

Prototipo para la automatización digital de objetos 3D apoyado en hardware y software libre

*Prototype for the digital automation of 3d objects supported in hardware
and free software*

*Protótipo para automação digital de objetos 3D suportados por hardware e
software livre*

Enrique Solis Cielo

Universidad Autónoma del Estado de México, México
e.solis92@hotmail.com

Oziel Lugo Espinosa

Universidad Autónoma del Estado de México, México
ozieluz@gmail.com

Joel Ayala de la Vega

Universidad Autónoma del Estado de México, México
joelayala2001@yahoo.com.mx

Alfonso Zarco Hidalgo

Universidad Autónoma del Estado de México, México
azarcox@hotmail.com

Resumen

El prototipo presentado en este artículo es una combinación de hardware y software libre para la captura de imágenes en tres dimensiones (3D). Se da prioridad al uso de hardware libre y la reducción de costos, con el objetivo primordial de asegurar que el producto pueda ser desarrollado o adquirido por cualquier institución u organización, ya sea de investigación o desarrollo, de la forma más accesible.

Para lograr el prototipo, este fue subdividido en: la creación de un estudio fotográfico; la automatización del movimiento de la cámara utilizando dos motores a pasos y, como elemento de control, la placa Arduino Mega 2560; la selección del software que se encarga de generar el modelo 3D; y, por último, el desarrollo del software que permite la interacción de todos estos elementos.

Palabras clave: automatización, hardware libre, modelado 3D (3 dimensiones), sistemas embebidos, software.

Abstract

The prototype presented in this article is a combination of free hardware and software for the capture of images in three dimensions (3D). Priority is given to using free hardware and reducing costs, with the primary objective of ensuring that the product can be developed or acquired by any institution or organization, be it research or development in the most accessible way.

To achieve the prototype, it was subdivided into: the creation of a photographic studio; the automation of the movement of the camera using two stepper motors, and as control element the Arduino Mega 2560 board; the selection of software that is responsible for generating the 3D model; and, finally, the development of software that allows the interaction of all these elements.

Keywords: automation, free hardware, 3D modeling, embedded systems, software.

Resumo

O protótipo apresentado neste artigo é uma combinação de hardware e software livre para capturar imagens em três dimensões (3D). É dada prioridade ao uso de hardware e redução de custos gratuitos, com o objetivo principal de garantir que o produto possa ser desenvolvido ou adquirido por qualquer instituição ou organização, seja de pesquisa ou desenvolvimento, da maneira mais acessível.

Para alcançar o protótipo, foi subdividido em: a criação de um estúdio fotográfico; a automação do movimento da câmera usando dois motores passo a passo e, como elemento de controle, a placa Arduino Mega 2560; a seleção de software que é responsável por gerar o modelo 3D; e, finalmente, o desenvolvimento de software que permite a interação de todos esses elementos.

Palavras-chave: automação, hardware livre, modelagem 3D (3 dimensões), sistemas embarcados, software.

Fecha Recepción: Octubre 2016

Fecha Aceptación: Mayo 2017

Introducción

Un modelo 3D es una representación digital de un objeto real que tiene la finalidad de reproducir fielmente las características físicas de este, como lo son volumen y forma (Mesa Múnica, Ramírez Salazar y Branch Bedolla, 2010).

En la actualidad, la posibilidad de digitalizar objetos del mundo real en 3D permite el desarrollo de aplicaciones científicas, de educación y entretenimiento que representan una potente herramienta con una amplia gama de enfoques para cada una de estas. Sin embargo, para muchas organizaciones, como las educativas o de investigación científica, adquirir un

dispositivo que sea capaz de recrear la estructura de un objeto de manera digital representa una inversión de alto costo que en muchas ocasiones no pueden permitirse.

Existen en el mercado diferentes alternativas de software que permiten llevar a cabo tareas de modelado en 3D a partir de diferentes técnicas. Algunos programas permiten modelar desde cero y a manera de “dibujo” objetos que podemos encontrar en el mundo real y generarlos de manera digital según la creatividad del usuario. Otros realizan el modelado a partir de un objeto físico mediante técnicas de obtención y digitalización de información para su posterior procesamiento.

Para obtener un modelado en 3D se tienen las siguientes técnicas: las basadas en iluminación utilizan la proyección de un haz de luz o laser sobre la superficie del objeto a digitalizar con la finalidad de resaltar puntos específicos de éste (Mesa Múnera, Ramírez Salazar, & Branch Bedolla, 2010); otras técnicas se basan en la fotogrametría, que es la obtención de características reales de un objeto a partir de un conjunto de imágenes fotográficas del mismo desde diferentes perspectivas (Rodríguez Jordana y Nuñez Andres, 2007); y, quizás, una de las técnicas más complejas sea el modelado por medio del uso de sensores ultrasónicos, que permiten determinar el tamaño y volumen de un objeto (Orozco Quinceno, Romero Acero, Marín Cano, & Jiménez Builes, 2014).

Al revisar todas las técnicas se seleccionó el modelado a partir de fotogrametría, ya que el objetivo principal del proyecto fue el diseño y desarrollo de una alternativa a bajo costo que permita realizar el modelado de un objeto en tres dimensiones y que sea de fácil uso para cualquier tipo de usuario. Dado que en caso de fallos en el sistema de movimiento y captura es mucho más fácil la adquisición y reemplazo de una cámara web que de un sensor ultrasónico o infrarrojo, o incluso un láser. Además, el procesamiento se simplifica en gran medida y evita tener que recalibrar las características físicas de cada uno de estos elementos en cada sustitución.

A lo largo de este documento se pretende abordar el proceso de diseño y desarrollo de esta plataforma iniciando con una breve introducción que nos habla acerca del concepto, además de algunas técnicas y aplicaciones de modelado 3D. Posteriormente se describe la metodología base que se utilizó para agilizar el diseño de este proyecto y cada una de las fases de desarrollo de ambos módulos de manera individual. Finalmente se describe el funcionamiento completo del proyecto una vez integradas ambas partes del mismo, así como se explican algunas de las pruebas llevadas a cabo sobre este con la finalidad de comprobar su funcionamiento y rendimiento en un entorno real.

