones.

En la figura 1-15, se exhiben tres patrones de franjas. El correspondiente a (1) está formado por franjas que poseen un enrejado de onda sinusoidal, el (2) es obtenido por superposición de dos patrones de franjas con enrejado sinusoidal de diferentes frecuencias, finalmente el (3) está constituido por franjas de enrejado rectangular [4].

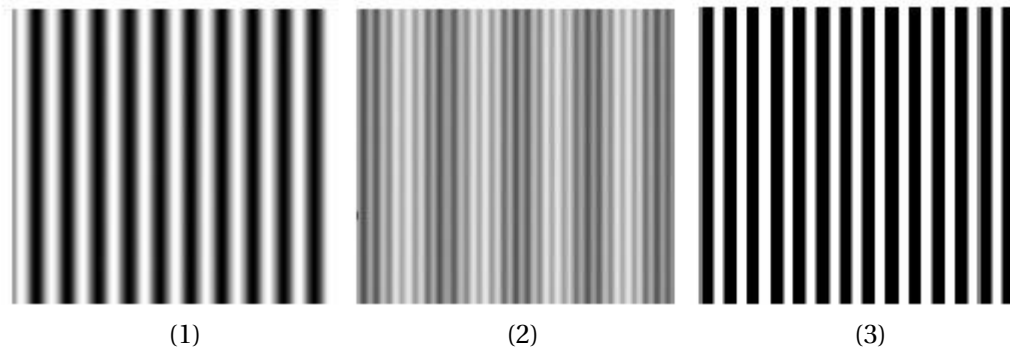


Figura 1-15: Patrones de franja para Luz estructurada [4].

En la siguiente figura 1-16, se muestra un ejemplo de este tipo de escáner.

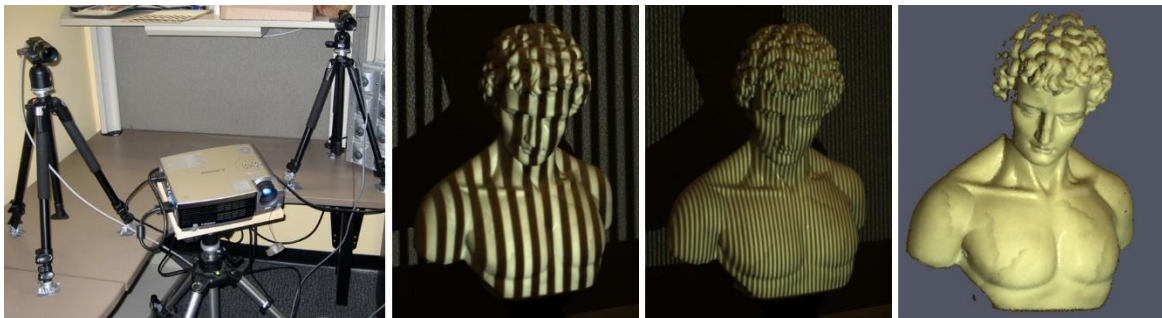


Figura 1-16: Ejemplo de escáner por Luz estructurada [4].

1.5.3 Escáner 3D por tiempo de vuelo

Este tipo de escáner obtiene la distancia del objeto mediante el uso de un rayo láser, luego se calcula el tiempo de viaje de la ida y vuelta de un pulso de luz. Como la velocidad de la luz (haz del láser) es conocida ("C") y, además como cronometra el tiempo de ida y vuelta del haz es posible calcular la distancia entre el escáner y el objeto.

La fórmula para calcular la distancia viene dada por $d = (C * T)/2$, donde d es la distancia, C es la velocidad de la luz (*aproximadamente* $3 * 10^8 \frac{m}{s}$) y T es el tiempo cronometrado entre el viaje de ida y vuelta.

En términos generales, un escáner de tiempo de vuelo no realiza otra cosa que medir la distancia, ángulo horizontal y vertical para cada posición en la que se encuentra. Lo anterior es conseguido moviendo el escáner en forma de cuadrícula en el plano horizontal en 360°, mientras que en el plano vertical es aproximadamente 330°.

Además, la efectividad del escáner vendrá dada por la precisión con la cual se mide el tiempo, considerando que aproximadamente en 3.3 picosegundos la luz viaja un 1 milímetro.

Un ejemplo de esta tecnología es mostrado en la figura 1-17.

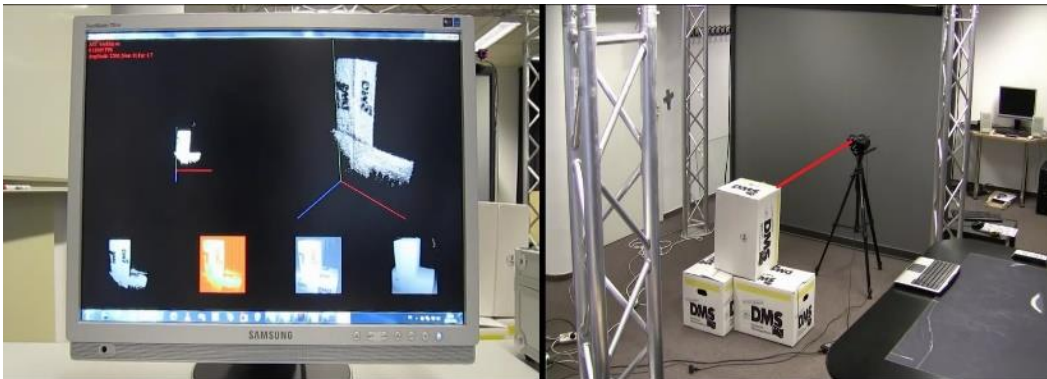


Figura 1-17: Ejemplo de escáner por tiempo de vuelo [5].

1.5.4 Comparativa de las diversas tecnologías

A continuación, se muestra la tabla 1-2 donde se exhibe las diversas tecnologías existentes para el escaneado en 3D, mostrando tanto las fortalezas como las debilidades que presentan.

Tabla 1-2: Comparativa de las diversas tecnologías 3D.

Tecnología	Fortalezas	Debilidades
Triangulación láser.	Relativamente simple. Buen rendimiento general, independiente de la luz ambiental. Alta velocidad, a la hora de adquirir los datos.	Alcance limitado. Limitación de seguridad, la cual es asociada a la fuente láser. Puede existir perdida de datos, por lo cual es posible encontrar sombras y oclusión. Costo.
Luz estructurada.	Alta velocidad, a la hora de adquirir los datos. Buen rendimiento general, pero es dependiente de la luz ambiente.	Limitación de seguridad, la cual es asociada a la fuente láser. Sistema de complejidad media (a nivel computacional). Puede existir perdida de datos, por lo cual es posible encontrar sombras y oclusión. Costo.
Visión estereoscópica.	Relativamente simple y económico. Alta precisión en los objetos que están bien definidos.	Alta demanda de computo. Datos dispersos. Limitación solo a escenas bien definidas. Baja velocidad a la hora de adquirir los datos.

Fotogrametría.	Relativamente simple y económico. Alta precisión en los objetos que están bien definidos.	Alta demanda de computo. Datos dispersos. Limitación solo a escenas bien definidas. Baja velocidad a la hora de adquirir los datos.
Tiempo de vuelo.	Gran capacidad para tomar medidas (rango). Alta velocidad, a la hora de adquirir los datos. Buen rendimiento general, independiente de la luz ambiental.	Costo. Precisión inferior a la triangulación.
Interferometría.	Alta precisión (en torno a los sub-micrón) en micro-intervalos.	La medición se limita a superficies cuasi planas. Poca aplicación a nivel industrial. Costo.

2 Escáneres 3D Open Source

Este capítulo se centrará principalmente en el estudio de diversos escáneres 3D de código abierto existentes en el mercado. Para ello se investigarán las distintas opciones y se indicarán las características correspondientes a cada modelo.

2.1 Escáneres disponibles

El día de hoy, se encuentra disponible un gran abanico de opciones para la elección de un escáner 3D, es por ello, que se necesita identificar las singularidades correspondientes a cada modelo en cuestión.

2.1.1 Spinscan

El año 2011, Tony Buser conocido por ser un genio en la impresión 3D crea el Spinscan, éste último es un escáner 3D que utiliza código abierto, el cual está basado en un láser y una cámara digital. Seguidamente, la empresa MakerBot utilizó las ideas de este prototipo de escáner para desarrollar un escáner 3D con un código cerrado. Gracias a este escáner, el cual aún está disponible, se han inspirado muchos de los escáneres 3D de la actualidad que emplean código abierto. El modelo de este tipo de escáner se representa en la figura 2-1, donde básicamente se tiene una cámara web en conjunto de un láser y un motor paso a paso el cual guía a una serie de engranajes que se mueven en la base (donde se coloca el objeto a escanear).

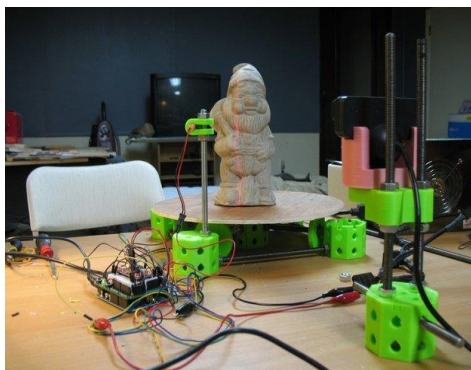


Figura 2-1: Escáner 3D Spinscan [6].

Para la figura 2-2 se puede observar los resultados dados por este escáner, los cuales no son muy buenos en términos generales, ya que si bien cumple el objetivo de escanear es posible ver la

pérdida de muchos puntos en la nube (cuando se fue realizando el escaneo) es por ello que la imagen se nota pérdida de información y las orejas del objeto no están completamente terminadas en el modelo digital.



Figura 2-2: Resultado del modelo de escáner 3D Spinscan [6].

Respecto al costo monetario de este prototipo, no existe mayor información en internet.

2.1.2 Fabscan

Hacia finales del año 2011, Fabscan comienza a concebirse con motivo de una tesis del alumno Francis Engelman en la universidad RWTH en Aachen (Alemania). En un inicio, Fabscan no utilizaba Raspberry Pi, pero con un código abierto que permitía que cualquier persona en el mundo pudiese fabricarlo. Cuando fue desarrollado este proyecto, se empleó una caja de madera, un motor paso a paso (stepper motor), un láser rojo de 5mW y una cámara web de resolución 720p. Además, se utilizaba un Arduino para controlar el motor paso a paso, tal cual como se muestra en la figura 2-3.

En términos generales, el primer prototipo utilizaba un plato giratorio, el cual era recorrido por un láser, así la cámara web supervisa el movimiento del láser y crea una nube de puntos de datos. Posteriormente estos datos son procesados para convertirlos en una malla 3D.

Sin embargo, muchas personas presentaban problemas con la compatibilidad del software del Fabscan con los diferentes sistemas operativos y configuraciones de hardware. Es por ello, que se integra el Raspberry Pi al proyecto, siendo reemplazada la cámara web por el módulo de cámara Pi. Esta nueva versión del FabScan es presentada en la figura 2-4.

En la figura 2-5, se muestra un resultado de este tipo de escáner, al igual que el anterior presenta pérdida de información (datos), lo que se traduce en modelos 3D digitales no completos.

En un inicio el Fabscan poseía un costo aproximado de 150 euros, mientras que la versión con Raspberry Pi está alrededor de 216 dólares. Ambas no están disponibles en Chile.



Figura 2-3: Primer prototipo del escáner 3D Fabscan [6].

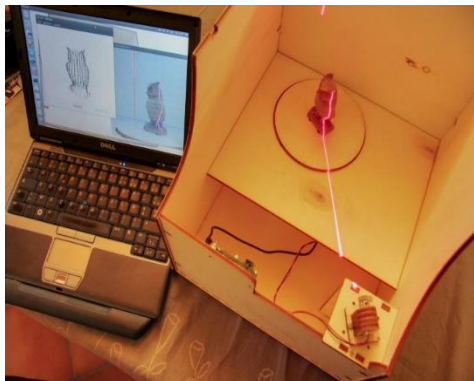


Figura 2-4: Segundo prototipo del escáner 3D Fabscan [6].

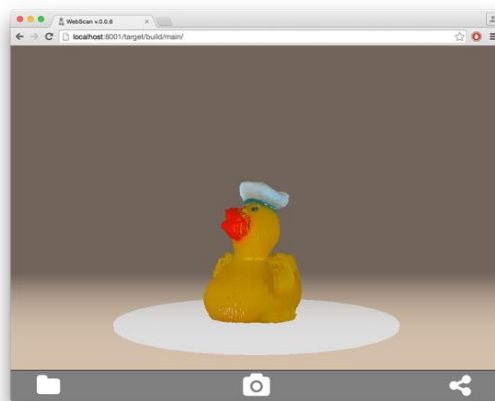


Figura 2-5: Resultado del modelo de escáner 3D Fabscan [6].

2.1.3 Atlas 3D

Atlas 3D surge a través de Uriah Liggett, quien es fundador de Murobo LLC. A fines del 2013, Liggett adquiere una impresora 3D para poder imprimir objetos cotidianos en su casa. Sin embargo, no encontraba los objetos (modelos) que necesitaba en internet. Esta última razón, resulta ser la motivación para que Liggett creará su propio escáner 3D para elaborar sus propios modelos. Dicho prototipo de escáner 3D se muestra en la figura 2-6 [7].

Este escáner utiliza un Raspberry Pi, en conjunto de un módulo de cámara para Raspberry. Además de usar 2 láser y un motor paso a paso.

Atlas 3D resultó ser un excelente escáner 3D en relación precio calidad, además utiliza un código abierto que permite a cualquier persona construirlo. Hasta el momento, Atlas 3D lleva 4 iteraciones de hardware y muchas de software (en ayuda a los usuarios de la web).

Los resultados de este prototipo fueron bastante mejores que los dos mencionados anteriormente, debido a que al utilizar dos láseres es menos probable tener problemas de oclusión, prueba de ello es la figura 2-7 y figura 2-8 las cuales muestran tanto el diseño 3D de un objeto escaneado por Atlas 3D como el objeto en cuestión.

El precio aproximado de este equipo es de 230 dólares (150.000 CLP) [7], dicho precio es solo en los componentes electrónicos, ya que el usuario debe imprimir las partes (en el precio se incluye los modelos para poder ser impresos en alguna impresora 3D). Este escáner no está disponible en Chile.



Figura 2-6: Prototipo del escáner Atlas 3D [7].



Figura 2-7: Resultado del modelo de escáner Atlas 3D [7].



Figura 2-8: Objeto original escaneado por Atlas 3D y replicación impresa en 3D [7].

2.1.4 Bq Ciclop

El escáner 3D Ciclop es un dispositivo que trabaja con código abierto y una plataforma rotativa. Este escáner ha sido desarrollado desde cero por la empresa BQ y fue lanzado con una licencia internacional “Creative Commons Attribution-ShareAlike 4.0” [8].

Este dispositivo está compuesto básicamente por una cámara web, dos láseres de línea, plataforma giratoria, motor paso a paso y una estructura impresa en una impresora 3D. Además, cabe señalar que este dispositivo utiliza un software llamado “Horus” el cual fue desarrollado desde cero al igual que el escáner. El diseño de este prototipo está mostrado en la figura 2-9.

Los resultados de este prototipo de escáner fueron bastante mejores a los primeros 2 escáneres mostrados en este capítulo, los resultados del Ciclop se muestran en la figura 2-10 y en la figura 2-11, este escáner entra a disputar lugar con el Atlas 3D, ya que ambos utilizan dos láseres de línea y además ambos obtienen muy buenos resultados. La diferencia entre el Ciclop y Atlas 3D radica en que la primera trabaja netamente con Arduino, mientras que el segundo lo realiza mediante un Raspberry Pi.

Este escáner tiene un precio aproximado de 349 dólares (227.000 CLP) [8]. Por dicho precio se envía el kit completo de Bq Ciclop, incluyendo las partes impresas por una impresora 3D. Dicho escáner no se encuentra disponible en Chile.



Figura 2-9: Prototipo del escáner 3D Ciclop [8].

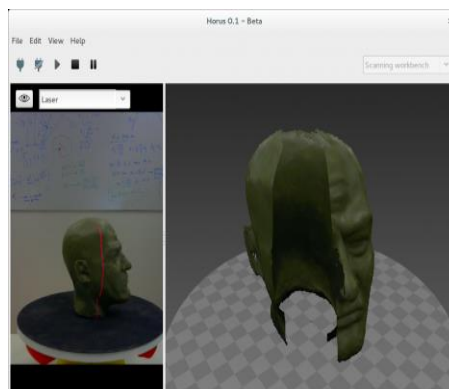


Figura 2-10: Resultado del modelo de escáner Ciclop [8].

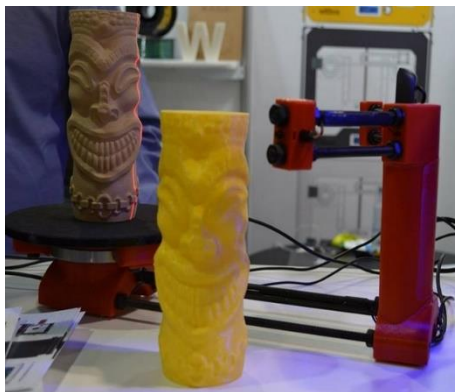


Figura 2-11: Objeto original escaneado por Ciclop y replicación impresa en 3D [8].

2.2 Estudio de los posibles componentes a utilizar

Como se ha visto en el capítulo anterior, básicamente existen dos plataformas de hardware para desarrollar este proyecto, las cuales son Raspberry y Arduino, éstas dos son presentadas en la figura 2-12. Raspberry Pi es una mini computadora íntegramente funcional, que posee un propósito general y que usualmente utiliza el sistema operativo Linux pudiendo ejecutar múltiples programas. Arduino es un microcontrolador que es una parte de una computadora, además puede ejecutar un programa a la vez, una y otra vez.

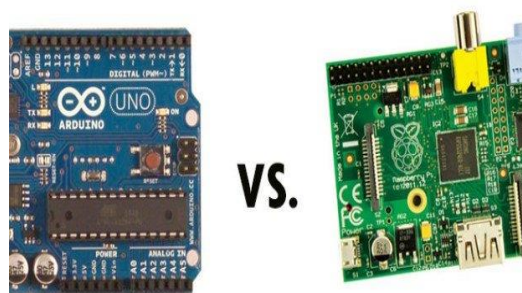


Figura 2-12: Placa Arduino y Raspberry Pi [9].

2.2.1 Arduino

Arduino es una plataforma que utiliza código abierto para construir diversos proyectos de electrónica. Este dispositivo es una placa de circuito integrado programable (microcontrolador) y además posee un software IDE (entorno de desarrollo integrado) el cual se ejecuta en un computador con el fin de escribir y cargar códigos desde el computador hacia el Arduino.

Este dispositivo, se ha vuelto muy popular a nivel mundial, debido a que muchas personas que recién se vienen integrando al mundo de la electrónica les resulta fácil aprender a programar porque utiliza una versión simplificada de C++. Una gran diferencia entre las versiones actualizadas de Arduino, con respecto a las anteriores es que actualmente no se requiere de una pieza de hardware externa para cargar un nuevo código en la placa (llamada programador), sino basta con solo utilizar el cable USB desde la placa Arduino al computador. Un ejemplo de ello es la figura 2-13 donde se muestra un Arduino conectado a una pantalla LCD, donde se le envía un mensaje vía el cable USB y es mostrado por dicha pantalla.

Tanto el hardware como el software de Arduino presta una valiosa ayuda para diversas áreas como diseño, arte, informática, entre otros. Para cualquier persona que desee crear objetos o entornos interactivos, Arduino resulta ser una buena herramienta. Este dispositivo es capaz de interactuar con leds, motores, parlantes, GPS, cámaras, botones, teléfonos inteligentes y hasta un televisor.

Arduino es económica y además de ser fácil de aprender el software y hardware. Ésta placa sea muy utilizada, teniendo una gran cantidad de usuarios que han contribuido al código (porque es código abierto) y, por lo tanto, existe posee una gran variedad de proyectos basados en este dispositivo.

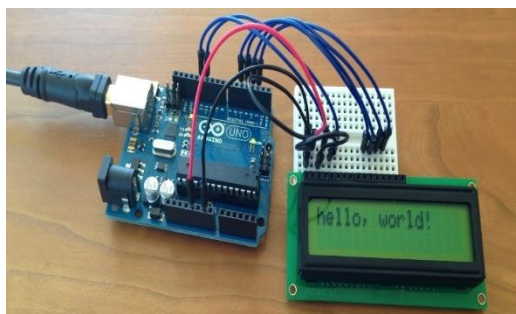


Figura 2-13: Placa Arduino y pantalla LCD [10].

2.2.1.1 Composición del arduino

En la actualidad, existen varias variaciones de Arduino, en la siguiente figura 2-14 se presenta un Arduino Uno, en el cual se señalará los componentes de la placa y que, a su vez, comparte con la mayoría de los diversos Arduino que posee el mercado.

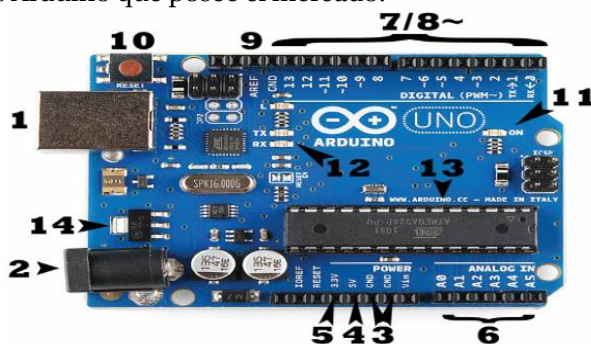


Figura 2-14: Placa Arduino identificada con sus principales componentes [10].

Alimentación (1 y 2):

- Los terminales 1 y 2 corresponden a la alimentación de la placa Arduino, esta placa puede funcionar alimentándose vía USB (1), la cual proviene desde el computador ó a su vez, puede alimentarse de una fuente de poder continua (2), donde esta varía dependiendo del modelo entre los 6 a 12V. Cabe señalar, que la función del USB (1) es enviar los códigos hacia la placa Arduino.

Pines GND, 5V, 3,3V, Analógico, Digital, PWM, AREF (3,4,5,6,7,8,9):

- Los pines 3, 4, 5,6,7,8,9 corresponden a los lugares donde se puede conectar cables para construir algún circuito deseado, poseen una cabeza de plástico de color negro que permite una fácil conexión entre un cable a otra placa que se utilice para realizar alguna función específica. En este contexto se especificará a continuación el detalle de cada componente:
- GND (3): este pin corresponde a la tierra (ground) que posee la placa Arduino.

- 5V (4) y 3.3V (5): Corresponden a las salidas que suministran 5V y 3.3V para poder alimentar los componentes que se pueden emplear en conjunto con un Arduino.
- Analógico (6): Aquí se tiene los pines desde A0 al A5, los cuales son pines de entrada analógica. Estos permiten leer una señal de un sensor analógico y transformarlo en un valor digital que puede ser leído.
- Digital (7): Estos pines van desde el 0 al 13 (para el Arduino UNO). Pueden ser utilizados para una entrada digital como para una salida digital.
- PWM (8): Los pines correspondientes a PWM tienen al costado del número el siguiente símbolo ~ (3,5,6,9,10 y 11 en este caso), estos poseen la función anterior (7) y a la vez se pueden usar para PWM.
- AREF (9): Representa la referencia analógica de la placa Arduino.

Botón reset (10):

- El botón de reseteo (10) al mantenerlo presionado reiniciará cualquier código cargado en el Arduino.

Indicador Led de encendido (11):

- En el costado de la palabra “UNO” de esta placa existe un led que dice “ON”. Este led debe estar encendido siempre que la placa Arduino esté conectada a una fuente de alimentación, ya sea el (1 ó 2). En caso de no encender, se hace necesario revisar el circuito.

Leds de transmisión y recepción (12):

- La placa Arduino muestra la sigla TX para identificar la transmisión, mientras que el RX es para la recepción, ambos son los encargados de la comunicación serial que posee la placa. Estos leds son útiles a la hora que se esté cargando un nuevo programa o código a la placa, ya que con estos se puede identificar visualmente.

Circuito integrado principal (13):

- Se podría afirmar que este es la especie de “cerebro” de la placa Arduino, este circuito integrado puede variar dependiendo del Arduino con el cual se esté trabajando (UNO, MEGA, RedBoard, entre otros), usualmente estos son fabricados por la compañía ATMEL y son pertenecientes a la línea ATmega. Se hace necesario conocer el modelo de circuito integrado, ya que se requiere para el software de Arduino.

Regulador de voltaje (14):

- Tal cual dice su nombre, este posee la función de regular la tensión de entrada al Arduino para evitar que se quemé, sin embargo, el voltaje máximo de entrada es de 20V.

2.2.2 Sensores

Actualmente, Arduino posee una gran cantidad de sensores con los que puede trabajar, tal como se muestran en la figura 2-15, basta con crear un código simple para poder controlarlos. Dentro de los sensores que puede controlar se encuentran:

- Sensor de luz.
- Sensor de temperatura.
- Sensor de proximidad.
- Sensor de monóxido de carbono.
- Sensor de aceleración.
- Sensor de grado de flexión.
- Sensor de humedad.
- Sensor de radioactividad.
- Sensor de presión barométrica.
- Otros.

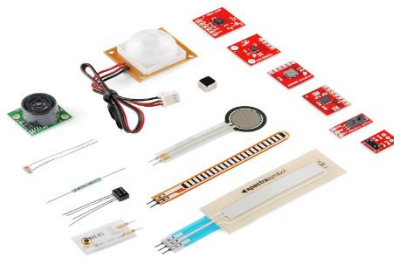


Figura 2-15: Muestra de algunos sensores disponibles en el mercado para Arduino [11].

2.2.3 Shields

Los shields o escudos (en español) hacen referencia a placas preconfeccionadas que son colocadas sobre la placa Arduino y que suministran nuevas funciones como, por ejemplo:

- Control de motores.
- Conexión a internet.
- Control de pantalla LCD.
- Comunicación inalámbrica.
- Entre otras.

Algunos ejemplos de dichas placas preconfeccionadas se muestran en la figura 2-16.

La comunicación entre el Arduino y las Shields puede ser a través de los pines digitales o analógicos. Además, puede ser conectado por puerta serial o I2C, SPI y también se puede emplear pines como interrupción. Usualmente, estos Shields son alimentados por los pines de 5V y GND.

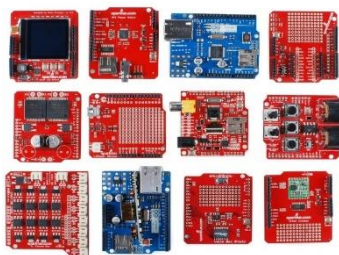


Figura 2-16: Muestra de algunas Shields disponibles en el mercado para Arduino [12].

2.2.4 Raspberry Pi

Raspberry Pi es básicamente un computador del tamaño de una tarjeta de crédito, el cual fue diseñado originalmente para la educación. Este dispositivo fue creado por Eben Upton y su principal inspiración para su creación fue el BBC Micro (comercializado en 1981), el principal objetivo del creador era que el Raspberry fuera una plataforma de bajo costo, que pudiera mejorar las habilidades de programación y comprensión del hardware en un nivel preuniversitario.

El tamaño y precio de este dispositivo indujo una rápida adquisición por parte de diversos fabricantes y aficionados a la electrónica para diversos proyectos. Sin embargo, este dispositivo no es una herramienta tan rápida como un computador de escritorio (actual), pero aun así a través de Linux puede proporcionar muchas herramientas con un bajo consumo energético.

En términos generales, este dispositivo es de hardware abierto (open hardware) exceptuando el chip principal de él, el Broadcom SoC (System on Chip). Este chip es el encargado de ejecutar muchos componentes como el CPU, gráficos, memoria, controlador USB, entre otros. Además de ser un hardware abierto, muchos proyectos realizados con el también son abiertos y están documentados para poder ser replicados y modificados por los usuarios a nivel mundial.

El Raspberry Pi, fue diseñado sobre el sistema operativo Linux y muchas variantes de este sistema operativo poseen una versión optimizada para utilizada en este dispositivo. Las opciones más utilizadas son Raspbian y Pidora, el primero está basado en sistema operativo Debian, mientras que el segundo en el Fedora. Otra opción es el software NOOBS (New Out Of Box Software) esta opción permite experimentar múltiples variantes de Linux.

Este dispositivo (Raspberry Pi) se muestra en la figura 2-17, donde se muestra realizando la misma tarea que en Arduino.

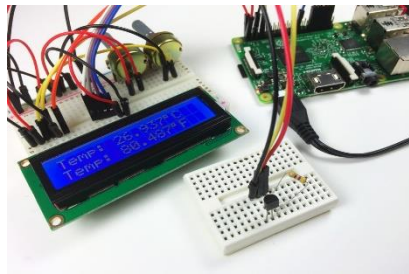


Figura 2-17: Placa Raspberry Pi y pantalla LCD [13].

2.2.4.1 Composición del Raspberry Pi

A continuación, se presenta un Raspberry Pi 2, para este modelo se mostrará un resumen de las funciones que posee cada parte de este dispositivo. Cabe señalar que existen 3 versiones disponibles para la compra, sin embargo, en esta ocasión se mostrará la “model B Pi 2” en la figura 2-18.

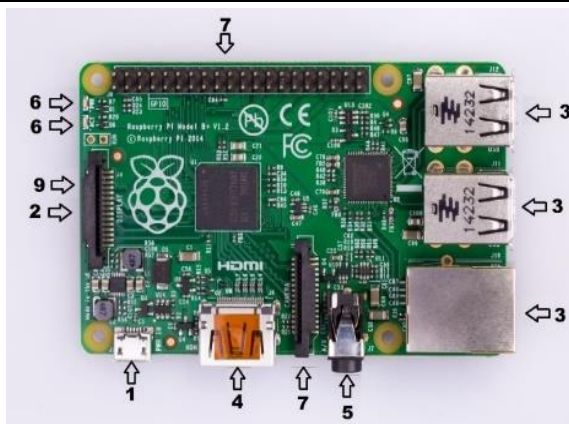


Figura 2-18: Placa Raspberry Pi 2 identificada con sus principales componentes [13].

Alimentación (1):

- Este terminal es indicado en la placa como PWR IN. Es la alimentación del Raspberry Pi, se alimenta por un cargador de 5V, el cual debe tener un conector micro-USB tipo B. La corriente varía de acuerdo con el modelo de Raspberry Pi, en el caso del modelo B es 700mA a 3.5W, mientras que el modelo A es 500mA a 2.5W. Un transformador compatible con este dispositivo se muestra en la figura 2-19.



Figura 2-19: Transformador para Raspberry Pi [14].

Almacenamiento (2):

- Este dispositivo cuenta con una ranura para microSD, donde este último es un dispositivo de almacenamiento extraíble de estado sólido. Esta tarjeta microSD es requerida para ejecutar los diversos sistemas operativos en el Raspberry Pi, esto es debido a que esta placa no posee almacenamiento de datos. Es recomendable el uso de una tarjeta clase 10, ya que ésta posee una mayor velocidad de datos por segundo. La ranura va colocada en la parte posterior a la figura mostrada en 2-6, es por ello que dicha ranura se muestra en la figura 2-20.



Figura 2-20: Ranura para tarjetas microSD en Raspberry Pi.

Puertos USB e internet (3):

- Este dispositivo cuenta con 4 puertos USB 2.0, estos puertos permiten tener conectividad con otros dispositivos como teclado, mouse, Wifi dongle, USB -Hub, entre otros.
- En el caso del puerto de internet, este permite conectar un cable RJ45 con soporte para CAT 5/6, así se puede habilitar la conexión a internet para obtener actualizaciones de software o instalar aplicaciones.
- Para ambos casos (USB y puerto de internet) se muestra en la figura 2-21 ambos.



Figura 2-21: Puerto de internet y USB en Raspberry Pi [15].

Puerto HDMI (4):

- El puerto HDMI o también conocido como “interfaz multimedia de alta definición”, permite que el Raspberry Pi pueda conectarse a un televisor o pantalla de alta resolución admitiendo una resolución máxima de 1920 x 1200. Un ejemplo de ello es la figura 2-22, en donde se muestra el Raspberry Pi conectado a una pantalla LCD mediante un cable HDMI.

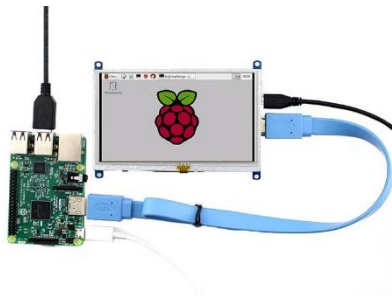


Figura 2-22: Pantalla LCD conectada mediante HDMI en Raspberry Pi [16].

Audio (5):

- El Raspberry Pi posee dos salidas de audio, una es el HDMI y el otro es el Jack (5), de forma que se puede intercambiar cualquiera de los dos en cualquier momento.
- En el caso de tener una pantalla conectada, el audio se emitirá por los parlantes de esta a través del HDMI, sin embargo, se puede emplear también el conector Jack de audio, en el caso de usar audífonos o parlantes.

