Universidad Técnica Federico Santa María
Departamento de Informática
Valparaíso, Chile

Mejora de un microscopio de bajo costo con SuperScanner Software para apoyar las clases STEM de informática educativa de la Escuela Aliro Lamas Castillo

José David Marroquín Toledo

Memoria para optar al título de Ingeniero de Ejecución en Informática

Profesor guía: Cecilia Reyes Covarrubias

Correferente: Eduardo Peña

Marzo de 2018

El difunto profesor de ciencias de la computación de Carnegie Mellon Randy Pausch consideraba la inspiración la más útil de todas las herramientas para explotar al máximo el potencial humano... y yo también lo creo así. Tiene más valor que el dinero en la lucha contra la pobreza.

Resumen

Esta memoria aborda el desarrollo de Microscopx, un microscopio que es una adaptación de otro de bajo costo que mejora en a lo menos tres veces su poder de resolución y cuatro veces el de magnificación, y cuatro veces la resolución de imagen de las fotografías que captura su cámara mediante la técnica computacional Fourier Ptychography. Esta fue implementada en el lenguaje Python con *notebooks* Jupyter bajo la segunda versión de la Licencia Pública General de GNU (GPLv2) como parte del proyecto SuperScanner Software (S2H) que fue dado a conocer en la XXIV Feria de Software en noviembre de 2016. El resultado logrado en esta memoria viene a satisfacer la necesidad de poseer una herramienta de apoyo afín a la metodología de enseñanza STEM para las clases de informática educativa que la profesora Claudia Rojas imparte en la escuela centenaria Aliro Lamas Castillo de la Comuna de Diego de Almagro, Región de Atacama, Chile.

Palabras clave: microscopía, Fourier Ptychography, STEM, Python, Jupyter, *software* libre, *open-source*, NeoPixel, impresión 3D.

Abstract

This report deals the development of Microscopx, a microscope that is an adaptation of another low cost one that improves its resolving power at least three times and four times its magnification, and four times the image resolution of the photographs captured by its camera through Fourier Ptychography imaging technique. This was implemented in Python language with Jupyter notebooks under the second version of the GNU General Public License (GPLv2) as part of the SuperScanner Software (S2H) project that was unveiled at the XXIV Feria de Software in November 2016. The result of this report satisfies the need to have a support tool related to the STEM methodology for the educational computing classes taught by Professor Claudia Rojas at the Aliro Lamas Castillo centennial school of the Diego de Almagro Commune, Atacama Region, Chile.

Keywords: microscopy, Fourier Ptychography, STEM, Python, Jupyter, free software, open-source, NeoPixel, 3D printing.

Resumen **2**

Abstract **2**

A Lista de figuras **5**

B Lista de tablas **6**

Glosario **7**

Introducción **8**

1 Definición del problema **9**

1.1 Contexto 9

1.1.1 Descripción de la situación actual 9

1.1.2 Importancia de la programación en la educación STEM 10

1.3 SuperScanner Software 11

1.3.1 Objetivos 11

1.3.2 Descripción de los componentes del framework 11

1.3.3 Fourier Ptychography 13

1.4 Justificación de una solución 14

2 Diseño del hardware de la solución **17**

2.1 Caracterización del microscopio de bajo costo 17

2.1.1 Resolución de imagen 18

2.1.2 Poder de resolución 19

2.1.2 Rango del campo de visión 19

2.2.3 Rango de magnificación 21

2.3 Diseño del iluminador LED 22

2.3.1 NeoPixels LEDs 22

2.3.2 Diagrama electrónico 23

2.3.3 Soporte imprimible para los componentes del iluminador 24

2.3.4 Soporte imprimible para las muestras 27

2.3.5 Arduino sketch para el control del iluminador 28

3 Creación de los módulos Jupyter **31**

3.1 Requerimientos 31

3.2 Módulo microscopx 31

3.3 Módulo fputilities 35

3.4 Creación de una sintaxis para todos los eventos 36

4 Resultados **38**

4.1 Obtención de la imagen en alta resolución 38

4.2 Caracterización del nuevo instrumento 41

5 Conclusiones 43

5.1 Resultados alcanzados 43

5.2 Proyecciones futuras 45

6 Referencias 47

A Lista de figuras

1 *Framework* del SuperScanner 12

2 Reconstrucción de la imagen en alta resolución en el dominio de Fourier usando Fourier Ptychography 14

3 Imagotipo que identifica al proyecto Microscopx 16

4 U500X Digital Microscope junto al disco de instalación y la regla de calibración incluidas en el empaque 18

5 Sección de la regla de calibración usada para calcular el FoV para el aumento mínimo del microscopio 20

6 Sección de la regla de calibración usada para calcular el FoV para el aumento máximo del microscopio 20

7 NeoPixel LED 23

8 Diagrama de conexiones entre el Arduino y el iluminador 24

9 Soporte imprimible en 3D para el iluminador (dos NeoPixel Rings de 16 y 24 elementos y un NeoPixel Jewel de 7 Pixels) 25

10 Iluminador de NeoPixels usado para las primeras pruebas de Fourier Ptychography 26

11 Base imprimible en 3D para el soporte iluminador de NeoPixel LEDs 27

12 Soporte imprimible en 3D para muestras compatible con el microscopio de COLEMETER 28

13 Diagrama simplificado para el diálogo entre el PC y el Arduino que controla los Pixel LEDs 34

14 Foto de la regla de calibración con el Pixel LED del centro del iluminador encendido 38

15 Foto de la regla de calibración con un Pixel LED del anillo de 16 NeoPixels encendido (53 mm del centro aprox.) 38

16 Foto de la regla de calibración con un Pixel LED del anillo de 24 NeoPixels encendido (105 mm del centro aprox.) 39

17 Foto de la regla de calibración iluminada con la lámpara de la cámara 39

18 Secuencia de iluminación sucesiva usando tres NeoPixels 39

19 Reconstrucción de la escena a partir de 47 fotografías iluminadas con diferentes ángulos de incidencia 40

20 Microscopx durante el proceso de captura de fotografías de baja resolución 42

B Lista de tablas

1 Puntajes obtenidos por la Escuela Aliro Lamas Castillo en el CNIE de 2012 en uso, gestión e infraestructura de TIC 9

2 Cálculo del poder de resolución de U500X Digital Microscope usando luz visible 19

3 Registro de diez mediciones de los valores de longitud horizontal y vertical visible a través del microscopio para los valores mínimo y máximo de aumento del U500X Digital Microscope 21

4 FoV promedio para los valores mínimo y máximo del aumento del microscopio 21

5 Rango de magnificación para distintas densidades de píxeles de monitores del U500X Digital Microscope 22

6 Poder de resolución del Microscopx para los LEDs rojo, verde y azul de los Pixels 41

7 Rango de magnificación del Microscopx para distintas densidades de píxeles de monitores 42

Glosario

- **Apertura numérica (NA)**: en un microscopio, es la medida de la capacidad para recoger luz y resolver detalle finos de la muestra que se observa a una distancia fija. Se calcula como el producto del índice de refracción del medio y el seno de la mitad del ángulo del cono de luz que puede entrar al lente.
- **Banda de espacio (SBP)**: producto entre el poder de resolución y el campo de visión.
- **Campo de visión (FoV)**: en microscopía, es el área visible a través del objetivo o en la fotografía.
- **Fourier Ptychography**: técnica de imagen computacional que aumenta la banda de espacio de un microscopio combinando fotografías de baja resolución iluminadas con distintos ángulos de incidencia.
- **Magnificación**: medida de la habilidad de un lente o un instrumento óptico para magnificar. Para un sensor de imagen (CCD), se puede calcular como el cociente entre número de píxeles de ancho y el producto entre la densidad de píxeles del monitor desde donde se observa la muestra y la longitud horizontal del campo de visión.
- **Número f (f#)**: razón entre la distancia focal del lente y el diámetro de la pupila de entrada.
- **Poder de resolución**: habilidad de un instrumento óptico de diferenciar dos puntos o líneas en un objeto. Se expresa en unidades longitud como µm.
- **Pupila**: lente. A la composición de más de una de ellas, se le conoce como objetivo.
- **Síntesis de apertura**: en Fourier Ptychography, es la combinación de las imágenes de baja resolución en el dominio de Fourier.
- **STEM**: metodología de enseñanza que integra las disciplinas de ciencia, tecnología, ingeniería y matemáticas en un paradigma de aprendizaje cohesivo basado en problemáticas del mundo real.

Introducción

En la Escuela Aliro Lamas Castillos de la Comuna de Diego de Almagro, Región de Atacama, la asignatura de Informática Educativa que imparte la profesora Claudia Rojas está basada en STEM. En ella, resuelven problemáticas del mundo real integrando las disciplinas de ciencia, tecnología, ingeniería como las son propuestas en la First LEGO League o como aquellas que surgieron en su comunidad tras el aluvión que los golpeó el 25 de marzo de 2015. La profesora afirma que STEM ha facilitado el desarrollo de habilidades como inferencia, evaluación y conclusión en sus alumnos. Además, que con un instrumento afín a esa metodología como SuperScanner, que fue presentado por el autor en una visita a finales de octubre de 2016, cree que podría mantener la capacidad de asombro de sus niños e incitar a sus colegas a la adopción de STEM en sus asignaturas (C. Rojas, comunicación personal, 30 de mayo de 2017).

SuperScanner es un proyecto *open-source* y de *software* libre que da la funcionalidad de microscopio y escáner 3D a una impresora 3D. En la XXIV Feria de Software organizada por el Departamento de Informática de la UTFSM y que se llevó a cabo en Santiago el 18 de noviembre de 2016, la máquina mostró todas sus capacidades con un prototipo funcional, pero con datos de entrada simulados. Una de las característica más llamativas es la de la mejora de las fotografías que captura mediante una técnica de imagen computacional que combina fotos de una escena que se ilumina sucesivamente desde abajo por un LED distinto cada vez. La técnica es conocida como Fourier Ptychography y Microscopx, haciendo uso del *software* de SuperScanner, la implementa para un *set* de imágenes reales capturadas con la cámara de un microscopio de USD$28,99 (CLP$17.459[1]).

El desarrollo de Microscopx es lo que se aborda en esta memoria. Aquí, se trata la problemática identificada (cap. 1 Definición del problema), el diseño del *hardware* adaptable al microscopio U500X fabricado por COLEMETER (cap. 2 Diseño del *hardware* de la solución), la creación de las bibliotecas escritas como *notebooks* Jupyter (cap. 3 Creación de los módulos Jupyter); se exhiben valores cuantitativos que validan las mejoras sobre los resultados originales que entrega la cámara del microscopio de bajo costo (cap. 4 Resultados), una revisión y nuestras proyecciones que podrían mejorar el instrumento (cap. 5 Conclusiones).

Microscopx mantiene la licencia de Creative Commons (CC) Attribution-NonCommercial-ShareAlike 3.0 Unported para su *hardware* y Licencia Pública General de GNU (GPLv2) para su *software*. Cuando él se concluya, uno de ellos será entregado a la Escuela Aliro Lamas Castillo.

[1] Precio de venta en Amazon y equivalencia de divisas del 12 de marzo de 2018.

1 Definición del problema

1.1 Contexto

1.1.1 Descripción de la situación actual

La Escuela Aliro Lamas Castillo de la Comuna de Diego de Almagro, Provincia de Chañaral, Región de Atacama, es una institución educativa municipal gratuita que instruye a sus alumnos en los niveles de educación parvularia y básica. Cuenta con una matrícula de 737 estudiantes distribuidos en cursos de 23 alumnos [1]. En el año 2017, celebró su primer centenario.

El Censo Nacional de Informática Educativa (CNIE) intenta reflejar el grado en que las escuelas han desarrollado las condiciones de uso, gestión e infraestructura de Tecnologías de Información y Comunicación (TIC). Según el último censo del 2012, la Escuela Aliro Lamas Castillo "posee condiciones aceptables [y] requiere gestión para potenciar fortaleza [en el uso y gestión de TIC]" y, en infraestructura de TIC, tiene la "capacidad para instalar innovaciones" [1].

Índice	Puntaje obtenido	Categoría	Rango de la categoría
Uso	24,52	Intermedia	[17,55–64,75]
Gestión	75,15	Intermedia	[41,18–75,85]
Infraestructura	78,27	Avanzada	[76,70–100]

Tabla 1. Puntajes obtenidos por la Escuela Aliro Lamas Castillo en el CNIE de 2012 en uso, gestión e infraestructura de TIC. Elaboración propia.

Adicionalmente, Claudia Rojas, profesora de computación y robótica en la escuela, ha detectado que muchos de sus alumnos «tienden a pensar [más] en incapacidad». Para abatir eso, ha incitado la capacidad de asombro de los niños, por ejemplo, consiguiendo que grupos de estudiantes representen a su comunidad en competencias científicas y tecnológicas como la First LEGO League obteniendo podios como el "Premio al Campeón" en el año 2014 [2], la fabricación de bloques a partir de una mezcla de molido de botellas PET y cemento para reconstruir la comuna después del aluvión que los azotó el 25 marzo de 2015 y que chilenos sobresalientes en distintas áreas los visiten. Todo ello en el dominio de la enseñanza STEM que ha implementado en sus clases y que desearía que todo el cuerpo docente de la escuela hiciera uso (C. Rojas, comunicación personal, 30 de mayo de 2017).