Metodología

Para el desarrollo de este prototipo se usó la metodología de desarrollo de hardware y software embebidos propuesta por Perez, Urkidi, Berreteaga, Ruiz de Olano y Pérez (2006). Esta metodología fue seleccionada debido a que se enfoca en optimizar el proceso de diseño y desarrollo del proyecto permitiendo a la vez la retroalimentación del mismo y la pronta evolución conforme se van identificando deficiencias en este.

El concepto de embebido se refiere al hardware y software que trabajan en conjunto como un todo y de manera “invisible” para el usuario. Esta metodología es conocida como Metodología V y se compone de las siguientes etapas de desarrollo:

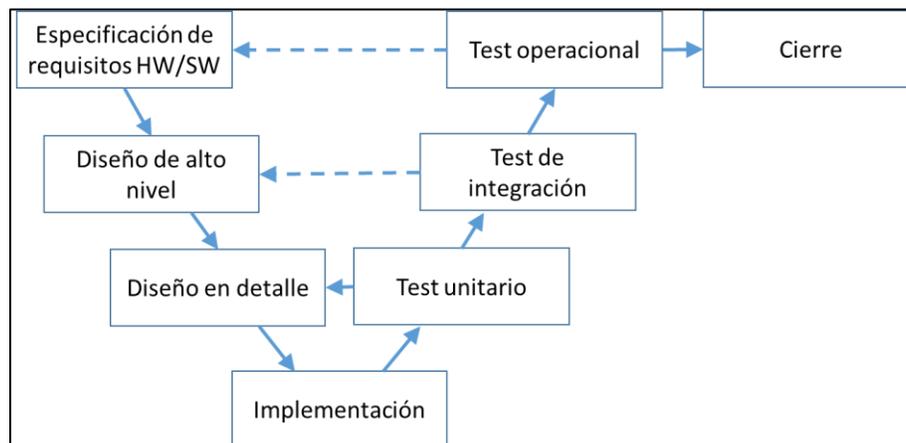
- Especificación de requisitos HW/SW. En esta primera etapa de la metodología se debe definir y documentar los diferentes requisitos que se pretende que cumpla el sistema a desarrollar, identificando en medida de lo posible la mayor cantidad de valores numéricos concretos.
- Diseño de alto nivel. Durante esta etapa se busca obtener una visión general de las funciones del sistema.
- Diseño en detalle. Consiste básicamente en detallar cada una de las funciones definidas en la etapa anterior.
- Implementación. Aquí se debe materializar el diseño en detalle.

- Test unitario. En esta etapa se verifica cada módulo que conforma la parte de hardware y software de manera unitaria para medir su funcionamiento.
- Test de integración. Consiste, como su nombre lo indica, en la integración total de los diferentes módulos que componen al sistema con la finalidad de comprobar el correcto funcionamiento del sistema y, en caso de encontrar fallos, realizar las correcciones pertinentes para subsanar el problema. Además, se debe comprobar el cumplimiento de los requisitos establecidos durante la primera y segunda etapa.
- Test operacional. El sistema es probado en un entorno real y, de ser posible, en su ubicación final.
- Cierre. Entrega del sistema.

En la

se muestra el diagrama que representa el ciclo de vida de la metodología V.

Figura 1. Ciclo de vida de la metodología V.



Fuente: Perez *et al.* (2006).

1. El prototipo

Para desarrollar el proyecto se trabajaron seis elementos por separado:

- La creación del estudio fotográfico.
- La portabilidad de la estructura.
- El sistema de iluminación.
- El hardware que permite los movimientos y disparos de la cámara fotográfica.
- El desarrollo del software para poder controlar los movimientos y fotografías tomadas en el estudio fotográfico.
- La integración del software para modelar en 3D.

1.1 Creación del estudio fotográfico

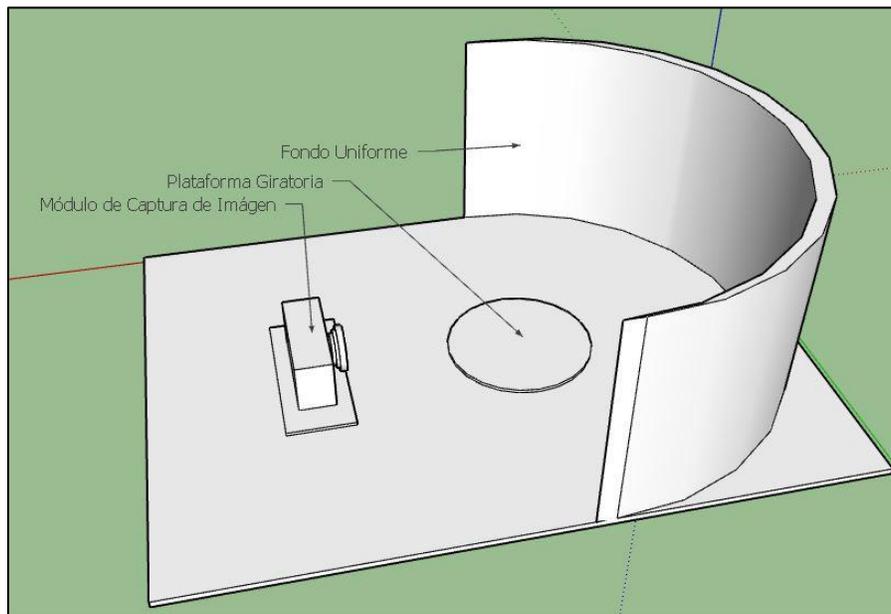
Para la creación del prototipo del estudio fotográfico, donde se toman las imágenes digitales del objeto que se desea modelar en 3D, se utilizó madera triplay y se pintó de color blanco en un tono mate con la finalidad de evitar reflejar luz. En la cara inferior del cubo se realizó una pequeña ranura a través de la cual pasa el eje del motor encargado del giro de la plataforma. De igual manera, en la cara lateral izquierda de la estructura se realizó otra ranura para que al igual que con la cara inferior pasara el eje del motor encargado de elevar la posición de la cámara. El fondo curvo se sujeta a la cara inferior y trasera del cubo mediante velcro, lo cual le permite montarse y desmontarse fácilmente, y el sistema de iluminación queda móvil por si se necesita colocar en algún punto en específico que facilite la iluminación de las caras del objeto.