Luces indicadoras (6):

- ACT: La función principal del led ACT es mostrar el estado de la tarjeta. Normalmente parpadea al existir actividad en la tarjeta microSD.
- PWR: La función de este led es indicar la alimentación. Cuando este led está encendido es porque la Raspberry Pi está encendida y se mantiene encendida hasta que se apague el dispositivo.

GPIO (7):

- El GPIO o “salida-entrada de propósito general” facilita la conexión de todo tipo de dispositivos periféricos al Raspberry Pi. Existen 40 pines, donde 26 son para entradas o salidas digitales, mientras que 9 de 14 pines GPIO son entradas-salidas dedicadas que facilitan el uso de SPI bus, I2C, entre otros. En la siguiente figura 2-23 se ilustran todos los pines y sus funciones.

Raspberry Pi 2 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1	2	5.0 VDC Power	
8	GPIO 8 SDA1 (I2C)	3	4	5.0 VDC Power	
9	GPIO 9 SCL1 (I2C)	5	6	Ground	
7	GPIO 7 GPCLK0	7	8	GPIO 15 TXD (UART)	15
	Ground	9	10	GPIO 16 RxD (UART)	16
0	GPIO 0	11	12	GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13	14	Ground	
3	GPIO 3	15	16	GPIO 4	4
	3.3 VDC Power	17	18	GPIO 5	5
12	GPIO 12 MOSI (SPI)	19	20	Ground	
13	GPIO 13 MISO (SPI)	21	22	GPIO 6	6
14	GPIO 14 SCLK (SPI)	23	24	GPIO 10 CE0 (SPI)	10
	Ground	25	26	GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27	28	SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29	30	Ground	
22	GPIO 22 GPCLK2	31	32	GPIO 26 PWM0	26
23	GPIO 23 PWM1	33	34	Ground	
24	GPIO 24 PCM_FS/PWM1	35	36	GPIO 27	27
25	GPIO 25	37	38	GPIO 28 PCM_DIN	28
	Ground	39	40	GPIO 29 PCM_DOUT	29

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Figura 2-23: Diagrama de GPIO en Raspberry Pi [17].

Conector de cámara CSI (8):

- El Raspberry Pi cuenta con un procesador industrial móvil de interfaz (MIPI) en conjunto con una cámara serial con interfaz tipo 2 (CSI-2). Donde éste último, facilita la conexión entre una pequeña cámara y el procesador Broadcom.
- El objetivo de la interfaz es estandarizar la conexión de los módulos de la cámara a los procesadores de la industria móvil. MIPI CS-2 admite 4 líneas de datos, donde cada carril posee 1 Gbps de ancho de banda.
- En la figura 2-24, se puede observar la conexión entre la cámara al Raspberry Pi.



Figura 2-24: Conexión entre módulo de cámara y Raspberry Pi [18].

Conector de pantalla DSI (9):

- El Raspberry Pi cuenta con una interfaz serie de pantalla (DSI) que permite conectar una pantalla LCD que posee un cable de 15 pin.
- Dicho conector puede ser observado en la figura 2-25.



Figura 2-25: Conector de pantalla DSI en Raspberry Pi.

2.2.5 Sensores

En la actualidad, Raspberry Pi posee una gran cantidad de sensores, de los cuales se pueden compartir desde Arduino a esta plataforma. Estos sensores son una gran ayuda a áreas como domótica, kits de robots, estaciones meteorológicas, entre otros.

Algunos de los sensores a nombrar son:

- Sensor de luz.
- Sensor de temperatura.
- Sensor de proximidad.
- Sensor de monóxido de carbono.
- Sensor de aceleración.
- Sensor de grado de flexión.
- Sensor de humedad.
- Sensor de radioactividad.
- Sensor de presión barométrica.
- Otros.

Algunos de estos sensores son presentados en la siguiente figura 2-26.



Figura 2-26: Ejemplo de algunos sensores disponibles para Raspberry Pi [19].

2.2.6 Motor paso a paso (Stepper motor)

El motor paso a paso o también conocido en inglés como “Stepper motor” son motores de corriente continua (CC) que se mueven a pasos discretos, es decir, gira bajo un determinado ángulo. En términos generales, este tipo de motores convierte los impulsos eléctricos en movimientos mecánicos (de giro) que son controlados, a tal punto que su desplazamiento puede variar entre 1.8° hasta 90° , convirtiéndose en un motor muy preciso.

A medida que los pulsos digitales del controlador incrementan en frecuencia, el movimiento escalonado que lleva el motor se convierte en una rotación continua que es directamente proporcional a la frecuencia de los pulsos enviados por el controlador.

Debido a la precisión que posee este tipo de motores, es que tiene muchas aplicaciones a la hora de trabajar en áreas de control, robótica, impresoras, tecnología aeroespacial, entre otras. Además, dentro de sus características, posee una alta confiabilidad y alto torque a bajas velocidades. Este tipo de motor es ilustrado en la figura 2-27 [20].

En términos generales, este tipo de motores son buenos en 3 ejes que son:

- **Posicionamiento:** Debido al movimiento que realizan (paso a paso) y de manera precisa, éstos sobresalen en aplicaciones de movimientos precisos como impresoras 3D, CNC, Plotters, además de que algunas unidades de discos también los emplean para colocar el cabezal de lectura/escritura.
- **Control de velocidad:** El incremento preciso de la rotación del motor provee de un excelente control de velocidad de rotación para la robótica y automatización de procesos.
- **Torque a baja velocidad:** Usualmente los motores normales de corriente continua (CC) no poseen mucho torque o par a bajas velocidades. Sin embargo, un motor paso a paso tiene un torque o par máximo cuando está a bajas velocidades, es por ello que es muy útil a la hora de trabajar a baja velocidad con alta precisión.



Figura 2-27: Ejemplo de motores paso a paso disponibles actualmente [20].

2.2.7 Motor servo

Un servomotor es un dispositivo eléctrico que puede empujar o girar objetos con precisión. Este tipo de motor son ocupados ampliamente y pueden encontrarse desde juguetes hasta electrodomésticos, autos y aviones. En el caso de modelos a escala de autos y aviones, estos motores controlan la dirección y ajustan las alas.

En la actualidad, los motores servos son utilizados en la industria automotriz, pudiendo controlar la velocidad de los automóviles, ya que el pedal del acelerador funciona similar al control de volumen de una radio, al presionar el acelerador se envía una señal eléctrica al computador del automóvil que indica cuanto fue la presión que se ejerció sobre éste. El computador del automóvil calcula esa información en conjunto con otra que es proporcionada por otros sensores del automóvil y se envía una señal al motor servo para ajustar la velocidad del motor.

Una de las características que posee este tipo de motores es su simpleza, por lo cual se hacen confiables. Si se desarma uno de estos motores, se verá que el corazón de este es un motor pequeño de corriente continua (CC), el cual es muy parecido a los típicos motores de juguetes que se encuentran disponibles en la mayoría de los locales de electrónica. Cabe destacar que estos motores giran a altas RPM (rotaciones por minuto), pero con un torque o par muy bajo (fuerza de torsión para realizar trabajo).

Este pequeño motor en el interior va acoplado a un sistema de engranajes semejante a una transmisión, la que tiene el fin de favorecer el torque. El tener dicho sistema permite que, al mover el eje del motor se perciba una inercia mayor a la de un motor común. En la siguiente figura 2-28 se presenta un servo motor abierto, mostrando sus componentes internos.

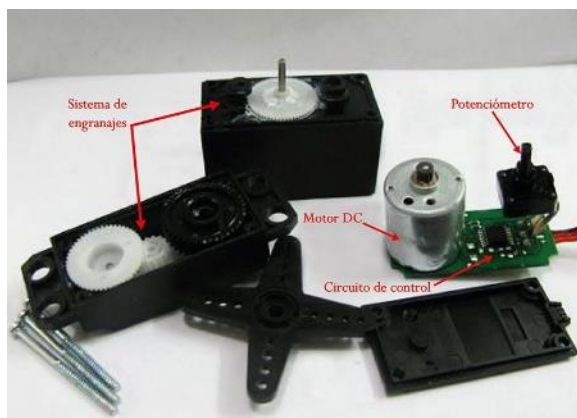


Figura 2-28: Sistema interno de un servomotor [21].

Para el correcto funcionamiento de este motor, se hace necesario tener una señal de control por ancho de pulso (PWM) para que el circuito de control trabaje apropiadamente. Para tener una mejor comprensión sobre el sistema interno de este motor, la figura 2-29 siguiente muestra en teoría como funciona.

DIAGRAMA DE BLOQUE DEL SERVOMOTOR

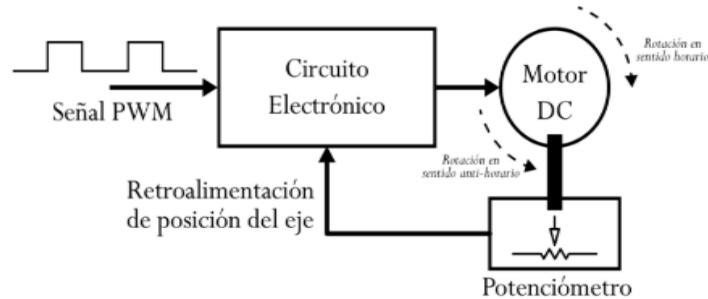


Figura 2-29: Sistema interno de un servomotor.

En diagrama de bloque de la figura 2-29 muestra el servo motor como un sistema. En dicho diagrama, se puede apreciar que para mover el motor DC, tiene que ser enviada una señal PWM, la cual es procesada por el circuito electrónico para mover el motor. El eje del motor tiene enlazado un potenciómetro, el cual forma un divisor de tensión. La tensión en la salida de dicho divisor cambia de acuerdo con la posición del eje del motor, así el circuito electrónico de control puede retroalimentarse con la posición exacta del motor.

Usualmente, las señales PWM son similares en los diversos modelos de servo motores. A continuación, se presenta la figura 2-30 la cual muestra una señal PWM y su correspondiente posición en función del ancho del pulso.

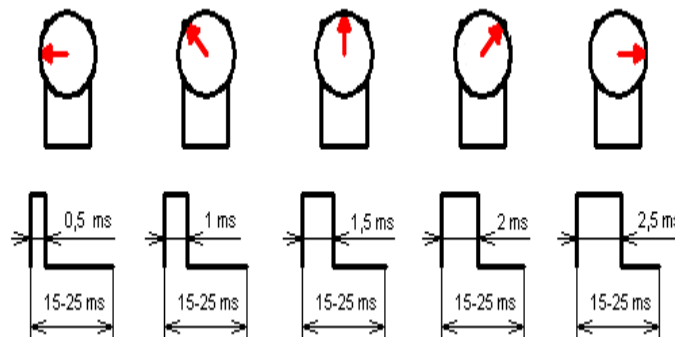


Figura 2-30: Pulsos PWM y su relación con la posición del motor.

Para poder mantener una rotación fija, las señales mostradas en la figura 2-30 deben repetirse en el tiempo. Para mantener dichas señales, puede utilizarse un circuito oscilador o también pueden ser por microcontrolador.

2.2.8 Láseres

Para un escáner 3D de triangulación es importante el uso de un láser de línea, ya que este tipo de láser le entrega al sensor CMOS una banda de puntos que recorren el objeto que se está escaneando. Dicho esto, existen principalmente 3 colores de láser de línea disponibles para emplear en el escáner los cuales son: rojo, azul, verde. La diferencia entre ellos será la aplicación o tipo de objetos en los cuales serán empleados.

Los láseres de color rojo poseen una longitud de onda entre los 630nm y 680nm, siendo un estándar en la industria, sin embargo, posee más distorsión que las otras opciones disponibles.

Este tipo de láser al tener una mayor longitud de onda en relación a los láseres azul y verde, funcionan de muy buena forma en superficies de baja reflectividad o mate y además en objetivos de rápido movimiento.

Los láseres azules tienen una longitud de onda que varía entre los 405nm y los 500nm, este tipo de láser se comporta muy bien a la hora de trabajar con objetos que sean muy brillantes o pulidos, dado su longitud de onda más corta resulta en niveles de ruido mucho menores que al utilizar láseres rojos. Sin embargo, dada la longitud de onda a la cual trabajan, resultan nocivos para los ojos y es más caro (económicamente) que las otras tres opciones.

Finalmente, los láseres verdes disponen de una longitud de onda de 532nm, estos láseres son una opción intermedia entre las dos anteriores (rojo y azul), pueden ser potencialmente dañino y a su vez, dependiendo de la potencia pueden llegar a quemar objetos. Posee un ruido nivel intermedio para objetos que poseen mucho brillo (en relación con los otros), sin embargo, su valor económico es mucho mayor a los de color rojo.

2.3 Estudio de plataformas Arduino y Raspberry Pi

Actualmente, es posible fabricar un escáner 3D por triangulación mediante de estas plataformas Arduino y Raspberry Pi; Ambas poseen tanto ventajas como desventajas y en este capítulo se analizarán y compararán para dar a conocer con cual se realizará este proyecto.

2.3.1 Raspberry Pi 3

Raspberry Pi 3 es un computador completamente funcional lanzado al mercado en febrero del 2016, este dispositivo posee el tamaño de una tarjeta de crédito, siendo desarrollado para la educación y es de bajo costo. Este dispositivo posee software de código libre u open source, donde el sistema operativo para el Raspberry es el Raspbian (una versión adaptada de Debian), no obstante, también permite la instalación de Windows 10.

En el caso de esta versión posee un procesador más rápido que las versiones anteriores, específicamente un chip ARM Cortex con una velocidad de 1.2 GHz y cuatro núcleos de 64 bits. Este dispositivo destaca respecto a las versiones anteriores debido a su capacidad para conectarse a redes Wifi y también con Bluetooth 4.1.

En la figura 2-31 se muestra una placa Raspberry Pi 3, es posible observar también que es muy similar a la Raspberry Pi 2 mostrada en el informe anterior.



Figura 2-31: Raspberry Pi 3 [13]

Tal como se observa en la figura 2-31, los componentes utilizados en este dispositivo son muy parecidos a su versión anterior (Raspberry Pi 2). Este posee su puerto de alimentación, almacenamiento por micro sd, puertos de internet y USB, HDMI, Jack de audio, GPIO (40 en total), conectores de pantalla y cámara.

Las características importantes de este modelo de Raspberry Pi, se resumen en la siguiente tabla 2-1:

Tabla 2-1: Características de Raspberry Pi 3.

Características	Valores
Modelo	Raspberry Pi 3
Procesador	BCM2837
Velocidad	1.2 GHz
Cantidad de núcleos	4
Memoria RAM	1 Gb
Cantidad de USB	4
Cantidad de pines	40
Compatibilidad accesorios con versiones anteriores	Sí
Voltaje de Entrada	5 V
Precio mercado (CLP)	34.000
Tamaño (mm)	85 x 56 x 17

2.3.2 Arduino R3

El Arduino es una placa de circuito integrado programable (microcontrolador) que posee un software IDE, esta plataforma está basada en ATmega 328 y sus proyectos están basados en código libre u open source. Esta versión utiliza una mejora para tener transferencias más rápidas y más memoria (se reemplaza el 8U2 por ATmega16U2 para el manejo USB) en relación con las versiones previas a esta. Una de las novedades de este modelo, es que uno de sus pines (IOREF) le permite a las Shields adaptarse al voltaje dado por el Arduino.

Este dispositivo posee una gran fama a nivel mundial, dado su lenguaje de programación (versión simplificada de C++), es realmente fácil de programar para los expertos como atractivo para los aficionados de la electrónica. En conjunto con una gran comunidad de soporte y además de un tremendo conjunto de bibliotecas de soporte.

En la figura 2-32 se puede apreciar la placa Arduino uno R3, la cual a diferencia de los modelos antecesores posee el botón de Reset cercano al USB, lo cual le da una ventaja al trabajar con una Shield (es más accesible).

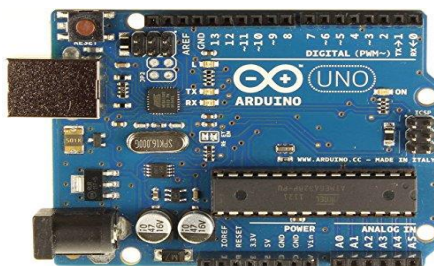


Figura 2-32: Arduino Uno R3 [10].

Respecto a la composición, posee un USB de alimentación y vez permite la conexión con el computador mediante el software Arduino IDE. En términos generales, esta placa puede utilizarse de dos maneras, la primera es recibiendo y ejecutando información a través del cable USB (llegando la información desde el computador). La segunda, es mediante una tarjeta microSD, donde se pueden cargar los datos para que este dispositivo realice las acciones que uno necesite.

Las características importantes de este modelo de Arduino se resumen en la siguiente tabla 2-2:

Tabla 2-2: Características de Arduino Uno R3.

Características	Valores
Modelo	Arduino Uno R3
Microcontrolador	ATmega328
Interfaz	USB
Memoria Flash	32KB
Velocidad de reloj	16MHz
Cantidad de pines	20
Cantidad de pines digitales	14
Cantidad de entradas análogas	6
Voltaje de Entrada	7-12V
Precio mercado (CLP)	10.000
Tamaño (mm)	68.6 x 53.4

2.3.3 Tabla comparativa

Tabla 2-3: Ventajas y desventajas de las plataformas.

Plataforma	Ventajas	Desventajas
Arduino Uno R3	<ul style="list-style-type: none"> - Muy fácil para iniciar. - Se puede emplear para aplicaciones en tiempo real para hardware, software y el IDE es de código abierto. - No se requiere de mucho conocimiento de programación para realizar cosas básicas. - Facilidad para añadir funciones nuevas, debido a sus escudos (Shields) y bibliotecas contribuidas por los usuarios. - Gran cantidad de escudos, convirtiéndolo en muy atractivo para proyectos. - De bajo costo. 	<ul style="list-style-type: none"> - No es poderoso, en relación con un Raspberry Pi. - Necesita programarse solo en Arduino o C/C++. - Conectarse a internet es algo difícil, pero no imposible.
Raspberry Pi 3	<ul style="list-style-type: none"> - Facilidad para acceder a internet. - La programación puede realizarse en una gran cantidad de lenguajes. - Disponibilidad de software gratuito (Linux): - Hardware con excelentes prestaciones en relación a su tamaño. 	<ul style="list-style-type: none"> - El acceso al hardware no es a tiempo real, en el caso de estar ocupado el CPU, puede existir retraso entre la interfaz y el hardware. - No posee suficiente potencia para abastecer cargas inductivas. - No posee un convertidor análogo- digital. - Hardware no es de código abierto, lo cual puede representar algún problema para ciertos usuarios. - Valor económico elevado.

2.3.4 Plataforma que desarrollará el proyecto

Dada la tabla comparativa anterior y para dar cumplimiento al objetivo principal de este proyecto, que es desarrollar un escáner 3D de bajo costo, se propone realizar este proyecto en base a Arduino, cuya plataforma en términos económicos es bastante accesible en relación al Raspberry Pi 3. Ambas plataformas cumplen la misma función y el desarrollo del escáner más que depender de la plataforma, depende de la iteración de hardware externo y software externo. Por esta razón, se puede escoger el Arduino por sobre el Raspberry Pi, ya que éste último resulta realmente útil a la hora de trabajar con software y paquetes de softwares complejos.

Además de lo expuesto anteriormente, en términos de programación resultará más fácil trabajar con Arduino, lo cual permitirá que en un futuro cercano este proyecto pueda ser modificado y mejorado más fácilmente por otro alumno que desee continuar en esto.

2.4 Shield BQ ZUM SCAN

El escudo o Shield BQ ZUM SCAN será el elegido para este proyecto, éste es mostrado en la figura 2-33 y permitirá controlar los elementos de potencia. Específicamente el motor paso a paso (mediante el uso de drivers) y los módulos laser.

El diseño de la Shield permite conectar 2 drivers con sus respectivos motores paso a paso, además de 4 salidas para módulos de láser y dos sensores analógicos. Los diseños de esta Shield se encuentran publicadas en GitHub, siendo liberados bajo la licencia Creative Commons Attribution Share-Alike [8].

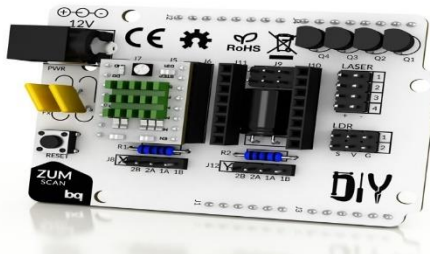


Figura 2-33: Shield BQ ZUM SCAN [8].

Las características importantes de esta Shield son resumidas en la tabla 2-4.

Tabla 2-4: Tabla resumen de la Shield Bq ZUM SCAN.

Características	Valores
Modelo	Bq ZUM SCAN
Cantidad de módulos láser	4
Cantidad de drivers	2
Cantidad de sensor analógico	2
Otros	Conector Jack.
Otros	Botón Reset
Protección Motores	Sí
Voltaje nominal	12V

Características	Valores
Voltaje de Entrada	8V-16V
Precio mercado (CLP)	10.000
Tamaño (mm)	67 x 54

*Nota: el precio señalado anteriormente corresponde a importar un kit desde el exterior, ya que en Chile no se encuentra disponible.

2.5 Drivers A4988

Debido a que la placa Arduino no puede alimentar directamente al motor paso a paso, se hace necesario el uso de drivers. El driver A4988 permite controlar el motor y posee protección para delimitar la corriente que circula por éste, además de sobre temperatura. Todo lo anterior, sumado a la capacidad de realizar micro pasos hasta 1/16 de paso, por ejemplo: Si se posee un motor que gira cada 1.8° grados, al implementar el driver se podrán realizar 16 micro pasos ($\frac{1}{16} * 1.8^\circ = 0.1125^\circ$), es decir, se puede dividir la cantidad de grados de giro original del motor y dividirlo en 16 partes, así se obtendrán 0.1125° de giro por cada señal enviada por el Arduino, donde el último indicará cuando se avance un paso.

En términos muy simples, se puede resumir el funcionamiento del driver a que el Arduino envía una señal y el driver permite el paso de corriente desde una fuente de alimentación a la bobina del motor, así queda separado la alimentación del Arduino y la de las bobinas del motor. Este driver va montado sobre la Shield y es mostrado en la figura 2-34.

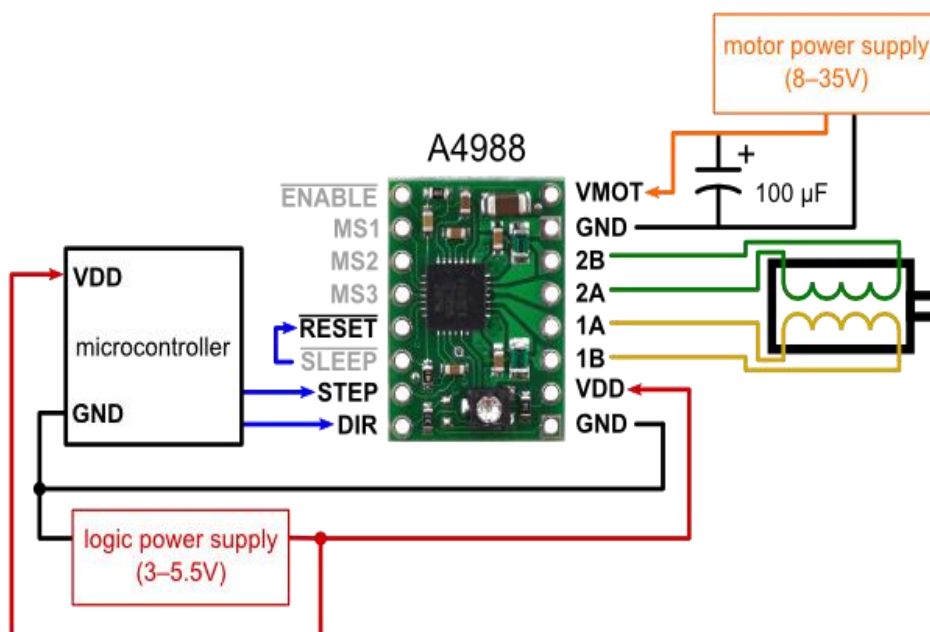


Figura 2-34: Driver A4988 [22].

Seguidamente, se presenta la tabla 2-5, la que resume las principales características de este driver.

Tabla 2-5: Tabla resumen de driver A4988.

Características	Valores
Modelo	A4988
Tensión del controlador	3V-5.5V
Alimentación del motor	8V-35V
Posibilidad de paso	1,1/2,1/4,1/8,1/16
Control corriente de salida	Sí
Protección	Corriente, Tensión y temperatura
Protección Motores	Sí
Precio mercado (CLP)	3.000
Tamaño (mm)	20 x 15.2 x 1

2.6 Motor Nema 17

La familia de motores Nema 17 es la más empleada a la hora de trabajar con diversos proyectos, tales como: CNC, impresoras 3D y escáneres 3D. El motivo por el cual poseen gran fama a nivel mundial es por su torque que va desde los 0.6 Kg/Cm hasta los 4.8 Kg/cm y además de su bajo costo.

El nema 17 es un motor paso a paso es un motor de corriente continua (CC) que se mueve a pasos discretos, es decir, gira un determinado ángulo. En términos generales, este tipo de motor convierte los impulsos eléctricos en movimientos mecánicos (de giro) que son controlados, en este caso el motor elegido recorre por paso 1.8°, es decir, para completar un giro se requieren 200 pasos. El motor específico a utilizar es el JK12HS40-1704-13 el cual es mostrado en la figura 2-35 y además se añade una tabla resumen de las principales características de este motor.



Figura 2-35: Motor Nema 17 JK12HS40-1704-13 [23].

Tabla 2-6: Tabla resumen de motor Nema 17 JK12HS40-1704-13.

Características	Valores
Modelo	JK12HS40-1704-13
Pasos	1.8°
Alimentación motor	12V
Cantidad de bobinas	2
Corriente por bobina	1.7A
Resistencia por bobina	1.5 Ω
Torque	4.2 Kg-cm
Temperatura de trabajo	-20°C a 50°C
Precio mercado (CLP)	9.000
Tamaño (mm)	42.3 x 42.3 x 66.3

*Nota: el precio señalado anteriormente corresponde a importar desde el exterior, ya que en Chile no se encuentra disponible.

2.7 Láseres

Los láseres de línea de color rojo son ampliamente ocupados en diversos proyectos existentes en la web. El principal motivo por el cual son muy utilizados es por el precio (es muy accesible) y también realizan menos daño físico a los objetos, ya que manejan una potencia mucho menor a los otros colores que existen en el mercado como lo son el azul y verde.

Es posible ver el módulo en la figura 2-36, este láser (HLM 1230) es ideal para proyecto, ya que genera una línea láser recta que puede enfocarse a una distancia determinada. Dentro de las características generales, este módulo posee una potencia de 5mW con una longitud de onda de salida a 650nm y es clasificado como un láser de clase IIIa, esto significa que si una persona mira directamente al láser es posible provocar un daño a la visión. Utilizar un láser de mayor potencia resulta peligroso, es más, la Agencia de Protección de la Salud del Reino Unido advierte que una potencia mayor a la de este láser es extremadamente peligroso y no deben ser aptos para la venta al público.



Figura 2-36: Módulo láser HLM 1230 [24].

A continuación, se exhibe la tabla 2-7, la cual muestra el resumen de las principales características de este láser.

Tabla 2-7: Tabla resumen del láser.

Características	Valores
Modelo	HLM-1230
Potencia	< 5mW
Longitud de onda	645-655 nm
Tensión de alimentación	3.5-4.5V
Corriente para operar	< 25 mA
Diámetro del rayo	0.75 mm
Divergencia	> 0.5 mRad
Temperatura de trabajo	-10°C a 50°C
Precio mercado (CLP)	1.700
Tamaño (mm)	12 x 30.6

*Nota: el precio señalado anteriormente corresponde a importar desde el exterior, ya que en Chile no se encuentra disponible.

2.8 Estudio de las cámaras

Es conocido que en los escáneres de triangulación 3D, las cámaras juegan un papel trascendental a la hora de poder obtener la nube de puntos, ya que esta es la encargada de ver en qué posición exacta del plano se encuentra el objeto para registrar los puntos en su propio plano (cámara) que se quiere analizar en cuestión.

Al trabajar con una plataforma de Arduino, se hace necesario elegir una cámara web (no se trabaja con módulos) y es aquí donde resaltan dos modelos de una misma marca “Logitech”, esta marca proporciona constantes actualizaciones de software, lo cual nos permite tener equipos no desactualizados y a su vez, son compatibles diversos sistemas operativos como: Windows, Mac Os, Android entre otros.

2.8.1 Logitech C270

La Logitech C270 es una cámara web de relativo bajo costo y de bastante uso a nivel mundial. Este dispositivo cuenta con resolución de 720P y con la tecnología Logitech Vid HD que le permite obtener videos más fluidos e imágenes más nítidas, con colores más vivos. Otra de las tecnologías que resalta en esta cámara es la Logitech RightLight que permite ajustar automáticamente la cámara bajo condiciones de mala iluminación para así obtener una buena imagen. En la figura 2-37 es posible observar esta cámara.



Figura 2-37: Cámara Logitech C270 [25].

A continuación, se presentará una tabla resumen, con los factores más importantes de este dispositivo en la tabla 2-8.

Tabla 2-8: Características de Logitech C270.

Características	Valores
Modelo	Logitech C270
Resolución video	HD (1280x720 pixeles)
Fotos	Hasta 3 MPX (por software)
Micrófono integrado	Sí
Sistema operativo	Windows, MacOS, Chrome Os
Peso (gr)	75
Tamaño (mm)	71 x 31 x 24
Precio mercado (CLP)	22.990

2.8.2 Logitech C920

La Logitech C920 se perfila como una cámara de gama alta, la cual posee video Full HD a 1080p, siendo mucho más rápido y fluido que sus competidores. Al igual que la C270, posee la tecnología Logitech Fluid Crystal que le permite obtener videos fluidos con imágenes nítidas y colores vivos.

Una de las cualidades que destaca en este modelo, es el uso del formato H.264 para grabar en Full HD, esta es una tecnología de compresión que posibilita acelerar las cargas y emplear menos recursos en los computadores, ya que elimina los largos procesos de compresión.

Esta cámara incorpora al igual que la C270 el sistema de corrección de iluminación automática, permitiendo grabar y tomar fotos en condiciones de iluminación escasa. A continuación, es posible observar en la figura 2-38 esta cámara y además en la tabla 2-9 se resumen las características más importantes de este modelo.



Figura 2-38: Cámara Logitech C920 [26].

Tabla 2-9: Características de Logitech C920.

Características	Valores
Modelo	Logitech C920
Resolución video	FULL HD (1920x1080 píxeles)
Fotos	Hasta 15 MPX (por software)
Micrófono integrado	Sí
Sistema operativo	Windows, MacOS, Chrome OS
Peso (gr)	162
Tamaño (mm)	94 x 24 x 29
Precio mercado (CLP)	56.990

2.8.3 Cámara a elegir

De esta comparación y a pesar de que Logitech C920 es una excelente cámara con muy buenas prestaciones (gama alta) en parte debido a su lente Carl Zeiss, se opta por la C270. El precio en el mercado para la Logitech C920 es demasiado excesivo y estaría en riesgo de perder uno de los objetivos planteados para este proyecto.

A pesar de ello, ambas cámaras en teoría pueden funcionar y obtener buenos resultados para el desarrollo y elaboración del proyecto.

La cámara elegida Logitech C270 es bastante popular a nivel mundial, dado sus prestaciones y su bajo costo respecto a la competencia. Además, Logitech constantemente realiza actualizaciones de software para sus modelos.

2.9 Rodamientos

El uso de rodamientos acero permitirán un movimiento con menor roce para mantener un suave deslizamiento de la base que será rotada por el motor nema 17. Las principales características de estos elementos se resumen en la tabla 2-10, a su vez, pueden ser observados respectivamente en las figuras 2-39.

Tabla 2-10: Características de rodamientos.

Características	Valores
Modelo	623zz
Diámetro (mm)	10
Altura (mm)	4
Movimiento	Lineal
Material	Acero Inoxidable
Precio mercado (CLP)	3.890 (5 unidades)



Figura 2-39 Rodamiento 623zz [27].

2.10 Estudio del software

Si bien es importante los componentes para la construcción de este proyecto, otro aspecto que es necesario estudiar es el software para el correcto funcionamiento de este prototipo. Sin el correcto apoyo del software no se podrá calibrar este escáner 3D y sin este punto es imposible obtener buenos modelos 3D para posteriormente imprimirlos.

2.10.1 Horus

Horus es un software desarrollado por la empresa BQ que permite el control para los escáneres fabricados por la empresa BQ (el Ciclop) y los escáneres fabricados por usuarios que no posean algún fin comercial, es por ello que este software es de código libre. Este software a través de su interfaz gráfica permite realizar diversas tareas como: calibración, captura y sincronización de datos, procesamiento de imagen, generación y visualización de la nube de puntos.

Otra ventaja que posee el Horus, es ser un software multiplataforma lo cual permite el uso de él en Linux/GNU, Ubuntu, Windows y soporte para MacOS, por lo tanto, abarca un gran espectro de los usuarios a nivel mundial, ya que la mayoría utiliza algunos de estos sistemas operativos y con ello podrían trabajar sin mayores problemas de compatibilidad.