1.1.2 Importancia de la programación en la educación STEM

STEM (acrónimo en inglés para *Science, Technology, Engineering and Mathematic*) es una metodología de enseñanza que integra las disciplinas de ciencia, tecnología, ingeniería y matemáticas en un paradigma de aprendizaje cohesivo basado en problemáticas del mundo real [3] que pueden ser resueltas con herramientas como la programación.

La programación nació con los lenguajes computacionales en la década de los cincuenta como FORTRAN y COBOL. Estos fueron creados para que científicos e ingenieros, y gobiernos y grandes compañías, manejaran datos matemáticos. En la década de los sesenta, surgieron los lenguajes de alto nivel como BASIC que hicieron accesible la programación de computadores a personas sin conocimiento técnico [4].

La programación se ha popularizado tanto que hoy se imparte a alumnos de edad temprana. Por ejemplo, en el año 2015, en a lo menos 5 países europeos (Dinamarca, Eslovaquia, Finlandia, Francia y Portugal), en la educación secundaria, la enseñanza de programación era obligatoria [5]. En Chile, ella no forma parte del currículo educativo, pero en el año 2015, la cámara de diputados aprobó por unanimidad el proyecto de resolución n.º 392 [6] basado en "Pensamiento computacional y programación a nivel escolar en Chile" (PCyP) (Hitschfeld, Pérez y Simmonds, 2015) [7] en el que se solicita al Presidente de la República «abordar la implementación de las condiciones necesarias para que [..] [el PCyP] sea una rama en todas las escuelas del país»; al Ministerio de Educación, encabezar «los cambios curriculares e institucionales para la incorporación de PCyP»; incrementar «como mínimo en un 10% [...] los recursos destinados a la incorporación de las [TIC] [...] [y la] educación temprana en generación de tecnología digital». Pérez desconoce cuál es el estado actual del proyecto (J. Pérez, comunicación personal, 13 de marzo de 2018).

Actualmente, el Ministerio de Hacienda colabora en la organización de la versión chilena de "Hour of Code" que ha acercado a 250 mil niños chilenos al PCyP [8] de un total 10 millones de niños de todo el mundo [9]. Por otro lado, profesores como Claudia Rojas han elaborado programas educativos para impartir clases de computación y robótica a sus alumnos (C. Rojas, comunicación personal, 30 de mayo de 2017).

La programación ha hecho posible la existencia de todas las aplicaciones de *software* que corren en computadores de escritorio, dispositivos móviles y vestibles como teléfonos y relojes inteligentes. También, es la herramienta que los programadores usan para permitirnos el acceso limitado a grandes volúmenes de contenido multimedia en la nube, registrar el estado de múltiples sensores de sistemas electromecánicos complejos en tiempos inferiores a un segundo y procesarlos para apoyar el proceso de toma de decisiones con los usuarios de esa información o de modo autónomo como ocurre en una línea de producción automatizada, en la calle con los *self-driving cars* o en los sistema de manejo de los *rovers* que la NASA ha enviado al planeta Marte [10].

La profesora Rojas afirma que la programación y STEM han facilitado el desarrollo de las habilidades de inferencia, evaluación y conclusión en sus alumnos (C. Rojas, comunicación personal, 30 de mayo de 2017).

1.3 SuperScanner Software

1.3.1 Objetivos

SuperScanner Software (S3) es un ambiente de *software open-source* y gratuito que implementa técnicas de procesamiento de imágenes y visión computacional para dar la funcionalidad de microscopio y *scanner* 3D a una impresora 3D [11]. S3 es parte del proyecto SuperScanner que fue ideado para mejorar el proceso de control de calidad de piezas del AstroTaller del Observatorio Paranal. S3 fue presentado en la XXIV Feria de Software (2016) que organizó el Departamento de Informática de la Universidad Técnica Federico Santa María donde fue reconocido por el jurado como el proyecto más innovador.

S3 se aloja en GitHub[2] y está licenciado bajo la segunda versión de la Licencia Pública General de GNU[3] (GPLv2).

1.3.2 Descripción de los componentes del *framework*

SuperScanner Software tiene un hermano, SuperScanner Hardware (S2H). Este último es la representación material (*hardware* “real”) de la máquina diseñada en FreeCAD y ensamblada en Blender (*hardware* “simulado”). Ambas hacen uso de los mismos módulos de programación escritos en *notebooks* Jupyter para el procesamiento de la imágenes. Mejoras en el *hardware* simulado producen mejoras en el *hardware* real y viceversa.

S2H se aloja en Thingiverse[4] como un Work In Progress —trabajo en progreso— y está licenciado bajo Creative Commons (CC) Attribution-NonCommercial-ShareAlike 3.0 Unported[5] (CC BY-NC-SA 3.0).

[2] Visitar https://github.com/josemarroquintoledo/superscanner-software-s3
[3] Revisar en https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
[4] Visitar https://www.thingiverse.com/thing:1880928/apps
[5] Revisar en https://creativecommons.org/licenses/by-nc-sa/3.0/legalcode

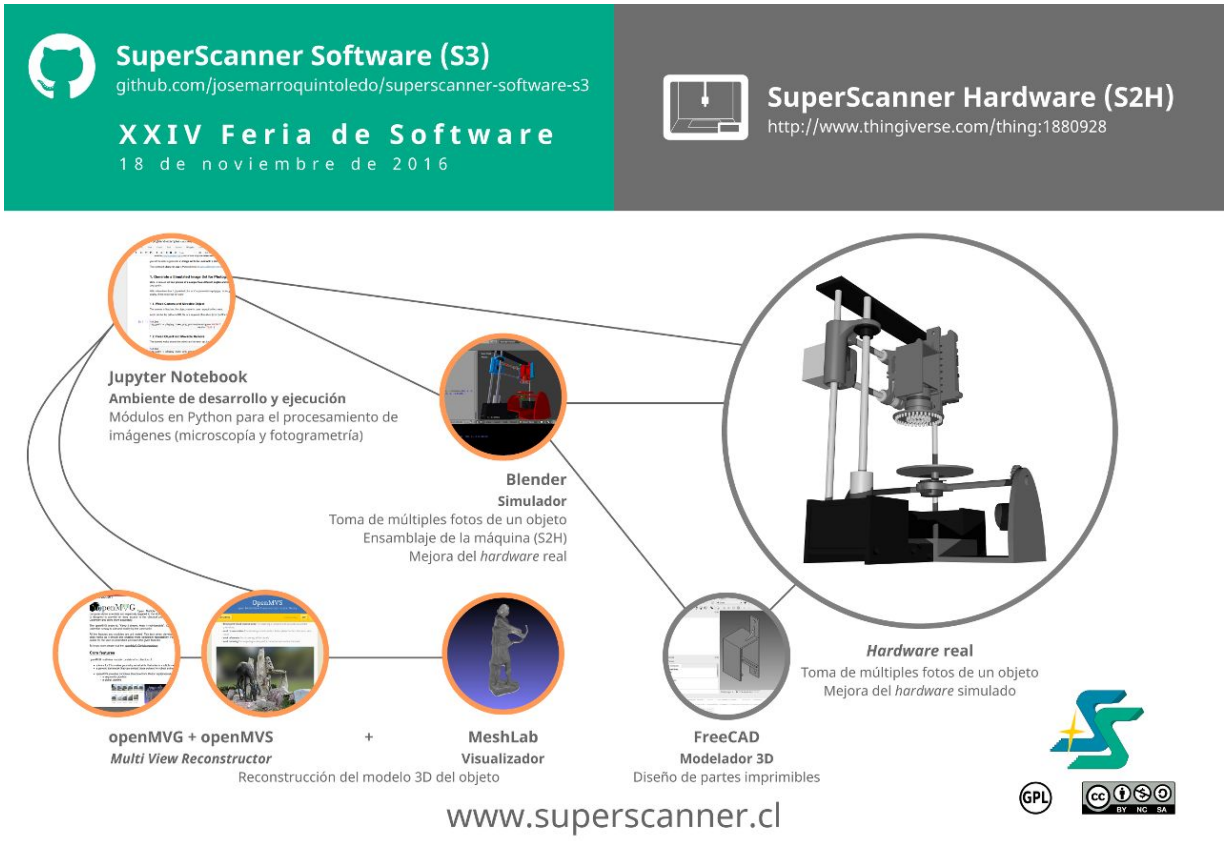

Figura 1. *Framework* de SuperScanner. Descripciones e interacciones entre los componentes de SuperScanner Software (S3) y SuperScanner Hardware (S2H). Elaboración propia.

A continuación, son detallados los componentes de *software* del *framework* del SuperScanner.

- Jupyter Notebook: es una aplicación *web open-source* que permite la creación y compartición de documentos que contengan, además de texto e imágenes, audio, videos, animaciones y código vivo como Python [15].

- Blender: es un *software* multiplataforma y *open-source* para la creación de gráficos 3D. Ofrece una API en Python para crear y modificar las propiedades de una escena y sus elementos, las preferencias del usuario, programar *scripts* y complementos (*add-ons*) [16].

- FreeCAD: es un modelador 3D paramétrico, *open-source* y multiplataforma. Se puede acceder a la mayoría de sus funcionalidades, automatizarlas y crear modelos 3D con Python [17].

- OpenMVG: es una biblioteca *open-source* para la recuperación de las posiciones de una cámara y generación de una nube de puntos a partir de un conjunto de fotografías mediante Multi View Geometry (MVG) y Structure-from-Motion (SfM) [18][19].

- OpenMVS: es una biblioteca o*pen-source* que, a partir de la posiciones de una cámara y la nube de puntos, recupera la superficie a ser reconstruida mediante Multi-View Stereo [18].

- MeshLab: es un *software open-source* multiplataforma para la visualización y edición de mallas de triángulos 3D.

En *notebooks* Jupyter, fueron escritos los módulos para el procesamiento de las imágenes tanto para microscopía como para fotogrametría. Haciendo uso de Blender Python API, fueron creados *scripts* para realizar la captura fotográfica de un objeto tridimensional ubicado en medio de una escena Blender con una cámara que describe trayectorias esféricas. También, con él, es ensamblado, manipulado y animado el modelo 3D del *hardware* de SuperScanner. En FreeCAD, son diseñadas nuevas partes imprimibles en 3D para SuperScanner. Con OpenMVG, OpenMVS y MeshLab en combinación es reconstruido el modelo tridimensional del objeto a partir del *set* de fotografías capturadas por la cámara en distintas posiciones y ángulos.

1.3.3 Fourier Ptychography

En microscopía, a la distancia más pequeña entre dos puntos de una muestra que son distinguibles por un observador como dos entidades separadas se le conoce como poder de resolución; al área circular visible a través del ocular de un microscopio, como campo de visión (FoV o F.O.V, por sus siglas en inglés); y al producto de esas cantidades, como producto de ancho de banda de espacio (SBP, por sus siglas en inglés) que representa la máxima cantidad de información que puede recogerse y transmitirse [20].

El incremento del valor de FoV ampliando el tamaño de los lentes conlleva el aumento del valor de SBP, pero acarrea aberraciones ópticas. Para corregirlas, pueden ser añadidas superficies ópticas, pero esto conlleva al aumento de los costos de producción y dificultad en el proceso de alineación [21]. Fourier Ptychography (FP) aborda esta problemática desde la imagen computacional.

La técnica de Fourier Ptychography consiste aumentar el SBP por medio de la combinación de fotografías de baja resolución de imagen iluminadas desde ángulos diferentes y capturadas desde una misma posición y ángulo de cámara. Para ello, FP emplea dos técnicas: una para la recuperación de la fase y otra para síntesis de apertura. La primera alterna la información capturada entre el dominio espacial (plano XY) y el de Fourier, y la segunda combina imágenes en el dominio de Fourier [22].

Las imágenes de baja resolución son obtenidas iluminando y fotografiando una escena que consiste en la muestra siendo alumbrada con un LED diferente de una matriz. La fotografía es

capturada con un objetivo de baja apertura numérica (NA, por sus siglas en inglés) para obtener un campo de visión amplio a expensas de la resolución.

La figura 2 da cuenta del estado de la composición de la fotografía de resolución mejorada de la muestra en el dominio de Fourier para un LED específico de la secuencia de iluminación. La secuencia sigue el camino indicado por la flechas de color rojo hasta haber encendido la última lámpara de la matriz LED. En la figura, la pupila cumple el rol de concentrar y dirigir la luz hacia la cámara y podría estar compuesta de más lentes como un objetivo.