1.2 Portabilidad

El primer diseño de la estructura se pensó como un entorno semicontrolado, conformado por un diseño de medio cilindro con un soporte fijo para la captura de las fotografías (ver Figura 2). Para facilitar el transporte del prototipo se cambió a una estructura en forma cubica con

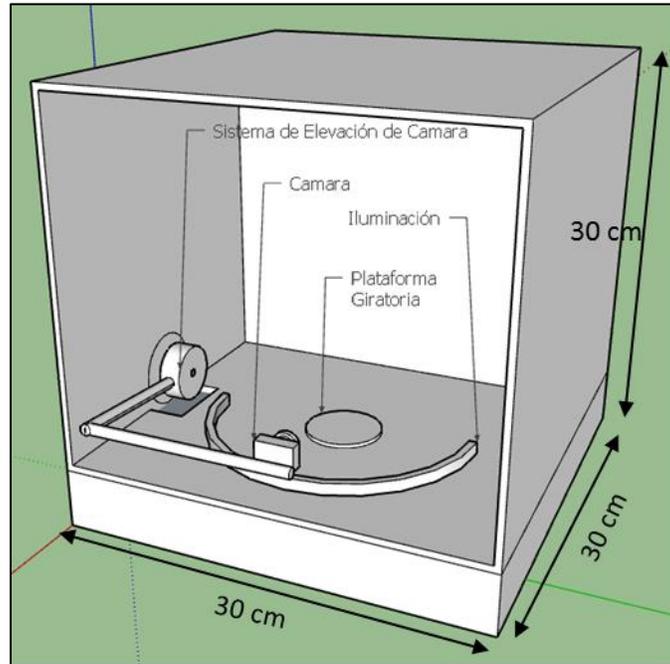
un compartimiento inferior para cubrir toda la circuitería y permitir montar y desmontar en poco tiempo el prototipo sin dañar ninguno de los componentes. En las Figuras 3 y 4 se muestran la estructura final del prototipo.

Figura 2. Estructura inicial del prototipo.



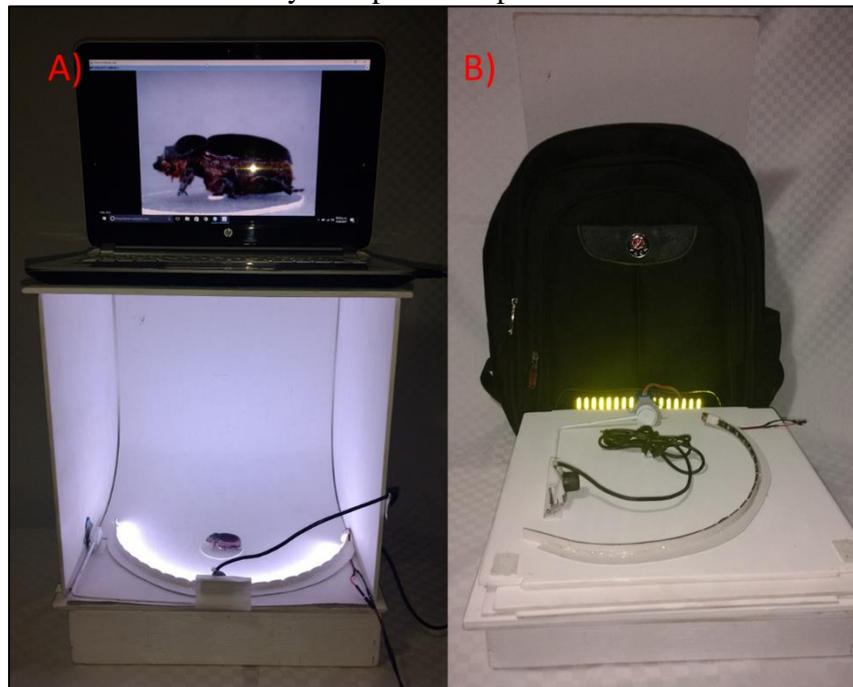
Fuente: elaboración propia.

Figura 3. Estructura final del prototipo.



Fuente: elaboración propia.

Figura 4. A) Estructura Final en funcionamiento B) Estructura final completamente desmantelada y lista para transportar en una maleta.



Fuente: elaboración propia.

1.3 Frecuencia de iluminación y entorno

Durante el diseño del prototipo, dos puntos se sometieron a diferentes pruebas con la finalidad de aumentar el rendimiento del mismo. El primer punto fue el sistema de iluminación; inicialmente las pruebas se llevaron a cabo mediante el uso de un flash ordinario. Sin embargo, se detectó que, al tomar las fotografías, este generaba una sombra del objeto en el fondo de la toma, lo cual suponía mayor dificultad para el software de modelado al identificar el volumen y la forma del objeto. Posteriormente, el flash se reemplazó por un sistema de iluminación constante a base de tiras de led de diferentes tipos, algunas de las más potentes generaban un efecto de líneas negras sobre la toma, debido a la baja frecuencia de refresco de cuadros de la cámara. La frecuencia (o tasa) de refresco se refiere a cuántas imágenes puede capturar la cámara por segundo, y su unidad de medida son los Frames Per Second (FPS) que, relacionado al problema propuesto, podríamos traducir a Hercios (Hz). Por lo tanto, si la tira de led tiene una frecuencia diferente a la de la cámara, esta generará un efecto de líneas negras debido a la nula sincronización entre los destellos de luz y la captura de imagen de la cámara (Vazquez, 2016).