Este software como se mencionó anteriormente es de código libre y está publicado en GitHub bajo la licencia GPL v2 [8].

Este software posee una interfaz relativamente simple tal como se muestra en la figura 2-40 y además permite a los usuarios al inicio del programa acceso rápido a diversas funciones como: control avanzado, calibración avanzada, escanear utilizando la última configuración y un modo “Wizard mode” (realiza paso a paso los procesos).

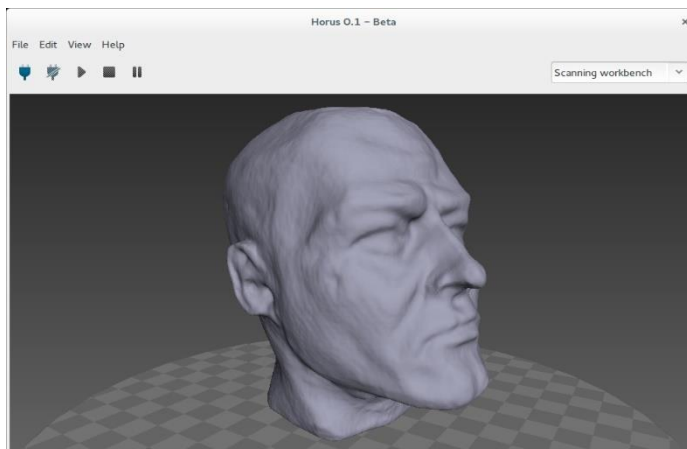


Figura 2-40: Software Horus [8].

El Wizard mode, permite realizar los procesos de conexión, calibración y sintonización de parámetros de forma muy simplificada, lo cual es una gran ayuda para los usuarios que no poseen mayor conocimiento técnico respecto al escáner.

En el control avanzado es posible tener un control total de los elementos que conforman el escáner, los cuales son el motor, láseres y cámara. Este modo permite ajustar manualmente diversos parámetros como: brillo, contraste, saturación, exposición, framerate, resolución, distorsión.

La calibración avanzada faculta al usuario para cambiar los parámetros de la cámara y poder así tener una calibración óptima. Existen parámetros importantes para calibrar en este punto como: calibración entre cámara y módulos laser, calibración entre cámara y plataforma, calibración entre el sensor de la cámara y el lente. Dentro de esto, la calibración de la cámara hace referencia a adquirir los parámetros de la cámara como: distancia focal, centro óptico y distorsión del lente. Mientras que la calibración de los láseres se puede obtener los planos de cada láser y finalmente la calibración del disco, la cual permite obtener la posición y rotación de éste.

3 Proceso de diseño

Este capítulo se centrará en el diseño para cumplir con los objetivos del proyecto.

3.1 Propuesta de diseño

Para el diseño del escáner fue necesario en un inicio obtener todas las hojas de especificaciones de los componentes electrónicos a utilizar para poder crear un modelo real en base a los tamaños de los componentes. Básicamente este diseño utiliza los componentes mencionados en los capítulos 1 y 2 de este informe, además para ofrecer una estructura más robusta, se elige un perfil de aluminio cuadrado de 20x 20 mm con un largo de 400 mm y al interior de este se colocarán los cables del motor Nema 17, así evitando cualquier posible daño accidental que pueda ocurrirle al motor durante el armado o cuando se encuentre trabajando (a nivel de cables), tal como se muestra en la figura 3-1.

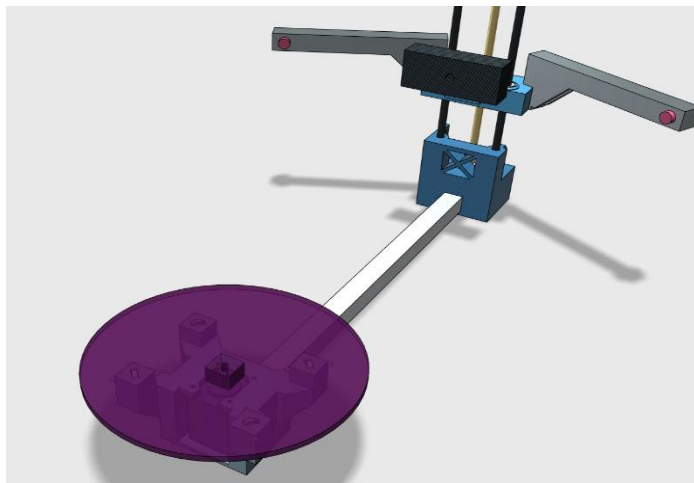


Figura 3-1: Diseño propuesto para el escáner 3D.

Cabe destacar que en este diseño se optó por variar manualmente la altura del conjunto cámara-láseres, con motivo de no limitar al usuario respecto a los tamaños de los objetos que desea analizar, para lograr esto se ideó que los brazos que sostienen los láseres sean independientes y unidos mediante encaje. El beneficio de tener un sistema así, es que en caso de fatiga o fractura de material es fácilmente reemplazable por una pieza equivalente.

En la figura 3-2 es posible observar el sistema de elevación para este prototipo, este diseño consta de 3 pilares que son 1 husillo de 8 mm en conjunto de dos barras lisas de 8 mm, el husillo permitirá desplazar el conjunto cámara-láseres en la altura que el usuario desee, así brindando una capacidad variable de escaneado, la cual será limitada por el largo del husillo (300 mm) y a su vez, ayudada por las barras lisas que al tener un bajo coeficiente de roce y además de tener rodamientos permitirá un suave deslizamiento hacia donde el usuario desee.

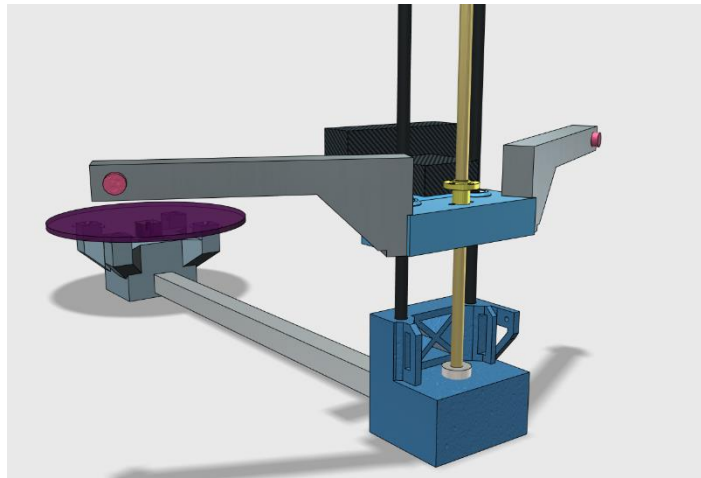


Figura 3-2: Sistema de elevación del prototipo.

En la actualidad, la mayoría de los escáneres 3D utilizan sistemas de rodamientos grandes, que poseen gran peso y son de alto costo. Para eliminar dichos problemas, se opta por 4 rodamientos más pequeños, pero igual de eficientes y a su vez, permiten reemplazarlos fácilmente en el mercado, ya que son muy utilizados a nivel mundial. Este sistema es posible observarlo en la figura 3-3 y 3-4.

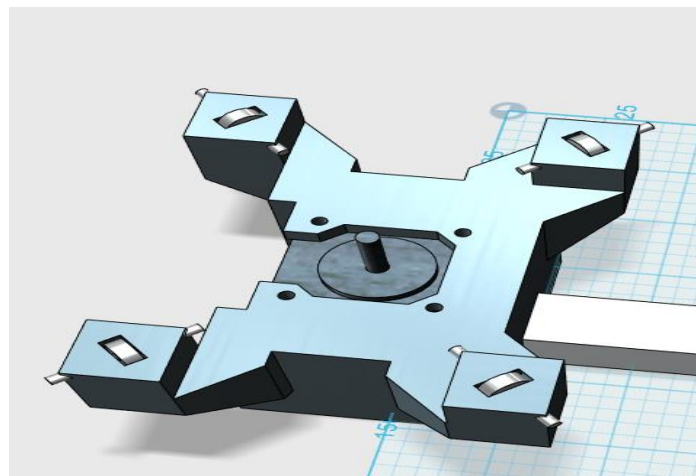


Figura 3-3: Sistema de rodamientos en la base.

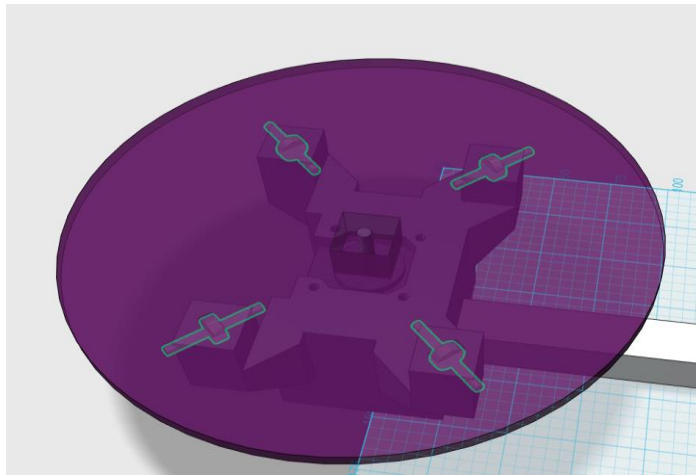


Figura 3-4: Sistema de rodamientos en la base con la base.

Finalmente, la posición de los láseres fue desarrollada y vista mediante software, se debe realizar pruebas con los modelos, con el fin de verificar el correcto funcionamiento de las partes con el software Horus, respecto al ajuste del sistema se puede visualizar en las figuras 3-5, 3-6.

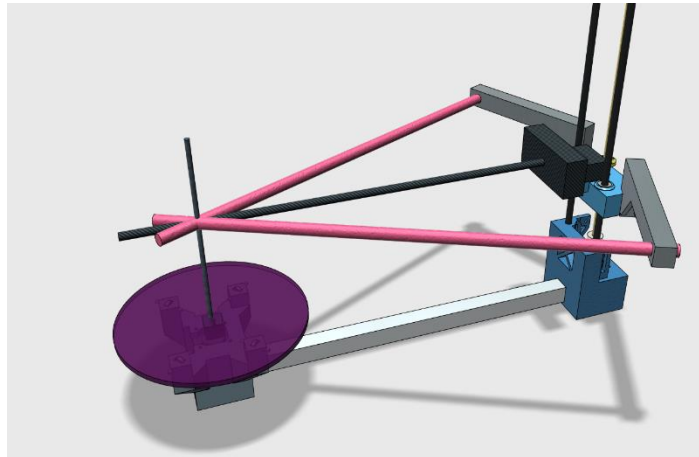


Figura 3-5: Comprobación del sistema.

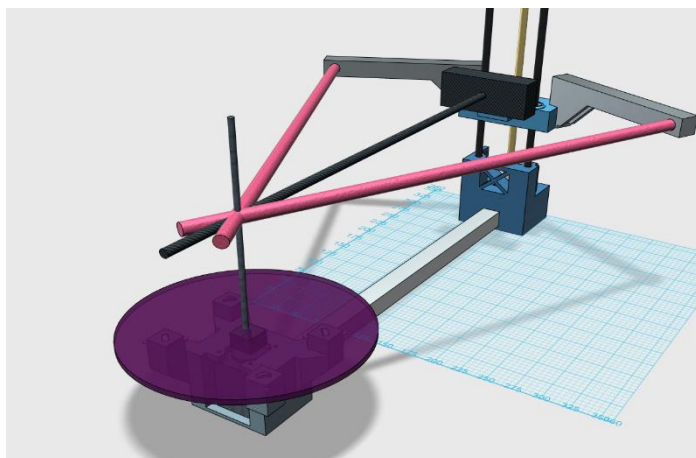


Figura 3-6: Verificación del sistema.

3.2 Cambios en el prototipo

Luego de realizar la propuesta de diseño, se procede a la impresión de este prototipo donde se encontraron diversas falencias en el diseño, con las cuales se debieron realizar cambios a nivel estructural, dado los problemas encontrados.

Uno de los primeros cambios a realizar es la ubicación de la cámara, para así obtener un correcto funcionamiento en el escáner. Si bien el software es bastante flexible en cuanto a parámetros físicos de los componentes electrónicos, éste fue diseñado para el uso de una cámara en posición vertical. El diseño anterior, el escáner se ve como se muestra en la figura 3-7.

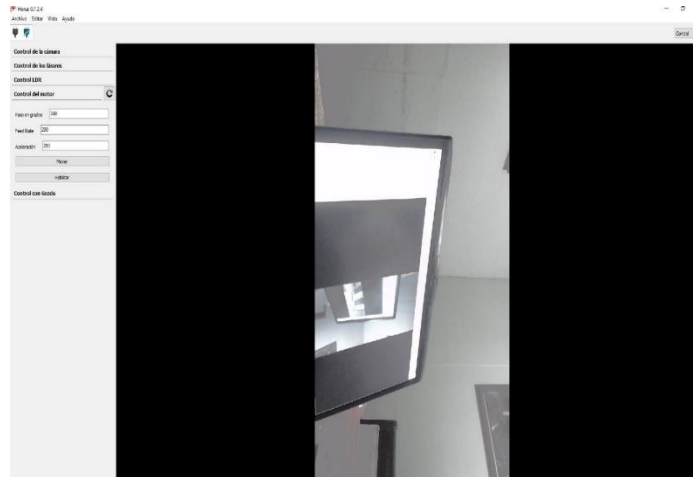


Figura 3-7: Cámara con posición horizontal.

Con el fin de arreglar el inconveniente de diseño, se tuvo que realizar un prototipo que considerara esta variable. Dado que la primera propuesta fue fabricada con brazos completamente impresos, se encontró una debilidad en los materiales. En términos generales, cualquier caída del escáner produce un quiebre en estos, lo cual se pretende evitar a toda costa, por esta razón se procede a cambiar estos brazos por un perfil de aluminio 2020, el cual le brinda solidez y robustez a la solución (lo que es deseado para el usuario final), estos cambios son mostrados en la figura 3-8 y 3-9, respectivamente.

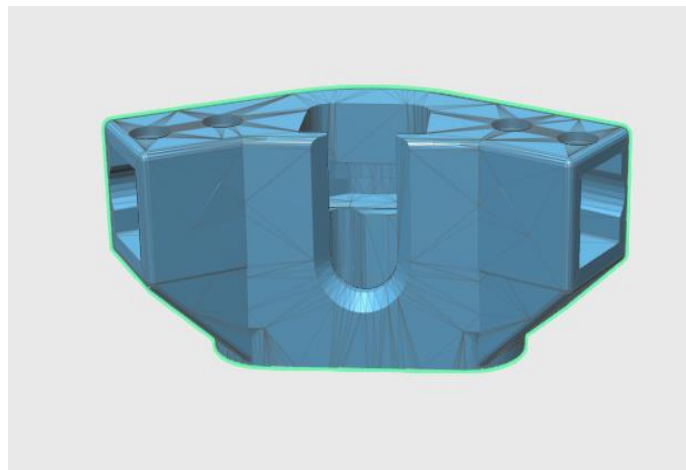


Figura 3-8: Cámara en vertical.

Además, luego de realizar diversas pruebas con las partes ya impresas

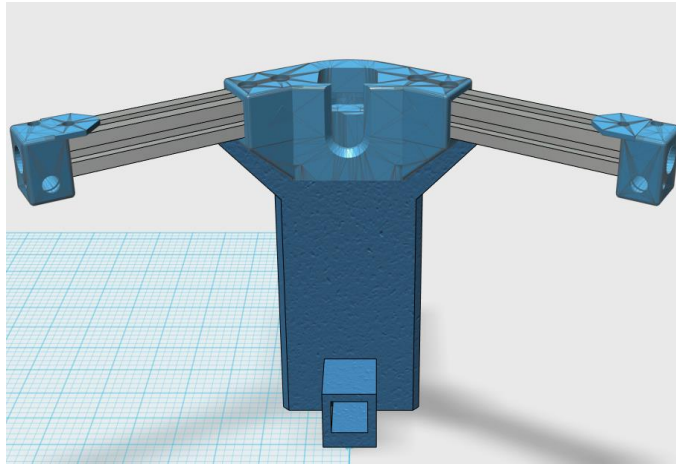


Figura 3-9: Nuevo diseño base escáner 3D.

Como se muestra en la figura 3-8, se corrige la posición de la cámara y a su vez, se modifica el centro del escáner, ya que el software “Horus” no permite modificar los parámetros intrínsecos desde el software, si no a través del código que está cargado en el arduino, por lo tanto sería muy poco práctico a la hora de utilizar el escáner, dado que deberían introducirse los parámetros cada vez que se quisiera utilizar. Por la razón mencionada anteriormente, se modifica el diseño original propuesto y se deja una base fija, con la cual no será necesario cambiar los parámetros cargados inicialmente al Arduino.

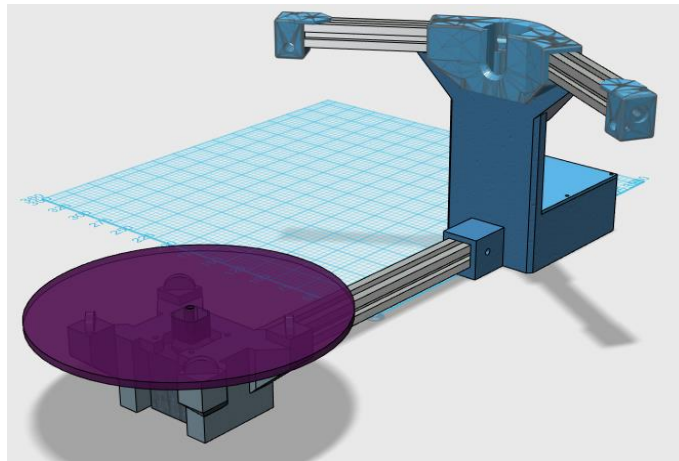


Figura 3-10: Diseño final escáner 3D.

En la figura 3-10, se puede visualizar el aspecto que tendrá finalmente el escáner, luego de realizar las pruebas de rigor al prototipo anteriormente mencionado, se exhibe el uso del perfil de aluminio 2020, tanto para los brazos que sostendrán los láseres del escáner como el soporte que sostiene y une la base donde está posicionado el objeto a analizar con la base donde está posicionada la cámara.

4 Calibración, Horus y Diseños obtenidos.

A continuación, se exhibirá desde el proceso de calibración del driver A4988, el cual permite la interacción entre el Arduino y motor, mostrando paso a paso los procesos necesarios para poner en marcha correctamente el motor sin inducir daños en él. Además, de la integración de la electrónica con el software Horus, lo cual conllevará a obtener los primeros diseños del prototipo de escáner 3D.

4.1 Procedimiento para calibrar

Antes de enumerar el procedimiento, se hace necesario entender lo que hace cada entrada que disponible el driver A4988. En la figura 4-1, se muestra un diagrama simplificado del driver en el cual se muestran los pines.

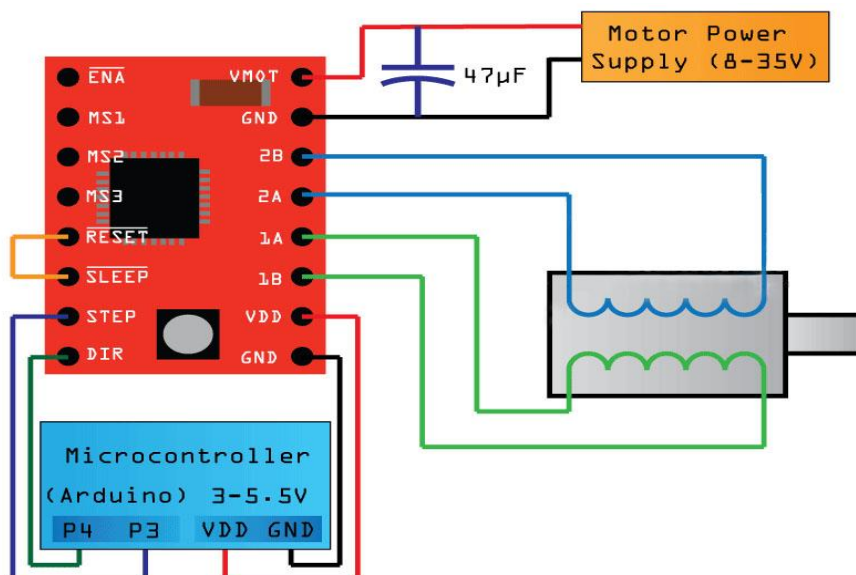


Figura 4-1: Diagrama simplificado del driver [22].

Como se muestra en la figura 4-1, se tienen básicamente dos tensiones en el driver, la primera es tensión de control de la placa (la que varía entre 3 a 5.5V para VDD) y es entregada por el Arduino. La segunda tensión es la correspondiente a VMOT la cual varía entre 8 a 35V y es la tensión que necesita el motor para funcionar, en este caso se usa 12V.

Para el caso de los pines 2B, 2A, 1A, 1B son para las dos bobinas correspondientes del motor paso a paso.

Para el costado izquierdo del driver A4988 tenemos el DIR y STEP, ambos son controlados en este caso por el Arduino y cumplen la función dar el sentido de giro y el segundo (STEP) controla los pasos, dependiendo de la cantidad de pulsos que se entreguen la posición del motor irá de un lado a otro (en este caso para completar una vuelta se necesitan 200 pasos).

En el caso de los pines RESET y SLEEP, el primero posee la función de activar o desactivar la función servo del motor, mientras que el segundo no será utilizado. Por otro lado, los pines MS1, MS2, MS3 son los encargados de realizar los pasos para el motor, lo cual se explicará más adelante.

Finalmente, el pin ENA posee la función de habilitar o deshabilitar externamente el driver, permitiendo así poseer un control externo de este.

Como se menciona anteriormente, los pines MS1, MS2 y MS3 son los encargados de controlar los pasos, para entender mejor como funciona se hace necesario observar el datasheet correspondiente al driver A4988, tal como se muestra en la tabla 4-1.

Tabla 4-1: Control de los pasos.

Table 1: Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

En este caso, como se explicó en la mesa número 3 se trabajará a paso completo o “Full Step”. Ya que, de esta forma se tiene el máximo par del motor. Tal como se muestra en la figura, los estados son L (“low”) para los pines MS1, MS2 y MS3, de esta forma no hay que conectar nada en estos pines para mantenerlos en ese estado, así se completará con 200 pasos una vuelta completa (recordando que el motor tiene un paso cada 1.8°). Por ejemplo, en caso de querer un octavo de paso, los estados para MS1 y MS2 son High y MS3 Low, para completar esto se necesitará 1600 pasos y entregar 5V en los pines MS1 y MS2 para hacer el estado “High” o alto.

A continuación, se necesita identificar el voltaje de referencia que tendrá el driver. Para ello, nuevamente se hace necesario recurrir al datasheet del equipo, en donde el fabricante entrega la siguiente fórmula para calcular el voltaje de referencia:

$$I_{TripMAX} = V_{REF} / (8 * R_S)$$

En donde el parámetro R_S corresponde a la resistencia de sensibilidad, las cuales están mostradas en la figura 4-2:



Figura 4-2: Driver A4988 con Resistencias de sensibilidad destacadas [22].

Como se logra apreciar en la figura 4.2 es posible encontrar las resistencias de sensibilidad (R_{100}), las cuales son resistencias en “chip” para montaje superficial, para calcular el valor de este resistor es posible utilizar calculadoras online como se muestra en la figura 4-3.

SMD resistor code calculator

Figura 4-3: Calculadora de resistores SMD [28].

De la formula anteriormente mencionada, se desprende:

$$V_{REF} = I_{TripMAX} * 8 * R_S$$

Donde, $I_{TripMAX}$ corresponde a la corriente máxima que puede circular por el motor. En este caso y según el datasheet es 1.7 A, luego la tensión de referencia será:

$$V_{REF} = 1.7 * 8 * 0.1 = 1.36 V$$

Luego según el fabricante, para tener paso completo se necesita aproximadamente un 70% de la tensión de referencia. Esto se muestra en la siguiente tabla 4.2

Tabla 4-2: Relación paso completo y corriente en las fases (%)

Full Step #	Half Step #	1/4 Step #	1/8 Step #	1/16 Step #	Phase 1 Current [% $I_{tripMax}$] (%)	Phase 2 Current [% $I_{tripMax}$] (%)	Step Angle (°)
	1	1	1	1	100.00	0.00	0.0
				2	99.52	9.80	5.6
			2	3	98.08	19.51	11.3
				4	95.69	29.03	16.9
		2	3	5	92.39	38.27	22.5
				6	88.19	47.14	28.1
			4	7	83.15	55.56	33.8
				8	77.30	63.44	39.4
1	2	3	5	9	70.71	70.71	45.0

Al revisar la tabla anterior, se muestra que, para tener un paso completo, es necesario tener tanto en la fase 1 como en la 2 un 70.71%.

Luego, la tensión de referencia para realizar un paso completo será:

$$V_{REF} = 1.36 * 0.7 = 0.952 V$$

Siendo dicho valor, el que se tiene que ajustar en el driver para realizar un paso completo.

Finalmente, para obtener dicho valor se conecta el sistema sin el motor conectado al driver. El pin correspondiente a GND del motor es conectado al terminal de un multímetro. Siendo, el negativo o "COM" (en el multímetro) y el positivo del multímetro es conectado a un destornillador (el cual ajusta el potenciómetro del driver) logrando así el voltaje de referencia correspondiente.

El software Horus es una potente herramienta que posee diversas funciones, donde se destaca:

- El firmware ha sido utilizado en diversos proyectos, incluyendo Marlin 1.1 (firmware usado en las impresoras 3D de código libre).
- Permite encender y apagar los láseres.
- Permite el control de la posición, velocidad y aceleración del motor paso a paso que es controlado mediante comandos Gcode.

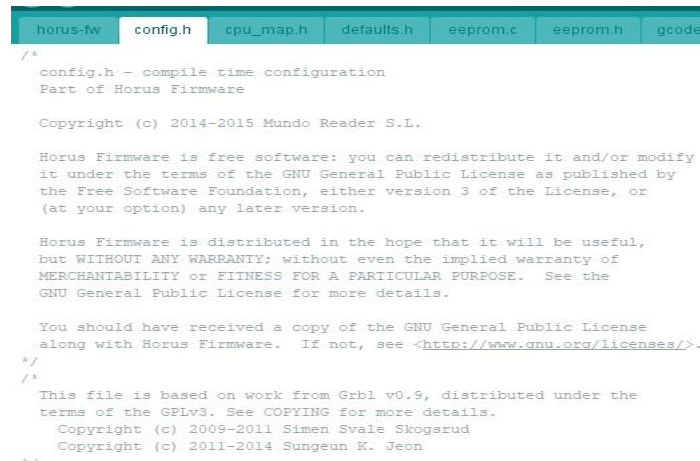
Como se menciona anteriormente es muy poderoso, a tal punto que es utilizado por otros proyectos en paralelo como es el caso de las impresoras 3D. Por esta razón es importante realizar un estudio a este firmware, ya que son muchas las funciones con las que se pueden trabajar y es necesario entenderlo.

4.2 Configuración de parámetros.

Unas de las primeras acciones a realizar es instalar en el computador el software de Arduino, para ello es necesario acceder al sitio web: <https://www.arduino.cc/en/Main/Software>. En ese sitio, se selecciona la versión de sistema operativo y posteriormente se descargará e instalará en el computador.

Seguidamente, se conectará la Shield al Arduino y se debe corroborar en que puerto COM es reconocida la placa en el computador, para así comenzar con la asociación entre el Arduino y el computador.

Una vez conseguido lo anterior, es necesario abrir el archivo: “horus-fw.ino.”. Dentro de este archivo se encontrarán la configuración de la mayoría de los parámetros empleados para hacer funcionar el escáner 3D, tal como se muestra en la figura 4-3.



```

horus-fw  config.h  cpu_map.h  defaults.h  eeprom.c  eeprom.h  gcode
/*
 * config.h - compile time configuration
 * Part of Horus Firmware
 *
 * Copyright (c) 2014-2015 Mundo Reader S.L.
 *
 * Horus Firmware is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Horus Firmware is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Horus Firmware. If not, see <http://www.gnu.org/licenses/>.
 */
/*
 * This file is based on work from Grbl v0.9, distributed under the
 * terms of the GPLv3. See COPYING for more details.
 * Copyright (c) 2009-2011 Simen Svale Skogsrud
 * Copyright (c) 2011-2014 Sungeun K. Jeon
 */

```

Figura 4-3: Software Arduino y firmware.

Uno de los parámetros importantes a considerar es la aceleración con la cual reaccionará el motor a la hora de realizar movimientos, el parámetro se denomina: `#define ACCELERATION_TICK_PER_SECOND 100`, tal como se deja comentado en el código, éste afecta directamente en la suavidad de la aceleración (que tan brusca es). En este sentido, es necesario considerar el factor de la inercia, cuando el motor stepper está en estado estacionario costará hacer girar el motor (pasar del estado de reposo a movimiento), así resulta más fácil realizar una aceleración progresiva hasta llega al valor de velocidad deseado, de igual forma con el frenado del motor, tal como se muestra en la figura 4-4, donde se hace la relación entre la velocidad en el tiempo.

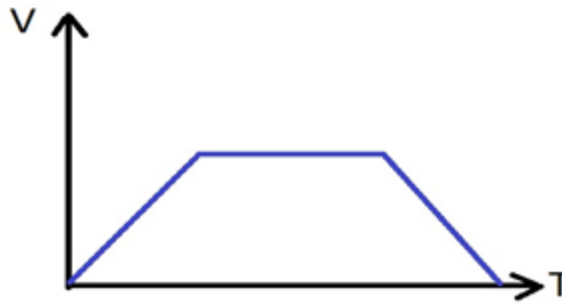


Figura 4-4: Relación velocidad y tiempo del motor.

Es importante destacar, que este firmware al ser utilizado por muchas máquinas, este valor puede cambiar. Sin embargo, en este caso al ajustarlo en 100 queda perfecto para el uso que se dará (como escáner 3D), ya que el avance del escáner es pequeño (no funciona a grandes RPM). Esta parte se muestra en la figura 4-5, donde se pueden apreciar de igual forma los comentarios utilizados en el firmware.

```
// -----
// Opciones de configuración avanzada:
// La resolución temporal del subsistema de gestión de aceleración.
// Un número más alto entrega una suave aceleración,
// particularmente notable en máquinas que funcionan
// con altas velocidades de avance, pero puede afectar negativamente
// impactando en el rendimiento. El valor correcto para este parámetro
// depende de la máquina, por lo que se recomienda
// establecer esto tan alto como sea necesario.
// Se aconseja utilizar un valor de 100 (medido mediante pruebas)
#define ACCELERATION TICKS PER SECOND 100 // 100
```

Figura 4-5: Definición de la aceleración del escáner.

Además de lo mencionado anteriormente, es igual de importante el AMASS (Adaptive Multi-Axis Step Smoothing) como dice su nombre, es un suavizador de pasos de multi ejes adaptable. En términos generales, una de las principales funciones es evitar el ruido y movimientos involuntarios (principalmente sacudidas) a nivel de motor, por esta razón es necesario añadir esta función al escáner, para así evitar problemas a la hora de trabajar con el modelo finalizado.

```
// Adaptive Multi-Axis Step Smoothing (AMASS)
// es una función avanzada que hace lo que su nombre indica,
// suavizar el paso de los movimientos multieje.
// Esta función suaviza el movimiento, particularmente a bajo nivel
// frecuencias inferiores a 10 kHz, donde el alias entre los ejes
// de movimientos multieje puede causar audible
// ruido y sacudir a la máquina. A frecuencias aún más bajas,
// AMASS se adapta y proporciona incluso mejores pasos (en suavizado)
#define ADAPTIVE MULTI AXIS STEP SMOOTHING
```

Figura 4-6: Definición de AMASS.

Resulta importante destacar el funcionamiento del motor al someterlo a funcionamiento continuo, anteriormente en la figura 4-4 se consideró el movimiento de 1 paso, sin embargo, para realizar el movimiento en su totalidad es requerido utilizar 200 pasos para completar una vuelta (360°), en este sentido es importante entender cómo se comporta la unión de los dos bloques, tal como se muestra en la figura 4-7.

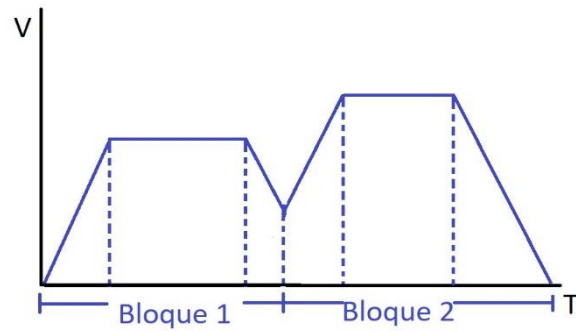


Figura 4-7: Velocidad por bloque.

Tal como se muestra en la figura 5-5, existe un “planificador” donde la memoria temporal (buffer) va planificando perfiles de velocidad y aceleración constantes y se van realizando uniones de bloques para ir logrando lo deseado. Es importante destacar que, el planificador va calculando activamente solo las velocidades de entrada para completar la velocidad óptima (elegida).