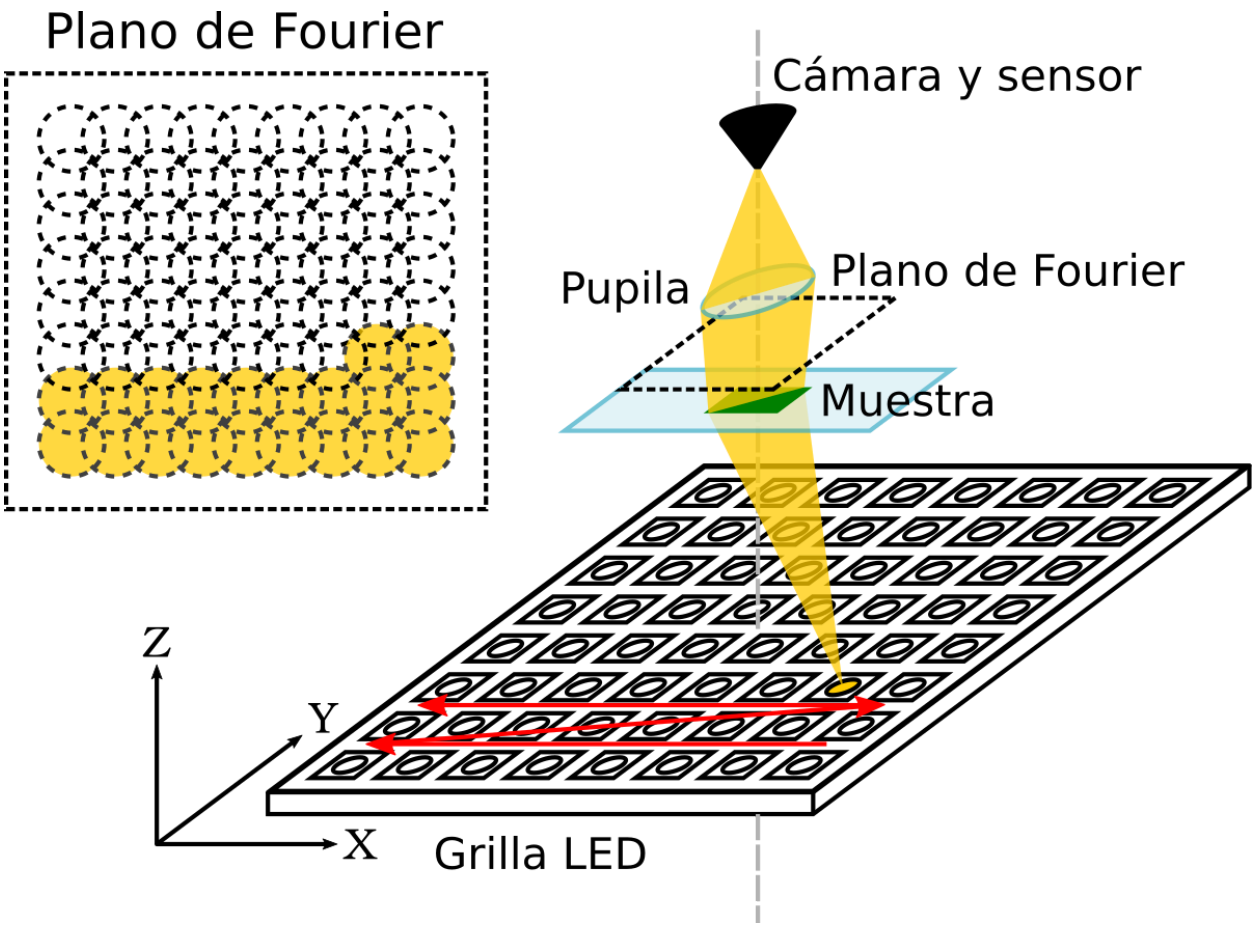

Figura 2. Reconstrucción de la imagen en alta resolución en el dominio de Fourier usando Fourier Ptychography. Elaboración propia.

1.4 Justificación de una solución

En el año 2017, Monica Retamal, directora ejecutiva de la Fundación Kodea y representante en Chile de Code.org®, una organización sin fines de lucro dedicada a expandir el acceso la ciencias de la computación e incrementar la participación de mujeres y minorías [9], afirmó que

la demanda laboral en tecnologías en STEM representa el 73% y que tan solo el 8% de quienes la realizan son egresados provenientes de especialidades afines a ese campo [8]. Esa declaración puede validar la afirmación «el cambio metodológico [en la enseñanza es algo] que tenemos claro que debemos realizar» (C. Rojas, comunicación personal, 30 de mayo de 2017) de la profesora Claudia Rojas.

Dada la calificación en el uso de la TIC de la Escuela Aliro Lamas Castillo, la transformación en la metodología de la enseñanza que la profesora Rojas lidera en esa institución educativa, la importancia de la programación que reconocemos desde la década de los 50 es que, ha modo de apoyar la clases STEM de informática educativa de la escuela, se ha decidido llevar a cabo el diseño y construcción de un módulo compatible con un microscopio de bajo costo y otro para el OpenFlexure de WaterScope, de partes imprimibles en 3D con licencia para ser modificadas y distribuidas libremente, y que haga uso del *software* libre del proyecto SuperScanner que mejora la calidad de la imágenes con la técnica Fourier Ptychography. Para lograrlo, son planteados los siguientes objetivos específicos como parte de este trabajo de memoria de título:

- Crear una interfaz gráfica de usuario (GUI) basada en notebooks Jupyter para establecer los parámetros de configuración para la obtención de una imagen de calidad mejorada con un microscopio o a partir de un directorio de fotografías fuente.

- Crear un contenedor de Docker para la automatización del proceso de instalación de SuperScanner Software (S3) y sus dependencias en máquinas que ejecuten distribuciones de GNU/Linux o versiones del sistema operativo Windows.

- Diseñar y construir una lámpara de múltiples LEDs independientes para el OpenFlexure Microscope u otro de bajo costo para iluminar la muestra a fotografiar.

- Crear un módulo de declaraciones y definiciones para compatibilizar S3 con el microscopio.

- Vincular SuperScanner Software con Omero, un *software* open-source para manipulación, visualización y análisis de imágenes de microscopía.

Figura 3. Imagotipo que identifica al proyecto Microscopx. Por F. Ojeda para Microscopx.

2 Diseño del *hardware* de la solución

2.1 Caracterización del microscopio de bajo costo

U500X Digital Microscope es el microscopio de bajo costo a utilizar, fabricado por COLEMETER. Al 11 de marzo de 2018, el producto aún podía ser comprado en Amazon por USD$28,99 (CLP$17.459[6]). Al proveedor del servicio de importación se le pagaron $23.610 (IVA incluido) por traerlo a Chile. El producto fue entregado entre el 23 y 24 de mayo del 2017.

El microscopio tiene una cámara que puede capturar fotografías a una resolución de imagen máxima de 640 píxeles × 480 píxeles (0.3 Mpx). Se conecta al PC mediante puerto USB. El rango de magnificación se puede ajustar manualmente entre los 50× y 500×, según el manual en inglés que trae el CD de instalación, o 25× y 200× [23], según otro manual disponible en Internet del mismo autor y la publicación en Amazon del fabricante. La distancia focal es entre 5 y 40 mm.

El sensor de imagen es un GC0308 [24] fabricado por GalaxyCore Inc. de resolución 640×480, tipo 1,65″ y con tamaño de píxel 3,4μm × 3,4μm [25].

Adosado al cuerpo de la cámara, existe un anillo de 8 LEDs conectados en serie. El brillo puede regularse manualmente. En las etapas del proceso de reconstrucción de la imágen en alta resolución a partir del *set* de baja resolución usando Fourier Ptychography, hacemos uso de esos LEDs solo para alinear la escena a capturar con el Pixel central del iluminador LED que se ubica entre la muestra y la base del microscopio[7].

El fabricante asegura el funcionamiento del dispositivo en ambientes de temperaturas que oscilan entre los 0 y 40 grados Celsius y de humedad relativa entre 45% y 85% [23].

[6] Equivalencia de divisas al lunes, 12 de marzo de 2018.
[7] Ver sección 2.3 Diseño del iluminador LED.

Figura 4. U500X Digital Microscope junto al disco de instalación y la regla de calibración incluidas en el empaque. Por COLEMETER[8].

2.1.1 Resolución de imagen

Para verificar la máxima resolución de imagen soportada, en Linux, utilizamos el comando `lsusb`. Con `lsusb`, se obtiene una línea en la salida como:

[8] Imagen disponible en http://colemeter.easyofficial.com/colemeter-digital-microscope-endoscope-magnifier-p16.html

```
Bus 001 Device 007: ID 1871:7670 Aveo Technology Corp.
```

Luego, ejecutando `lsusb -s 001:005 -v | egrep "Width|Height"`, en donde `001` es el número de bus y `005` el del dispositivo, se observó que el máximo valor para `wWidth` es `640` y para `wHeight` es `480`.

2.1.2 Poder de resolución

El poder de resolución de un instrumento óptico es medido por la habilidad de diferenciar dos líneas o puntos en un objeto. El valor máximo está asociado a la mínima distancia entre ellos que se puede distinguir. Este puede calcularse con la fórmula de Rayleigh [26]:

$$\varepsilon = 0,61 \times \frac{\lambda}{NA}$$

en donde ε es el poder resolución, λ es la longitud de onda de luz y NA el valor de la apertura numérica del objetivo.

El valor de NA en caso de que no sea proporcionado por el fabricante se puede determinar a partir del número f, $f\#$, que equivale a la razón entre la distancia focal del lente, F, y el diámetro de la pupila de entrada, D[9][27]:

$$NA = \frac{1}{2f\#}\ ,\ f\# = \frac{F}{D}\ .$$

La tabla 2 resumen el cálculo del poder de resolución, ε , para el U500X usando la longitud de onda de la luz visible para las distancia focales especificadas por el fabricante COLEMETER (5 mm y 40 mm).

λ	$f\#$	NA	ε
0,55 µm	1,77	0,28	1,19 µm
	14,13	0,04	8,39 µm

Tabla 2. Cálculo del poder de resolución de U500X Digital Microscope usando luz visible. Elaboración propia.

2.1.2 Rango del campo de visión

Para calcular el campo de visión (FoV), fue girado el tambor que varía el aumento del microscopio hasta el punto mínimo y luego el máximo. Para cada escenario, diez veces, fue

[9] El diámetro de la pupila de entrada es 2,83 mm. Fue medido con un pie de metro digital.

desplazado el microscopio a través del tubo vertical hasta percibir con nuestros propios ojos un grado de nitidez aceptable y, a partir de la cantidad de líneas verticales y horizontales visibles de la sección de Coordinate Shaped de una regla de calibración (figuras 5 y 6), fue determinada una longitud visible con un error de ±0.05 mm. Luego, a cada lista de valores (tabla 3), le fue aplicada la media como medida de tendencia central debido a la inexistencia de valores extremos. Los resultados son resumen en la tabla 4.

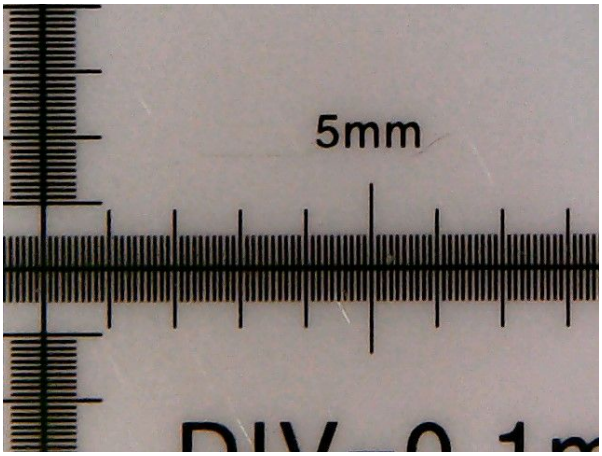

Figura 5. Sección de la regla de calibración usada para calcular el FoV para el aumento mínimo del microscopio. Elaboración propia.

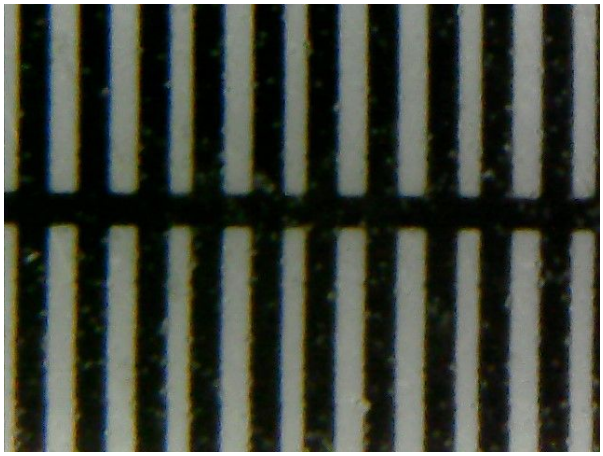

Figura 6. Sección de la regla de calibración usadas para calcular el FoV para el aumento máximo del microscopio. Elaboración propia.

Aumento	Mínimo		Máximo	
N.o de medición	Longitud horizontal [mm]	Longitud vertical [mm]	Longitud horizontal [mm]	Longitud vertical [mm]
1	9,0	6,5	1,0	0,7
2	8,9	6,7	1,0	0,7
3	9,0	6,6	1,0	0,7
4	9,0	6,6	1,0	0,7
5	8,9	6,8	1,0	0,7
6	8,9	6,6	1,0	0,7
7	9,0	6,7	1,0	0,7
8	8,9	6,6	1,0	0,7
9	9,0	6,7	1,0	0,7
10	8,9	6,7	1,0	0,7

Tabla 3. Registro de diez mediciones de los valores de longitud horizontal y vertical visible a través el microscopio para los valores mínimo y máximo de aumento del U500X Digital Microscope. Elaboración propia.

Aumento mínimo		Aumento máximo	
Long. horiz. prom. [mm]	Long. vert. prom. [mm]	Long. horiz. prom. [mm]	Long. vert. prom. [mm]
8,95	6,65	1,0	0,7

Tabla 4. FoV promedio para los valores mínimo y máximo del aumento del microscopio. Elboración propia.

2.2.3 Rango de magnificación

Para el cálculo de la magnificación (M), es posible utilizar una ecuación que tiene por datos la densidad de píxeles del monitor (DP) desde donde se observa la muestra bajo el microscopio,

la longitud horizontal física de la escena a fotografiar (FoV) y el número de píxeles de ancho (W) [28]:

$$M = \frac{1}{FoV} \times \frac{W}{DP}$$

DP [PPI]	N.o de píxeles de ancho (W)	FoV [mm]	Magnificación (M)
166	640	8,95	10,9×
		1,0	97,9×
96		8,95	18,9×
		1,0	169,3×
72		8,95	25,2×
		1,0	225,8×

Tabla 5. Rango de magnificación del U500X Digital Microscope para distintas densidades de píxeles de monitores. Elaboración propia.