Para solucionar este problema se probó con tiras de led de frecuencia baja hasta poder seleccionar una similar, por lo cual se optó por la tira de led de la marca 3M a 20 Hz, con lo que se logró evitar este problema.

Un segundo punto que fue sometido a pruebas es el entorno de las tomas, dado que se pretende eliminar en la mayor medida posible el ruido en las imágenes. Se optó por implementar un entorno controlado con un fondo semicurvo que permite incorporar a las tomas un efecto óptico conocido como “fondo infinito”, una técnica que es comúnmente usada en la mercadotecnia para la fotografía de productos y permite simular un fondo sin fin detrás del objeto en cada toma (Guasco, 2012).

1.4 El hardware para el estudio fotográfico

En cuanto al sistema de movimiento, se emplearon dos motores a pasos 28BYJ-48, cada uno con su correspondiente driver del tipo ULN2003A. Estos motores son habituales como piezas clave en la enseñanza de pequeños robots y posicionadores caseros programados con Arduino. Las características técnicas con las que dicho motor cuenta son (Prometec, 2017):

- Tensión nominal de entre 5V y 12V.
- 4 fases de excitación.
- Resistencia de 50Ω.
- Par motor de 34 N/m.
- Consumo aproximado de 55 mA.
- 8 pasos por vuelta.
- Reductor de 1/64.

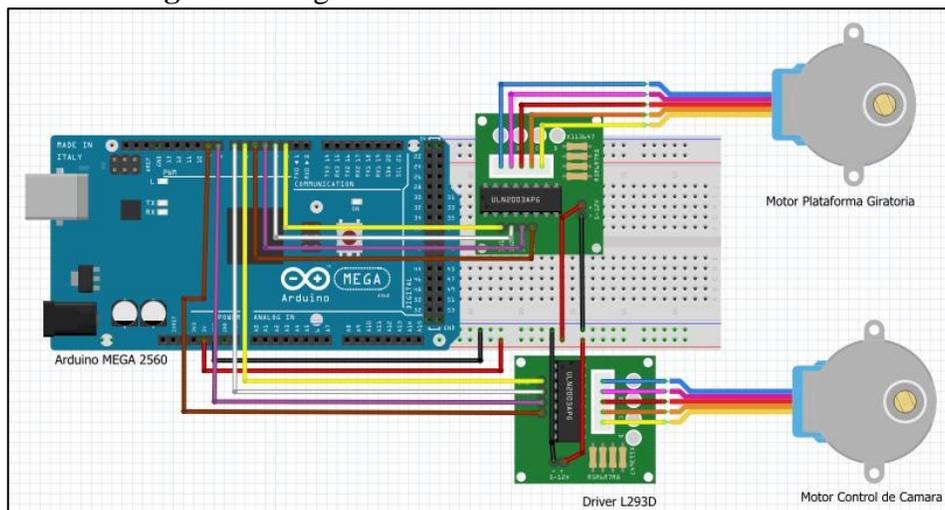
Traduciendo los dos últimos puntos de manera más comprensible, podemos deducir que multiplicando el número de pasos por vuelta (8) por el índice de reducción (64) da como resultado un total de 512 impulsos eléctricos para completar una vuelta completa del motor.

El primer motor se colocó en la parte lateral izquierda del prototipo y se encarga de calibrar la elevación de la cámara para permitir la captura de imágenes a diferentes ángulos del objeto sin perder estabilidad. El segundo motor está colocado en la parte inferior central del prototipo mediante un sistema de plataforma donde permite que el objeto realice un movimiento giratorio de 360 grados sobre su eje vertical para capturar el contorno total de este. Ambos motores son controlados por medio de la placa Arduino Mega 2560.

Arduino es una plataforma de hardware libre basada en un tablero de entradas y salidas (I/O), y un ambiente de desarrollo que implementa el lenguaje Processing (Fry y Reas, 2017). Arduino puede ser utilizado para desarrollar objetos interactivos autónomos o puede conectarse vía software con una computadora. La plataforma puede ser ensamblada a mano o puede adquirirse ya funcionando; el Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) puede ser descargado gratis de la página oficial de Arduino: www.arduino.cc (Banzi, 2011).

El modelo Mega 2560 está compuesto por un microcontrolador ATmega2560 a 16 MHz y con una memoria flash de 256 KB. Además, provee un total de 54 pines digitales entrada/salida, y 16 entradas análogas con su propio entorno de desarrollo para la programación del microcontrolador (Perea, 2015). Su esquema de conexión se muestra en el diagrama físico (ver Figura 5).

Figura 5. Diagrama de conexiones físicas del circuito.



Fuente: elaboración propia.

En la Tabla 1 se presenta el costo unitario de cada elemento que compone la estructura propuesta para este prototipo.

Tabla 1. Costo unitario y total de producción del prototipo.

Elemento	Costo Unitario
Estructura de Triplay	\$150.00
Placa Arduino Mega 2560	\$740.00
Motor a pasos 28BYJ-48 (x2)	\$150.00
Cámara web Ele-gate HD 18 Mpx	\$180.00
PlastiAcero	\$35.00
Pintura Acrilica Blanco Mate (x2)	\$45.00
Driver Motor a Pasos (x2)	\$124.00
Tira de Led	\$30.00
TOTAL	\$1,454.00 MN

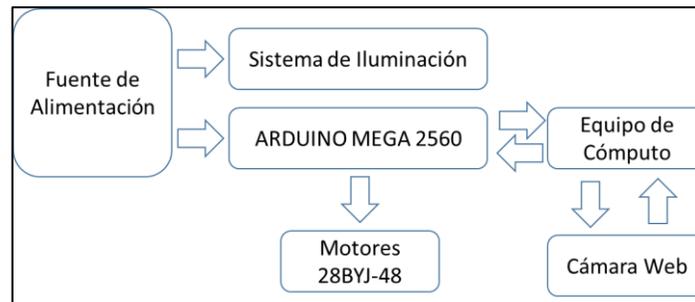
Fuente: elaboración propia.