4.3 Ejecución del Software.

Antes de ejecutar el programa Horus, se hace necesario enchufar la fuente de poder de 12v al suministro eléctrico, el cual que trae consigo el escáner. Además de conectar a el computador los cables USB correspondientes a la placa Arduino y el de la cámara web. Es importante respetar dicho orden, ya que si se hace al revés existe posibilidad de quemar la electrónica que conforma el escáner.

Una vez completados los pasos mencionados anteriormente, se hace necesario ejecutar el programa. En el caso del sistema operativo Windows 10, podemos concurrir al menú inicio y buscar el software “Horus” (previamente instalado), tal como se muestra en la figura 4-8., una vez encontrado el software, se procederá a realizar un clic izquierdo sobre el icono del “Horus” de dicha forma se podrá ejecutar el software.

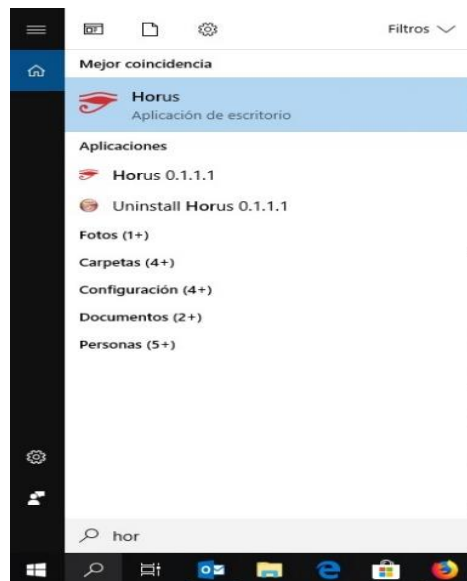


Figura 4-8: Inicio de Horus.

Una vez abierto el software Horus, se abrirá una ventana en donde se encontrarán 4 opciones disponibles para ejecutar, tal como se muestra en la figura 4-9. Donde la opción elegida dependerá de lo que desee realizar el usuario al momento de ejecutar el programa.

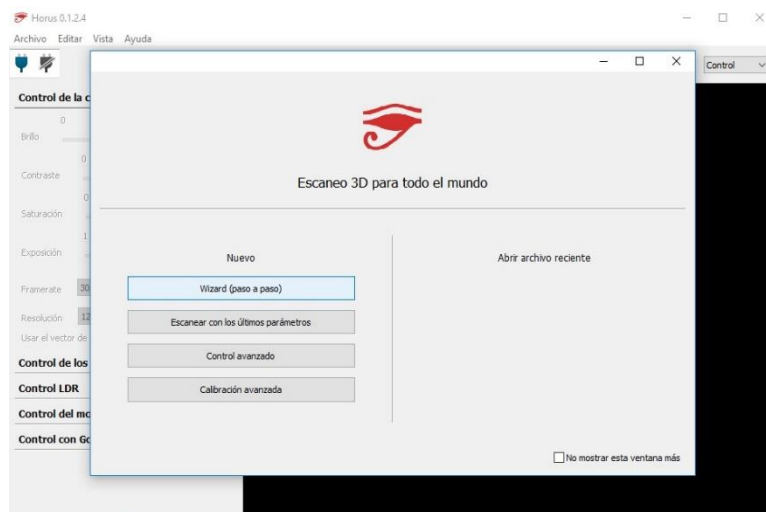


Figura 4-9: Arranque de Horus.

De acuerdo con la figura 4-9, se tienen 4 opciones disponibles (Wizard, Escanear con los últimos parámetros, Control avanzado, calibración avanzada).

Wizard mode, permite realizar los procesos de conexión, calibración y sintonización de parámetros de forma muy simplificada, lo cual es una gran ayuda para las personas que no poseen mayor conocimiento técnico respecto al escáner.

En el control avanzado es posible tener un control total de los elementos que conforman el escáner, los cuales son el motor, láseres y cámara. Este modo permite ajustar manualmente diversos parámetros como: brillo, contraste, saturación, exposición, framerate, resolución, distorsión.

La calibración avanzada faculta al usuario para cambiar los parámetros de la cámara y poder así tener una calibración óptima. Existen parámetros importantes para calibrar en este punto como: calibración entre cámara y módulos laser, calibración entre cámara y plataforma, calibración entre el sensor de la cámara y el lente. Dentro de esto, la calibración de la cámara hace referencia a adquirir los parámetros de la cámara como: distancia focal, centro óptico y distorsión del lente. Mientras que la calibración de los láseres se puede obtener los planos de cada láser y finalmente la calibración del disco, la cual permite obtener la posición y rotación de este.

Mientras que, en la columna de la derecha, se muestran los modelos escaneados recientemente donde se pueden abrir estos últimos. También se encuentra una opción en la esquina inferior derecha que permite ocultar la ventana cada vez que se inicia el software Horus.

En este caso, si no se es un usuario avanzado, se procederá al Wizard mode donde se encontrará una interfaz bastante amigable, en donde se mostrará todos los pasos simplificados para obtener una correcta calibración y podamos obtener los modelos escaneados que requiere el usuario.

Finalmente, esta pestaña mostrará un mensaje “Connecting board COM4 115200”, en donde se conecta la placa con el software “Horus”, mostrando en este caso que la placa se está conectando en el puerto de serie COM4 y está trabajando a un “Baud Rate” de 115200.

Posterior a esto, se concurre a la pestaña editar ajustes, en donde se mostrarán parámetros como el tipo de luminosidad, invertir la dirección del motor (en caso de que sea necesario), distancia del patrón en (mm). Los parámetros mencionados anteriormente, dependerán tanto de la luz ambiente en donde está inserto escáner (posee 3 opciones: alta, media y baja), una forma fácil de observar si se eligió bien la luminosidad, es visualizando los patrones de colores que se forman en la tabla (mostrados en la figura 4-11), lo que se busca es que dichos patrones se visualicen desde distintos ángulos, no solamente desde frente a la tabla de ajedrez.

En el caso de invertir la dirección del motor, dependerá de la posición en la cual esté girando el motor, puede darse el caso (dependiendo del modelo específico del Nema 17), el cual debe girar en sentido antihorario mirado desde arriba el escáner. Finalmente, el parámetro de la distancia del patrón hace referencia a la distancia en mm medida desde la base de la tabla de ajedrez a el primer cuadrado que forma la tabla. Todos estos parámetros anteriormente mencionados son mostrados en la figura 4-12.

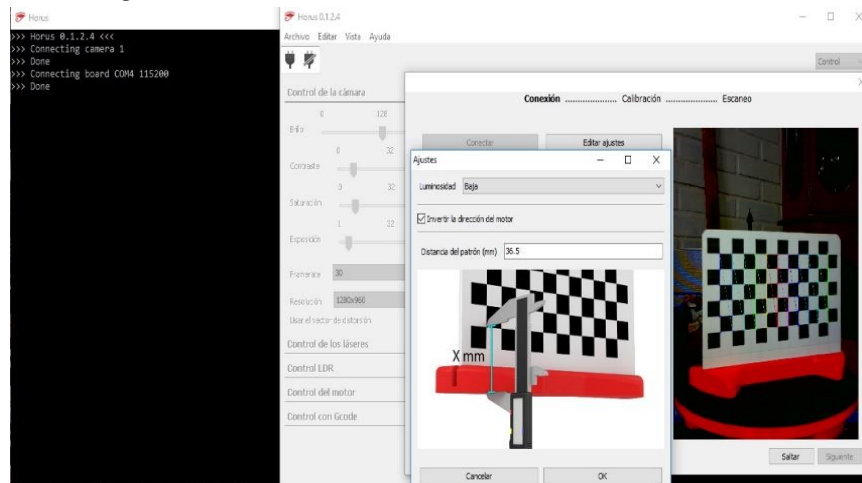


Figura 4-12: Ajuste de parámetros en Wizard Mode.

Tal como se muestra en la figura 6-5, una vez que se eligen todos los parámetros se procede a dar clic en el botón “OK”. A continuación, se procede dar clic en siguiente (se mostrará la interfaz mostrada en la figura 4-11) y se mostrará la interfaz mostrada en la figura 4-13.

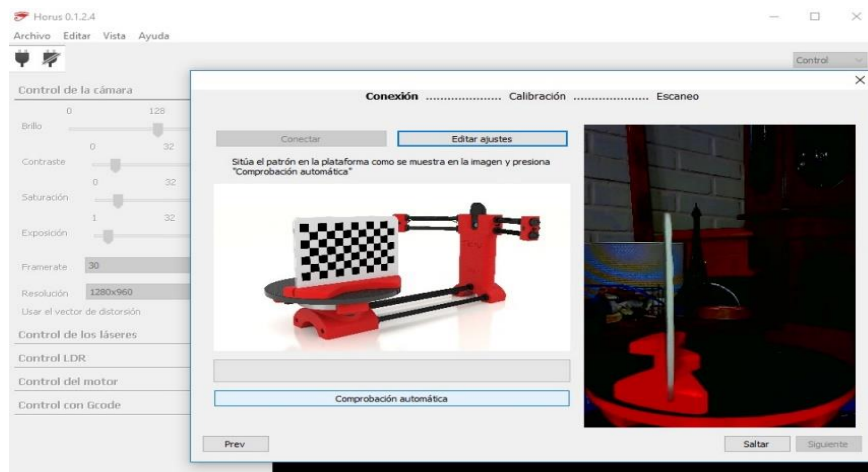


Figura 4-13: Calibración en Wizard Mode.

En la interfaz mostrada en la imagen 4-13 se muestra la comprobación automática, en donde el software “Horus” realiza una comprobación de los dispositivos que conforman el escáner, es decir, los láseres, motor, cámara. Este paso es importante, ya que entrega un indicio de que esté todo bien para realizar una correcta calibración, para realizar la comprobación se hace necesario colocar la tabla de ajedrez en paralelo a la cámara, tal cual como se muestra en la figura 6-6.

Una vez que el proceso de comprobación automática finaliza, se mostrará una interfaz como la mostrada a continuación en la figura 4-14:

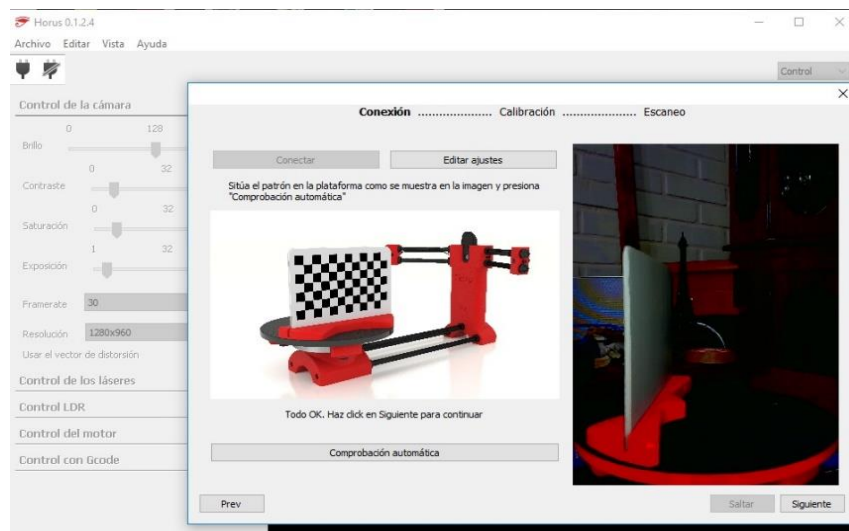


Figura 4-14: Comprobación automática en Wizard Mode.

Como se muestra en la figura 4-14, al realizar correctamente la comprobación automática, se exhibe un mensaje debajo de la imagen que indica “Todo OK. Haz clic en Siguiente para continuar”, con lo cual se irá a la siguiente interfaz que corresponde a la calibración del escáner.

Una vez que se está en la interfaz de calibración, tal cual como se muestra en la figura 4-15 se exhibe nuevamente la posición en donde debe ser ubicada la tabla de calibración de ajedrez, es importante conservar dicha posición, ya que así se podrá calibrar el escáner.

Después de ubicar dicha tabla en la posición correcta, se realiza un clic en la opción calibrar en donde el escáner realizará un giro donde se van tomando muestras y se estima la posición de los láseres (planos), coordenadas de la cámara, motor, entre otras.

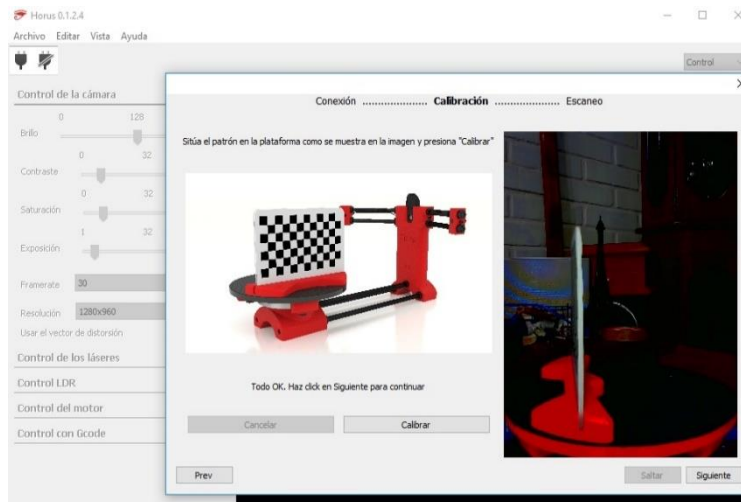


Figura 4-15: Calibración en Wizard Mode.

Finalmente, una vez que finaliza el proceso realizado en la figura 4-15, se mostrará un nuevo menú, donde se exhiben 3 opciones que son: Resolución, Láser y Escaneo.

En la primera opción, se debe seleccionar el tipo de resolución se utilizará (alta, media, baja), se hace necesario considerar que, a mayor resolución, más tiempo se empleará en el proceso, ya que a mayor resolución se tomarán más muestras y por ende se tendrá una nube de puntos mayor a que si fuera mediante resolución baja.

En el apartado de Láser, se pueden elegir 3 opciones: Izquierda, derecha y ambos, donde se puede elegir trabajar con solo un láser (izquierdo o derecho) y ambos. Se hace recomendable utilizar ambos, ya que se obtendrán resultados más fiables y es menos probable encontrar problemas de oclusión a la hora de tener los resultados.

Finalmente se tiene la opción de escaneo con textura o simple. El simple utiliza una imagen por ciclo para obtener una nube de puntos, sin capturar el color del objeto, mientras que el con textura utiliza dos imágenes para capturar el láser, así muestra los colores reales del objeto. Todas estas opciones son mostradas a continuación en la figura 4-16.

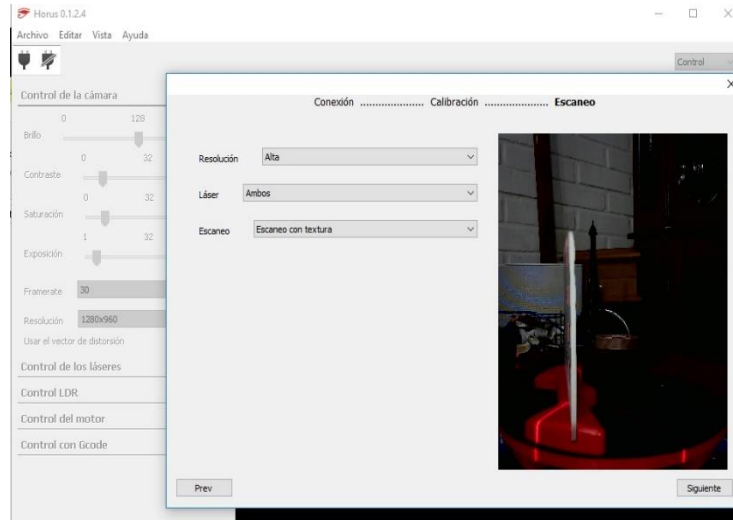


Figura 4-16: Escaneo en Wizard Mode.

Posteriormente, se obtendrá una interfaz como la mostrada en la figura 4-17, en donde se encontrarán diversas opciones como: Parámetros de escaneo, Plataforma giratoria, Adquisición de imagen, Segmentación de imagen, Generación de la nube de puntos.

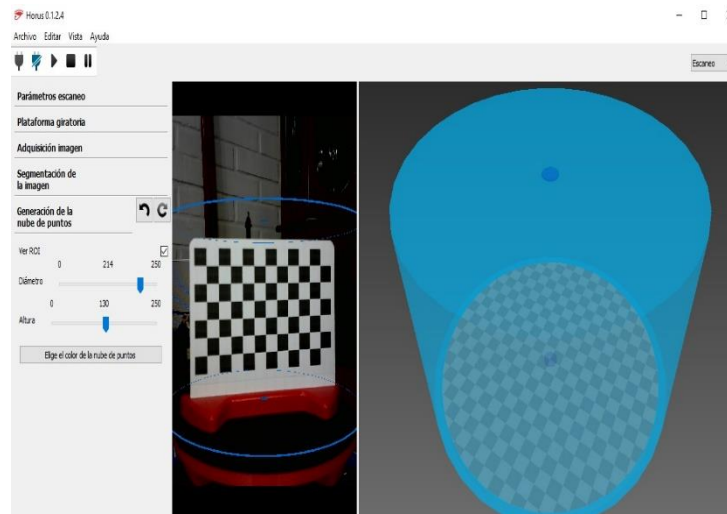


Figura 4-17: Parámetros antes de escanear.

4.4 Primeros modelos escaneados.

El primer objeto a escanear para este proyecto fue la misma tabla de ajedrez, con la cual se realizó la calibración del escáner. Cabe señalar, que este primer resultado fue malo, ya que como se puede apreciar en la figura 4-18, prácticamente no se observa bien lo que es el objeto que se escaneó, por lo tanto, no es un modelo bueno. Uno de los factores que influyó en obtener resultados así fue la cantidad de luz que le está entrando a la cámara, lo que influyó directamente en el mal resultado, ya que no se pudo distinguir bien el objeto y la consecuencia fue el resultado mostrado en la figura que se menciona anteriormente.

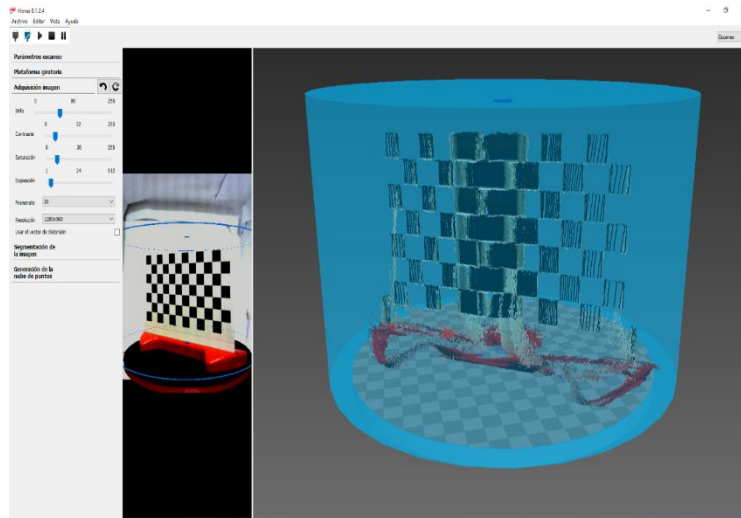


Figura 4-18: Primer objeto escaneado.

En una segunda instancia, se disminuyó la cantidad de luminosidad que existía en el lugar (se tenía un foco apuntando directamente a el objeto que se estaba escaneando, al disminuir la cantidad de luz y al probar con un objeto que poseía irregularidades para verificar el funcionamiento del prototipo de escáner, se obtuvo el resultado mostrado en la figura 4-19.

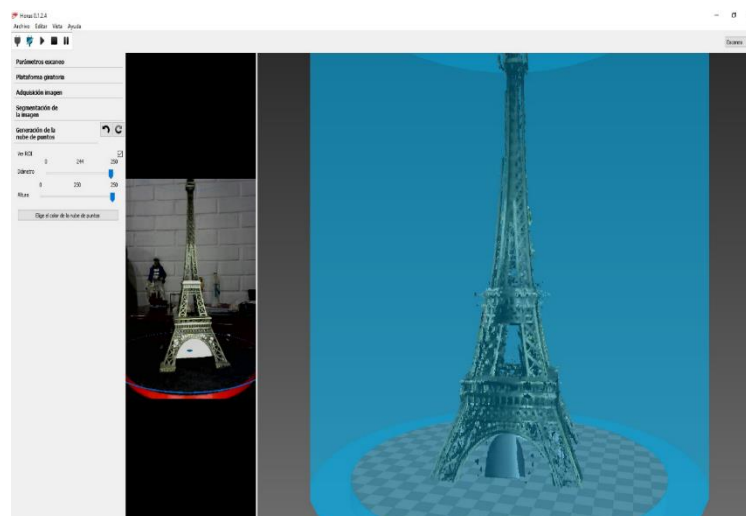


Figura 4-19: Segundo objeto escaneado.

Como se muestra en la figura anterior, existe una notable mejoría en el diseño obtenido, con solo bajar la luminosidad que estaba teniendo en aquel momento el escáner, además es posible observar que hablamos de una figura mucho más compleja, ya que posee muchas irregularidades en la superficie lo cual puede ocasionar en muchas ocasiones el fenómeno de la oclusión. Lo cual muestra un avance significativo y demuestra a su vez, que era posible obtener un buen modelo 3D con dos láseres en vez de utilizar solamente uno.

Seguidamente, se comienza a probar con los filtros que estaban por defecto en el software en el área de segmentación de imagen, aquí existían dos opciones: Umbral y Open.

El parámetro umbral corresponde a un algoritmo que filtra el ruido y lo elimina. De forma que filtra de acuerdo a un valor umbral definido, aquellos puntos que su intensidad es menor a la definida los elimina, mientras que los que están sobre ese valor los permite pasar.

Por otro lado, el parámetro Open filtra el ruido difuminado de los puntos aislados y une a los puntos cercanos. Es importante realizar la prueba de estos parámetros en vivo, ya que así se pueden obtener resultados. Un ejemplo de estas pruebas es la siguiente figura 4-20

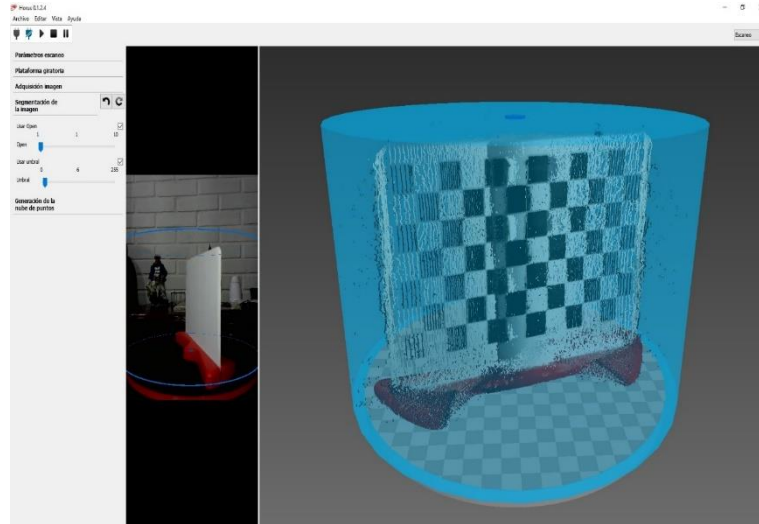


Figura 4-20: Prueba de filtros.

A continuación, se muestra una última figura 4-21 donde se muestra un modelo bastante optimizado en relación con los anteriores. Se calibró con los parámetros anteriormente mencionados.

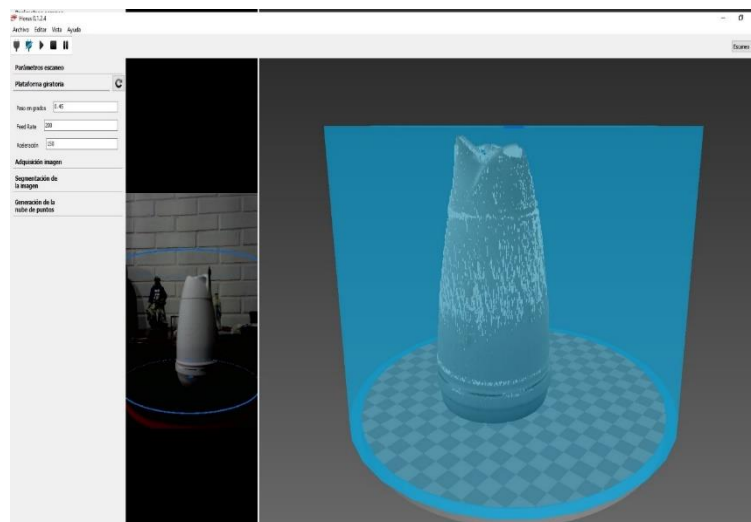


Figura 4-21: Pruebas finales con filtros.

Discusión y conclusiones

Como se ha mostrado a lo largo de este trabajo, los escáneres 3D resultan muy útiles tanto a los aficionados a las impresoras 3D, como también a instituciones. Siendo una gran herramienta que está logrando expandirse a nivel mundial muy rápidamente, dada esta última razón es que estos equipos han ido disminuyendo su precio dada la masificación que ha tenido esta tecnología, sin embargo, el precio sigue siendo elevado para la mayoría de los usuarios, lo que dificulta de gran manera el acceso a esta tecnología tan innovadora y de gran utilidad.

Resulta fundamental identificar bien el tipo de tecnología a emplear cuando se requiera desarrollar alguna labor, ya que de la elección dependerá el desempeño que se va a obtener y además de la calidad de los modelos. Por esta razón, se fue resaltando los problemas que posee cada tecnología de escaneo 3D, para así demostrar el porque se eligió esa tecnología para desarrollar este proyecto siendo el escáner de triangulación la mejor opción en este caso.

Dentro del mundo de los escáneres 3D se hace necesario elaborar formatos que sean universales, ya que muchas empresas fabricantes de escáneres 3D se basan en programas y formatos propios, los cuales restringen en cierta medida a los usuarios. Debido a que este tipo de tecnología se puede complementar muy bien con diversas aplicaciones, por esta razón al crear normas generales podría trabajarse de mejor manera, sin poner obstáculos a los usuarios finales.

Por otro lado, el desarrollo de los diversos escáneres 3D de bajo costo en la actualidad, ha contribuido a ir resolver paulatinamente el problema mencionado anteriormente (formatos), principalmente debido a la necesidad de los usuarios por obtener modelos para replicar los cuales no se encuentran fácilmente en internet y a su vez, al diseñar cada uno de estos se pierde bastante tiempo cuando se habla de realizar una replicación de un objeto.

Es importante destacar también que existen muchos factores que influyen en la realización de un correcto escaneo, dentro de estas variables tenemos en si misma el diseño del prototipo, como la interacción entre el software Horus con las dimensiones físicas que posee el prototipo de escáner 3D propuesto.

A la hora realizar la calibración del driver, es importante tener en cuenta la hoja de especificaciones que corresponden al driver utilizado, ya que muchas personas realizan calibraciones erróneas lo cual produce daños tanto a el mismo driver, como a la Shield que se está utilizando. Para evitar estos problemas, es de suma importante utilizar los datos entregados por el fabricante y además emplear los conocimientos aprendidos en los diversos cursos que forman parte de la carrera en la Universidad.

Consiguiendo, es importante destacar la importancia de entender el firmware Marlin 1.1 que es universal para diversos tipos de máquinas, como CNC, impresoras 3D, escáner 3D, entre otros. Es importante destacar que el uso del correcto del firmware permitió obtener un excelente resultado, respecto a la movilidad del motor de la base de este escáner

Es importante mencionar que se tuvieron que hacer distintos ajustes a lo largo del desarrollo de este proyecto, concretando modelos que en un principio no funcionaron dado que era necesario imprimir y tener físicamente el prototipo para después realizar los ajustes con el software.

En el sentido anterior, realizar un escáner 3D móvil, con el software Horus se convierte en una tarea compleja, ya que en un principio se planteó dicha opción como una alternativa de mejora, sin embargo, no se pudo concretar debido a que Horus establece parámetros fijos. En otras palabras, se deberían cambiar los ajustes por cada objeto (debido al tamaño) en la placa Arduino (donde está inserto el código de programación) para así ajustarse a la altura, lo cual resultaría en una tarea engorrosa para el usuario y, además se excluiría a los usuarios que tuvieran poca experiencia a la hora de programar.

Es de suma importancia entender completamente las funciones que posee el software Horus, tanto para calibrar, como para desarrollar un correcto escaneo, ya que existen parámetros de segmentación de imagen, que en pocas palabras lo que hacen es ir aplicando filtros a el modelo en 3D, con el fin de ir reduciendo el ruido que se genera en la imagen final con lo cual se obtuvieron mejores resultados que los originales sin aplicar estos parámetros.

En este sentido, los dos parámetros utilizados son: Umbral y Open, donde el primero hace un filtraje de ruido eliminándolo de la imagen, esto gracias al valor definido de acuerdo a el umbral que viene definido por el láser (longitud de onda), los valores sobre a el umbral emitido por el láser los conserva y los inferiores son eliminados, Mientras que la segunda opción "Open", permite eliminar el ruido difuminado, es decir aquellos puntos dentro de la nube de puntos generada por el Horus, que queden aislados (sin ningún punto vecino cercano a él), son eliminados mientras que los cercanos son unidos.

La elaboración de este proyecto representó un gran reto desde varias perspectivas, las cuales van desde softwares hasta la calibración completa del equipo. A pesar de ello, es un sistema prometedor en todo sentido, ya que en un futuro cercano será de mucha utilidad para los alumnos y también puede ser explotado en diversas áreas, que pueden ser diferentes a la que se está proponiendo en la problemática de este proyecto, siendo posible la colaboración con otras ramas de la Universidad, tal cual como se ha mostrado en las aplicaciones posibles para los escáneres 3D presentadas en este informe.

Si se utilizara como base este escáner 3D para otros proyectos, a juicio del alumno existen muchas mejoras que se pueden ir añadiendo al sistema ya diseñado, una de las que se podría ver a corto plazo es la utilización de luces que permitan dejar un ambiente más regular en términos de luminosidad, ya que como bien se muestra a lo largo de este informe, uno de los pilares a la hora de trabajar con el software es la cantidad de luz ambiente que rodea al objeto. Como se visualiza anteriormente, cuando se tiene un ambiente lumínico adecuado para el escáner 3D, se obtiene la visualización correcta de los patrones de calibración del escáner.

Quizás una de las posibles mejoras más difíciles de desarrollar, sería la elaboración de un modelo que sea adaptable en altura para así no limitar el área de escaneado del objeto, si se va a utilizar la misma plataforma (Horus), se debería automatizar el proceso mediante el uso de sensores, ya que en definitiva, como queda demostrado a lo largo del proyecto “Horus” requiere conocer los parámetros intrínsecos como extrínsecos de las variables que conforman al área de escaneado, ya sea distancia de la cámara al objeto, planos de los láseres (ubicación del plano), plano del giro de motor, entre otras. Una vez, resuelto esto el proyecto podría tener una excelente escalabilidad y ser de gran utilidad para diversas tareas, no solo recreativas.

Bibliografía

- [1] Dr. Mostafa Abdel-Bary Ebrahim. 3D Laser Scanners: History, Applications, and Future, 2011.
- [2] Gerald F. Marshall, Glenn E. Stutz. Handbook of Optical and Laser Scanning, 23 Julio 2004.
- [3] Sansoni, Trebeschi, Docchio. State of the art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation, 2009.
- [4] Douglas Lanman, Gabriel Taubin, Build Your Own 3D Scanner: 3D Photography for Beginners, 5 Agosto 2009.
- [5] Andreas Behr. (2011, Mayo, 29). Time of Flight 3D Scanner. Recuperado de <https://www.youtube.com/watch?v=XN4fynAlxqc>.
- [6] Ponoko, <<Ponoko,>> 2011. [En línea]. Available: <https://www.ponoko.com/blog/digital-manufacturing/fabscan-open-source-3d-scanner/>.
- [7] Kickstarter, <<Kickstarter,>> 2015. [En línea]. Available: <https://www.kickstarter.com/projects/1545315380/atlas-3d-the-3d-scanner-you-print-and-build-yourself?lang=es>.
- [8] Jesús Arroyo, «www.diwo.bq.com», 23 Enero 2015. [En línea]. Available: <http://diwo.bq.com/zum-scan-released/>.
- [9] Makezine, <<Makezine,>> 2015. [En línea]. Available: <https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/>.
- [10] Blog lenovo, <<Blog lenovo,>> 2016. [En línea]. Available: <https://www.bloglenovo.es/arduino-cursos-gratuitos-pago/>.