2.3 Diseño del iluminador LED

2.3.1 NeoPixels LEDs

'NeoPixel LEDs', 'NeoPixels', 'Pixel LEDs' o simplemente 'Pixels' son dispositivos optoelectrónicos diseñados por Adafruit que, además de uno o más LEDs, incorporan un microcontrolador embebido para su control independendiente cuando integran, por ejemplo, una tira o anillo [29]. Esas características sumadas al hecho de que el fabricante ofrece una biblioteca en Arduino para programar los NeoPixels influenciaron en la decisión de elegirlos como fuentes luminosas para el iluminador que se encarga de alumbrar la muestra bajo el microscopio. Específicamente, fueron adquiridos dos NeoPixel Rings de 16 y 24 Pixels y un NeoPixel Jewel de 7 NeoPixels[10] por USD$11,95 (CLP$7.192), USD$19,95 (CLP$12.008) y USD$6,95 (CLP$4.183), respectivamente[11].

[10] Ver figura 9.

[11] Además, fueron comprados un *pack* de 10 NeoPixels LEDs por USD$4,5 (CLP$2.709) y un motor *stepper* de 32 pasos por USD$4,95 (CLP$2.979). El costo del envío por UPS fue de USD$42,18 (CLP$23.385). Equivalencias de divisas al martes, 13 de marzo de 2018.

Figura 7. NeoPixel LED. En la mitad superior se encuentran el microcontrolador y los LEDs rojo, verde y azul; en la mitad inferior, el LED de color blanco. Elaboración propia.

2.3.2 Diagrama electrónico

La figura 8 muestra las conexiones necesarias para controlar el iluminador (borde izquierdo) compuesto de NeoPixels. El dispositivo del borde inferior es un Arduino UNO R3. Los cables de color negro dan cuenta de conexiones a tierra (GND); los cables rojos, conexiones a la salida de 5 V de la fuente de alimentación de corriente continua como un cargador de pared (*wall wart*, en inglés); los verdes, la ruta por la cual circula los datos de entrada a través del *pin* 6 en el Arduino —podría ser otro del mismo tipo.

El condensador (16 V, 1.000 μF) conectado entre los terminales + y - de la fuente de alimentación previene daños a los Pixels por el efecto del *peak* de corriente en el inicio [30].

El Arduino reconoce a la unión de las configuraciones como una única cadena de NeoPixels, ya que el *pin* digital 6 está conectado a la entrada (IN) del NeoPixel Jewel y la salida de ese, a la entrada del anillo de 16 elementos y la salida (out o OUT) de este a la entrada del segundo.

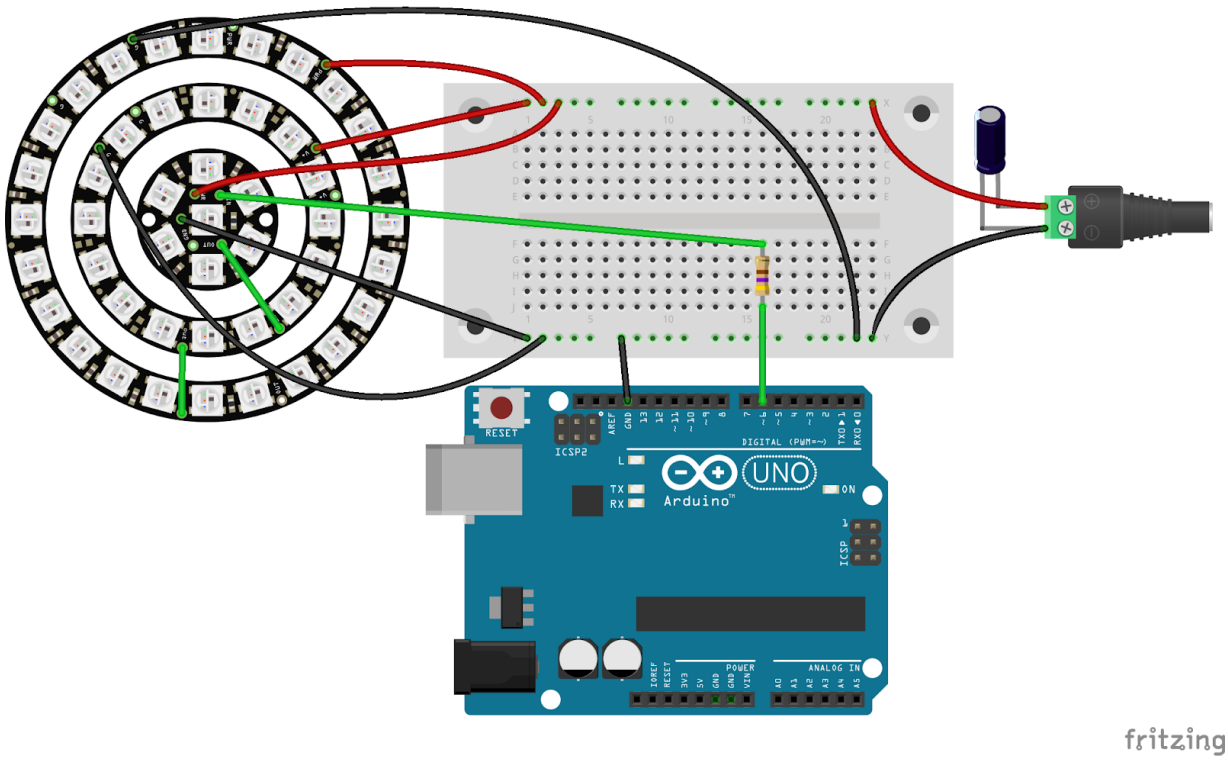

Figura 8. Diagrama de conexiones entre el Arduino y el iluminador. Elaboración propia con Fritzing.

2.3.3 Soporte imprimible para los componentes del iluminador

En Fourier Ptychography, se hace uso de las coordenadas (x, y) de los LEDs que conforman el iluminador para, por ejemplo, determinar los vectores de onda incidentes. Un modo de asegurar que la distancia radial sea la misma para cada Pixel de un anillo es construyendo un soporte con canaletas como el de la figura 9. El diseño fue hecho en FreeCAD y fueron utilizados los archivos EagleCAD PCB descargables que ofrece pública y gratuitamente el fabricante de los NeoPixels LEDs en su sitio *web*[12].

[12] Visitar https://learn.adafruit.com/adafruit-neopixel-uberguide/downloads

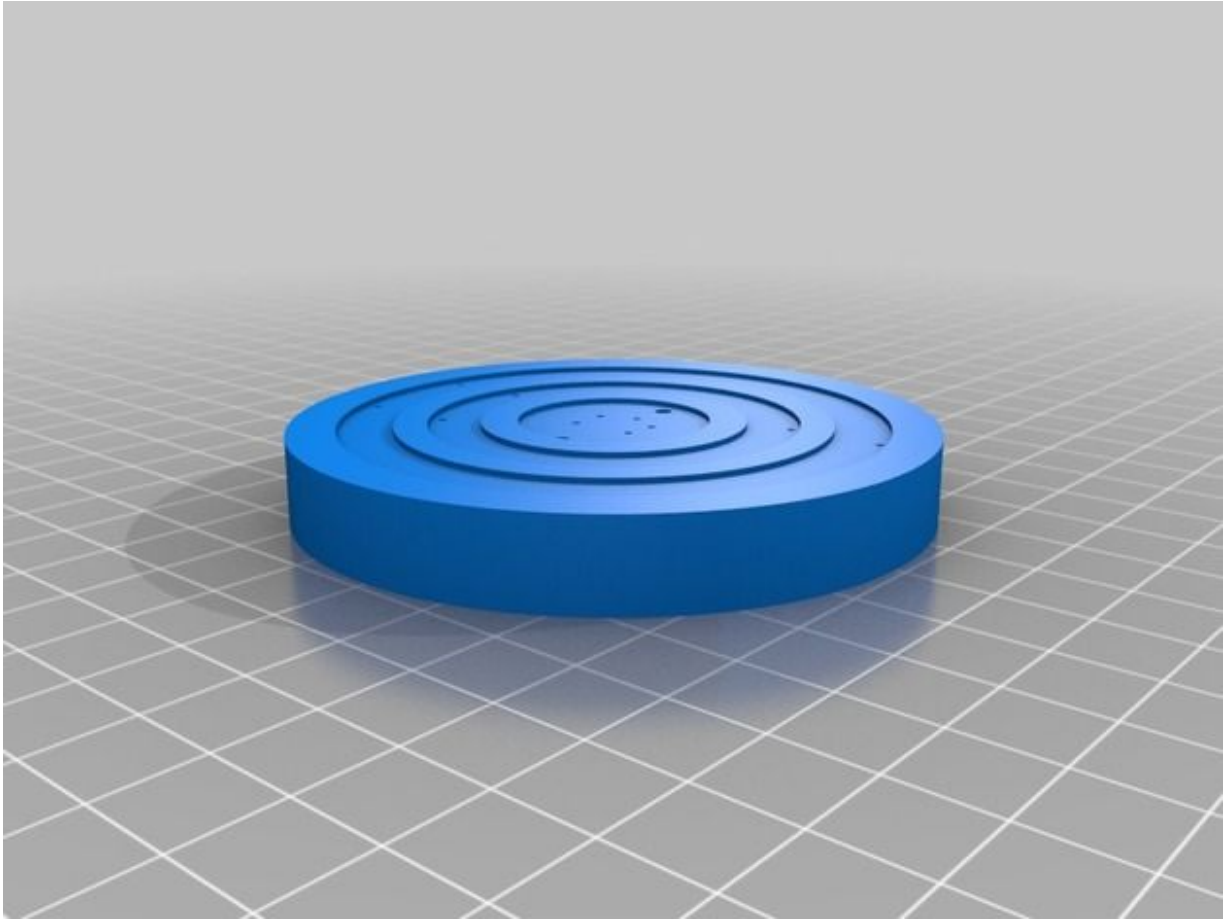

Figura 9. Soporte imprimible en 3D para el iluminador (dos NeoPixel Rings de 16 y 24 elementos y un NeoPixel Jewel de 7 Pixels)[13].

En las primeras pruebas que fueron realizadas para probar la técnica Fourier Ptychography con un *set* de imágenes capturado por el microscopio COLEMETER, las distintas configuraciones de Pixels yacían sobre una tapa de un envase cilíndrico de papas fritas como aparece en la figura 10. Si bien, se consiguió asegurar la distancia interconfiguración, fue especialmente dificultoso alinear el iluminador y mantener su posición. Por ello, es que fue diseñada la base de la figura 11. En ella, el cilindro inferior es de menor diámetro que el del área circular en desnivel sobre la cual se asienta. Las aletas facilitan la manipulación y pueden ser usadas para fijar el soporte con un trozo de cinta adhesiva. El soporte de la figura 9 se empotra sobre la apertura superior de la base.

[13] Captura realizada desde la página de Microscopx en Thingiverse en https://www.thingiverse.com/thing:2819042

Figura 10. Iluminador de NeoPixels usado para las primeras pruebas de Fourier Ptychography. Elaboración propia.

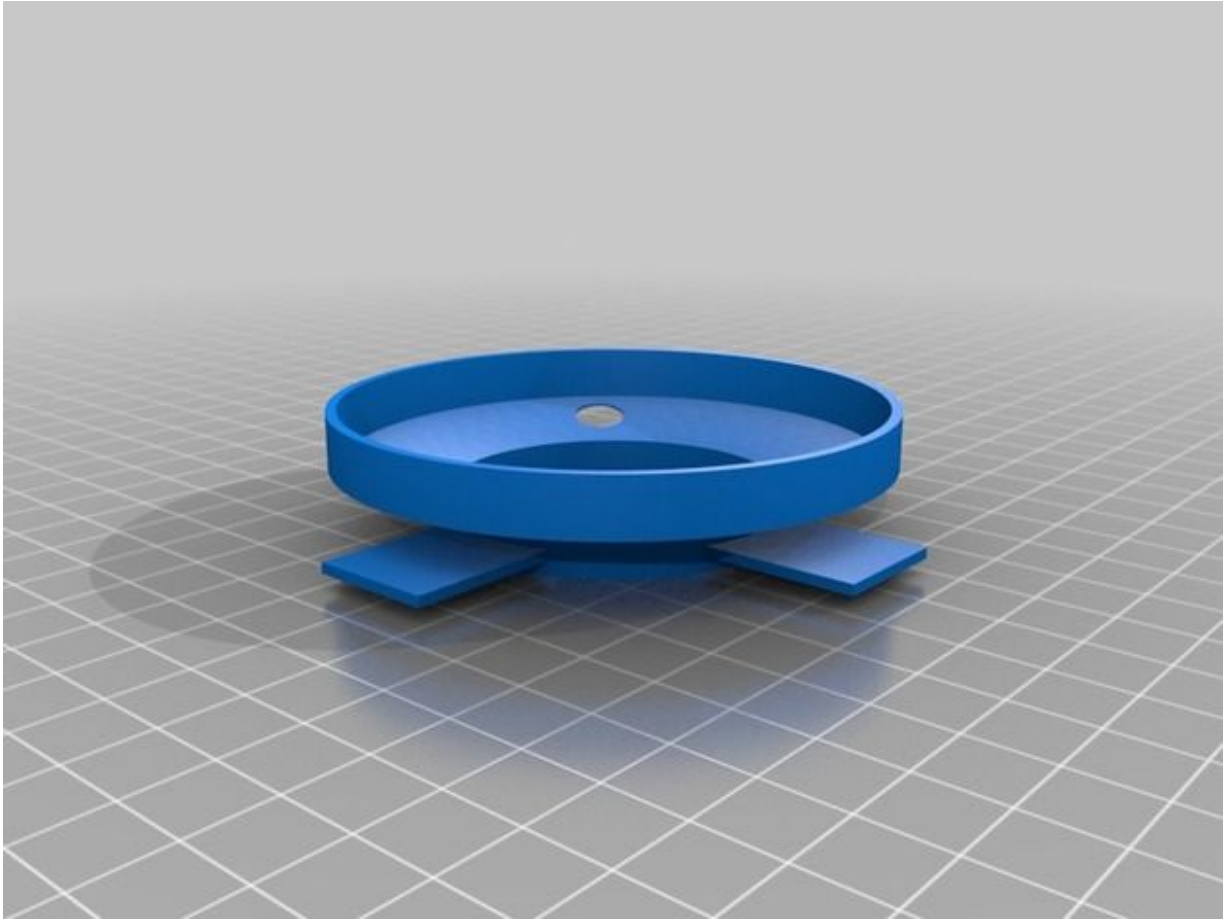

Figura 11. Base imprimible en 3D para el soporte iluminador de NeoPixel LEDs[14].