Existen diferentes alternativas de Hardware destinados a llevar a cabo la tarea del modelado. Alguno de los ejemplos más económicos que podemos encontrar a la venta son el escáner 3D “Matter and Form 3D” de la empresa Matter and Form Inc. con un precio estimado de \$22,000 M.N. y el escáner de mano Sense desarrollado por la compañía 3D Systems con un precio de \$15,000 M.N.

1.5 El software para control del estudio fotográfico

El algoritmo generado para llevar a cabo el control de los motores está basado en instrucciones simples de bloques repetitivos (bucles) y en declaraciones básicas de variables, además del uso de las entradas y salidas analógicas de la placa Arduino (Torrente Artero, 2013). El funcionamiento del circuito electrónico se describe en el diagrama de bloques de la Figura 6.

Figura 6. Diagrama de bloques del circuito eléctrico.



Fuente: elaboración propia.

En cuanto al software, se desarrolló un sistema mayormente generado en Java, que permite llevar a cabo la calibración de la cámara en tiempo real y la captura del conjunto de imágenes final de manera automática mediante el control sincronizado de ambos motores y la comunicación directa con la cámara.

Finalmente, se incrusto dentro de este software un script de modelado conocido como MeshRecon que permite, de manera nativa, llevar a cabo la tarea de digitalización con las imágenes obtenidas en el paso anterior.

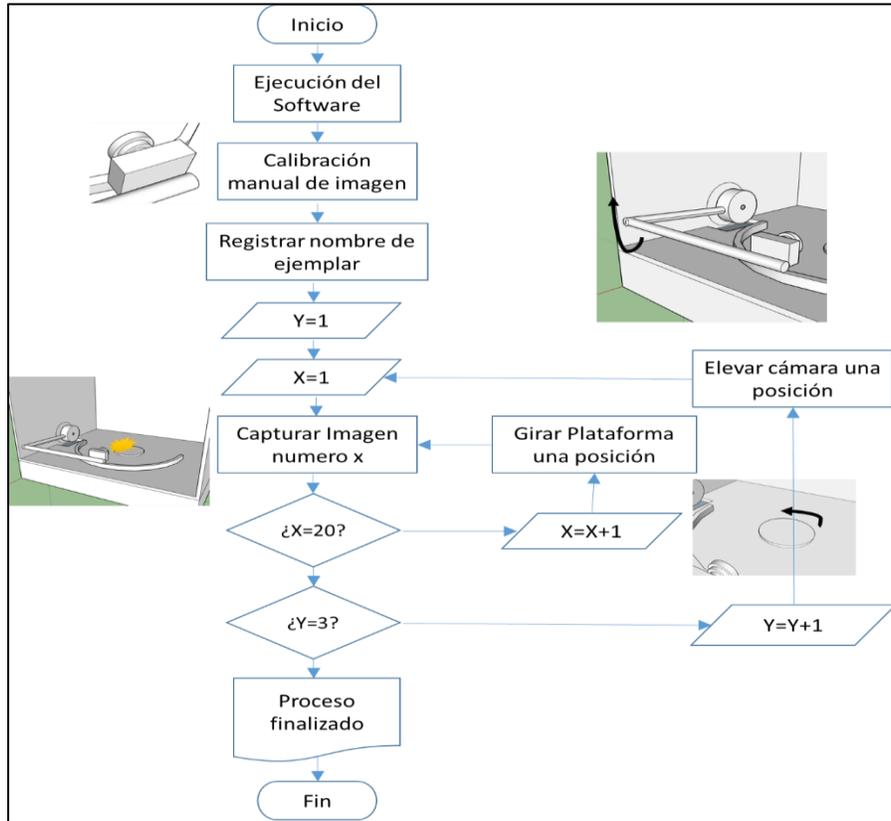
El flujo de procesos llevados a cabo para capturar la totalidad de imágenes requeridas para el modelado es el siguiente:

- 1) Iniciar la ejecución del programa.
- 2) Llevar a cabo la calibración manual de la cámara (enfoque y ubicación).
- 3) Etiquetar ejemplar.
- 4) Establecer el inicio de la primera sesión de tres en un ángulo de 0° .
- 5) Establecer el inicio de la primera toma de 20 (1/20).
- 6) Capturar la imagen.
- 7) Si la imagen capturada es la numero 20/20 se avanza al proceso 10, de lo contrario se avanza al proceso ocho.
- 8) Establecer la captura de la imagen siguiente del conjunto de 20.
- 9) El objeto se gira un ángulo de 18° y se continúa al proceso seis.

- 10) Si la sesión actual es la 3/3 se finaliza el proceso de captura, de lo contrario se pasa al proceso 11.
- 11) Establecer una nueva sesión de capturas.
- 12) Elevar la cámara en un ángulo de 20° respecto a su posición actual y se continúa en el proceso número cinco.

Una vez terminado el proceso de captura, el software continúa con el procesamiento de las imágenes obtenidas para que a partir de estas genere un modelo 3D del objeto en cuestión (ver Figura 7).

Figura 7. Diagrama de flujo general del prototipo.



Fuente: elaboración propia.

1.6 El software para el tratamiento de imágenes

Existen softwares comerciales para digitalizar los objetos en 3D, y ReMake® es uno de ellos. Es el más reciente sucesor de “123D Catch®”. Fue desarrollado por la reconocida empresa Autodesk. Este software genera un modelo 3D de algún objeto del mundo real proporcionando un conjunto de entre 50 y 120 fotografías desde diferentes ángulos del objeto en cuestión y enviándolas a través de Internet a los servidores de Autodesk por medio de la misma aplicación de software. ReMake® es un software de licencia gratuita, pero posee extensiones de paga para aumentar sus características de procesamiento. ReMake® puede ser adquirido desde la página principal de Autodesk: <https://remake.autodesk.com/about> (Autodesk, 2015).