-
- [11] Aprendiendo arduino, <<Aprendiendo arduino,>> 2018.[En línea]. Available: <https://aprendiendoarduino.wordpress.com/category/sensores/>.
- [12] Radiorhin, <<Radiorhin,>>.[En línea]. Available: <http://www.radiorhin.com/D/product/arduino-shields/>.
- [13] Wikipedia, <<Wikipedia,>>.[En línea]. Available: https://es.wikipedia.org/wiki/Raspberry_Pi/.
- [14] Raspberry Pi org, <<Raspberry Piorg,>>. [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-universal-power-supply/>.
- [15] Raspberry-Valley, << Raspberry-Valley,>>. [En línea]. Available: <https://raspberrypi.azurewebsites.net/Raspberry-Pi-Gentle-Introduction/>.
- [16] Seedstudio, <<Seedstudio,>>. [En línea]. Available: [https://www.seedstudio.com/Raspberry-Pi-HDMI-LCD-\(5-inch\)-p-2737.html](https://www.seedstudio.com/Raspberry-Pi-HDMI-LCD-(5-inch)-p-2737.html).
- [17] Pi4j, << Pi4j,>>. [En línea] 2018. Available: <http://pi4j.com/pins/model-2b-rev1.html>.
- [18] Core-electronics, <<Core-electronics,>>. [En línea] 2018. Available: <https://core-electronics.com.au/raspberry-pi-wide-angle-camera-module-seeed-studio.html>.
- [19] Stacksocial, <<Stacksocial,>>. [En línea] 2018. Available: <https://stacksocial.com/sales/37-sensors-starter-kit-for-raspberry-pi-pi-3b-included>.
- [20] Learn.adafruit, <<Learn.adafruit,>>. [En línea]. Available: <https://learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor>.
- [21] Panamahitek, << Panamahitek,>>. [En línea]. Available: <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>.
- [22] Pololu, << Pololu,>>. [En línea]. Available: <https://www.pololu.com/product/1182>.
- [23] Reprap, << Reprap,>>. [En línea]. Available: <https://reprap.org/wiki/File:RepRap-NEMA-17.jpg>.
- [24] Seedstudio, << Seedstudio,>>. [En línea]. Available: <https://www.seedstudio.com/document/HLM1230.pdf>.
- [25] Logitech, << Logitech,>>. [En línea]. Available: <https://www.logitech.com/es-es/product/hd-webcam-c270>.
- [26] Logitech, << Logitech,>>. [En línea]. Available: <https://www.logitech.com/es-mx/product/hd-pro-webcam-c920>.

- [27] Impresoras 3D, << Impresoras 3D,>>, 2018. [En línea]. Available:
<https://www.impresoras3d.com/producto/rodamiento-623zz/>.
- [28] Hobby-Hour, « Hobby-Hour ». [En línea]. Available:
<https://www.hobby-hour.com/electronics/smdcalc.php>.
- [29] Smartray, « Smartray ». [En línea]. Available:
<https://www.smartray.com/>.

A Manual de usuario

En este capítulo se abarcará la instalación del software Horus, el cual es de código libre y permite la interacción entre la electrónica y el código insertado en el Arduino. Este software es el que finalmente obtendrá los datos y posteriormente los procesará para crear el modelo 3D final. Es importante destacar que este software es compatible con: Windows, OS X, Ubuntu, lo cual abre un gran abanico de compatibilidad para los diversos usuarios finales que quieran trabajar con él.

A.1 Instalación en Windows.

Para poder instalar Horus en cualquiera de los sistemas operativos anteriormente mencionados, se hace necesario recurrir al siguiente enlace (“<https://www.bq.com/es/support/ciclop/support-sheet>”), el cual permitirá escoger al usuario que plataforma descargar. Tal como se muestra en la figura A-1.

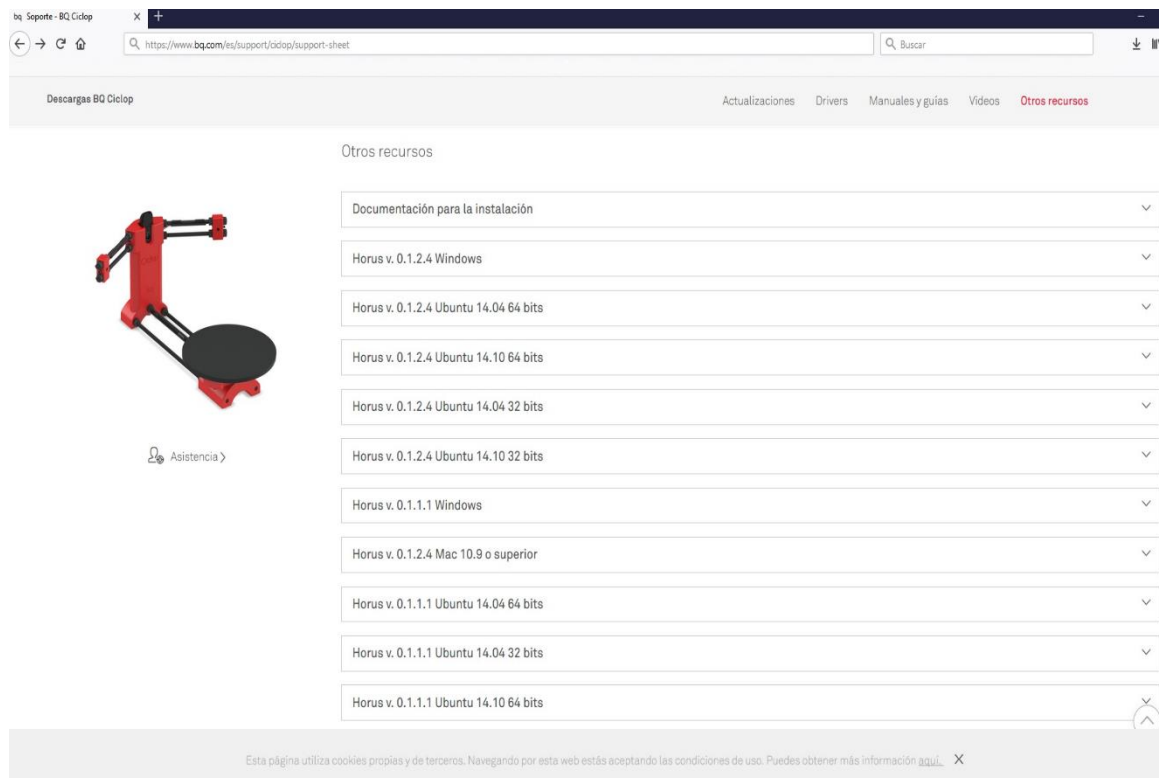


Figura A-1: Descarga del software Horus.

Tal como se muestra en la figura A-1, al abrir el link mencionado anteriormente encontramos las 3 plataformas disponibles para descarga, de acuerdo con la versión del sistema operativo que tenga el usuario.

Para este caso se recurre a el enlace Horus 0.1.2.4 Windows, en donde se desplegará una ventana mostrando los idiomas disponibles (Español, Inglés, Alemán, Italiano, Portugués, Francés), de igual forma como se muestra en la figura A-2.

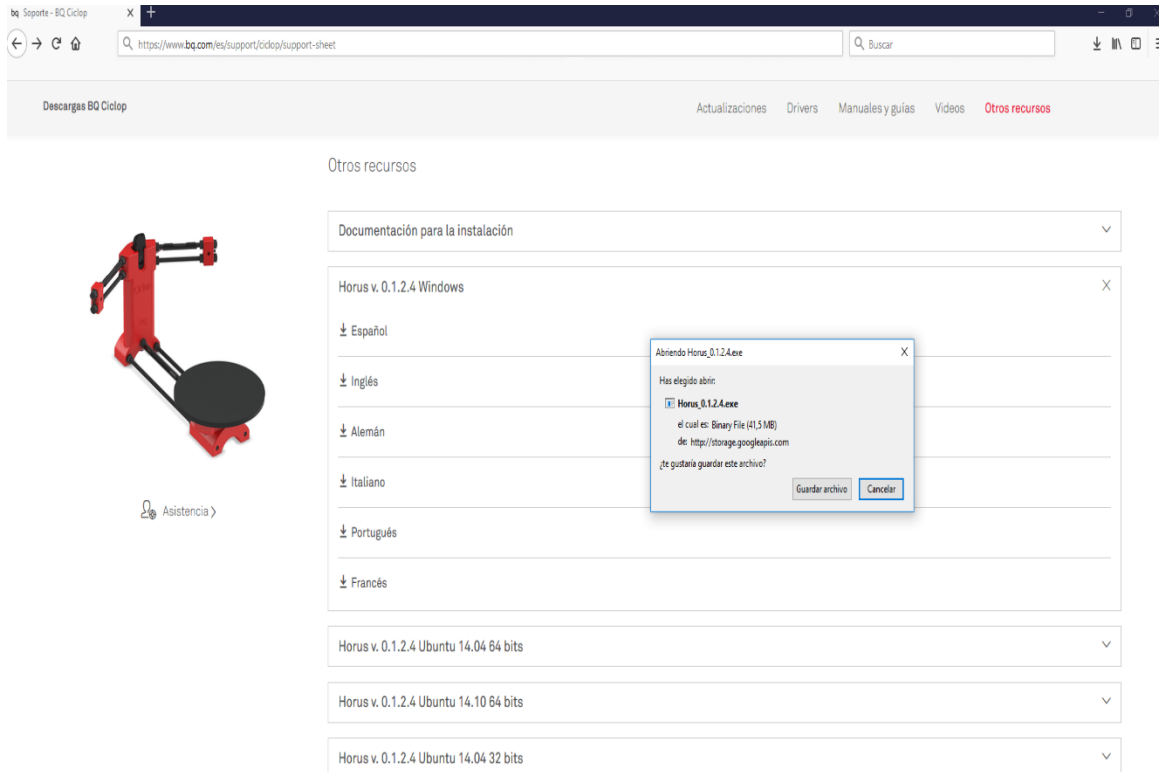


Figura A-2: Horus para Windows.

Una vez descargado el archivo del Horus, se procede a abrir el instalador y posteriormente seguir los siguientes pasos, tal como se muestra en las figuras A-3, A-4 y A-5.

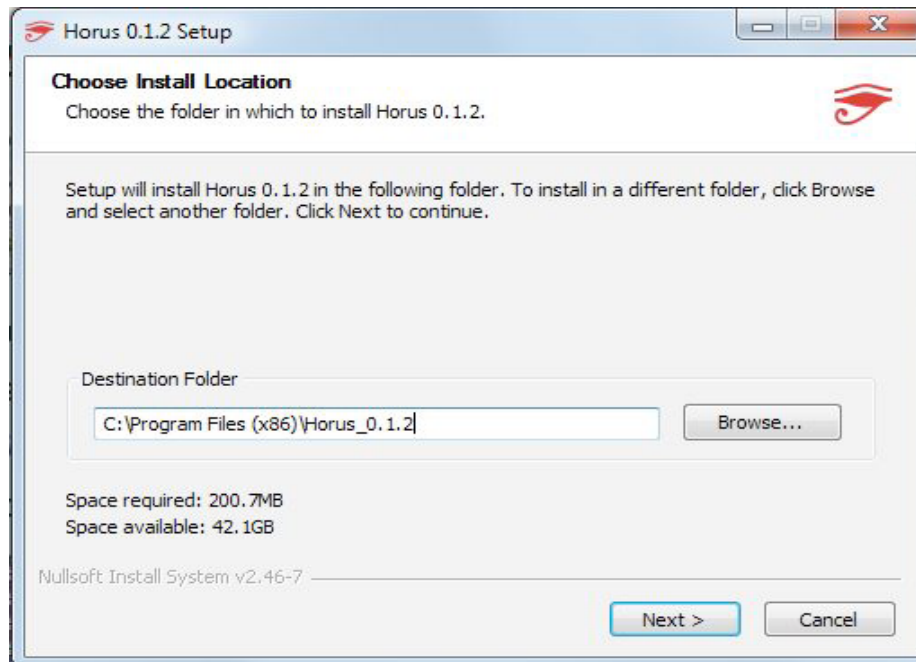


Figura A-3: Pasos de instalación para Horus

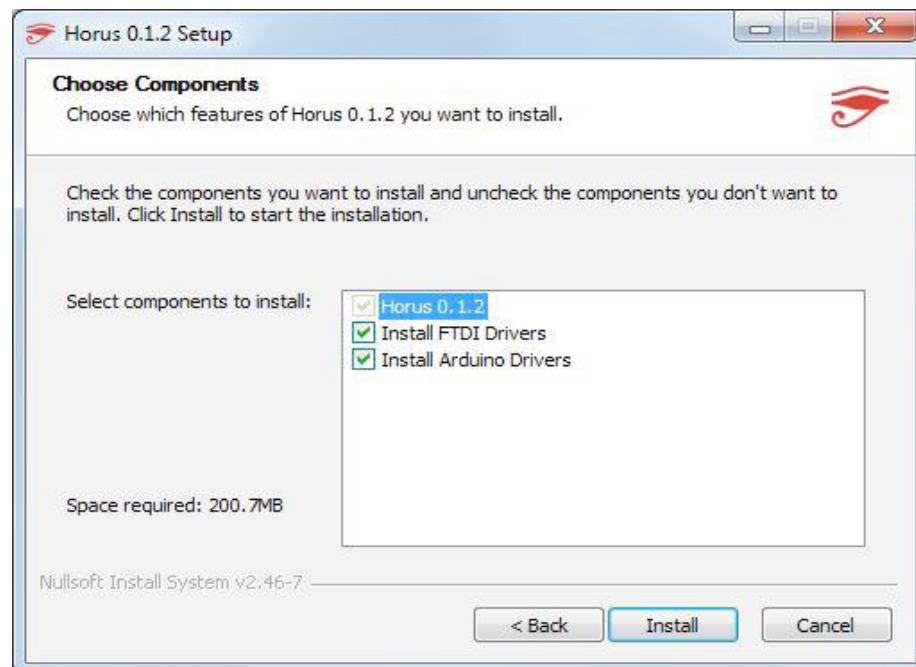


Figura A-4: Pasos de instalación para Horus.

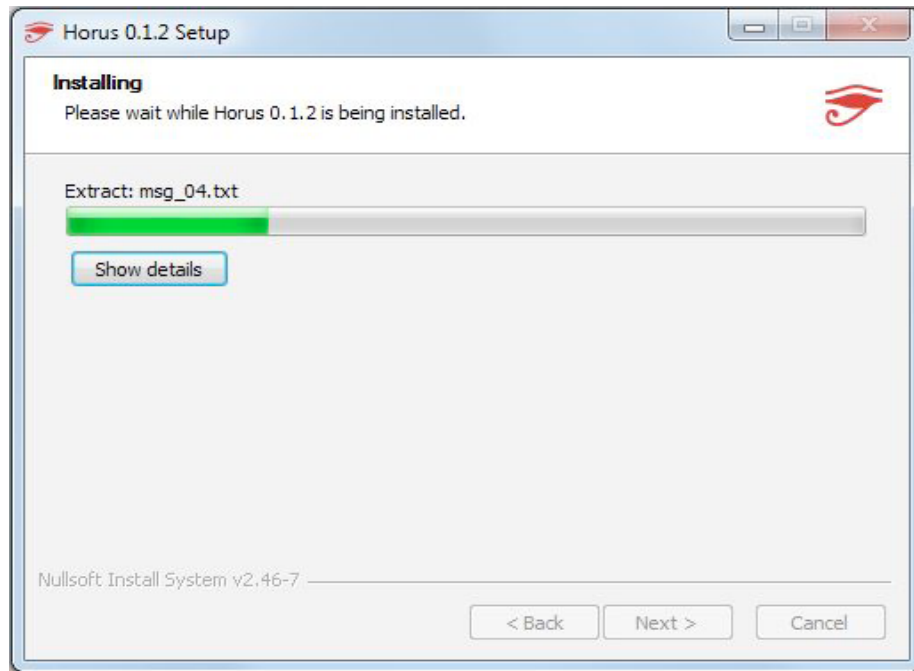


Figura A-5: Pasos de instalación para Horus.

En las figuras anteriormente mostradas, se expone los pasos para tener una correcta instalación del software, luego de abrir el instalador del programa Horus en Windows. En la figura A-3, el software pedirá elegir el directorio de instalación, posteriormente el programa brindará la opción de instalar FTDI y Arduino driver, los cuales deben quedar instalados (no se realizará ningún cambio) según la figura A-4. Finalmente, una vez concluido los procesos anteriormente mencionados, se comienza con la instalación del software Horus en el computador del usuario (utilizando Windows).

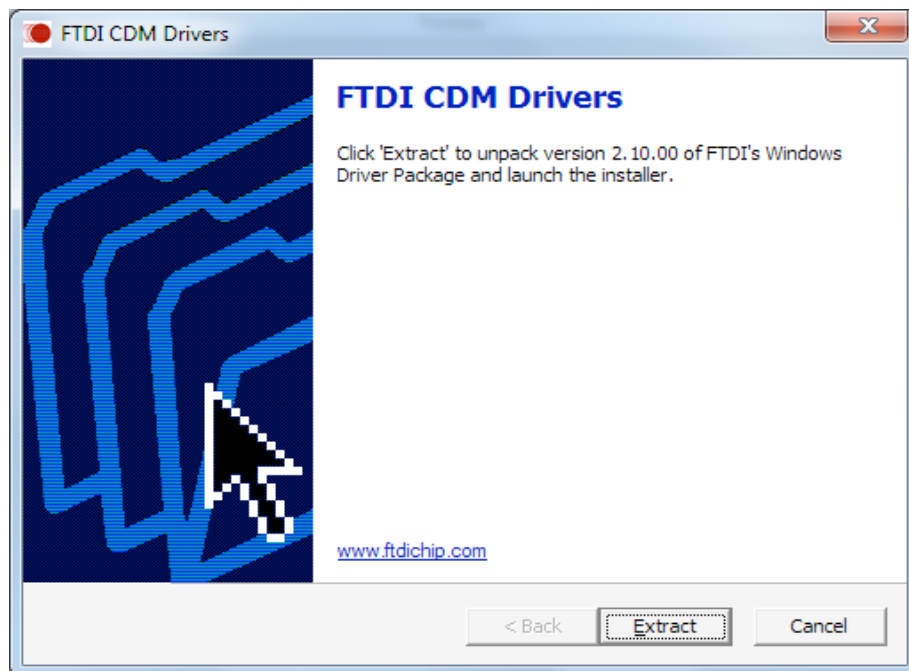


Figura A-6: Pasos de instalación para Horus

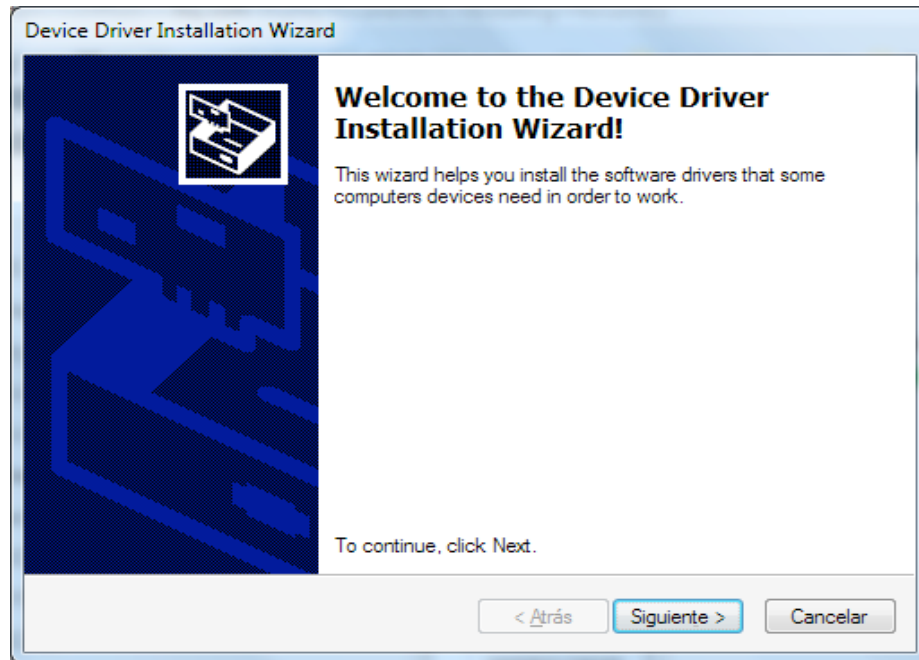


Figura A-7: Pasos de instalación para Horus

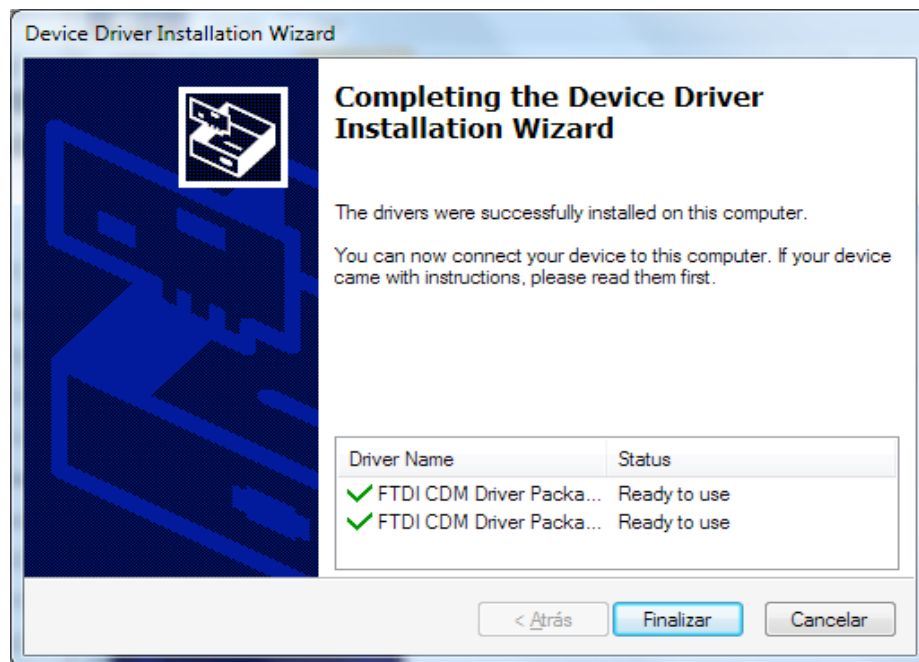


Figura A-8: Pasos de instalación para Horus

A mitad de proceso de instalación, comenzará la instalación de los archivos consultados en la figura A-4, anteriormente vista. Se identificará el proceso, ya que aparecerá una nueva imagen como la mostrada en la figura A-6,

La funcionalidad de este driver es permitir la comunicación entre la placa y el computador, ya que transforma las transmisiones RS-232 o TTL en señales USB. Para proseguir con la instalación, se dará clic en “Extract” tanto en las ventanas de la figura A-6 como en la A-7, así se encontrará la ventana mostrada en la figura A-8 la cual muestra que la instalación de este driver concluyó de manera exitosa.

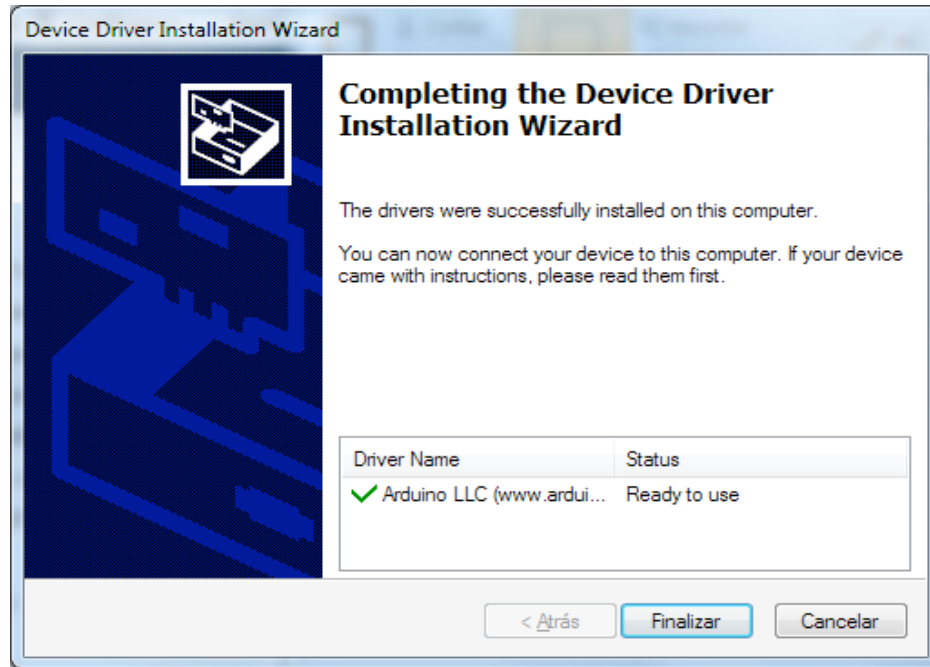


Figura A-9: Pasos de instalación para Horus

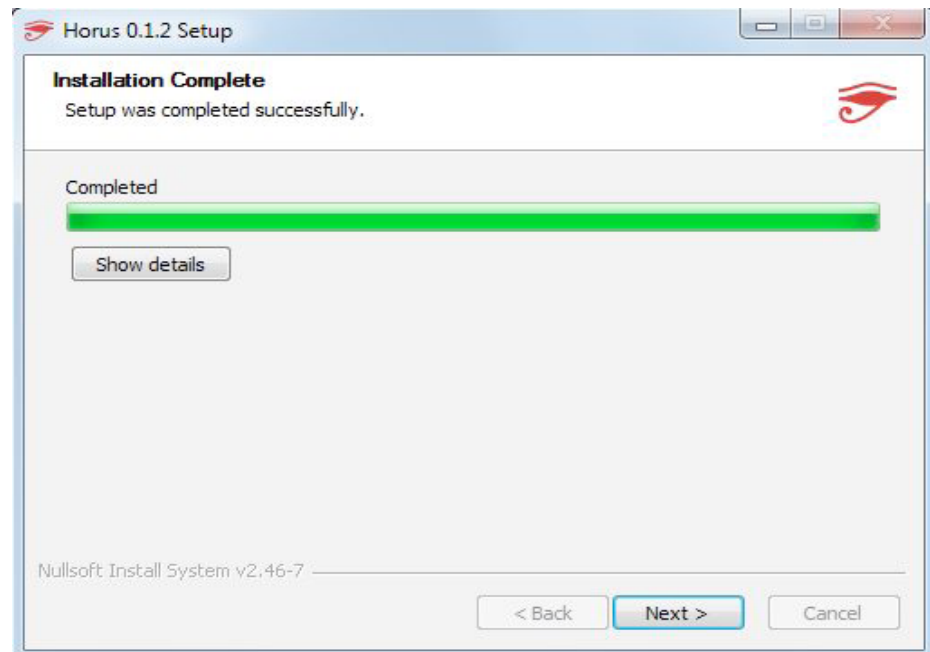


Figura A-10: Pasos de instalación para Horus

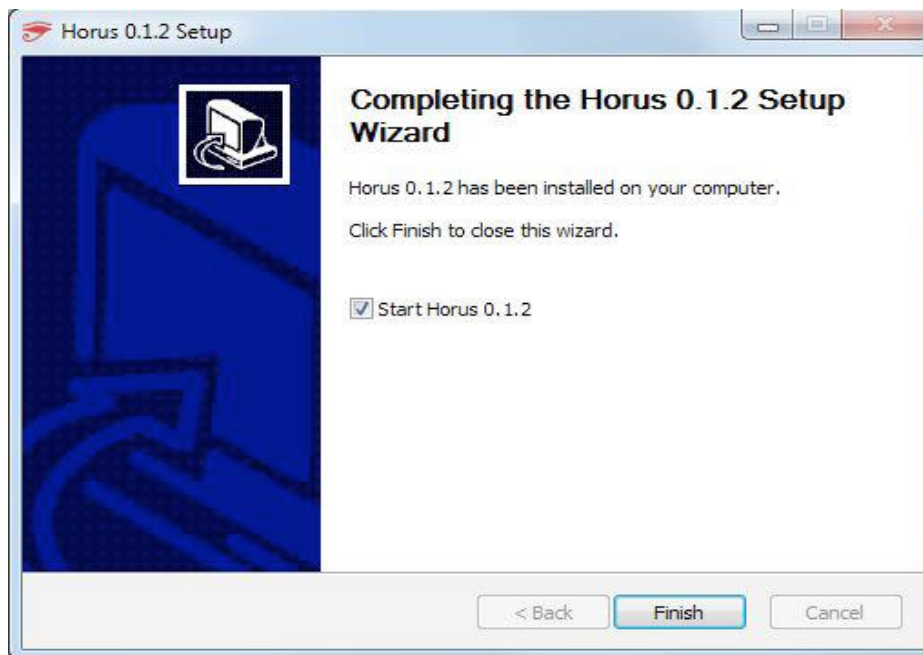


Figura A-11: Pasos de instalación para Horus

En la figura A-10 se muestra cuando acaba de instalar el último driver, correspondiente al último punto al driver del Arduino, una vez completado esto se da clic en finalizar y aparecerá un menú nuevo parecido al de la figura A-10, en donde se hará clic en la opción “next” y así dará paso al menú de la figura A-11 el cual otorga la finalidad de cerrar el instalador solamente, o ejecutar inmediatamente el Horus al terminar el instalador.

A.2 Driver para la cámara.

En el caso de que la cámara no sea reconocida por el PC, se hace necesario descargar los controladores (drivers) pertenecientes a la cámara elegida. En informes previos a este, se expuso que se sugerían dos modelos de cámaras para desarrollar este proyecto, las cuales son: Logitech C270 y Logitech C920. Es importante señalar que el controlador es el mismo para ambos modelos, por lo tanto, basta descargar cualquiera de los dos para tener funcionando ambos equipos.

Para descargar el driver de la cámara Logitech C270, se recurre al siguiente link: http://support.logitech.com/en_us/product/hd-webcam-c270. A continuación, se presenta la figura 2.1.1, la cual muestra la ventana y opciones que nos presenta Logitech antes de descargar el controlado.

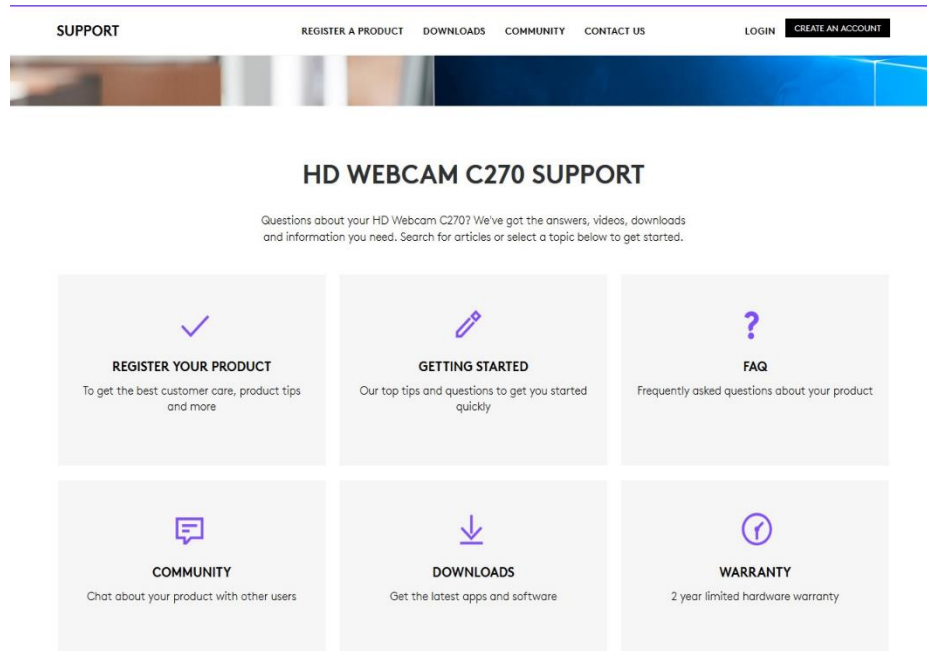


Figura A-12: Software Logitech.

Una vez abierto el enlace anteriormente mencionado (mostrado en la figura A-12), se debe concurrir a la sección de “downloads”, luego de hacer clic en dicha sección se abrirá una ventana como la mostrada en la figura A-13.

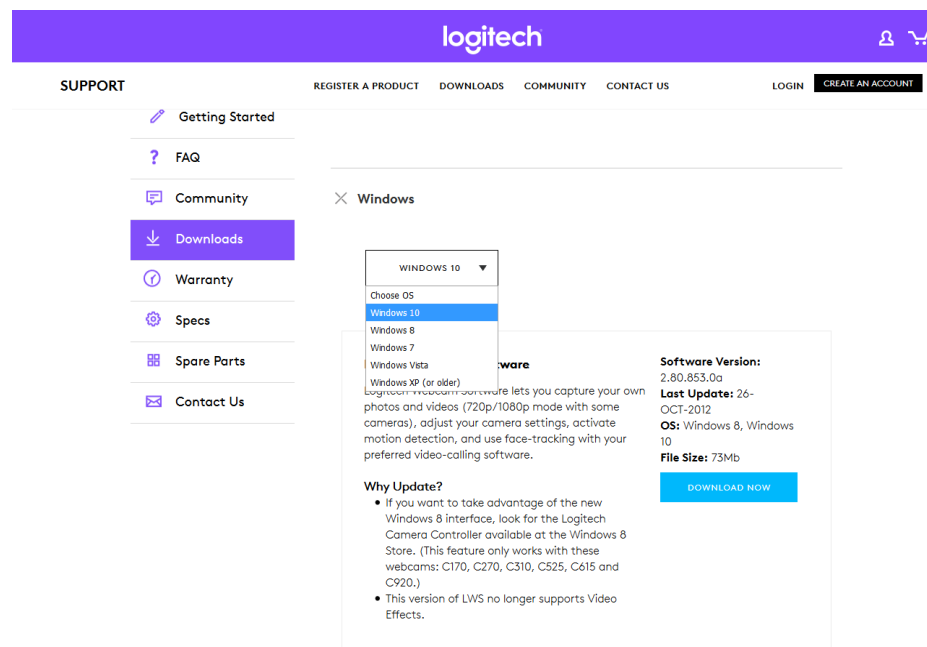


Figura A-13: Logitech Webcam Software.