2.3.4 Soporte imprimible para las muestras

En el microscopio de COLEMETER (figura 4), el espécimen se ubica sobre una área circular de color blanco que es parte de la base. Ese diseño impide que, por ejemplo, la muestra a observar sea iluminada desde abajo de ella —la lámpara original del microscopio es un anillo de 8 LEDs inserto en el cuerpo de la cámara—. Como es requerida la iluminación inferior, fue diseñado un soporte cilíndrico de una tapa de 58 mm de alto, 130 mm de diámetro exterior y 1.5 mm de espesor (figura 12). En la tapa, existe un hueco circular de 30 mm de diámetro que permite el paso de la luz desde el iluminador hacia la muestra. En el manto del cilindro, hay dos ranuras de 4 mm para pasar los cables que van hacia el Arduino o la fuente de poder. Las conexiones también pueden atravesar el cilindro a través de sus patas.

[14] Captura realizada desde la página de Microscopx en Thingiverse en https://www.thingiverse.com/thing:2819042

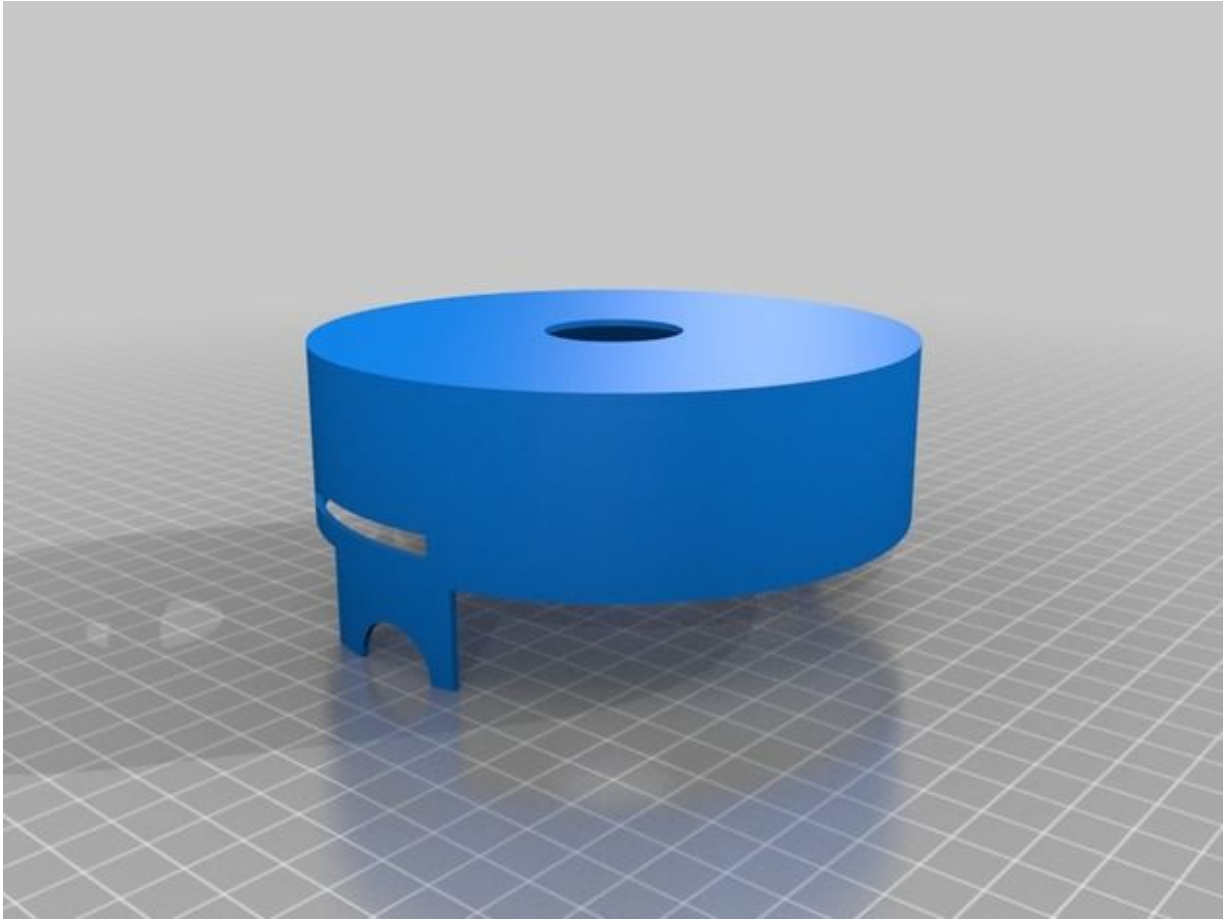

Figura 12. Soporte imprimible en 3D para muestras compatible con el microscopio de COLEMETER[15].

2.3.5 Arduino *sketch* para el control del iluminador

Sketch es el nombre que recibe un programa en Arduino. Como el fabricante de los NeoPixels ofrece una biblioteca en Arduino para controlarlos, fue creado un *sketch*, turnpixelon.ino. A continuación, se detallan las funciones que lo componen:

[15] Captura realizada desde la página de Microscopx en Thingiverse en https://www.thingiverse.com/thing:2819042

`setLightingSequence`*()*

Crea una secuencia de iluminación personalizada que se exhibe por pantalla para reemplazar la de ejemplo (`const int PIXE_SEQ[]`).

Después de la llamada de esta función, son válidas las siguiente escrituras en el puerto serial: **`f`** (por el vocablo en inglés ***f**orward*) enciende el siguiente Pixel de la secuencia real y apaga el actual, **`b`** (por ***b**ackward*) enciende el anterior y apaga el actual, **`jf`** (por ***j**ump **f**orward*) hace lo mismo que **`f`**, pero tantos Pixels hacia adelante como el valor de la constante `JUMP_STEPS` (`const int`) definida dentro de la función; **`jb`** tiene el mismo comportamiento de **`jf`**, pero hacia atrás; **`push`** hace un inserción del índice que identifica al Pixel en la secuencia real en el *array* que almacena la secuencia personalizada, **`pop`** remueve el último Pixel de la secuencia personalizada, **`show`** llama a **`showSequence()`** y **`q`** finaliza este proceso.

- Retorno:
    - `0` (`int`) cuando se escribe **`q`** en el puerto serial.

`printLightingSeq`*()*

Imprime en pantalla la secuencia de iluminación personalizada como una lista entre llaves ('{' y '}') de números que representan los índices de los Pixels en la secuencia real.

`showSequence`*(currentPixelIdx)*

Enciende uno a uno los Pixels de la secuencia de iluminación personalizada.

- Parámetros:
    - **`currentPixelIdx`**: (`int`) valor del índice en el array del último Pixel añadido mediante la operación `push`.

testPixels*()*

Ejecuta una rutina para que el usuario verifique visualmente el funcionamiento de todos los Pixel LEDs que componen el iluminador. El número total de ellos se identifica en el *sketch* con `#define MAX_NUM_OF_PIXELS <n.o total de Pixels>`.

loop*()*

Es el *loop* principal que corre el microcontrolador del Arduino. Enciende o apaga un NeoPixel cuando se escribe en el puerto serial **<índice que identifica al Pixel>,<nivel de brillo>,<modo>**. Los modos aceptados son **r**, **g**, **b**, **w**, **rgb** y **bgr** en donde **r** enciendo el LED rojo; **g**, el verde; **b**, el azul; **w**, el blanco y **rgb** y **bgr** prenden el rojo, verde y azul al mismo tiempo.

Cuando se escribe **useseq**, se usa la secuencia de iluminación personalizada (`const int PIXEL_SEQ[]`). El uso se interrumpe con **noseq**, el valor inicial. Otras escrituras aceptadas son **clear** para apagar todos los NeoPixels, **setseq** para llamar a `setLightingSequence()` y **test** para `testPixels()`.

3 Creación de los módulos Jupyter

3.1 Requerimientos

1. Del proyecto SuperScanner presentado en noviembre de 2016, las funciones utilizadas para generar la imagen en alta resolución a partir del *set* de imágenes de baja resolución simuladas deben funcionar con el *set* imágenes reales capturadas con la cámara del microscopio.

2. La matemática detrás del algoritmo de Fourier Ptychography es una caja negra para el usuario.

3. El control del microscopio y cualquier otro dispositivo se debe realizar por medio de un *notebook* Jupyter escrito en Python 3.

4. Los nuevos módulos o bibliotecas deben escribirse como *notebooks* y ser importables fácilmente desde otros.

5. Para las salidas en pantalla, se debe establecer una sintaxis común a los eventos que ocurren durante la comunicación de dispositivos y las etapas de la aplicación de la técnica de imagen computacional, y la lectura y escritura de archivos.

3.2 Módulo `microscopx`

El módulo `microscopx` es un *notebook* Jupyter, escrito en Python 3 e importable que permite:

1. Calcular las coordenadas (x, y) de todos los Pixel LEDs que componen un iluminador tipo anillo o joya haciendo uso de las especificaciones técnicas publicadas por el fabricante Adafruit.

2. Tomar una fotografía con la cámara del microscopio a una muestra por cada Pixel que se enciende del iluminador.

'Microscopx' hace referencia al vocablo en inglés para 'microscopía', ***microscopy***, y la abreviación para '*pixel*', **px**, por 'NeoPixel', el nombre de los LEDs utilizados para construir el iluminador. A continuación son detalladas las funciones que componen este módulo:

```
get_pixels_xy(rings={7: 0, 16: 0, 24: 0}, clockwise=True, cm=True)
```

Calcula las coordenadas (x, y) de los Pixel LEDs de un anillo (NeoPixel Ring) o una joya (NeoPixel Jewel). Retorna las coordenadas como elementos de una lista para cada iluminador.

- Parámetros:
    - **`rings`**: diccionario (`dict`) en donde la clave (`int`) del par clave-valor es el número de Pixels que posee el anillo (NeoPixel Ring) o joya (NeoPixel Ring) a usar y el valor (`float`), el ángulo en grados sexagesimales medido desde eje X al primer Pixel a encender.
    - **`clockwise`**: (`boolean`) si es `True`, fija la dirección del encendido de los Pixel LEDs de iluminador en sentido horario; si es `False`, en sentido antihorario.
    - **`cm`**: (`boolean`) si es `True`, establece al centímetro como unidad de longitud para el cálculo de las coordenadas (x, y); en cambio, si es `False` hace uso de la pulgada (2,54 cm).
- Retorno:
    - Un diccionario (`dict`) de valores tipo lista (`list`) de coordenadas (`tuple`) y llaves tipo número entero (`int`) para el par clave-valor. Por

`take_photos`*(cam, n_photos, baud=9600, brightness_level=225, cam_wait=0, img_prefix='IMG_', img_format='TIF', outpath='')*

Ilumina una muestra ubicada bajo un microscopio con un NeoPixel LED a la vez y realiza una captura fotográfica de la escena con la cámara del instrumento.

- Parámetros:
    - **`cam`**: (`int`) índice de la cámara identificado por el kernel de Linux.
    - **`n_photos`**: (`int`) número de capturas fotográficas a realizar. Para la primera fotografía se usa el primer Pixel LED de la secuencia de iluminación personalizada; para la segunda, el segundo, y así sucesivamente.
    - **`baud`**: (`int`) tasa de datos en bit por segundo para la transmisión serial.
    - **`brightness_level`**: (`int`) nivel de brillo de los Pixel LEDs. El rango es entre 0 y 255. El valor 0 apaga el Pixel.
    - **`cam_wait`**: (`float`) tiempo de espera medido en segundos previo al envío de la señal de apagado del último Pixel encendido.
    - **`img_prefix`**: (`string`) prefijo común para el nombre de todos las fotografías.

- **img_format**: (string) formato de archivo a usar para almacenar las fotografías capturadas por la cámara del microscopio en el PC. La extensión de archivo se genera automáticamente a partir del valor de este argumento.
- **outpath**: (string) ruta del directorio en donde se desea almacenar las fotografías.