Otra alternativa como software de modelado es PhotoScan®, producto de la compañía Agisoft®, que es un avanzado software de modelado 3D, funciona de manera local en el

computador y genera una representación digital de un objeto a partir de fotografías del mismo (Hagedorn-Saupe, Fresa, Liestol, Rajcic y Grussenmeyer, 2016). A diferencia de ReMake®, PhotoScan® es un software totalmente de paga con un periodo de prueba de 30 días; posterior a este periodo la licencia del software puede ser adquirida desde \$3,205 M.N o se puede adquirir una licencia especial comprobando los fines educativos de la misma desde \$1,056 M.N. PhotoScan® puede ser adquirido desde la página principal de Agisoft: <http://www.agisoft.com/> (Agisoft, 2017)

Estos tipos de software son los que permiten usar este prototipo, ya que de manera sencilla se ofrece la capacidad de obtener un conjunto de imágenes con las características adecuadas para ser procesadas posteriormente por alguno de ellos o por el complemento de modelado que contiene de manera nativa esta propuesta.

De manera nativa, el software diseñado trabaja con un Script de modelado conocido como MeshRecon. Este Script, propuesto por el Dr. Zhuoliang Kang (2015), es una adaptación de una plataforma de software diseñada en 2012 para el modelado de ciudades en 3D a partir de fotografías aéreas, conocidos como VisualSFM (Wu, 2014), el cual a su vez es una GUI (Graphic User Interface/ Interfaz Gráfica de Usuario) con la capacidad de modelar estructuras a partir de una trayectoria de movimiento gracias a la implementación de la tecnología de paralelismo de múltiples núcleos incorporado por las tarjetas gráficas Nvidia. El proyecto MeshRecon puede ser consultado desde la página: <http://zhuoliang.me/meshrecon.html>

Resultados

En conjunto, la plataforma funciona como un sistema autónomo que permite la captura de imágenes para su posterior procesamiento digital a un modelo 3D del mismo.

En un inicio el software se conecta de manera automática a la placa Arduino, que a su vez lleva el control de los motores. Este sistema se desarrolló en Java, un lenguaje de programación de alto nivel que tiene la característica de permitir diseñar y desarrollar software robusto y portable entre diversas plataformas de sistemas operativos (Horstmann, 2010).

Inicialmente, al ejecutar el software, se mostrará el menú principal del sistema (ver Figura 8); este menú muestra las posibles acciones necesarias para completar con éxito el modelado.

Figura 8. Menú principal del módulo de Software.



Fuente: elaboración propia.

El primer botón permite abrir la ventana de calibración para enfoque de imagen (ver Figura 9) mediante la cual se comprueba la posición y el enfoque del objeto; el segundo botón permite tomar una fotografía de prueba del objeto; el tercer botón comienza el proceso de movimiento de los motores simulando la captura de las fotografías para comprobar que éstos funcionen adecuadamente; un cuarto botón inicia el proceso de captura de imágenes para su posterior procesamiento y, finalmente, el quinto botón permite comenzar la reconstrucción del objeto mediante el script de modelado que se incluye de manera nativa.

Figura 9. Ventana de calibración de enfoque de la imagen.



Fuente: elaboración propia.

La plataforma captura un total de 60 fotografías desde diferentes ángulos del objeto, que son los recomendados por la mayoría de alternativas de modelado 3D a partir de imágenes.

Finalmente, el prototipo se sometió a una serie de pruebas con la finalidad de medir el tiempo necesario para la captura de imágenes, y el tiempo de procesado necesario para generar un modelo 3D. Los resultados obtenidos indican que el prototipo tarda un aproximado de 2 minutos en tomar un total de 60 fotografías. Cada una de estas imágenes tiene un peso de almacenamiento de entre 15 KB y 35 KB, dando un promedio aproximado de 1.50 MB de almacenamiento para la galería final que posteriormente fue sometida al procesamiento por medio de diferentes ejemplos de software de modelado.

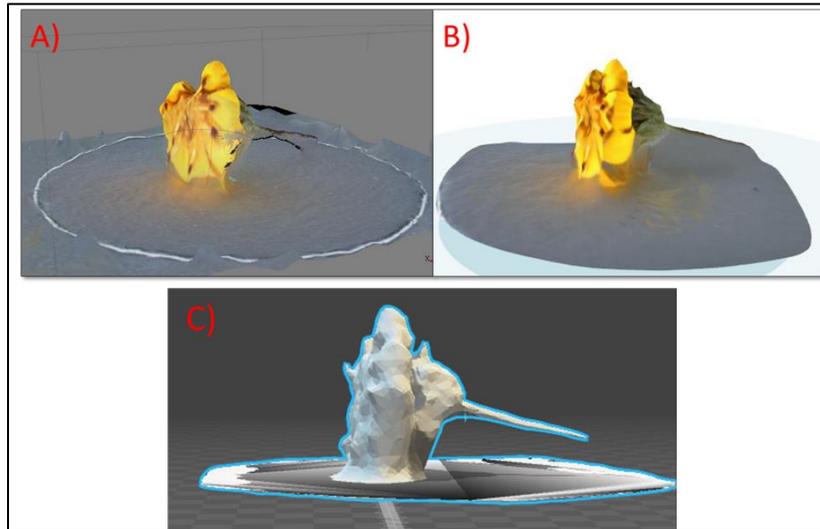
A continuación se muestra una tabla comparativa con los resultados obtenidos de dos ejemplos de plantas y una figurilla de plástico que fueron sometidas al proceso de digitalización en la cual se pueden apreciar de manera más clara las diferencias de peso y tiempos de procesamiento (Tabla 2). En las Figuras 10, 11 y 12 se muestra un ejemplo obtenido por cada software de modelado 3D.