En la ventana mostrada en la figura A-13, se puede observar que la página del fabricante de la cámara nos permite elegir varias ediciones de Windows (10, 8,7, Vista, XP).

Esto brinda una excelente compatibilidad dentro de la misma plataforma de Windows. Luego de elegir la edición de Windows que posee el usuario, hay que dirigirse a la sección de “Download Now”, donde se abrirá una nueva pestaña con un archivo denominado “lws280.exe” al cual se debe hacer clic en guardar y posteriormente abrir el archivo.

Una vez abierto el archivo, se mostrarán las siguientes ventanas como se muestra en la figura A-14., A-15, A-16

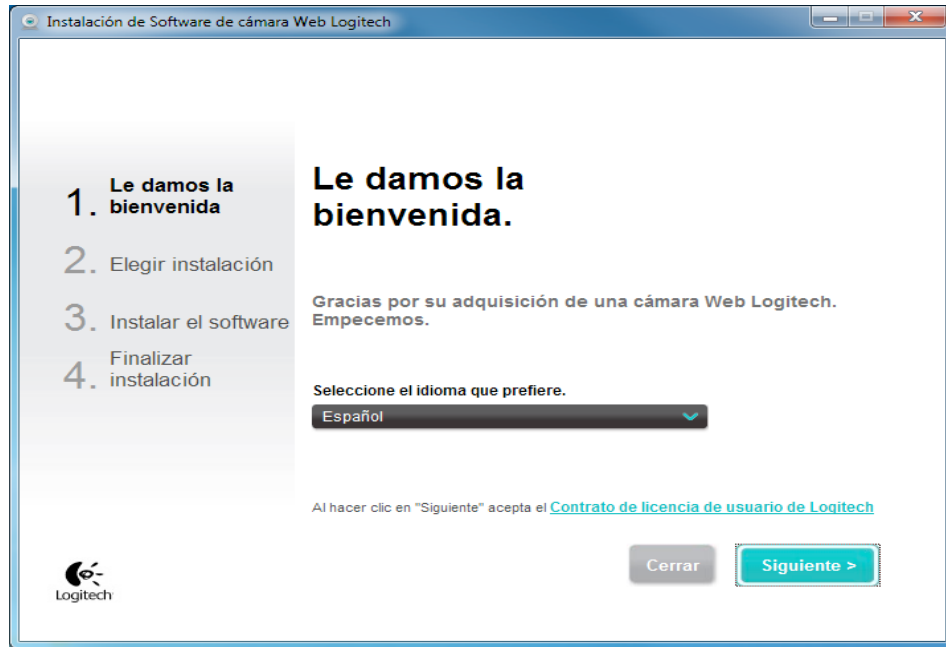


Figura A-14: Logitech Webcam Software.



Figura A-15: Logitech Webcam Software

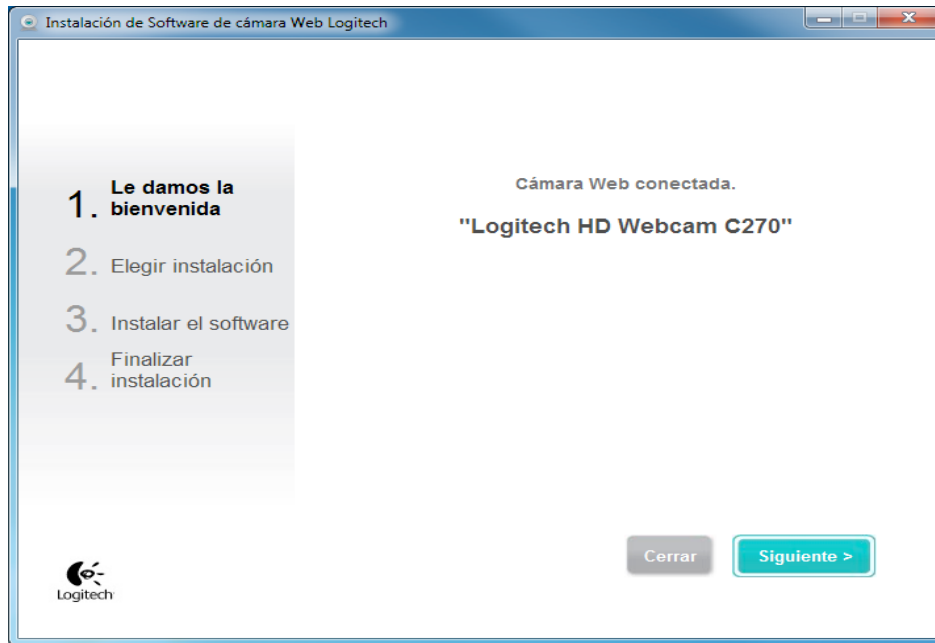


Figura A-16: Logitech Webcam Software

Como se visualiza en las figuras anteriores (A-14., A-15, A-16), los pasos para completar correctamente la instalación son bastantes sencillos, lo importante es que, al momento de abrir el software del fabricante, es tener la cámara desconectada, dada la figura A-14 se debe determinar el idioma a utilizar para el menú (en este caso español), luego se da clic en el botón siguiente. Posteriormente, se mostrará una nueva ventana como se muestra en la figura A-15 en donde se solicitará la conexión de la cámara al computador mediante el cable USB que esta posee. Seguidamente, se expondrá lo mostrado en la figura A-16 donde el software mostrará la cámara reconocida (en este caso una Logitech HD Webcam C270) y mostrará la opción “siguiente” para continuar con el proceso en caso de que sea reconocida correctamente la cámara.

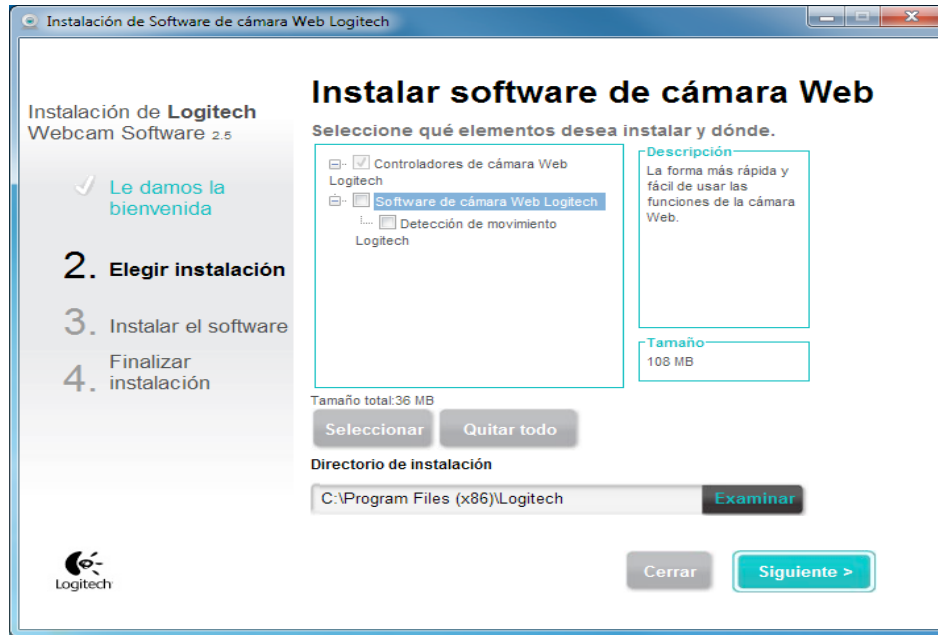


Figura A-17: Logitech Webcam Software

Al dar clic en la opción de la Figura A-16, se le mostrará al usuario una nueva que exhibe las opciones disponibles para instalar en el computador (Ver Figura A-17). Como se necesita solo los controladores de la cámara web, no es necesario mantener el ticket para instalar el “Software de Cámara Web Logitech”, por lo tanto, vendría siendo opcional.

En esta misma ventana es posible cambiar el directorio de instalación (en la opción “Examinar”), donde es importante verificar esta opción sobre todo con los computadores que posee más de una unidad de disco duro (almacenamiento), una vez elegido el directorio de instalación del controlador, se prosigue dando clic en la opción “Siguiente”. Posterior a esto, se mostrará una nueva ventana donde se exhibirá el progreso de la instalación del software como se exhibe en la figura A-18.



Figura A-18: Logitech Webcam Software

Finalmente, una vez que termine el proceso de instalación se mostrará una última ventana que indicará que el proceso ha terminado correctamente, esta ventana es mostrada en la figura A-19.

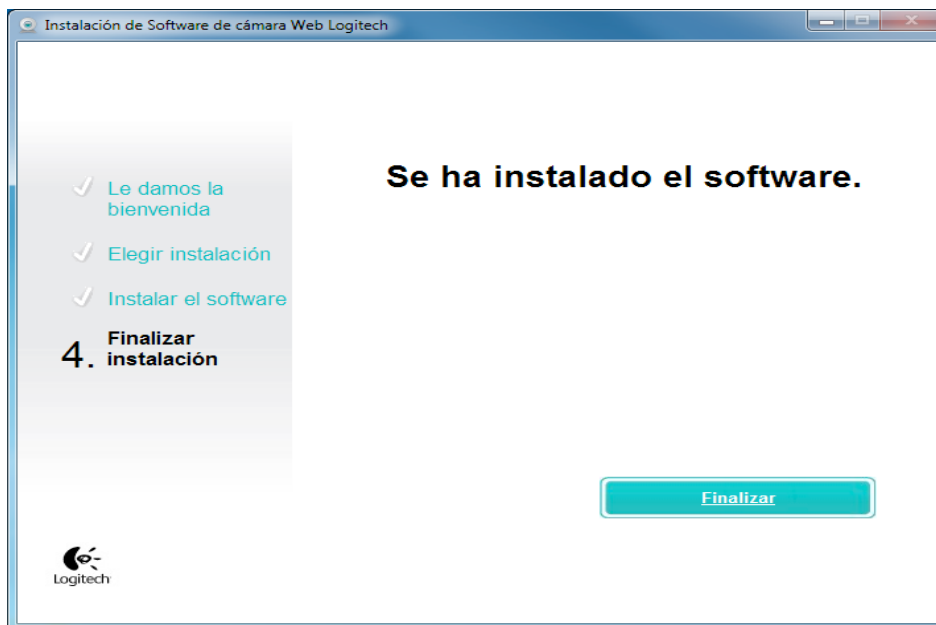


Figura A-19: Logitech Webcam Software

A.3 Instalación en Ubuntu.

Para trabajar en el sistema operativo Ubuntu, básicamente se trabajará con el comando “apt-get”, el cual es un comando extenso e instala paquetes que consigue desde internet. Para acceder a la consola de comandos, el usuario debe dirigirse a la esquina superior izquierda de él y realizando un clic en el logo de Ubuntu, se podrá acceder a la búsqueda dentro del sistema operativo, en esta ventana se escribirá “terminal” para buscar la consola de comandos, tal como se muestra en la figura A-20.

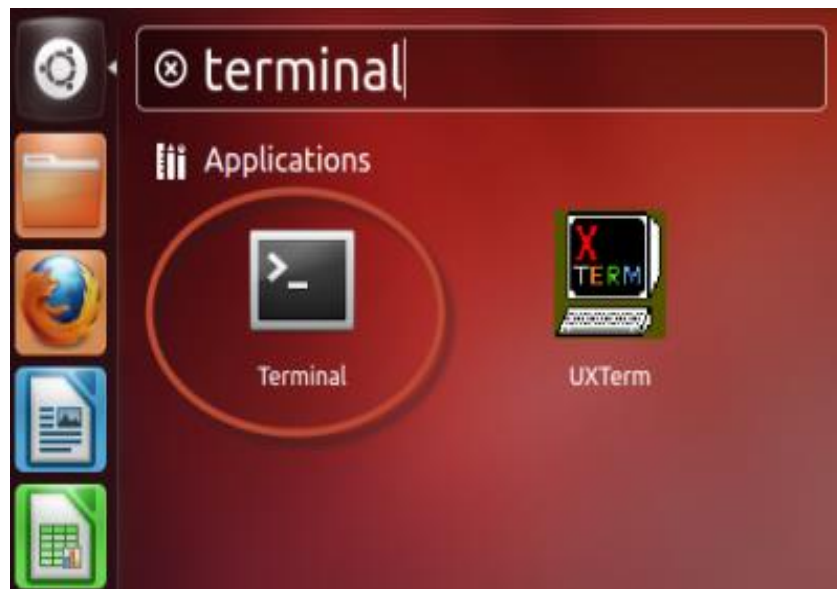


Figura A-20: Acceso a la consola de comando

Luego, se abrirá la consola de comandos la cual está representada por la figura A-21.

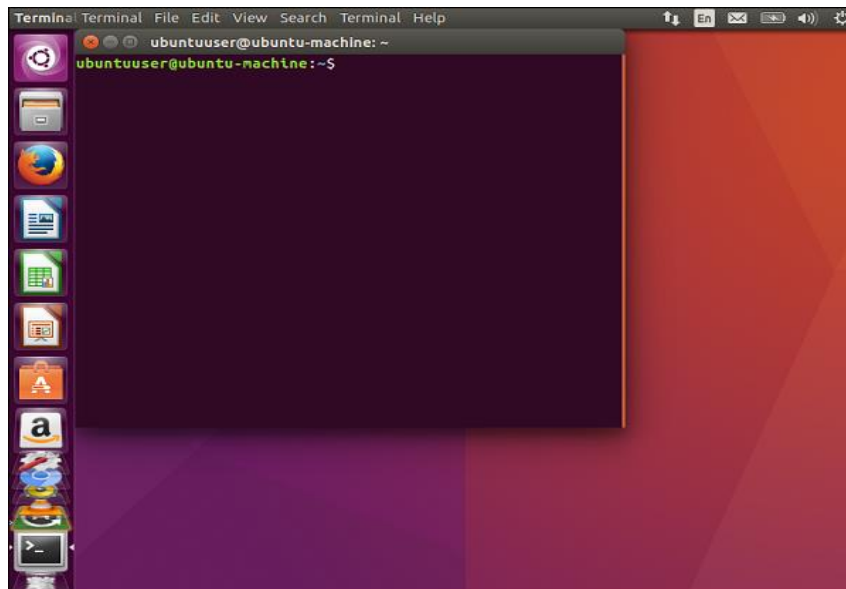


Figura A-21: Consola de comando.

En la consola de comando, mostrada en la figura A-21 se escribirán los siguientes comandos para instalar Horus en este sistema operativo:

- `sudo add-apt-repository ppa:bqopensource/opencv.`
- `sudo add -apt-repository ppa:bqopensource/horus.`

Luego para poder actualizar los paquetes, es necesario introducir los siguientes comandos:

- `sudo apt-get update.`
- `sudo apt-get dist-upgrade.`

Una vez terminada la actualización de los paquetes, se introduce el comando que se mostrará a continuación para poder instalar Horus en Ubuntu:

- `sudo apt-get install horus.`

En caso de no poseer permiso para el puerto de serie, es necesario introducir el siguiente comando;

- `sudo usermod -a -G dialout $USER.`

En caso de querer desinstalar el programa, es necesario repetir el procedimiento mostrado en la figura A-20 y A-21, una vez accediendo a la consola de comandos, basta escribir el siguiente comando:

- `sudo apt-get remove horus.`

A.4 Instalación en OS X.

Antes de comenzar con el procedimiento de instalación, es necesario destacar que Horus es compatible desde la versión 10.9 en adelante (Mavericks).

Para realizar una correcta instalación del sistema, se hace necesario descargar los drivers de FTDI (ídem que en Windows). Para ello, se recurre al siguiente enlace: <http://www.ftdichip.com/Drivers/VCP.htm>

En dicho enlace se encontrará una ventana como se muestra en la figura A-22

The screenshot shows the 'VCP Drivers' page on the FTDI website. It includes a navigation menu on the left with links like 'Firmware', 'Support', 'Android', 'EVE', 'MCU', 'Sales Network', 'Web Shop', 'Newsletter', 'Corporate', and 'Contact Us'. The main content area is titled 'VCP Drivers' and contains introductory text and a disclaimer. Below this is a search bar and the heading 'Currently Supported VCP Drivers:'. A table follows, detailing supported OSes and processor architectures.

Operating System	Release Date	Processor Architecture							Comments
		x86 (32 bit)	x64 (64 bit)	PPC	ARM	MIPSII	MIPSIV	SH4	
Windows*	2017-08-30	2.12.28	2.12.28	-	-	-	-	-	WHQL Certified. Includes VCP and D2XX. Available as a setup executable. Please read the Release Notes and Installation Guides.
Linux	-	-	-	-	-	-	-	-	All FTDI devices now supported in Ubuntu 11.10, kernel 3.0.0-19. Refer to TN-101 if you need a custom VCP VID/PID in Linux. VCP drivers are integrated into the kernel.
Mac OS X 10.3 to 10.8	2012-08-10	2.2.18	2.2.18	2.2.18	-	-	-	-	Refer to TN-105 if you need a custom VCP VID/PID in MAC OS
Mac OS X 10.9 and above	2017-05-12	-	2.4.2	-	-	-	-	-	This driver is signed by Apple
Windows CE 4.2-5.2**	2012-01-06	1.1.0.20	-	-	1.1.0.20	1.1.0.10	1.1.0.10	1.1.0.10	
Windows CE 6.0/7.0	2016-11-03	1.1.0.22 CE 6.0 CAT CE 7.0 CAT	-	-	1.1.0.22 CE 6.0 CAT CE 7.0 CAT	1.1.0.10	1.1.0.10	1.1.0.10	For use of the CAT files supplied for ARM and x86 builds refer to AN_319
Windows CE 2013	2015-03-06	1.0.0	-	-	1.0.0	-	-	-	VCP Driver Support for WinCE2013

*Includes the following version of the Windows operating system: Windows 7, Windows Server 2008 R2 and Windows 8, 8.1, Windows server 2012 R2, Windows Server 2016 and Windows 10. Also, as Windows 8 RT is a closed system not allowing for 3rd party driver installation our Windows 8 driver will not support this variant of the OS. You must use the Windows RT build for this platform.

Figura A-22: Driver FTDI.

Tal como se muestra en la figura A-22, existen diversas versiones de driver dependiendo de la arquitectura del procesador que se disponga en el equipo (x86, x64 PPC), quedando como obligación del usuario corroborar en que arquitectura está trabajando su equipo.

Una vez descargado el driver, se procede a la instalación de este. El proceso es bastante similar al mostrado en Windows (mencionado anteriormente), a continuación, se muestra la ventana del software FTDI en la figura A-23



Figura A-23: Instalación de software FTDI.

Para instalar correctamente el software FTDI, basta seguir las instrucciones mostradas en pantalla (ídem con las de Windows). Posteriormente, se recurre a descargar el software Horus para Mac en el siguiente enlace:

- <https://www.bq.com/es/support/ciclop/support-sheet>.

Luego se busca “Horus v.0.1.2.4 Mac 10.9 o superior” esta ventana es mostrada en la figura A-24 (mostrada anteriormente).

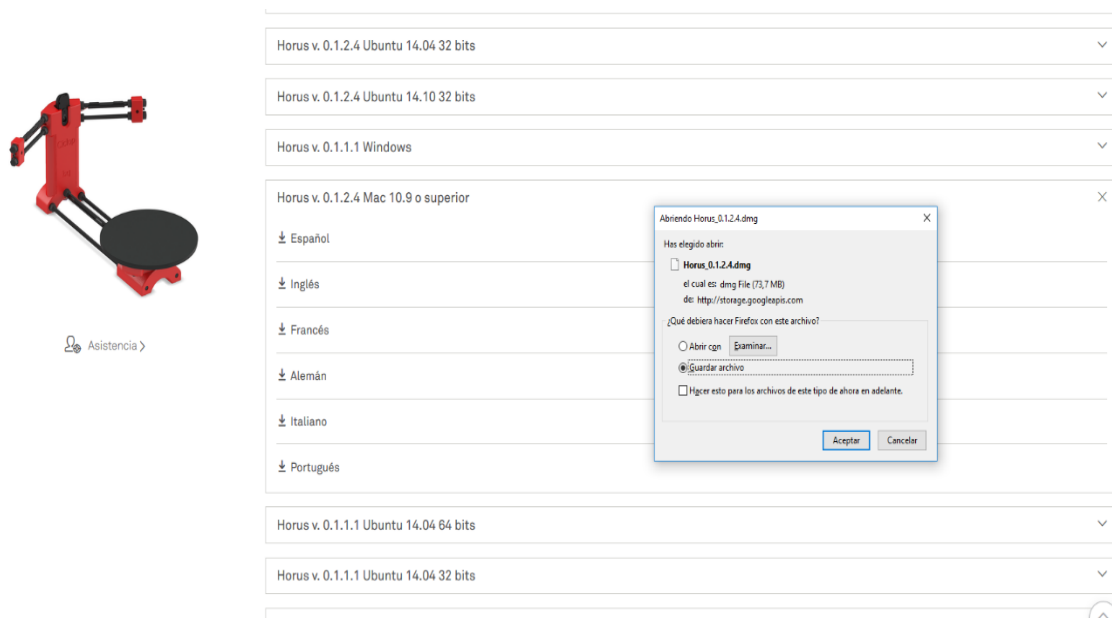


Figura A-24: Descarga de Horus para Mac.

Tal como se muestra en la figura A-24, al buscar el idioma que elija el usuario, aparecerá el archivo de descarga con el nombre “Horus_0.1.2.4.dmg”. Para poder instalar el software Horus basta con ejecutar el fichero descargado anteriormente, sin embargo, si no es posible instalar es necesario recordar que Mac deja instalar aplicaciones solo de Mac App Store (por defecto), por esta razón se hace necesario abrir la sección: “Preferencias del sistema> Seguridad y Privacidad”. En esta sección se hace necesario recurrir a la parte de “Abrir igualmente”, la cual es mostrada en la siguiente figura A-25.

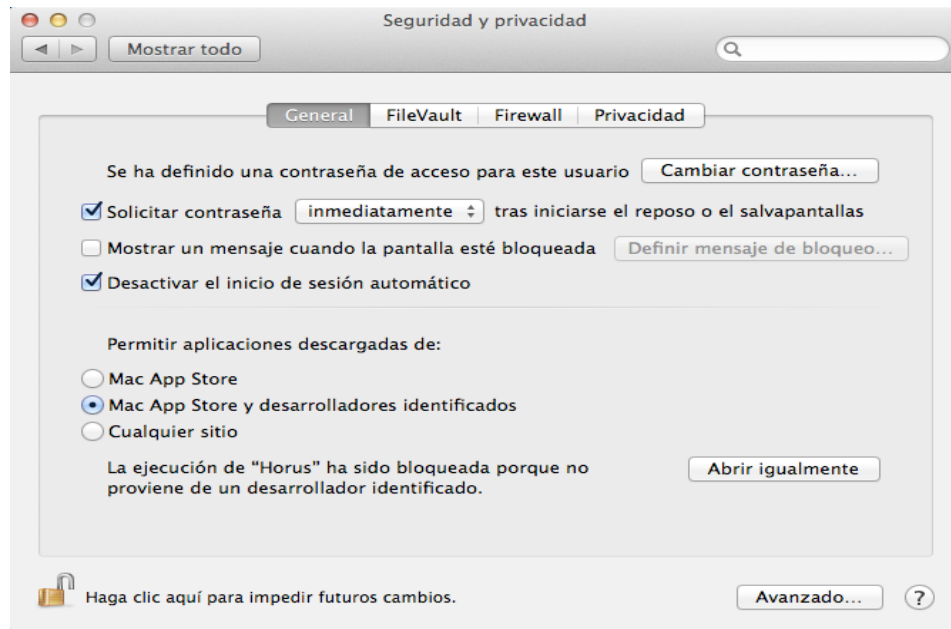


Figura A-25: Ejecución del instalador de Horus.

Finalmente, una vez finalizado el proceso mostrado en la figura A-25 aparecerá la ventana mostrada A-26 y esta sección quedaría mover el icono de Horus a la carpeta “Applications”.

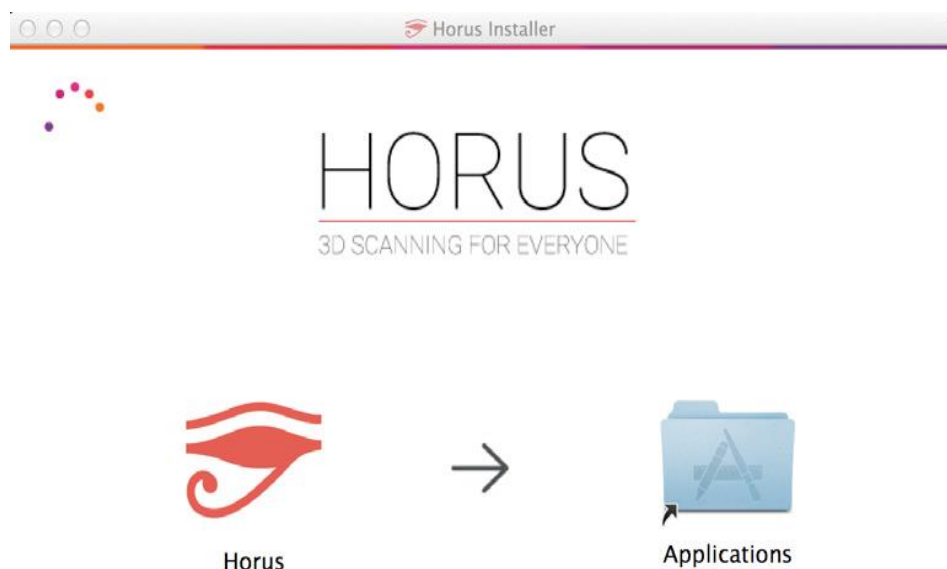


Figura A-26: Instalación de software Horus.

B Guía de montaje

En esta sección se detallarán los pasos empleados para la construcción del escáner 3D propuesto al inicio de este proyecto (Figura B-1), en donde se detalla desde la construcción hasta el montaje de los componentes electrónicos que componen el escáner.

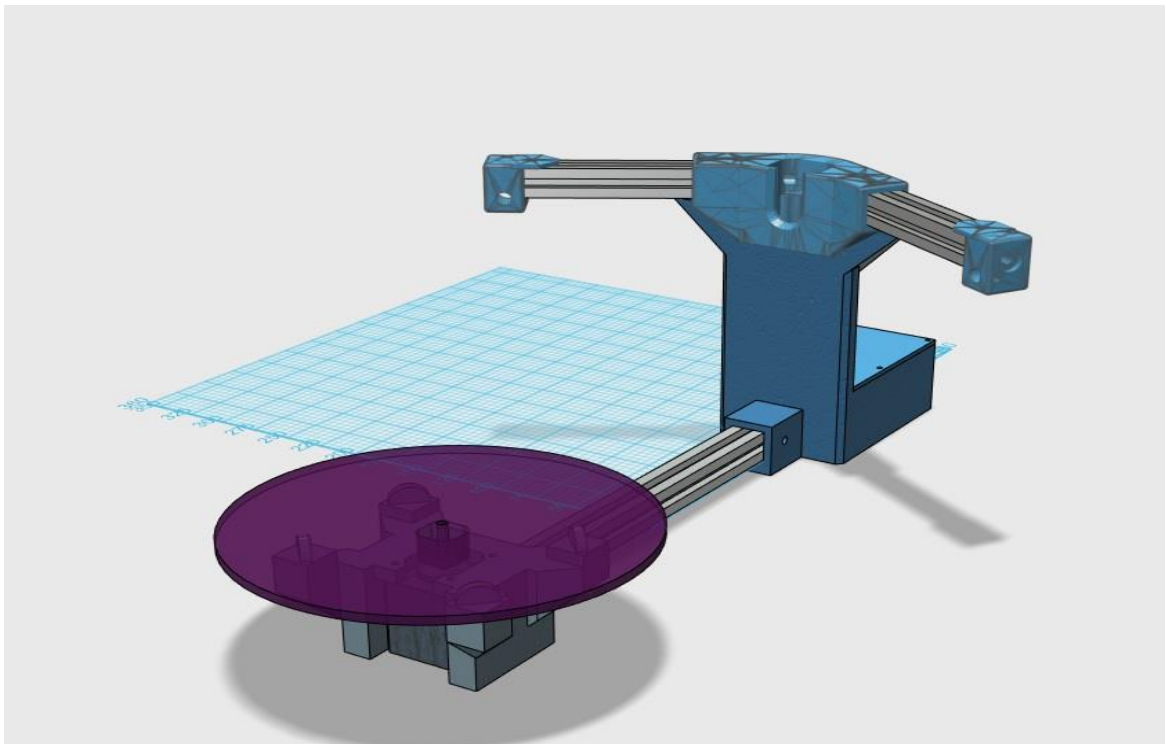


Figura B-1: Diseño final del prototipo.

B.1 Componentes

A continuación, se presenta una breve tabla explicativa que muestra los componentes que conforman en su totalidad el escáner.

Foto	Descripción	Cantidad
	Caja que posee la electrónica y soporte de la cámara.	1
	Soporte para los láseres.	2
	Perfil de aluminio 2020 de 290mm utilizado para la base del escáner.	1
	Perfil de aluminio 2020 de 130mm utilizado para los láseres.	2
	Base del motor.	1
	Componente cámara.	1

	Base (Disco de la plataforma).	1
	Motor.	1
	Tapa de electrónica.	1

B.2 Indicaciones

A continuación, se presenta la imagen correspondiente a la caja que posee la electrónica y que es el soporte de la cámara, dentro de la figura B-2 se exhibe circunferencias rojas, que corresponden a los orificios donde se atornillará la placa Arduino para el montaje de la electrónica.

En una primera instancia se procede a atornillar la placa con los tornillos suministrados (M3x10 mm) a la base del escáner (orificios en rojo).

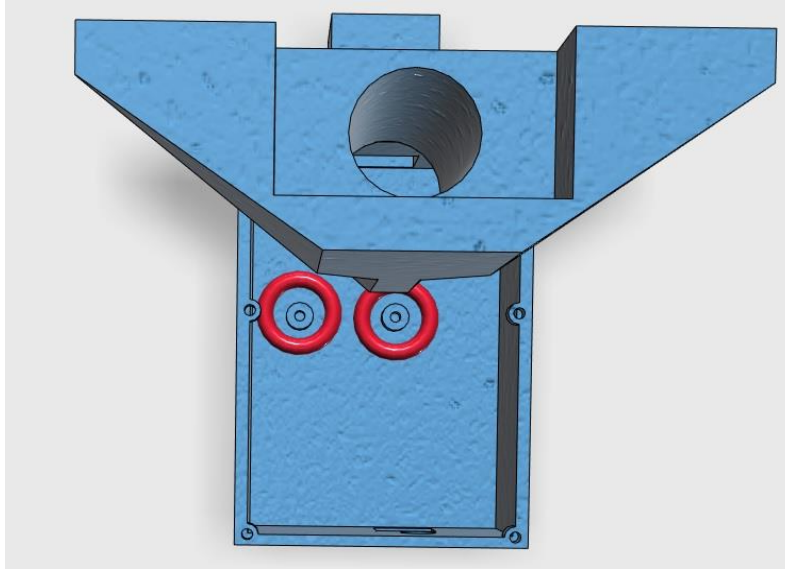


Figura B-2: Diseño de la base del motor (prototipo).

Posteriormente, se procede a instalar la Shield en el Arduino tal como se muestra en la figura 1.3. Es importante, considerar que, para evitar mayores problemas con cortocircuitos, se recomienda utilizar cinta aislante, recubriendo el conector USB de la placa Arduino, ya que la cercanía entre la Shield (conector de 12v) con el conector USB, en algunos casos ha producido cortocircuito entre las placas, ya que el conector es metálico.



Figura B-3: Placa instalada en escáner.

A continuación, se debe instalar los jumpers que controlan la cantidad de pasos que dará el motor Nema 17, en este caso para el controlador A4988 para obtener los 16 micro pasos, se requiere conectar 3 jumpers debajo del driver a4988. Tal como se muestra en la figura B-4, se destaca con rojo la posición donde deben estar ubicados estos.



Figura B-4: Shield con jumpers.

Una vez finalizado el proceso anterior, se procede a instalar el driver a4988 con la posición mostrada en la figura B-5. Donde se muestra claramente que el tornillo del potenciómetro con el cual se regula la tensión queda en la misma posición que el conector de 12V (alimentación).



Figura B-5: Shield con Driver.

En la figura B-5, se muestra además el cable del motor el cual va en la parte inferior del driver. Es necesario respetar el mismo orden de colores que están en el motor (donde va unido), un mal conexionado inducirá en una inversión en el sentido de giro del motor.

Luego, se procede a la instalación de los brazos de 130mm, los cuales se deben introducir en el espacio disponible en el centro de la base, basta introducirlos con algo de presión para posteriormente atornillarlo con los 4 tornillos que se incluyen (M8x 25mm), tal como se muestra en la figura B-6.