- Retorno:
  - La ruta al directorio donde fueron almacenadas las fotografías (string).
- Levantamiento de excepciones:
  - **KeyboardInterrupt**: cuando se interrumpe la ejecución de la función o cuando finaliza el proceso de toma de fotografías.
  - **serial.SerialException**: cuando no se detecta un Arduino conectado al PC.
  - **NameError**: cuando la cámara del microscopio no se encuentra o no está disponible.

Para asegurar la sincronía entre el PC y el Arduino en el proceso de captura fotográfica, una vez que el primero encuentra un Arduino y una cámara conectadas a él, crea el directorio para las fotografías —y los padres de ser necesario—, envía una señal de reinicio (**reset()**) al segundo y espera a que este escriba **<ArduinoReady>** en el puerto serial. Si el Arduino se desconectara en ese momento u otro, se levantaría la excepción **serial.SerialException** y se cerraría la conexión con la cámara; si el usuario cancelara la ejecución de la función, ocurriría **KeyboardInterrupt** y serían cerrados el puerto serial y la cámara.

Ya con el Arduino listo y todos los Pixel LEDs apagados, PC informa a Arduino que hará uso de la secuencia de iluminación personalizada con **useseq**[16], espera por la confirmación (**<CustomSeqUse>**) y tantas veces como el valor de **n_photos** (int):

1. Solicita el encendido de los LEDs rojo, verde y azul (**rgb**) al mismo tiempo del Pixel **x** a un nivel de brillo **brt** en modo RGB con **{x},{brt},rgb** en donde **{x}** es el valor de la variable de control en el ciclo *for* y el índice del Pixel en la secuencia personalizada.
2. Arduino escribe **<Arduino-PixelRGBOn-{y}>** y enciende el Pixel **y**. **{y}** es la posición del LED en la secuencia real[17].
3. PC espera por el segmento **<Arduino-PixelRGBOn-** para tomar una fotografía y almacenarla. Transcurridos **{cam_wait}** segundos, solicita el apagado del Pixel **x** escribiendo **{x},0** en el puerto serial.

[16] Ver sección 2.3.5 Arduino *sketch* para el control del iluminador.
[17] *Ibídem*.

4. Arduino apaga el Pixel LED y escribe **`<Arduino-PixelRGBOff-{y}>`**.

5. PC espera por **`<Arduino-PixelORGBff-`** para continuar con la siguiente iteración.

La figura 13 presenta el diálogo entre las entidades PC y Arduino en un diagrama simplificado que también podría representarse mediante una máquina de estados. En el lado izquierdo de la línea de PC, son mostrados los mensajes de la salida en pantalla del *notebook* Jupyter que llama la función `take_photos()`; en el centro, la señal y los mensajes que son intercambiados entre PC y Arduino.

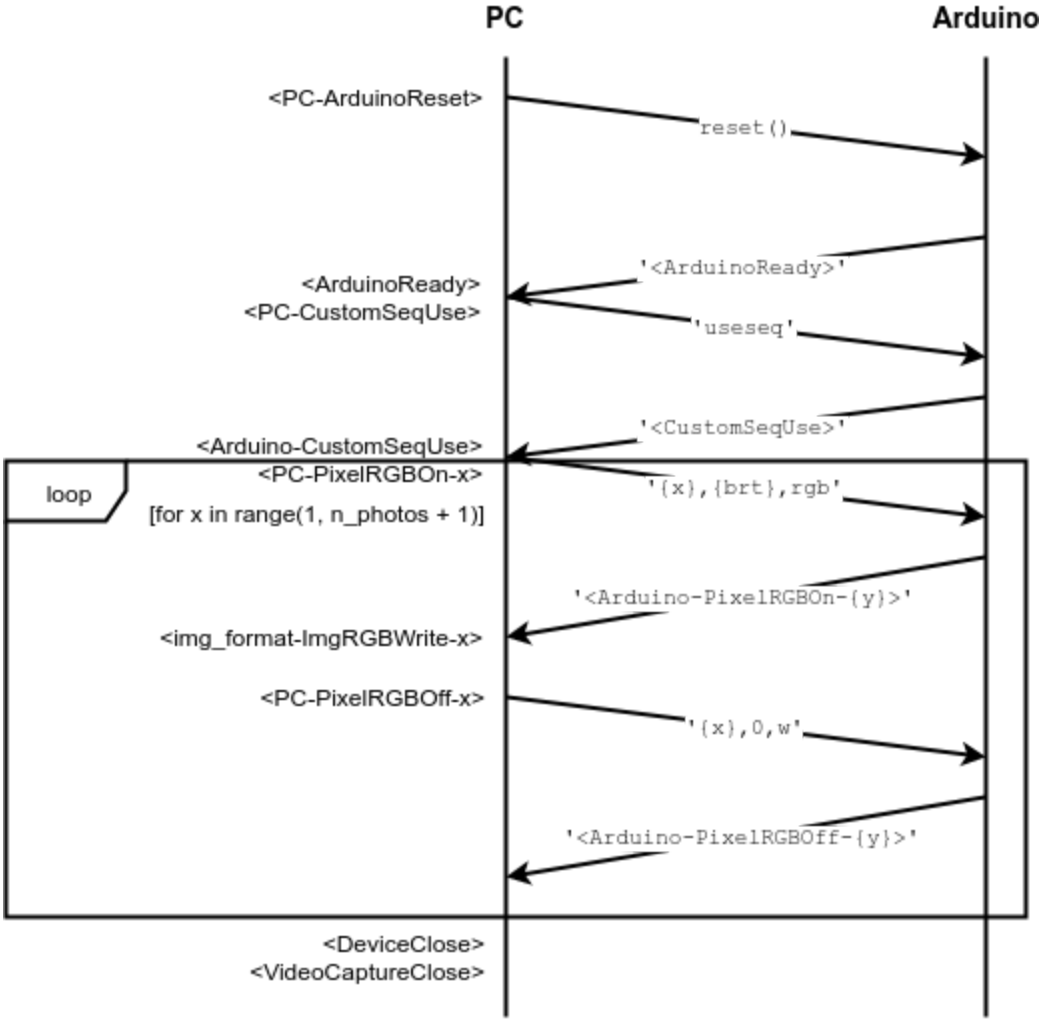

Figura 13. Diagrama simplificado para el diálogo entre el PC y el Arduino que controla los Pixel LEDs. Elaboración propia.

3.3 Módulo `fputilities`

El módulo `fputilities` contiene funciones comunes a las etapas de captura de fotografías de la muestra tanto reales como simuladas y la de reconstrucción en Fourier Ptychography (FP). Como los demás, este módulo es un *notebook* Jupyter escrito en Python 3. Son detalladas, a continuación, las funciones que lo conforman:

`read_lores_img_set`*(img_set_path, n_leds, img_prefix='IMG_', img_format='TIF_')*

Importa el *set* de imágenes de baja resolución que es el resultado de sucesivas capturas fotográficas de una muestra bajo un microscopio que se alumbra con diferentes ángulos de incidencia. Retorna el *set* como una pila de imágenes en forma de *array*.

Solo pasan a formar parte del *set* de imágenes importadas aquellas cuyos nombres de archivo coincidan con la expresión regular que se compone a partir de un prefijo común, un número y una extensión de archivo.

El *set* de baja resolución puede obtenerse por simulación con fwdimaging.ipynb[18]

- Parámetros:
    - **`img_set_path`**: (`string`) ruta al directorio donde se almacenan las fotografías a importar.
    - **`n_leds`**: (`int`) número de LEDs usados en el proceso de captura fotográfica. Cada fotografía se asocia a un LED diferente partiendo desde el primero, por ejemplo, con el primer LED del iluminador, se captura la primera fotografía; con el segundo, la segunda, y así sucesivamente.
    - **`img_prefix`**: (`string`) prefijo común del nombre del archivo de todas las fotografías a importar.
    - **`img_format`**: (`string`) formato de archivo de todas las fotografías a leer. La extensión del archivo es generada con ese valor.
- Retorno:
    - Un *array* (`numpy.ndarray`) de imágenes también como *array*.

[18] Para más información sobre los módulos involucrados en el algoritmo Fourier Ptychography, visitar https://github.com/josemarroquintoledo/superscanner-software-s3

`round_half_up`*(num)*

Redondea un valor numérico como lo hace la función `round()` de Python 2.

En Python 3, la función `round()` cambió. Por ejemplo, `round(2.5)` retorna `2` (`int`) como `round(1.5)` y no `3`.

- Parámetros:
    - **`num`**: (`int`) número a redondear.
- Retorno:
    - El número entero (`int`) que resulta del redondeo del valor de `num`.

3.4 Creación de una sintaxis para todos los eventos

La creación de una sintaxis simple y común para informar eventos ocurridos durante las etapas del proceso que conduce a la obtención de una imagen de mayor resolución cobra especial importancia:

1. En el diálogo que acontece entre el PC y el Arduino o entre el PC y la cámara en el proceso de captura fotográfica.
2. En la exhibición de un único tipo de salida en pantalla en los notebooks Jupyter.
3. Durante la inspección que un usuario realiza a la salida de un programa.

Por considerarla simple y de fácil comprensión, se optó por adaptar la sintaxis de los eventos Tkinter [31], un módulo Python para la creación de interfaces gráficas:

```
<modificador-tipo-detalle>
```

`tipo` es es una acción que ocurrió, transcurre o está por suceder. Esta se compone de un sujeto y una acción o característica, en inglés y con la primera letra en mayúsculas. Por ejemplo, **`PixelRGBOn`** que quiere decir *encender Pixel en modo RGB*; **`ArduinoReady`**, *Arduino listo*; **`MicroscopeNotAvailable`**, *microscopio no disponible*; **`DeviceClose`**, *cerrar dispositivo*.

Como se observa en el diagrama de la figura 13, el tipo **`PixelRGBOn`** es común tanto para el mensaje que imprime el PC como para el que escribe el Arduino, entonces, en ese tipo de casos es cuando se hace uso del **`modificador`**. El **`modificador`**, por ejemplo, puede diferenciar la fuente de la acción: **`Arduino-PixelRGBOn`** puede interpretarse como que *Arduino dice "encender el Pixel en modo RGB"*; **`PC-PixelRGBOn`**, *PC dice "encender el Pixel en modo RGB"*. Finalmente, **`detalle`** precisa la acción, por ejemplo, **`PixelRGBOn-1`** significa *encender el Pixel 1 en modo RGB*; **`PixelRGBOn-47`**, *encender el Pixel 47 en modo RGB*. Entonces, pueden

construirse mensajes como **<PC-PixelRGBOff-1>** o **<Arduino-PixelRGBOff-1>**, incluyendo los símbolos menor que, <, y mayor que, >.

4 Resultados

4.1 Obtención de la imagen en alta resolución

Las figuras 14, 15 y 16 son capturas fotográficas de una escena en la que la regla de calibración es iluminada por un NeoPixel diferente con los LEDs rojo, verdes y azul al mismo tiempo. En la figura 17, la luz proviene de la lámpara adosada al cuerpo de la cámara que incluyó el fabricante.

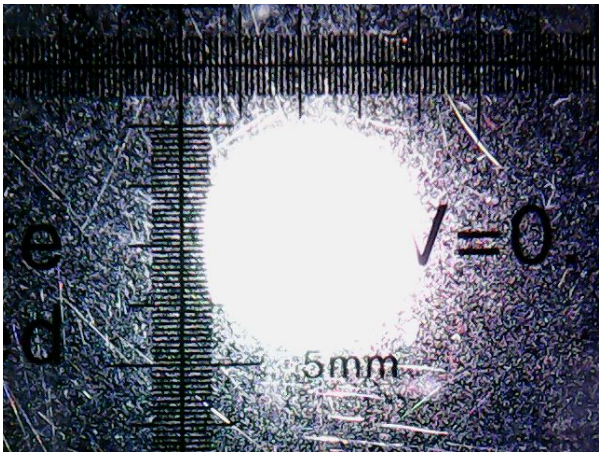

Figura 14. Foto de la regla de calibración con el Pixel LED del centro del iluminador encendido.

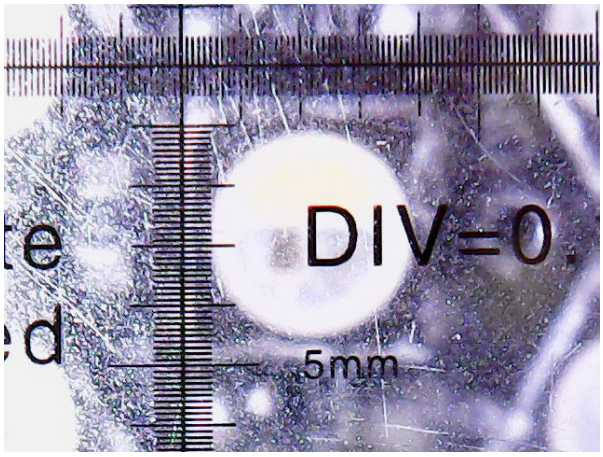

Figura 15. Foto de la regla de calibración con un Pixel LED del anillo de 16 NeoPixels encendido (53 mm del centro aprox.).