Tabla 2. Tabla comparativa de los resultados obtenidos a partir de pruebas con diferentes objetos.

Software	Ejemplo	Tiempo de captura	Tamaño promedio de imagen	Tipo de procesamiento	Tiempo de procesamiento	Tipo de archivo final	Tamaño de archivo final
PhotoScan	Flor Amarilla	2 Minutos	21.6 KB 640 x 480 px	Local	10 minutos	.psx	12.2 MB
	Flor Roja		25.8 KB 640 x 480 px		12 minutos		12.2 MB
	Figurilla de Plástico		22.1 KB 640 x 480 px		14 minutos		13.3 MB
ReMake	Flor Amarilla	2 Minutos	21.6 KB 640 x 480 px	Remoto	15 minutos	.rcm	1.47 MB
	Flor Roja		25.8 KB 640 x 480 px		19 minutos		1.34 MB
	Figurilla de Plástico		22.1 KB 640 x 480 px		17 minutos		2.1 MB
MeshRecon (software nativo)	Flor Amarilla	2 Minutos	21.6 KB 640 x 480 px	Local	9 minutos	.ply	616 KB
	Flor Roja		25.8 KB 640 x 480 px		8 minutos		456 KB
	Figurilla de Plástico		22.1 KB 640 x 480 px		9 minutos		588 KB

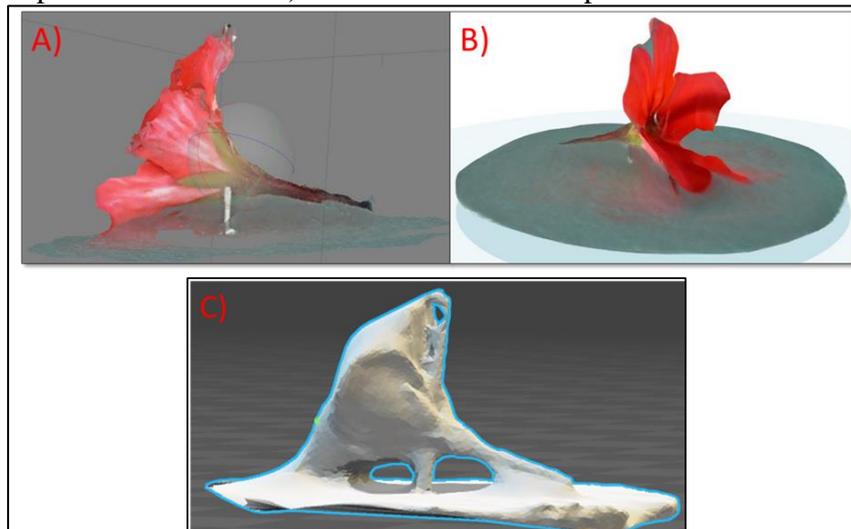
Fuente: elaboración propia.

Figura 10. Ejemplo 1 A) Resultado obtenido a partir de PhotoScan. B) Resultado obtenido a partir de ReMake C) Resultado obtenido a partir de MeshRecon.



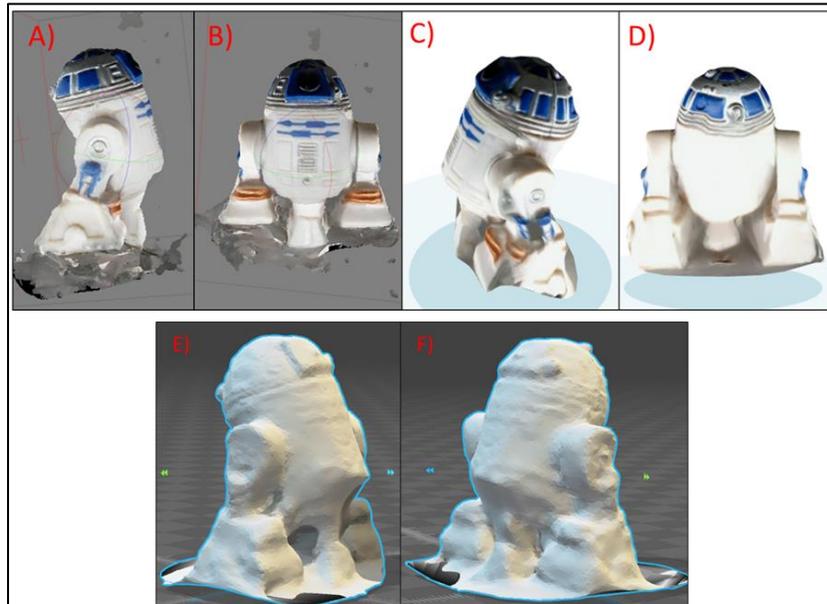
Fuente: elaboración propia.

Figura 11. Ejemplo 2 A) Resultado obtenido a partir de PhotoScan. B) Resultado obtenido a partir de ReMake C) Resultado obtenido a partir de MeshRecon.



Fuente: elaboración propia.

Figura 12. Ejemplo 3 A, B) Resultado obtenido a partir de PhotoScan (Vista lateral derecha y frontal). C, D) Resultado obtenido a partir de ReMake (Vista lateral izquierda y trasera) E, F) Resultado obtenido a partir de MeshRecon (Vista lateral derecha e izquierda).



Fuente: elaboración propia.

Tomando en cuenta que la velocidad de subida con la que se realizaron las pruebas al momento de cargar las imágenes al servidor de Autodesk fue de 0.25 Mbps y, dado que el tamaño total de la galería es de 1.52 MB, el tiempo de subida estimado es de $(1.52/0.25) 6.08$ minutos. Sin embargo, cabe mencionar que la velocidad de la red puede variar dependiendo del uso de esta y la saturación de información de la misma, por lo cual este tiempo puede verse incrementado.