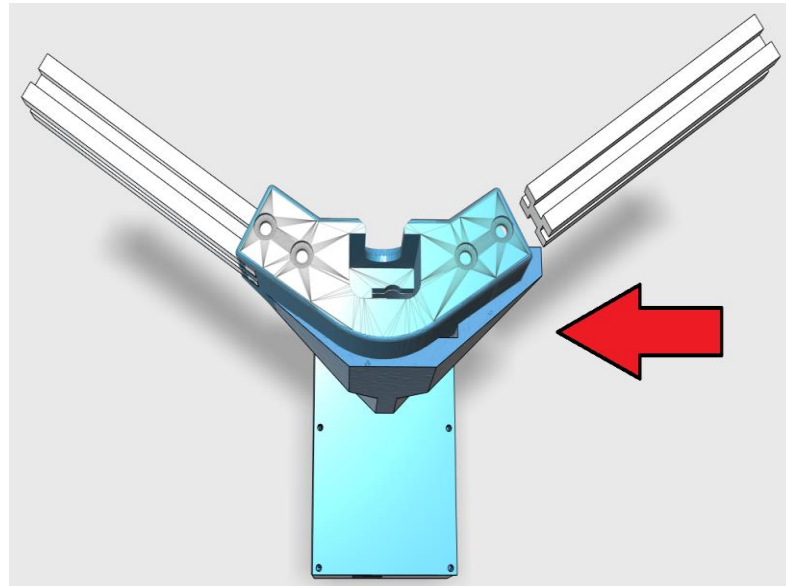


Figura B-6: Montaje de brazos.

Cuando se termina el proceso de instalación de los brazos, comienza la instalación de los láseres, para ello se montará en el perfil 2020 (brazos), los soportes para los láseres. Para poder dejarlos fijos, se emplearán tornillos M8x25mm, los cuales deben ser atornillados en los brazos para poder dejarlos fijos, tal como se muestra en la figura B-7.

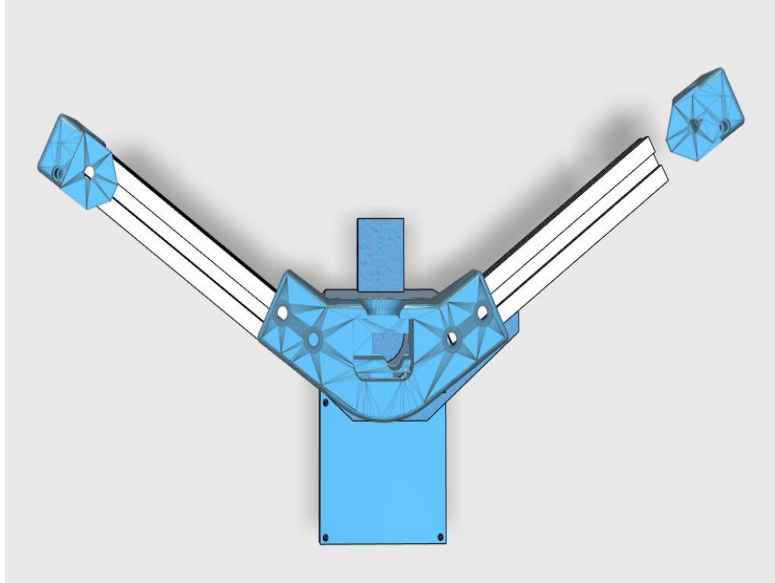


Figura B-7: Montaje de soporte de láseres.

Una vez instalados, se procede a introducir los láseres dentro del orificio que posee el soporte, basta con realizar un poco de fuerza para hacerlos entrar. Una vez que se pasen los láseres, por el costado del perfil 2020 se pasaran los cables correspondientes a los láseres (posee una ranura), además se introducirá por el centro de la base el cableado correspondiente a los láseres, para así llegar a la Shield. Es posible observar dicho componente en la siguiente figura B-8.



Figura B-8: Montaje de láseres.

A continuación, se procede con la instalación del perfil 2020 de 290mm. Este debe ser puesto en su lugar con un poco de fuerza, empujando tal cual como se muestra en la figura B-9 (flecha roja), para posteriormente, atornillar los tornillos (M3x12mm). Es importante que esta pieza quede en correcto funcionamiento, dado que será el soporte de unión entre la base de la cámara y el objeto.

La rigidez de esta le brindará una superior calidad y resistencia a la hora de recibir cualquier golpe o los movimientos propios que generará el motor al hacer a andar el sistema.

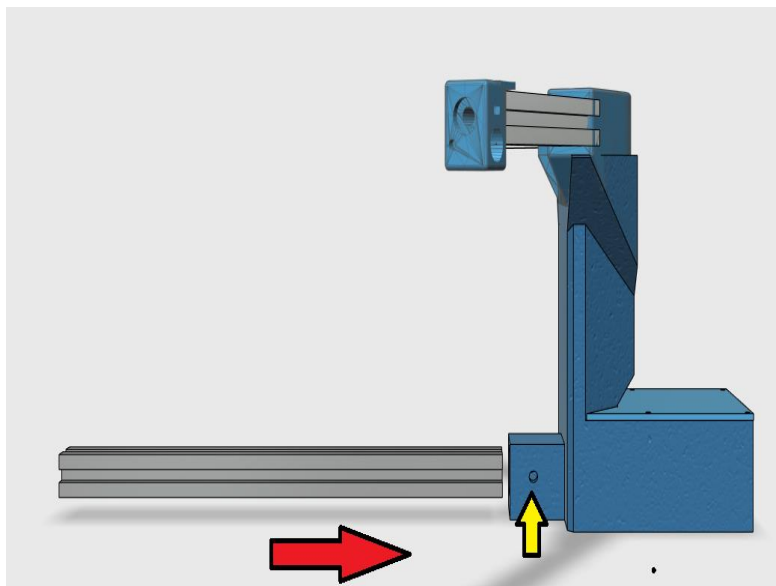


Figura B-9: Montaje perfil 2020 central.

Posteriormente, se procederá a atornillar la base del motor con el perfil 2020 que une la base de la cámara con la mencionada anteriormente, esto queda ilustrado en la figura B-10, en este punto también se utilizarán los tornillos M3x 12mm y son colocados en el punto donde se encuentra la flecha de color rojo.

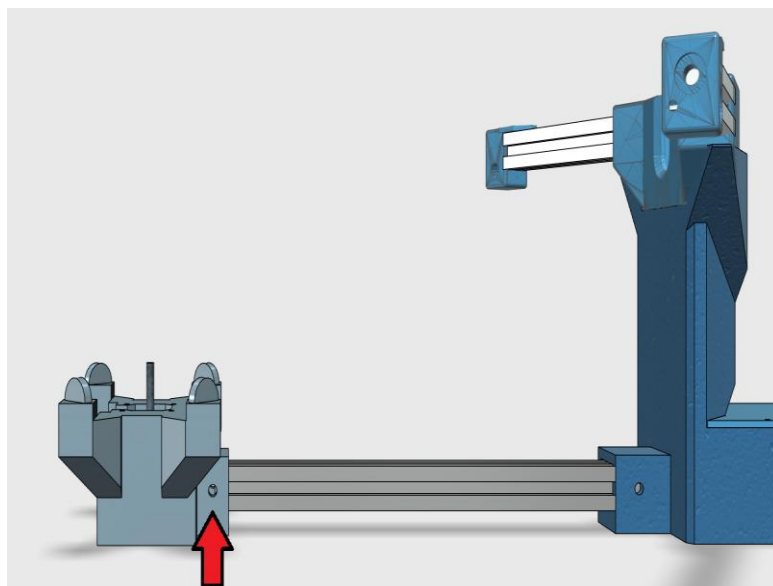


Figura B-10: Montaje perfil 2020 en base de motor.

Para finalizar con el armado del escáner se procederá a atornillar los correspondientes tornillos pertenecientes al motor dejándolo fijo y así se evitarán las posibles vibraciones que pueda generar el motor al girar. La ubicación de los tornillos queda exhibida en la imagen B-11 y lo último, es agregar sobre el eje del motor la base que soportará la masa del objeto que se quiera escanear el cual es mostrado en la figura B-12.

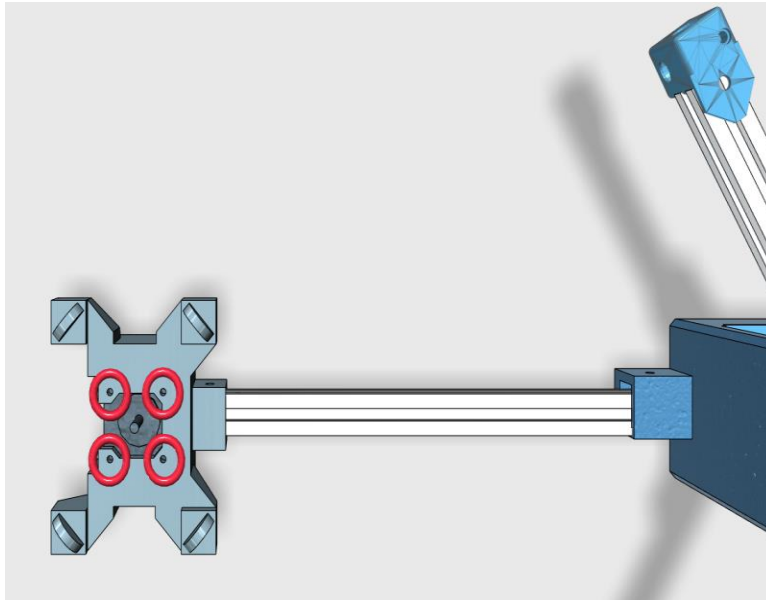


Figura B-11: Montaje de motor en base.

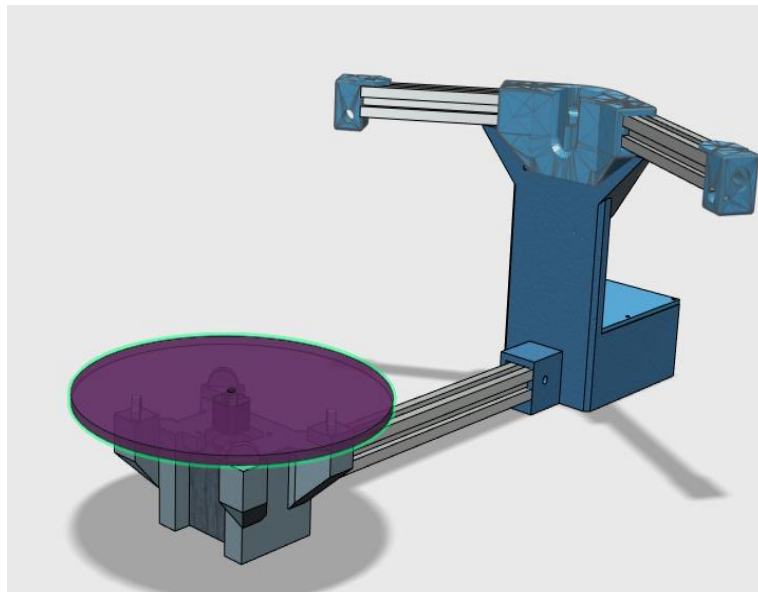


Figura B-12: Escáner terminado.

C Lista de costos

Se hace necesario a la hora de desarrollar el proyecto, elaborar una tabla que entregue un listado con el resumen de materiales utilizados y además de los costos aproximados de esto, con lo cual podemos inferir el rango de costo del escáner.

Tabla C-1: Presupuesto del proyecto.

Características	Valores
Motor Nema 17	14.990
Kit tornillo m3 10mm	5.000
Perfil de aluminio 2020 1m	14.000
Tornillo m6 30mm	1.500
Tornillo m3 30mm	2.000
PLA (material impresión)	35.990
Arduino	6.000
Shield	6.000
Cámara Logitech c270	22.990
Driver	3.000
Láseres	3.400
Total del proyecto (incluye impresión)	114.870

*Nota: El valor puede disminuir aún más, si es que todos los productos son importados en vez de ser comprados en el país, en el caso del Arduino y la Shield se colocó \$6.000 en cada uno, ya que el valor comercial es \$12.000 por el kit que incluye el Arduino con la Shield. Además, cabe destacar que todos los valores son en moneda chilena (CLP).

D Calibración

En este capítulo se mostrará todo lo relacionado a la interfaz del software Horus, los primeros pasos y también la forma en que se comienza a calibrar el escáner 3D. Es importante destacar que para realizar una correcta calibración existen parámetros externos, como la luz. Siendo este último un parámetro importante tanto en calibración, como posteriormente en la toma de datos y creación de la nube de puntos.

D.1 Ejecutando Horus por primera vez.

Para ejecutar Horus, basta recurrir al lugar de instalación o escritorio y haciendo doble clic en el logo de Horus, abriremos el programa. Cuando carga el programa se mostrará una ventana como la que se expone en la figura D-1

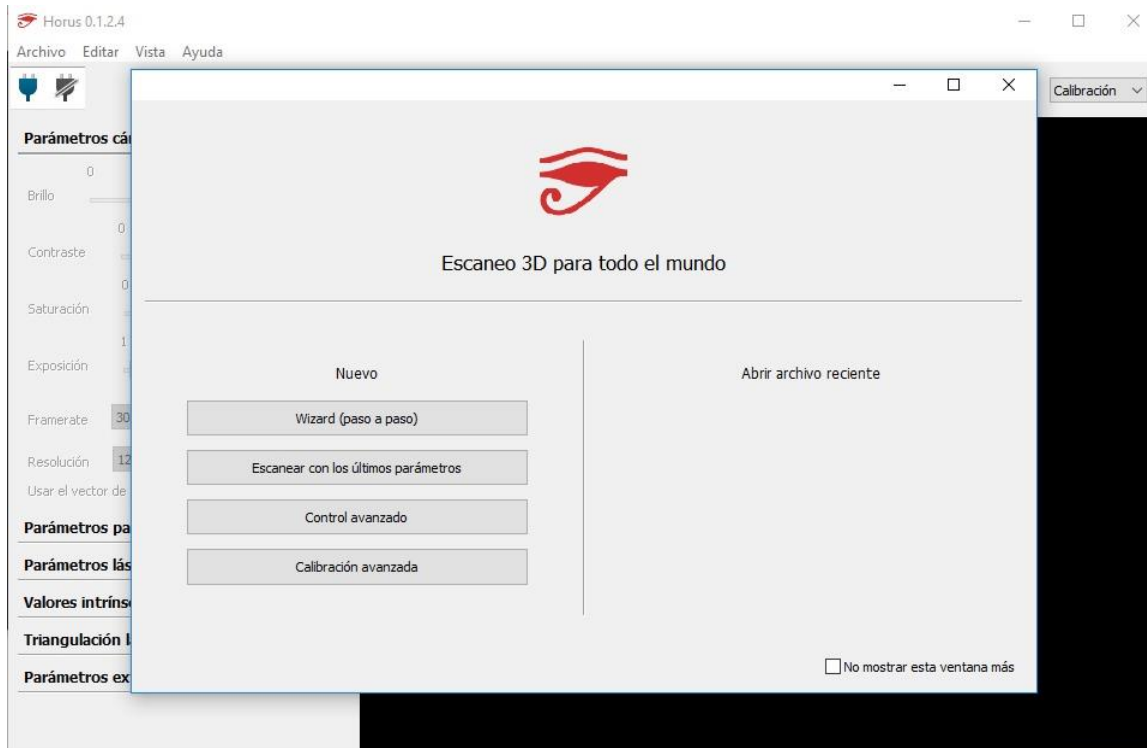


Figura D-1: Arranque del software Horus.

De acuerdo con la figura D-1 existen dos columnas disponibles, en la izquierda se tienen 4 opciones disponibles (Wizard, Escanear con los últimos parámetros, Control avanzado, calibración avanzada).

El Wizard mode, permite realizar los procesos de conexión, calibración y sintonización de parámetros de forma muy simplificada, lo cual es una gran ayuda para las personas que no poseen mayor conocimiento técnico respecto al escáner.

En el control avanzado es posible tener un control total de los elementos que conforman el escáner, los cuales son el motor, láseres y cámara. Este modo permite ajustar manualmente diversos parámetros como: brillo, contraste, saturación, exposición, framerate, resolución, distorsión.

La calibración avanzada faculta al usuario para cambiar los parámetros de la cámara y poder así tener una calibración óptima. Existen parámetros importantes para calibrar en este punto como: calibración entre cámara y módulos laser, calibración entre cámara y plataforma, calibración entre el sensor de la cámara y el lente. Dentro de esto, la calibración de la cámara hace referencia a adquirir los parámetros de la cámara como: distancia focal, centro óptico y distorsión del lente. Mientras que la calibración de los láseres se puede obtener los planos de cada láser y finalmente la calibración del disco, la cual permite obtener la posición y rotación de este.

Mientras que, en la columna de la derecha, se muestran los modelos escaneados recientemente donde se pueden abrir estos últimos. También se encuentra una opción en la esquina inferior derecha que permite ocultar la ventana cada vez que se inicia el software Horus.

D.2 Ventana Principal.

Dentro de la ventana principal encontramos tres elementos destacados en la figura D-2, Menú, Barra de tareas y Menú desplegable.

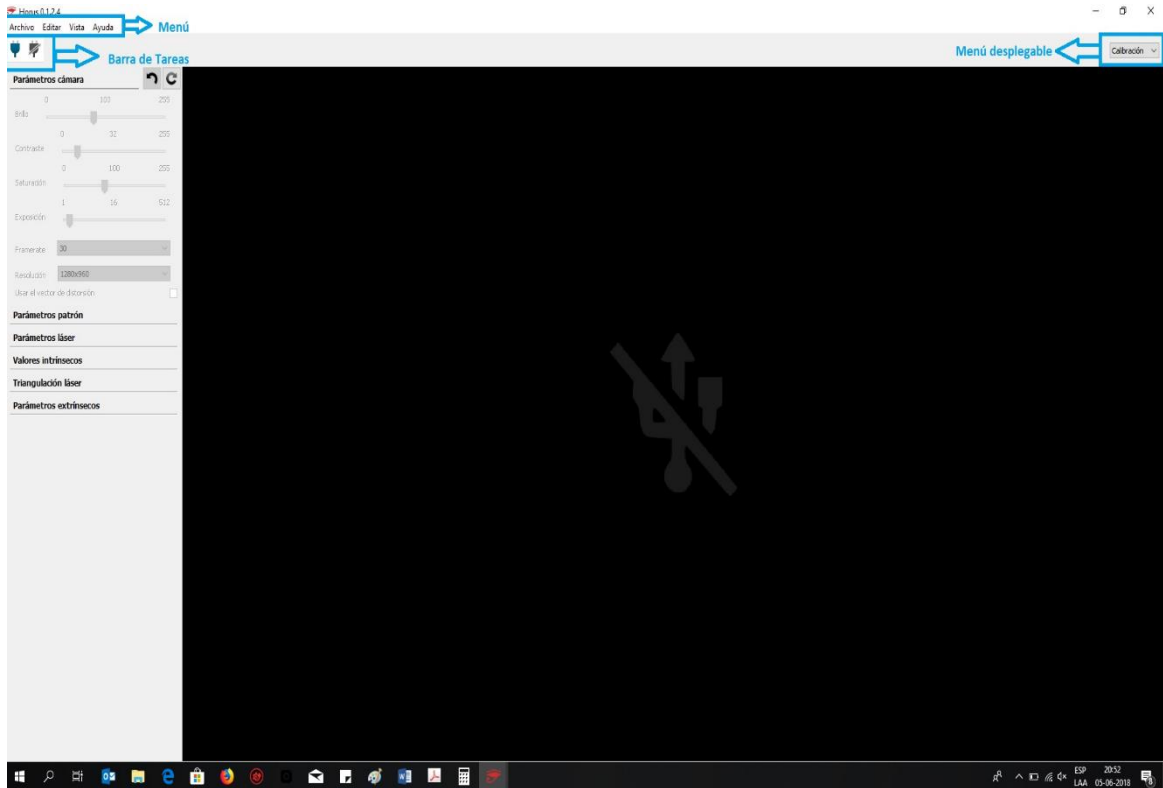


Figura D-2: Ventana principal del Horus.

D.2.1 Menú

Dentro de la sección denominada menú se 4 secciones definidas como: Archivo, editar, vista y ayuda.

- Archivo

En la parte de archivo, se pueden realizar las siguientes funciones:

- Ejecutar el Wizard (explicado anteriormente).
- Cargar, guardar y borrar modelos.
- Abrir, guardar y restablecer perfiles, donde los perfiles son archivos que poseen una extensión “.ini” donde se almacenan los parámetros que emplea el programa.

La sección “Archivo” se muestra en la figura D-2.1

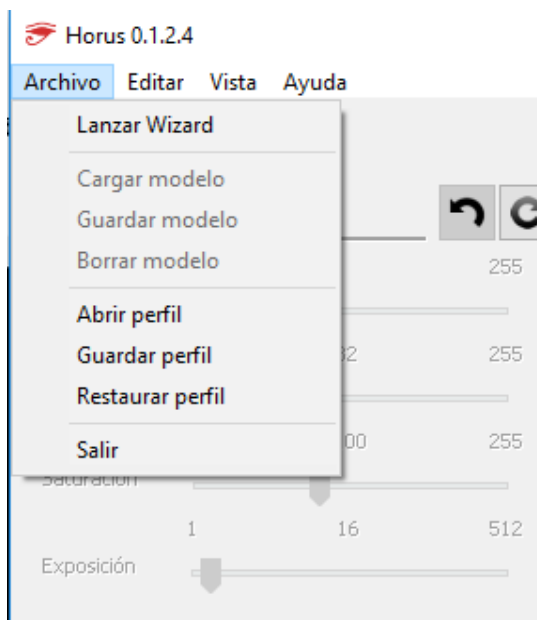


Figura D-2.1: Sección Archivo.

- **Editar**

En la parte de “Editar” encontramos las preferencias del Horus, en donde se abrirá una ventana que mostrará las siguientes opciones (mostradas en la figura D-2.2).

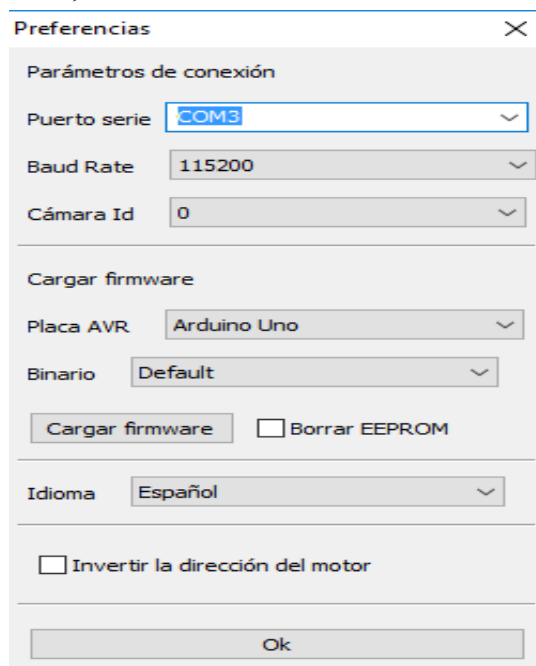


Figura D-2.2: Sección Editar (Preferencias).

Los “Parámetros de conexión” mencionados en la figura D-2.2, hacen referencia al puerto de serie que está utilizando el Arduino, la velocidad de la comunicación (Baud Rate) y el Id que identifica la cámara conectada en el computador. En el caso hipotético de no reconocido el Arduino o la cámara, se podrá observar que no existirán opciones como en la figura mostrada.

La opción “Cargar firmware”, se encuentra placa AVR donde se puede elegir si la placa es un Arduino Uno o un BT ATmega328 (opciones viables para trabajar con el software Horus), en la pestaña “cargar firmware” permite leer el código que contiene el Arduino para que el control lo tome el software Horus, mientras que “Borrar EEPROM” es utilizada para borrar los datos contenidos en la memoria del Arduino.

La opción “Idioma” permite cambiar el idioma del software, donde se puede elegir: Inglés, Español, Francés, Alemán, Italiano y Portugués.

Finalmente, la sección invertir la dirección del motor, como bien dice su nombre es para invertir el sentido de giro de la base (donde está el motor).

- Vista

En esta sección encontramos 3 opciones: Panel, Video, Escena (Solamente disponible en “Escaneo”). Dichas opciones son mostradas en la figura D-2.3.

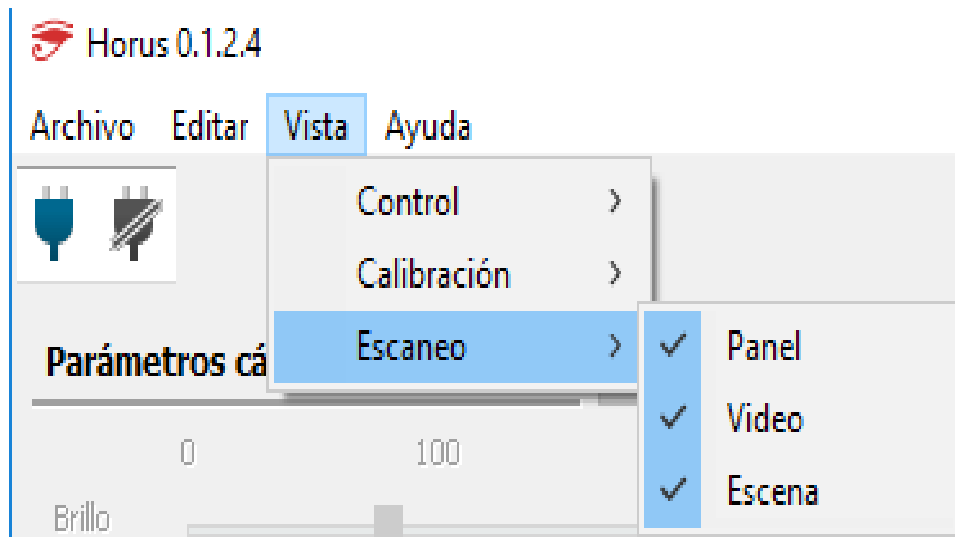


Figura D-2.3: Sección Vista.

En la opción Panel se exhiben las opciones y parámetros que pueden ser editables, al momento de situar el mouse en ciertas opciones y parámetros se muestra una descripción acerca de estos.

Por otro lado, Video muestra el video que está capturando en vivo la cámara, finalmente “Escena” es el visor 3D interactivo, donde se puede observar el modelo escaneado en las tres dimensiones.

- Ayuda

En esta sección se muestra información acerca de la aplicación, corroborar actualizaciones, acceder a la foto de BQ, Wiki y fuentes. Esta sección es mostrada en la figura D-2.4.

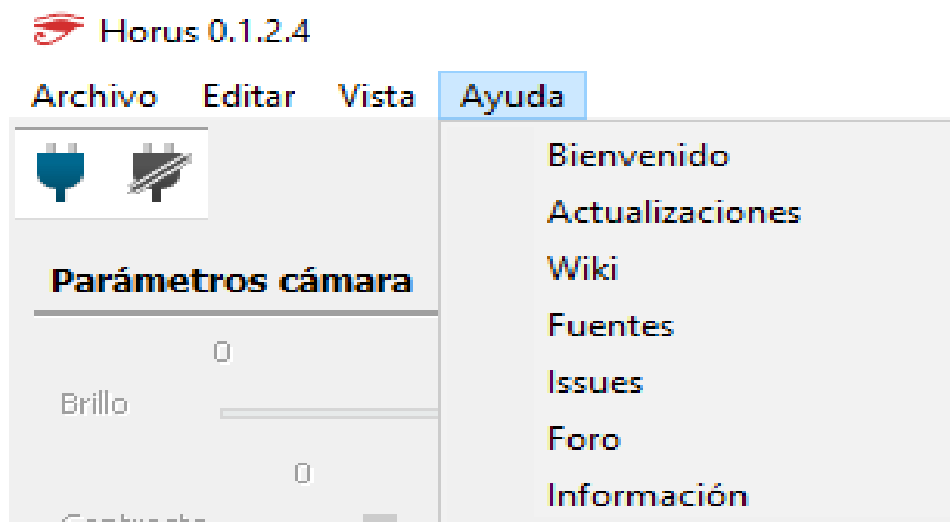


Figura D-2.4: Sección Ayuda.

D.2.2 Barra de tareas

En la barra de tareas encontramos dos botones, los cuales poseen la función de conectar o desconectar el Arduino al software Horus. Estos son mostrados en la siguiente figura D-2.5.

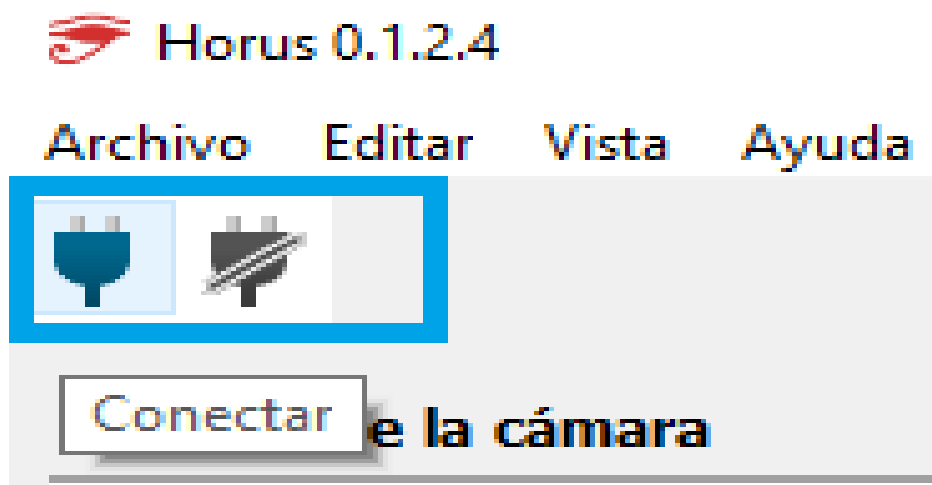


Figura D-2.5: Barra de tareas.

D.2.3 Menú desplegable

El menú desplegable permite acceder rápidamente a 3 opciones, las cuales son: Control, Calibración y Escaneo (ver figura D-2.6)

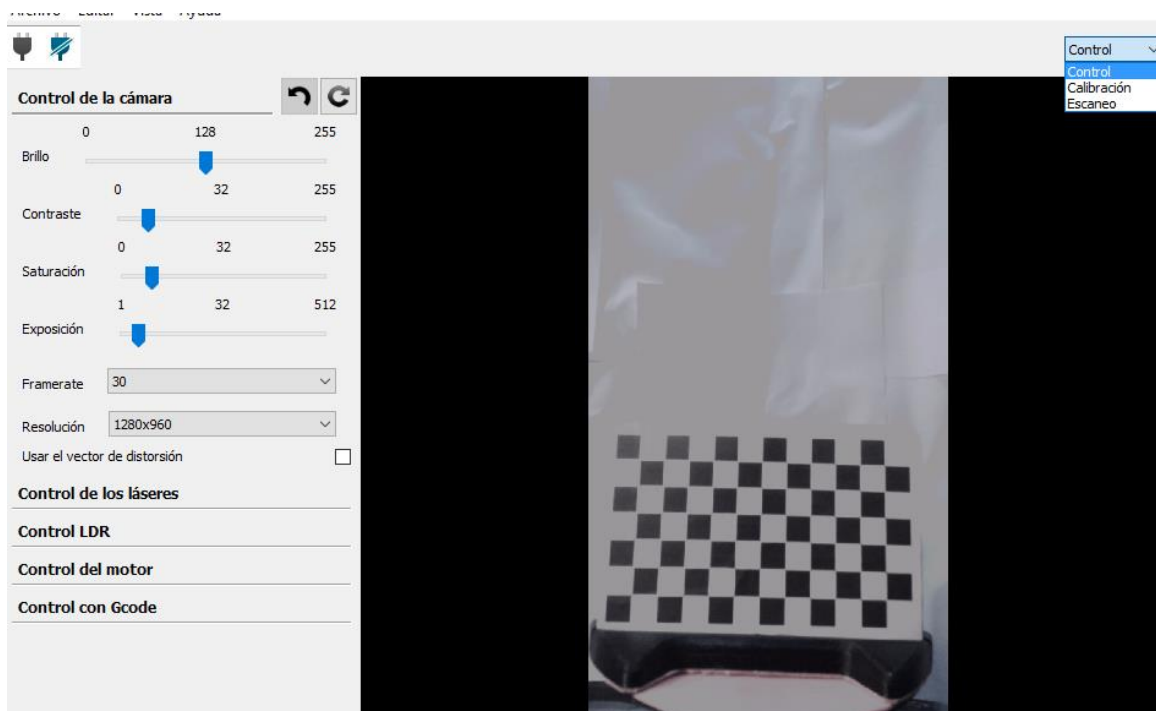


Figura D-2.6: Menú desplegable.

- Control

Esta sección permite controlar los componentes que conforman el escáner por separado (cámara, láseres, motor), tal como se muestra en la figura D-2.6.