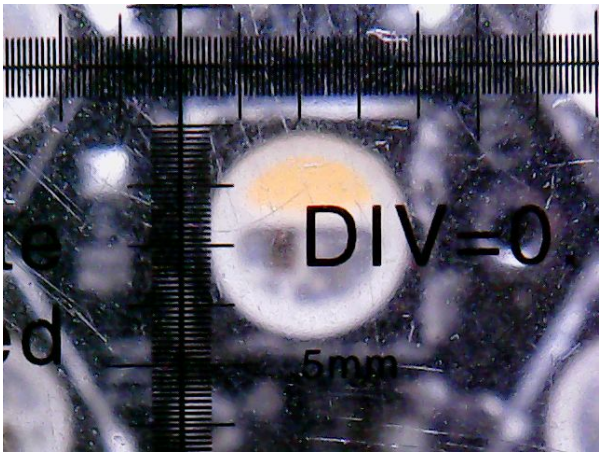

Figura 16. Foto de la regla de calibración con un Pixel LED del anillo de 24 NeoPixels encendido (105 mm del centro aprox.). Elaboración propia.

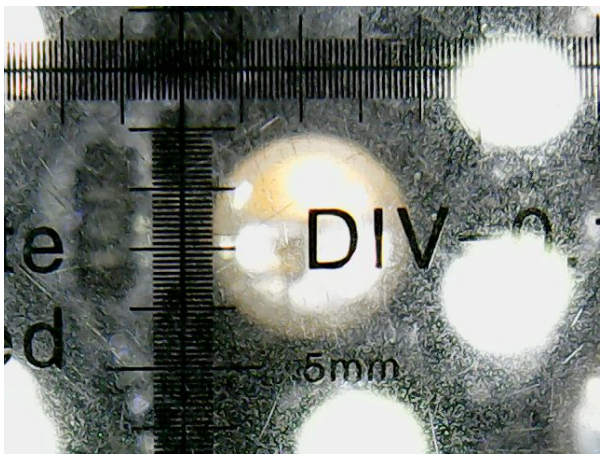

Figura 17. Foto de la regla de calibración iluminada con la lámpara de la cámara. Elaboración propia.

Los sensores de imagen, CCDs, solo pueden medir variaciones de intensidad de la onda de luz. La información de la fase, es decir, cuánto se retrasa la luz durante la propagación, es perdida en el proceso de captura. Fourier Ptychography recupera la fase haciendo uso de las diferentes mediciones de intensidad, es decir, las diferentes capturas fotográficas de la misma escena iluminada desde un punto diferente del espacio [26].

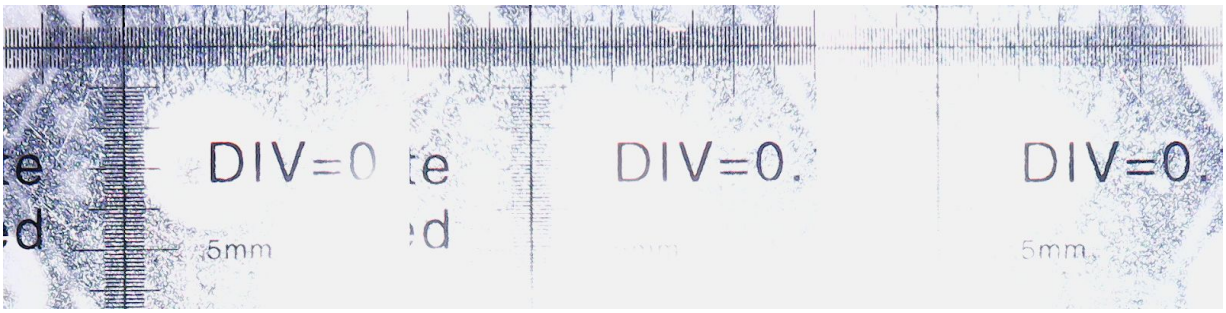

Figura 18. Secuencia de iluminación sucesiva usando tres NeoPixels. Elaboración propia.

La figura 18 enseña parte de la secuencia de iluminación sucesiva de la cual se obtuvieron 47 imágenes RGB en formato TIF de resolución 640×480. Se utilizó la mínima magnificación soportada por el microscopio, 25,2× a 72 ppi, ya que para ese valor la distancia focal es conocida, 5 mm.

En el proceso de reconstrucción usando la función `gen_hires_img()` del módulo Jupyter `phaseretrieval`[19], el *set* de imágenes de fuente es procesada por canales obteniéndose una reconstrucción por separado para el rojo (R), verde (G) y azul (B) para luego ser mezclados. El resultado es el que se muestra en la figura 19. Está escalado desde los 2560×1920.

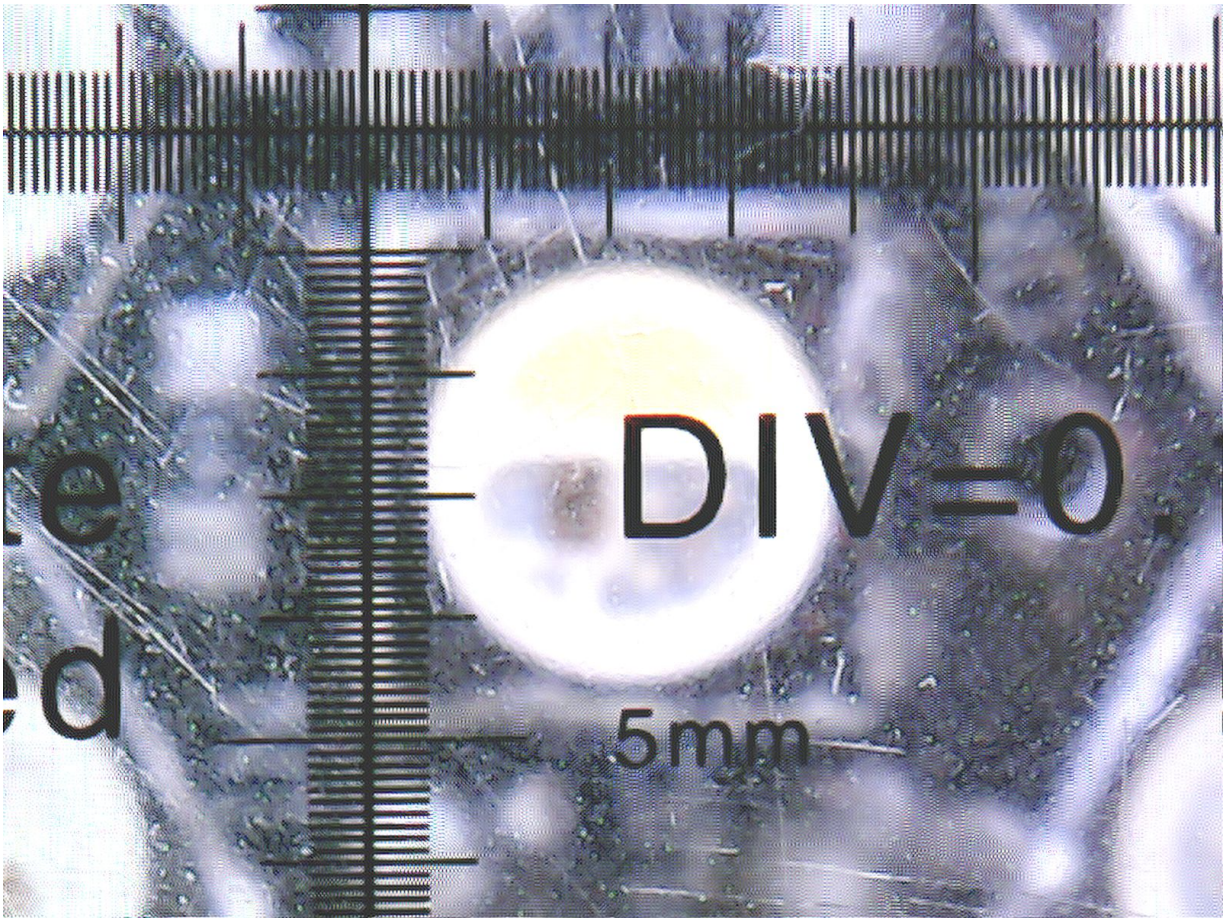

Figura 19. Reconstrucción de la escena a partir de 47 fotografías iluminadas con diferentes ángulos de incidencia. Elaboración propia.

El valor de la longitud de onda de la luz (λ) que ilumina la muestra es requerido por Fourier Ptychography. El fabricante de los NeoPixels, Adafruit, en el *datasheet* del NeoPixel, especifica el valor para los LEDs rojo, azul y verde. Es posible adquirir NeoPixels que tiene un LED blanco, pero λ tendría que determinarse con un instrumento de medición. Por ese motivo, es

[19] Para más información sobre los módulos involucrados en el algoritmo Fourier Ptychography, visitar https://github.com/josemarroquintoledo/superscanner-software-s3

que la escena se ilumina con luz roja, verde y azul proveniente del NeoPixel y la reconstrucción se realiza por canales.

4.2 Caracterización del nuevo instrumento

Cuando fue caracterizado el U500X Digital Microscope de COLEMETER[20], fueron determinadas la resolución de imagen, el poder de resolución, el rango del campo de visión y el rango de magnificación.

La nueva resolución de imagen es 2560×1920 que equivalen a 4,92 Mpx (5 Mpx).

Para el cálculo del poder resolución (tabla 6), hacemos uso de la fórmula de Rayleigh[21] con los valores de las longitudes de onda de los LEDs rojo, verde y azul de los Pixels, especificados por el fabricante, y el valor de la apertura numérica del sistema de iluminación. Este último es el seno del ángulo de incidencia respecto a la horizontal del Pixel más alejado.

λ	NA	ε
0,625 µm (rojo)	$sin(acos(\frac{17}{29,2})) = 0,8$	0,48 µm
0,522,5 µm (verde)		0,40 µm
0,470 µm (azul)		0,36 µm

Tabla 6. Poder de resolución del Microscopx para los LEDs rojo, verde y azul de los Pixels. Los valores para λ son el promedio de los valores extremos del rango especificado por el fabricante Adafruit [32]. Elaboración propia.

La tabla 7 resume el cálculo del rango de magnificación para el Microscopx con el nuevo valor de W.

[20] Ver sección 2.1 Caracterización del microscopio de bajo costo.
[21] Ver sección 2.1.2 Poder de resolución.

DP [PPI]	N.o de píxeles de ancho (W)	FoV [mm]	Magnificación
166	2560	8,95	×43,8
		1,0	×391,7
96		8,95	×75,7
		1,0	×677,3
72		8,95	×101,0
		1,0	×903,1

Tabla 7. Rango de magnificación del Microscopx para distintas densidades de píxeles de monitores. Elaboración propia.

Figura 20. Microscopx durante el proceso de captura de fotografías de baja resolución. Elaboración propia.

5 Conclusiones

5.1 Resultados alcanzados

En esta memoria se entrega el desarrollo de Microscopx, instrumento que fue concebido a partir de la funcionalidad de microscopía de SuperScanner. La problemática surgió después de una visita que el autor realiza a la Escuela Aliro Lamas Castillo en la Región de Atacama por invitación de la profesora C. Rojas entre los días 21 y 23 de octubre del año 2016, a un mes de la XXIV Feria de Software de la UTFSM, evento en el que fue presentado SuperScanner. En esta visita, se expone el estado de avance del proyecto SuperScanner y tanto los alumnos como los profesores y las autoridades de la institución mostraron gran entusiasmo por las características de la máquina (C. Rojas, comunicación personal, 30 de mayo de 2017). En el primer semestre de 2017, en el curso de Informática y Sociedad dictado por el profesor L. Contreras del Departamento de Informática de la UTFSM, el autor compromete como parte de una de las actividades de la asignatura, la de identificar un problema social real que puede ser resuelto con tecnología, la entrega del artefacto para microscopía a la escuela en el mediano plazo para apoyar las clases STEM de Informática Educativa de la profesora Rojas.

Microscopx corrobora el aumento del poder de resolución y magnificación de un microscopio y la resolución de imagen de una fotografía de un espécimen que es iluminado desde diferentes ángulos de incidencia mediante la técnica de imagen computacional Fourier Ptychography. Para ello, hace uso de las funciones de los módulos `fwdimaging` y `phaseretrieval` de SuperScanner Software (S2H), además de los nuevos `microscopx` y `fputilities`. A finales del segundo semestre de 2017, S2H reconstruía la imagen de mayor calidad a partir de un *set* de fotografías simuladas; hoy, éste es capaz de generarla con un *set* simulado y real.

Microscopx adapta el U500X Digital Microscope fabricado por Colemeter de USD$28,99 (CLP$17.459[22]) para Fourier Ptychography añadiendo un iluminador en la base del instrumento compuesto por configuraciones de 7, 16 y 24 NeoPixels de Adafruit y un soporte para la muestra a observar. Tanto el soporte para la muestra como el del iluminador son imprimibles en 3D. El iluminador es controlado por un Arduino Uno Rev3 que carga el *sketch* Arduino `turnpixelon`.

En la conversación que el PC y el Arduino llevan a cabo para sincronizar la cámara del microscopio con la iluminación consecutiva de los Pixels y en la notificación al usuario de los eventos transcurridos tales como la desconexión del Arduino o la cámara, el envío de las señales de encendido o apagado de un NeoPixel, la realización de una captura fotográfica, la importación de una imagen mediante la salida de pantalla de los *notebooks* Jupyter, existe una sintaxis común para los mensajes del tipo `<modificador-tipo-detalle>`, por ejemplo, `<PC-PixelRGBOn-6>`, `<Arduino-PixelRGBOn-2>`, `<TIF-ImgRGBWrite-06>`, `<PC-PixelOff-6>` que quieren decir: *PC envía la señal encender el Pixel 6 en modo RGB*, *Arduino enciende el*

[22] Precio de venta en Amazon y equivalencia de divisas 12 de marzo de 2018.