Discusión y conclusiones

Con respecto al objetivo principal del proyecto, esta propuesta demostró ser un medio viable para automatizar el proceso de digitalización de objetos 3D de una manera sencilla e intuitiva para cualquier tipo de usuario y en un entorno de software que puede ser ejecutado en diferentes sistemas operativos.

El prototipo final tiene la facilidad de permitir armar y desarmar la estructura con sencillez y rapidez. Se facilita el transporte de la estructura para evitar la adquisición de un ejemplar del prototipo para cada espacio de trabajo donde se requiera llevar a cabo la digitalización cumpliendo con el objetivo de la portabilidad.

Los tiempos promedio de respuesta para generar una digitalización completa son de 15 minutos con 60 imágenes y un peso de archivo final de entre 1 MB y 12 MB dependiendo del software. Esto permite al usuario obtener varios modelados desde el mismo conjunto de fotografías y seleccionar aquel que cumpla en mayor medida con sus expectativas.

La reducción de costos actual frente a un scanner 3D comercial es de un aproximado de \$13,500 MN comparado con el escáner de mano Sense de la compañía 3D Systems.

Cabe mencionar que, para el ejemplar actual del prototipo, los objetos a modelar pueden tener un tamaño máximo de 5 cm de alto y 5 cm de ancho. Sin embargo, este tamaño puede ser ampliado si el soporte de la cámara se aleja más del centro de rotación para ampliar la zona de captura.

Dado que Java permite el desarrollo y distribución de aplicaciones en los campos de la programación móvil e incrustada, videojuegos, contenido web y software empresarial y la ejecución de las aplicaciones en diversos entornos y sistemas operativos, esto contribuye a que el software creado para este proyecto también cuente con la ventaja de implementar portabilidad entre equipos de cómputo con diferentes características técnicas.

AGRADECIMIENTOS

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la asistencia brindada, así como a la Universidad Autónoma del Estado de México (UAEM), Centro Universitario Texcoco, por el apoyo aportado durante el desarrollo de esta investigación.

Bibliografía

- Agisoft. (2017). Agisoft. Obtenido de Agisoft: <http://www.agisoft.com/>
- Autodesk. (2015). Autodesk ReMake. Obtenido de <http://remake.autodesk.com/try-remake#system-requirements>
- Banzi, M. (2011). Getting Started with Arduino. O'Reilly Media, Inc., p. 2-6.
- Fry, B., & Reas, C. (08 de 2017). Processing. Obtenido de <https://processing.org/>
- Guasco, I. (2012). Tecnicas de Fotografia Profesional. Buenos Aires, Argentina: RedUSERS. pp. 32-38,112-116.
- Hagedorn-Saupe, M., Fresa, A., Liestol, G., Rajcic, V., & Grussenmeyer, P. (2016). Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection. Nicosia, Chipre: Springer International Publishing. pp. 401-408.
- Horstmann, C. (2010). JAVA For Everyone. Estados Unidos: John Wiley & Sons, Inc. pp. 6-11.
- Kang, Z. (29 de 04 de 2015). Mesh Reconstruction from Imagery. Obtenido de <http://zhuoliang.me/meshrecon.html>
- Mesa Múnera, E., Ramírez Salazar, J. F., & Branch Bedolla, J. W. (2010). Construcción de un modelo digital 3D de piezas precolombinas utilizando escaneo laser. Revista Avances en Sistemas e Informática 2010, 7(1), pp. 198-206.
- Perea, F. (2015). Arduino Essentials. Birmingham, UK: Packt Publishing. pp. 11-12

- Perez, A., Urkidi, A., Berreteaga, O., Ruiz de Olano, A., & Pérez, J. (2006). Una metodología para el desarrollo de Hardware y Software embebidos en sistemas críticos de seguridad. *Sistemas, cibernética e informática.*, 3(2), pp. 70-75.
- Prometec. (15 de 08 de 2017). Obtenido de Prometec: <http://www.prometec.net/motor-28byj-48/>
- Rodriguez Jordana, J. J., & Nuñez Andres, M. A. (2007). *Fotogrametría arquitectónica*. Barcelona, España: Ediciones UPC. pp. 23-30.
- Torrente Artero, O. (2013). *ARDUINO Curso Practico de Formacion*. Madrid, España: RC Libros. pp. 60-71, 129-130.
- Vazquez, A. (2016). La Frecuencia de refresco en las pantallas. Obtenido de <http://ledandgo.com/la-frecuencia-de-refresco-en-las-pantallas.html>
- Wu, C. (2014). VisualSFM : A Visual Structure from Motion System. Obtenido de <http://ccwu.me/vsfm/>

Rol de Contribución	Autor(es)
Conceptualización	Enrique Solis <Principal>;Oziel Lugo <Apoya>
Metodología	Enrique Solis <Principal>
Software	Enrique Solis <Principal>; Oziel Lugo <Apoya>; Joel Ayala <Apoya>
Validación	Oziel Lugo <Principal>; Joel Ayala <Apoya>; Alfonso Zarco <Apoya>
Análisis Formal	Joel Ayala <Principal>; Alfonso Zarco <Apoya>
Investigación	Enrique Solis <Principal>
Recursos	Enrique Solis <Principal>; Oziel Lugo <Apoya>
Curación de datos	Enrique Solis <Principal>; Oziel Lugo <Apoya>
Escritura - Preparación del borrador original	Enrique Solis<Principal>
Escritura - Revisión y edición	Joel Ayala <Principal>; Oziel Lugo <Apoya>; Alfonso Zarco<Apoya>
Visualización	Enrique Solis<Principal>
Supervisión	Oziel Lugo <Principal>; Joel Ayala<Apoya>
Administración de Proyectos	Oziel Lugo <Principal>; Joel Ayala <Apoya>
Adquisición de fondos	CONACYT<Principal>; Enrique Solis<Apoya>