D.2.3.1.1 Control de la cámara

D.2.3.1.2 Control de los láseres

D.2.3.1.2 Control del motor

Calibración

En esta sección tenemos 5 parámetros importantes a considerar, básicamente aquí se puede realizar una calibración para usuarios avanzados, ya que a diferencia del “Wizard” se trabajan todos los parámetros manualmente. Estos 5 parámetros importantes son:

- **Parámetros Cámara:** Modifica manualmente los factores mencionados anteriormente (Control de la cámara), su diferencia radica es que estos son para ser empleados en la calibración y no para probar la visualización de la cámara.
- **Parámetros Patrón:** Esta sección permite configurar los parámetros relacionados con el patrón de calibración, en esta sección aparece la tabla de ajedrez (como se muestra en la figura D-2.9). Por defecto, Bq ciclop fue calculado en base a cuadrados negros y blancos de 13 mm de lado. Además los valores de las filas y columnas corresponde a la cantidad de vértices reconocidos por la cámara, que son los que aparecen de color en la figura D-2.9. Dentro de esta sección es importante la distancia del patrón (ver figura D-2.9), la cual debe ser medida ojalá por un pie de metro, tal y como se muestra en la figura D-2.10, esta medida es la distancia en milímetros existente entre la base y el lado superior del primer cuadrado negro existente a la esquina inferior izquierda.

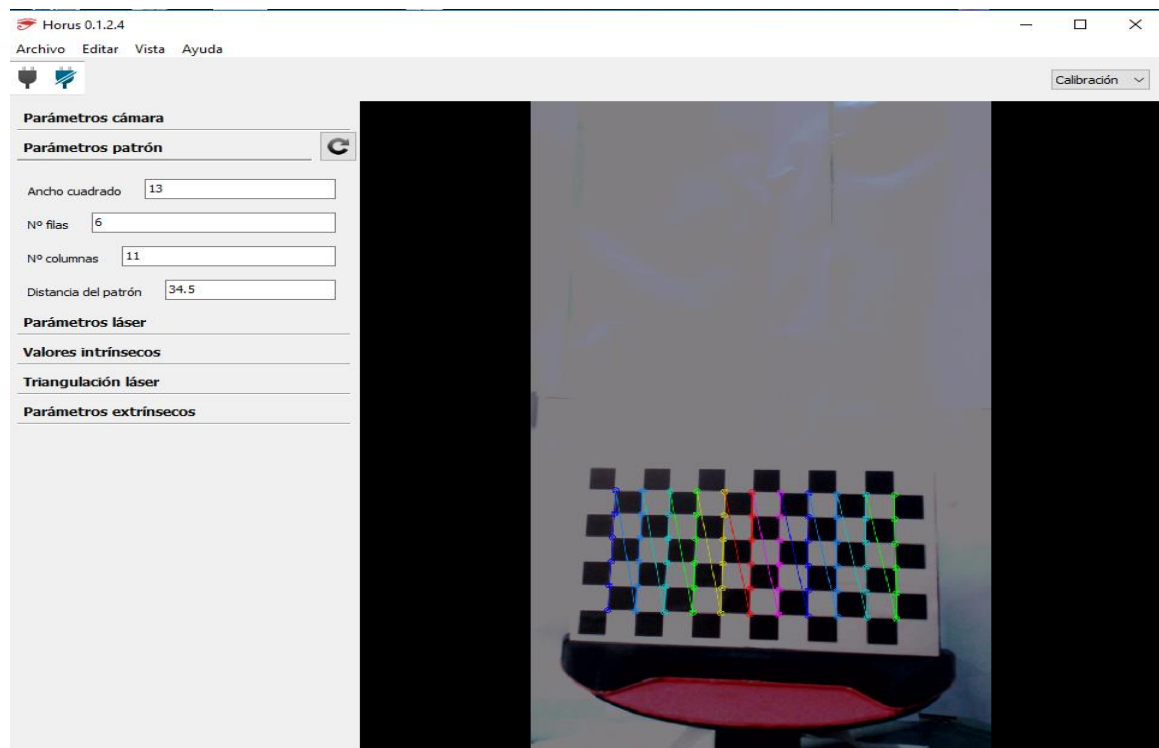


Figura D-2.9: Parámetros de Patrón.

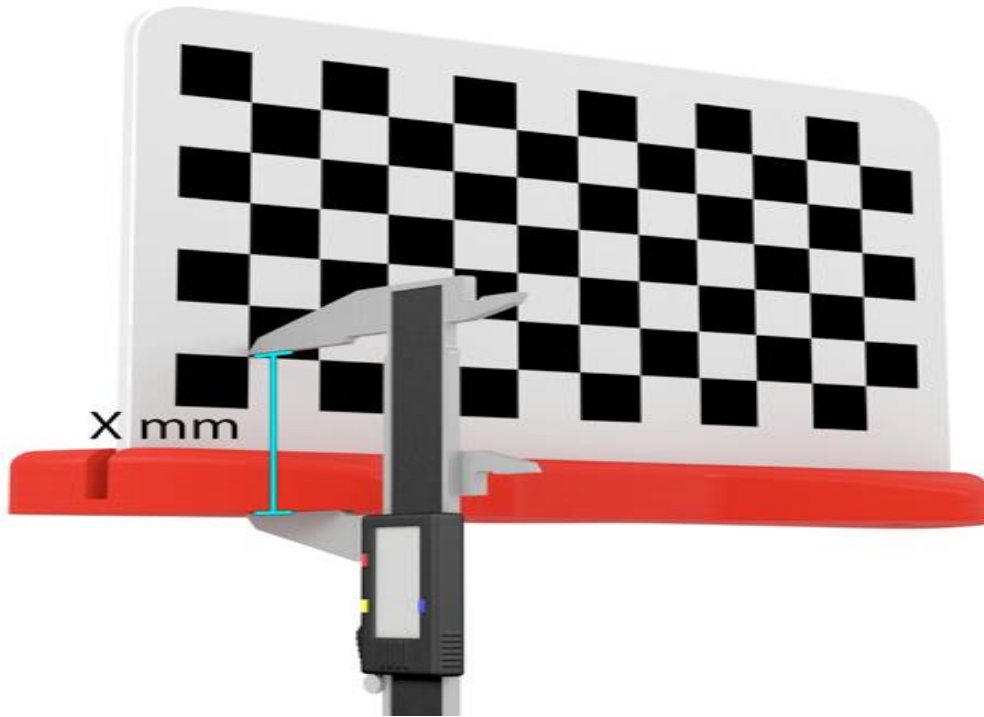


Figura D-2.10: Distancia del Patrón.

- **Parámetros Láser:** En esta sección al igual que la mencionada en la D-2.3.1.2, permite controlar los láseres del lado izquierdo y el derecho, también la verticalidad de estos.
- **Valores Intrínsecos:** Estos valores representan los valores internos de la cámara, tanto la distancia focal (Horizontal y vertical), centro óptico (Horizontal y vertical) y distorsión de la lente. Según Bq cliclop la modificación de estos valores es opcional, ya que los valores de la teoría a la práctica son muy cercanos. Sin embargo, se quieren calcular de todas formas, el proceso no difícil, basta con colocar en el botón iniciar y el programa solicitará realizar tomas de la tabla de ajedrez en distintas posiciones, tal como se muestra en la figura D-2.11 tomando 12 tomas en total, una vez que se termina el proceso, aparecerán los planos de las coordenadas de la cámara (ver figura D-2.12)

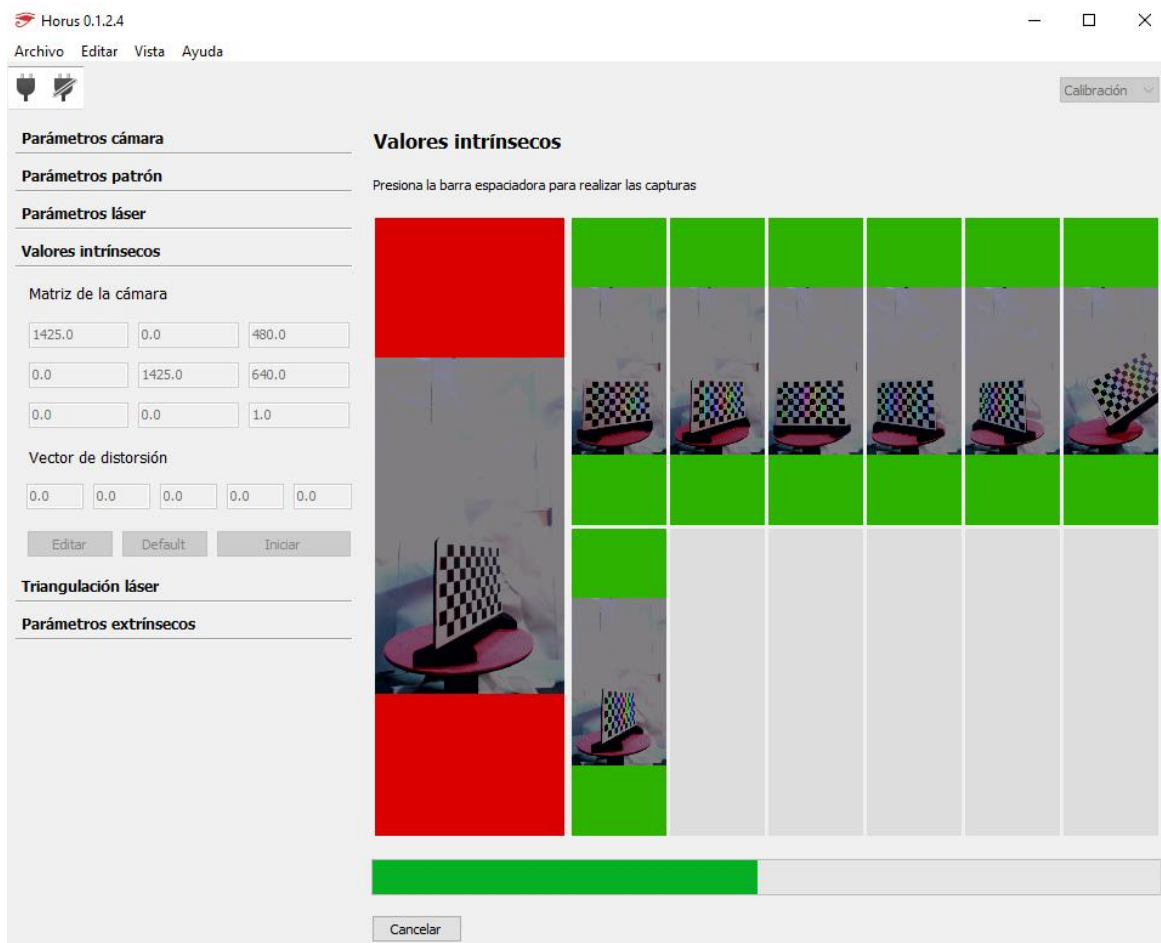


Figura D-2.11: Valores intrínsecos

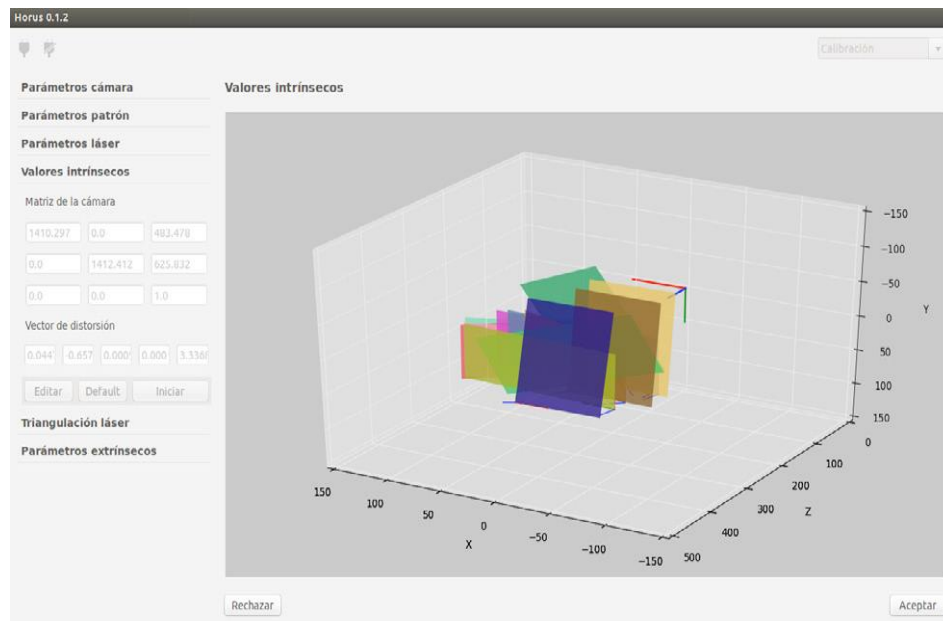


Figura D-2.12: Gráfico 3D con sistema de coordenadas de la cámara.

- **Triangulación Láser:** Este es un proceso bastante importante, ya determina la inclinación, distancia de la intersección entre los láseres y el plano central de la cámara. Para realizar la calibración, basta con realizar un clic en el botón iniciar para desplegar una ventana como la que se muestra en la figura D-2.13. Seguidamente, se realiza un clic en calibrar.

Es importante destacar que la base debe girar en sentido antihorario según se indica en la figura D-2.13.

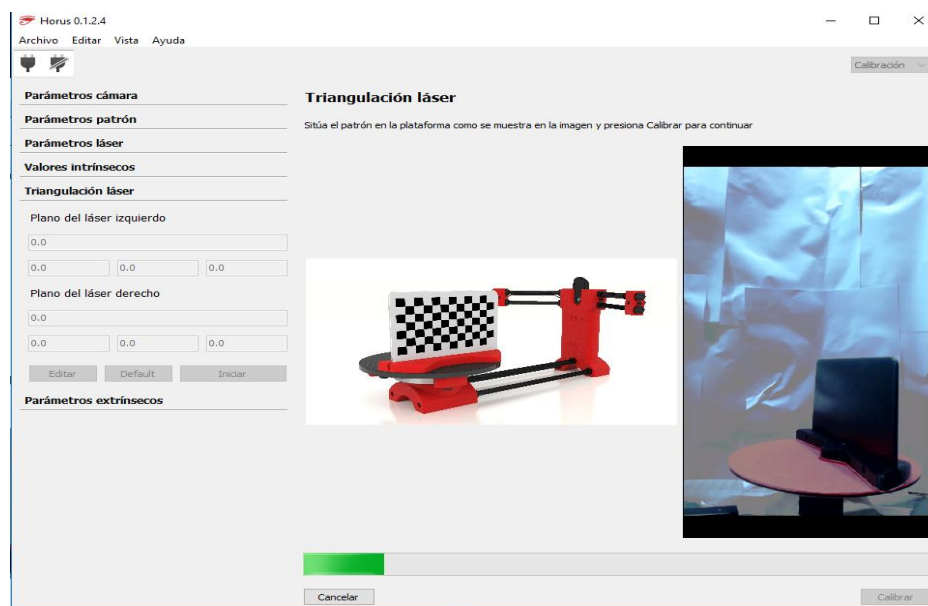


Figura D-2.13: Calibración del escáner.

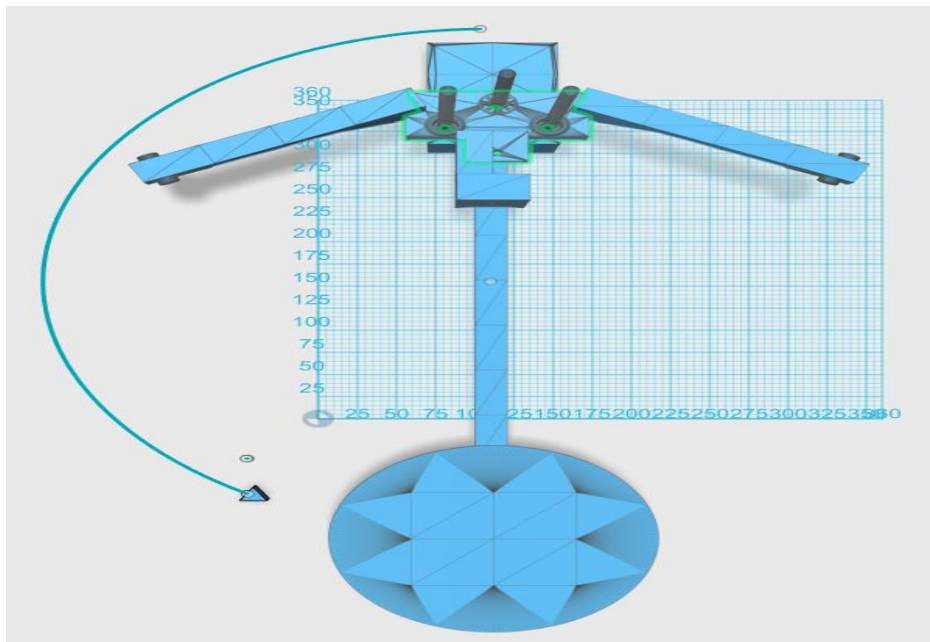


Figura D-2.14: Sentido de giro antihorario.

Una vez obtenida la calibración, se mostrarán los planos que definen la inclinación de los láseres y la profundidad que posee el patrón, tal como se muestra en la figura D-2.15.

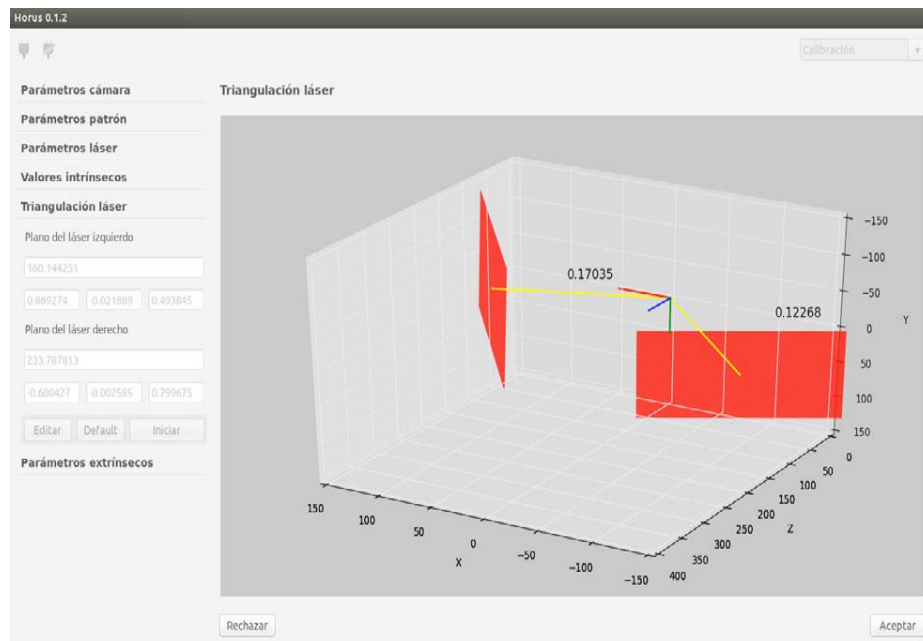


Figura D-2.15: Plano de los láseres en 3D.

- **Parámetros Extrinsecos:** Los parámetros extrínsecos son los que permiten obtener la posición y rotación del disco, por lo tanto también es fundamental al igual que el punto anterior. Para poder calcularlo, es necesario que la plataforma gire en sentido horario, es decir al contrario de lo mostrado en la figura D-2.14.

Para iniciar esta calibración se hace clic en el botón iniciar y a continuación en “calibrar”, esta ventana es mostrada en la figura D-2.16. Los resultados de esta calibración aparecen una vez finalizado el proceso, donde se muestra un gráfico que exhiben las muestras capturadas, como también el centro de rotación. Este resultado es mostrado en la figura D-2.17.

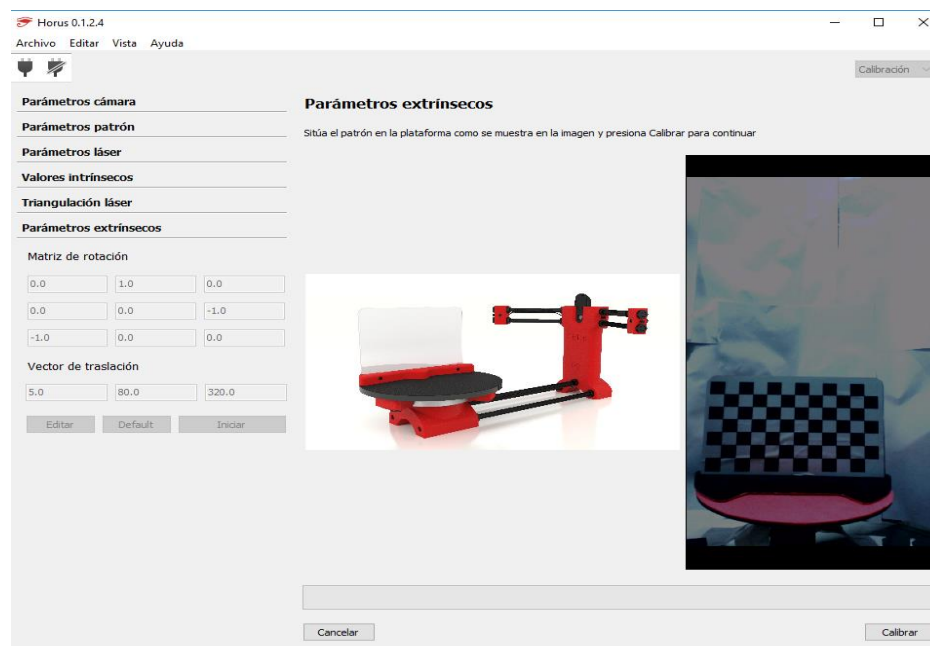


Figura D-2.16: Calibración de posición y rotación del disco.

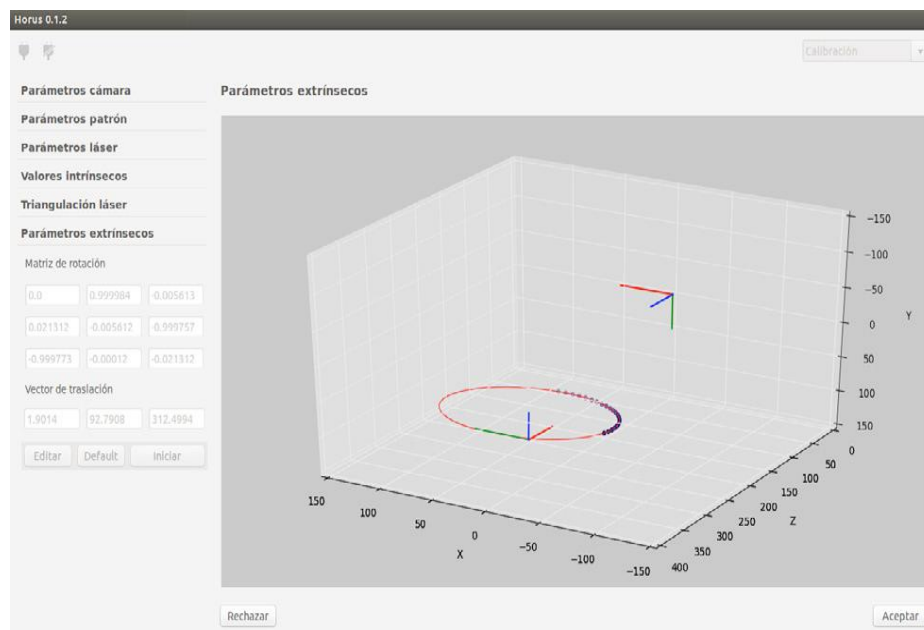


Figura D-2.17: Plano de posición y rotación del disco

D.2.4 Calibración mediante Wizard.

Para iniciar la calibración mediante Wizard, basta con ir al menú de “Archivo” y posteriormente a la sección de “Lanzar Wizard”. A continuación, se abrirá la siguiente ventana (mostrada en la figura D-2.17).

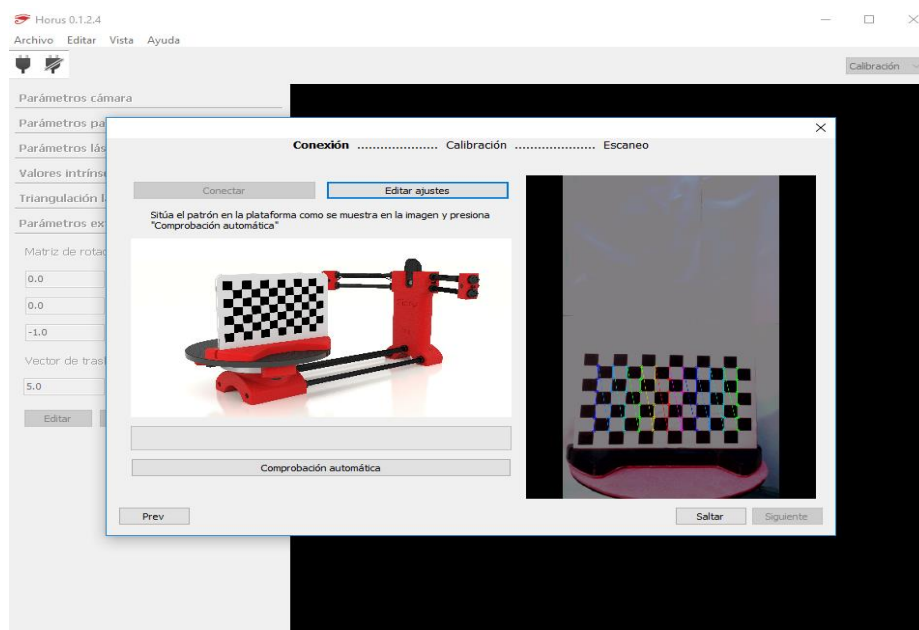


Figura D-2.17: Plano de posición y rotación del disco

Una vez que se abre la ventana mostrada anteriormente, se requiere colocar el tablero en la posición que muestra la imagen en la columna izquierda de la figura D-2.17.

Antes de comenzar con la comprobación automática es importante ir a la opción “editar ajustes”, ya que en esta podemos verificar la luminosidad y verificar la “distancia del patrón”. Tal como se muestra en la figura D-2.18

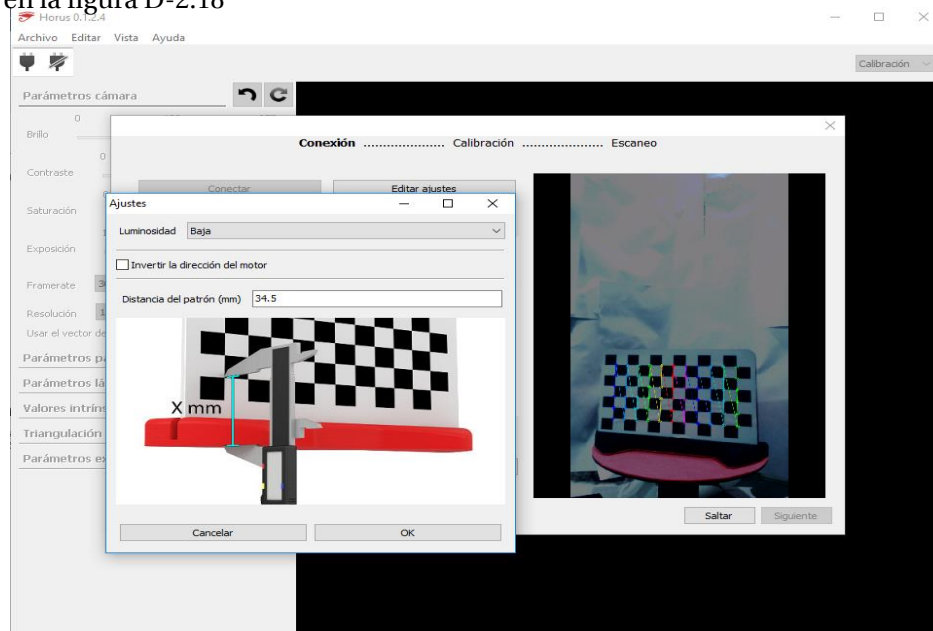


Figura D-2.18: Editar Ajustes.

La mejor opción de “luminosidad” será la que permita visualizar de mejor forma el patrón (donde se ven la línea de colores). Finalizado este proceso, se realiza la comprobación automática,

Si resulta correcta la comprobación automática se salta al ítem de calibración donde aparecerá una ventana como la mostrada en la figura D-2.19.

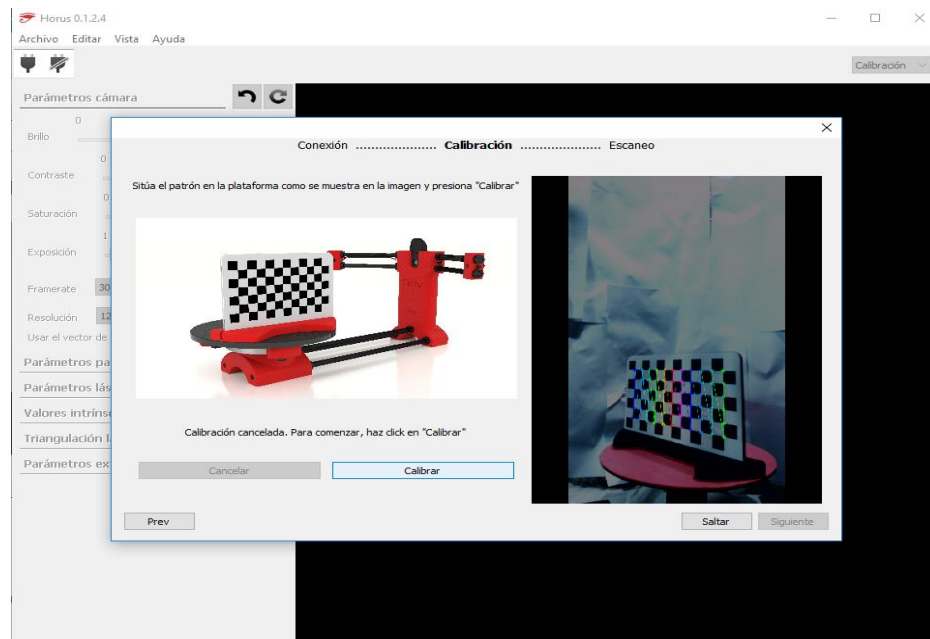


Figura D-2.19: Calibración.

En la calibración, basta con realizar un clic en el botón “calibrar” para que el programa solo comience con el proceso.

Para finalizar con el “Wizard”, si termina correctamente el proceso, se mostrará el ítem de “Escaneo”, donde se configurarán las preferencias del escaneo, existen disponibles 3 ítems importantes los cuales son:

- Resolución: Existen dentro de esta opción 3 posibilidades, “alta, media, baja” resolución. Es importante recordar, que a mayor resolución mayor tiempo se demorará el escaneo.
- Láser: El software permite trabajar con 1 solo láser (eligiendo cual escoger) o ambos, al igual que el caso anterior la utilización de dos láseres empleará mayor cantidad de tiempo para obtener los puntos.
- Tipo de Escaneo: Existen dos posibilidades, que sea con textura o simple. El simple utiliza una imagen por ciclo para obtener una nube de puntos, sin capturar el color del objeto, mientras que el con textura utiliza dos imágenes para capturar el láser, así muestra los colores reales del objeto.

Este último paso, queda mostrado en la figura D-2.20.

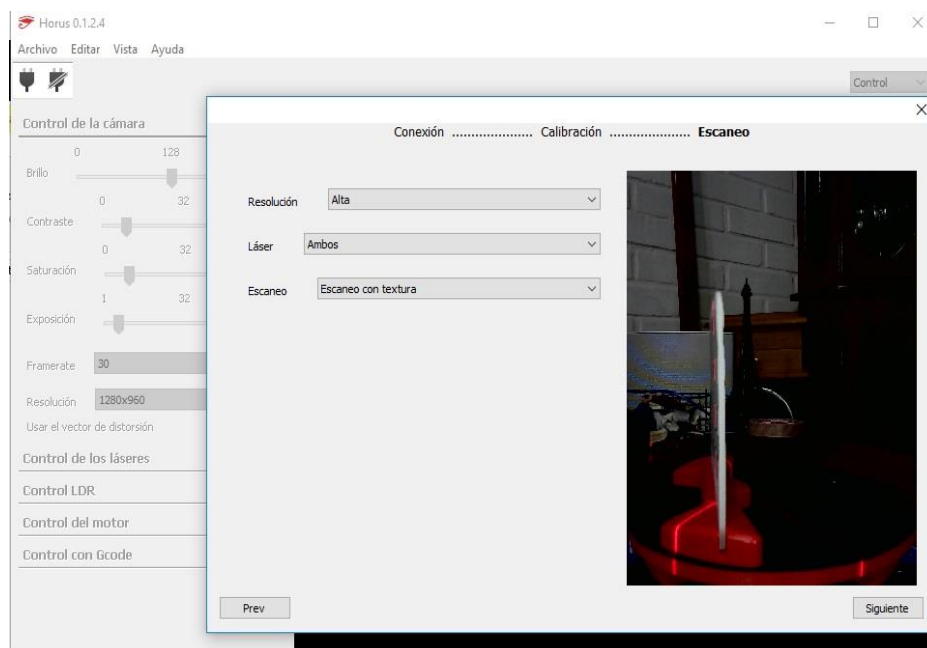


Figura D-2.20: Escaneado.