NeoPixel 2 en modo RGB que equivale al 6 en la secuencia de iluminación personalizada, *la cámara toma la fotografía 06 en modo RGB y la almacena como TIFF*, y *PC solicita apagar el Pixel 6*, respectivamente.

Para el U500X Digital Microscope se logra:

- Incrementar el poder de resolución de 1,19 µm para f1.77, el mayor valor, a 0,41 µm[23].
- Aumentar la máxima magnificación desde 225× a 903× (72 ppi).
- Mejorar la resolución de imagen desde los 640×480 (0,3 Mpx), la máxima soportada por la cámara del microscopio, a 2560×1920 (5 Mpx).

En ocasiones, el avance en desarrollo de Microscopx se vio escabroso debido a que:

1. Valores de parámetros que son requeridos por el algoritmo de reconstrucción, como el tamaño del píxel del sensor de imagen de la cámara, no son especificados por el fabricante COLEMETER o son incorrectos como el rango de magnificación.
2. La conexión entre el PC y la cámara de U500X se pierde antes o después de un número indeterminado de usos.
3. En el proceso de impresión en 3D de las partes del iluminador y la base para la muestra, la elección de parámetros que disminuyeran el tiempo de impresión generaron defectos en el acabado de las piezas.

Para resolverlos:

1. Se recurrió a recursos externos disponibles en Internet como un vídeo en que el YouTuber NFM desarma un microscopio idéntico en la forma al U500X Digital Microscope e identifica los componentes electrónicos como el sensor de imagen [24] y otro del canal visualchallenge en que recrean en 3D los componentes que conforman el sistema óptico del microscopio [33].
2. Fue estimada en 2 la cantidad de usos de la cámara previa desconexión automática. Con ello, puede planificarse en qué momento realizar una demostración. Desafortunadamente, la desconexión perdura y los motivos son desconocidos.
3. Con la asistencia de los estudiantes del FabLab UTFSM, fue elaborado un perfil de impresión para la Ultimaker 3 Extended que tiene por valores claves 0,4 mm; 0,8 mm, 230 °C y 70 °C para la altura de la capa, el espesor de los muros laterales e inferior y superior, la temperatura de impresión y la temperatura de la placa sobre la cual se imprime, respectivamente.

Respecto al cumplimiento de otros objetivos específicos:

- La interfaz de usuario (GUI) para el control del microscopio fue reemplaza por un *notebook* de ejemplo con llamadas a las funciones que conducen a la generación de una imagen de mayor resolución que las que captura la cámara del microscopio, ya que

[23] Representa el promedio de los valores para el poder de resolución para las distintas longitudes de onda de la tabla 6.

con Microscopx y también con SuperScanner, se busca incentivar el aprendizaje de la programación.

- El contenedor Docker fue pensado para facilitar el proceso de instalación del ambiente. Se requería, por ejemplo, para importar *notebooks* en otros, interrumpir el *notebook server*, exportar y modificar un *notebook* específico como *script* en Python, almacenarlo en una ubicación particular y volver a correr el servidor. Ese procedimiento podría conllevar a que el usuario no consiguiera ver el algoritmo en funcionamiento. Durante el desarrollo de esta memoria, se logró la importación de módulos Jupyter de SuperScanner Software en cualquier *notebook* tan solo con el Magic `%run` en una línea de código. Con ello, es posible prescindir de los Dockers.
- Sobre el OpenFlexure, se presentaron dificultades en la impresión del módulo óptico y las ruedas de enfoque, motivo por el cual se abandonó la adaptación para Fourier Ptychography. Para el primero, por ejemplo, no conseguimos el calce con la estructura de soporte principal. En ese entonces, usábamos una MBot 3D Printer. Creémos que pueden obtenerse mejores resultados con una Ultimaker 3 Extended como ocurrió con las partes de Microscopx.
- Respecto al módulo Omero, hoy, no se precisa la creación de él solo para importar las imágenes en Omero, ya que puede realizarse mediante el comando CLI `import` [34] desde la celda de un notebook con anteponiendo el símbolo `!`.

5.2 Proyecciones futuras

Desde un punto de vista

1. Diseñar un iluminador que se fije al cuerpo de la cámara para que:

    a. La muestra bajo el microscopio pueda ser opaca además de traslucida.

    b. Los ejes X del iluminador y del plano XY del sensor de imagen ya no requieran ser alineados paralelamente de modo manual.

    c. Una cámara pueda instalarse en el espacio que ocupa actualmente el iluminador en la base del microscopio para el proceso de establecimiento de la secuencia de iluminación personalizada.

    d. El usuario pueda fotografiar una muestra desde distintas posiciones y ángulos de cámara para más tarde reconstruir el modelo tridimensional del objeto mediante la técnica Multi-View Stereo con el *software* de SuperScanner.

2. Reemplazar la cámara del microscopio por la Raspberry Pi Camera Module (v2), ya que:

a. Por tener un tamaño de píxel más pequeño, puede recolectar mayor cantidad de información: 1,12 µm [36] frente a los 3,4 µm de la cámara del U500X.

b. Es de mayor resolución de pantalla: 8Mpx [35] versus los 0,3 Mpx del microscopio de COLEMETER.

c. Valores como la apertura numérica y distancia focal del objetivo son especificados por el fabricante.

d. La Raspberry Pi a la cual se conecta la cámara puede actuar como servidor y el PC que ejecuta los *notebooks* Jupyter, como cliente suprimiendo de la conexión USB.

e. Prescindir de la estructura de soporte del U500X Digital Microscope de Colemeter.

3. Diseñar una estructura imprimible en 3D compatible con el microscopio de Colemeter que permita desplazar la cámara a través de una trayectoria espiral esférica, por ejemplo, para fotografiar un objeto desde distintos puntos de ese camino y reconstruir el modelo tridimensional.

4. Reemplazar el NeoPixel Jewel de 7 elementos del iluminador[24] por uno que elimine el Pixel del centro para que la cámara mire a través de él.

5. Realizar el procesamiento de las imágenes en la nube.

[24] Ver sección 2.3 Diseño del iluminador LED.

6 Referencias

[1] Ministerio de Educación. (s.f.). *MIME - Ministerio de Educación de Chile*. Recuperado de http://www.mime.mineduc.cl/mvc/mime/ficha?rbd=386

[2] Fundación SparkTalen. (2014). *Ganadores - First Lego League*. Recuperado de http://www.firstlegoleague.cl/torneos/regionales/ganadores/2/3

[3] Hom, E. (2014). *What is STEM Education?* Recuperado de https://www.livescience.com/43296-what-is-stem-education.html

[4] Dutta, S. and Mathur, R. (2de abril de 2011). *Computer Programming – A Building Block of STEM*. 2011 IEEE Integrated STEM Conference (ISEC). Conferencia llevada a cabo en Ewing, New Jersey, Estados Unidos.

[5] Fábrega, R.; Fábrega, L.; Carreño, E.; Osorio, M. y Jara, M. (septiembre de 2016). *Enseñanza de lenguajes de programación en la escuela, ¿Qué están haciendo otros países?* Recuperado de http://www.fundaciontelefonica.cl/wp-content/uploads/descargas/1504030044-Enseñanza%20de%20Lenguajes%20de%20Programación%20en%20otros%20países.pdf

[6] Kodea. (23 de junio de 2015). *Se aprueba el Proyecto de Creación de Tecnologías Libres a Nivel Escolar*. Recuperado de http://www.kodea.org/se-aprueba-el-proyecto-de-creacion-de-tecnologias-libres-a-nivel-escolar/

[7] Mirosevic, V.; Fernández, M.; Espejo, S.; Sepúlveda, A.; Bellolio, J.; Jackson, G. y Boric, G . (18 de junio de 2015). *Proyecto de resolución N.º 392*. Recuperado de https://www.camara.cl/pdf.aspx?prmID=4142&prmTIPO=PACUERDO

[8] Kodea. (29 de septiembre de 2017). *Comienza la "Hora del Código" Chile 2017, el evento educativo más grande del mundo*. Recuperado de http://www.horadelcodigo.cl/rrss/2017/comunicado-oficial.pdf

[9] Code.org. (2018). *About Us | Code.org*. Recuperado de https://code.org/about

[10] Jet Propulsion Laboratory. (s.f.). *Mars Exploration Rover Mission: Technology*. Recuperado de https://mars.nasa.gov/mer/technology/is_autonomous_mobility.html

[11] josemarroquintoledo. (2016). *GitHub - josemarroquintoledo/superscanner-software-s3: SuperScanner Software (S3), part of SuperScanner project, is a open-source and completely free software environment to implement a low-cost scanner 3D, which is also a microscope and a 3D printer*. Recuperado de https://github.com/josemarroquintoledo/superscanner-software-s3

[12] Free Software Foundation. (2017). *GNU General Public License v2.0 - GNU Project - Free Software Foundation*. Recuperado de https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html

[13] marroquintoledo. (2016). *SuperScanner Hardware (S2H) by marroquintoledo - Thingiverse*. Recuperado de https://www.thingiverse.com/thing:1880928/apps

[14] Creative Commons Corporation. (s.f.). *Creative Commons Legal Code*. Recuperado de https://creativecommons.org/licenses/by-nc-sa/3.0/legalcode

[15] Project Jupyter. (2017). *Project Jupyter | Home*. Recuperado de http://jupyter.org

[16] Blender Foundation. (s.f.). *Quickstart Introduction — Blender 2.79.0 8ef39d5c882 - API documentation*. Recuperado de https://docs.blender.org/api/current/info_quickstart.html

[17] FreeCAD. (s.f.). *FreeCAD: An open-source parametric 3D CAD modeler*. Recuperado de https://www.freecadweb.org/wiki/About_FreeCAD

[18] cdcseacave. (s.f.). *OpenMVS by cdcseacave*. Recuperado de http://cdcseacave.github.io/openMVS/

[19] Zia, Zeeshan. (21 de junio de 2015). *Is there a fundamental difference between structure from motion and multi-view geometry?* [Mensaje en un foro online]. Recuperado de https://www.quora.com/Is-there-a-fundamental-difference-between-structure-from-motion-and-multi-view-geometry

[20] Spring, K.; Komatsu H.; Scott, M.; Schwartz, S.; Fellers, T.; Carr, K.; Parry-Hill, M y Davidson, M. (2017). *Microscopy Basics | MicroscopyU*. Recuperado de https://www.microscopyu.com/microscopy-basics

[21] UCLA Technology Development Group. (2017). *Computational Out-Of-Focus Imaging Increases The Space-Bandwidth Product In Lens-Based Coherent Microscopy*. Recuperado de https://techtransfer.universityofcalifornia.edu/NCD/28827.html

[22] Zheng, G. (2016). *Fourier Ptychographic Imaging: A MATLAB® tutorial*. San Rafael, CA: Morgan & Claypool Publishers.

[23] Cooling Tech. (24 de septiembre de 2009). *U500X Digital Microscope User Manual*. Recuperado de https://www.kissmegoodbuys.com/eBay/Product/Mobile/Accessories/Others/DMS-USB-500X-1_User_Manual.pdf

[24] NFM. (NFM). (2017). *ebay USB Microscope Teardown* [YouTube]. Recuperado de https://www.youtube.com/watch?v=Cvnf4S2GdU4&t=6s

[25] GalaxyCore Inc. (s.f.). *GC0308: product brief*. Recuperado de http://www.chinesechip.com/files/2015-05/e72f61e0-fdc8-4f1f-8059-c1b224375d31.pdf

[26] Nikon Instruments Inc. (2017). *The Diffraction Barrier in Optical Microscopy | MicroscopyU*. Recuperado de

https://www.microscopyu.com/techniques/super-resolution/the-diffraction-barrier-in-optical-microscopy

[27] Newport Corporation. (2018). *Light Collection and Systems Throughput*. Recuperado de https://www.newport.com/t/light-collection-and-systems-throughput

[28] March Ho. (1 de agosto de 2015). *Calculation of final magnification when using a CCD camera*. [Mensaje en un foro online]. Recuperado de https://biology.stackexchange.com/questions/36528/calculation-of-final-magnification-when-using-a-ccd-camera

[29] Adafruit. (s.f.). *NeoPixel RGBW LEDs w/ Integrated Driver Chip - Natural White [~4500K - Black Casing - 10 Pack] ID: 2761 - $5.95 : Adafruit Industries, Unique & fun DIY electronics and kits*. Recuperado de https://www.adafruit.com/product/2761

[30] Adafruit. (s.f.). *The Magic of NeoPixels | Adafruit NeoPixel Überguide | Adafruit Learning System*. Recuperado de https://learn.adafruit.com/adafruit-neopixel-uberguide/

[31] Lundh, F. (2014). *Events and Bindings*. Recuperado ×de http://effbot.org/tkinterbook/tkinter-events-and-bindings.htm

[32] Worldsemi. (s.f.). *WS2812B: Intelligent control LED integrated light source*. Recuperado de http://www.seeedstudio.com/document/pdf/WS2812B%20Datasheet.pdf

[33] visualchallenge. (visualchallenge). (2015). *Inside a USB Microscope 3d Animation* [YouTube]. Recuperado de https://www.youtube.com/watch?v=-87yvU58LFk&t=2s

[34] The Open Microscopy Environment. (2018). *Import images — OMERO 5.4.4 documentation*. Recuperado de https://docs.openmicroscopy.org/omero/5.4.4/users/cli/import.html

[35] Raspberry Pi Foundation. (s.f.). *Camera Module - Raspberry Pi Documentation*. Recuperado de https://www.raspberrypi.org/documentation/hardware/camera/README.md