

UNIVERSIDAD DE PANAMÁ

VICERRECTORÍA DE INVESTIGACIÓN Y POSGRADO

FACULTAD DE CIENCIAS NATURALES, EXACTAS Y TECNOLOGÍA

SISTEMA NOVEDOSO DE ÓPTICA REFRACTIVA INTEGRADA

Pablo Martín Weigandt Beckmann

**TESIS COMO REQUISITO PARA OPTAR POR EL TÍTULO DE DOCTOR EN
CIENCIAS FÍSICAS**

REPÚBLICA DE PANAMÁ

2022

DEDICATORIA

A mis padres, Ernesto y Norma.

AGRADECIMIENTO

A mi hermano Daniel, porque entre los dos hicimos posible este proyecto.

ÍNDICE GENERAL

RESUMEN	1
INTRODUCCIÓN	2
CAPÍTULO 1 - PROYECTO DE SENACYT FID16-010	3
1. Experiencias previas	3
1.1 Proyecto de SENACYT CVP11-017	3
1.2 Tesis de maestría.....	5
1.3 Dirección de investigación y asesoría de tesis	6
2. La convocatoria.....	7
3. Ejecución del proyecto.....	8
4. Divulgación.....	14
4.1 Artículos científicos.....	14
4.2 Congresos, visitas y página web	15
4.3 Medios de comunicación masiva	16
CAPÍTULO 2 - SNORRI: ESPECIFICACIONES TÉCNICAS	20
1. Construcción, armado y especificaciones técnicas del equipo	20
2. Comando central de operaciones	27
3. Pruebas de calidad	28
CAPÍTULO 3 - LOS DATOS: ADQUISICIÓN, CANALIZACIÓN Y CARACTERÍSTICAS	31
1. Ajuste y configuración del equipo	31
2. Adquisición y reducción de la data	33
2.1 Adquisición	33
2.2 Proceso general de reducción	34
2.3 Selección inicial de las imágenes	35
2.4 Solución astrométrica de las imágenes como modo de filtrado	36
2.5 Sustracción del fondo de cielo	38
2.6 Registro de las imágenes	38

3. Apilado	40
3.1 Calibración fotométrica	40
3.2 Combinación	43
3.3 Resumen del proceso	45
4. Resultados y análisis	48
4.1 Características astrométricas	49
4.2 Características Fotométricas	51
CONCLUSIONES	58
REFERENCIAS	59
ANEXOS	63

RESUMEN

Este proyecto consiste en una matriz de lentes súper telefoto y sus respectivas cámaras que se utilizan para observaciones astronómicas especializada en detectar estructuras u objetos celestes de poco brillo con un amplio campo de visión. Se describe la construcción de este "*Sistema novedoso de óptica refractiva integrada*" o SNORRI, como se ha dado a conocer, ubicado en Churuquita Chiquita en la República de Panamá. Se analizan diferentes aspectos del sistema relacionados con la calidad, los ajustes, la calibración y el procesamiento de las imágenes, y se presentan algunas características generales de los resultados iniciales.

SUMMARY

This project is a super telephoto matrix and their respective cameras that are used for astronomical observations specialized in detecting low surface brightness of celestial structures or objects with a wide field of view. The construction of this "*Novel System of Integrated Refractive Optics*", or in what has become known as SNORRI, is described, located at Churuquita Chiquita in the Republic of Panama. Different aspects of the system are analyzed related to quality, adjustments, calibration and image processing are covered, and some preliminary results are presented.

INTRODUCCIÓN

La Universidad de Panamá y la Universidad Autónoma de Chiriquí unen esfuerzos a través de la gestión y ejecución del proyecto FID16-010 financiado por la Secretaría Nacional de Ciencia, Tecnología e Innovación (SENACYT) de la República de Panamá. Consiste en una matriz de lentes súper telefoto y sus respectivas cámaras que se utilizan para observaciones astronómicas especializada en detectar estructuras u objetos celestes de poco brillo con un amplio campo de visión. Este Sistema Novedoso de Óptica Refractiva Integrada o SNORRI como se ha dado a conocer, fue concebido, planificado y ejecutado por los investigadores Pablo y Daniel Weigandt en la comunidad de Churuquita Chiquita de Coclé y ha generado un impacto positivo en la comunidad local y nacional.

Se describe la construcción del "Sistema novedoso de óptica refractiva integrada" (SNORRI). Este es un sistema de cuatro súper teleobjetivos alineados junto con cuatro cámaras CCD de última generación, integradas en una red. Softwares comerciales se utilizan para manejar el equipo, integrados en un centro de control de interfaz personalizado.

Este equipo astronómico fue construido en Churuquita Chiquita en el distrito de Penonomé de la provincia de Coclé en la República de Panamá. Su objetivo principal es la detección de estructuras de bajo brillo superficial u objetos tenues en el cielo con un amplio campo de visión. Se discute la buena calidad del sistema y algunos resultados preliminares.

CAPÍTULO 1 - PROYECTO DE SENACYT FID16-010

A continuación se dará una descripción general del Sistema Novedoso de Óptica Refractiva Integrada (SNORRI) y su impacto en la comunidad, en el contexto de lo que fue el Proyecto FID16-010, financiado por la Secretaría Nacional de Ciencia, Tecnología e Innovación de la República de Panamá (SENACYT). Antes, se quiere hacer notar la experiencia previa de sus creadores en gestión de proyectos de investigación independiente; así como de investigación en astrofísica, lo que llevó a concebir la participación y en una las convocatorias de la SENACYT en el 2016, que fue adjudicada dentro del 13% de los muchos proponentes, y que dio paso a la creación de SNORRI y de esta tesis.

1. Experiencias previas

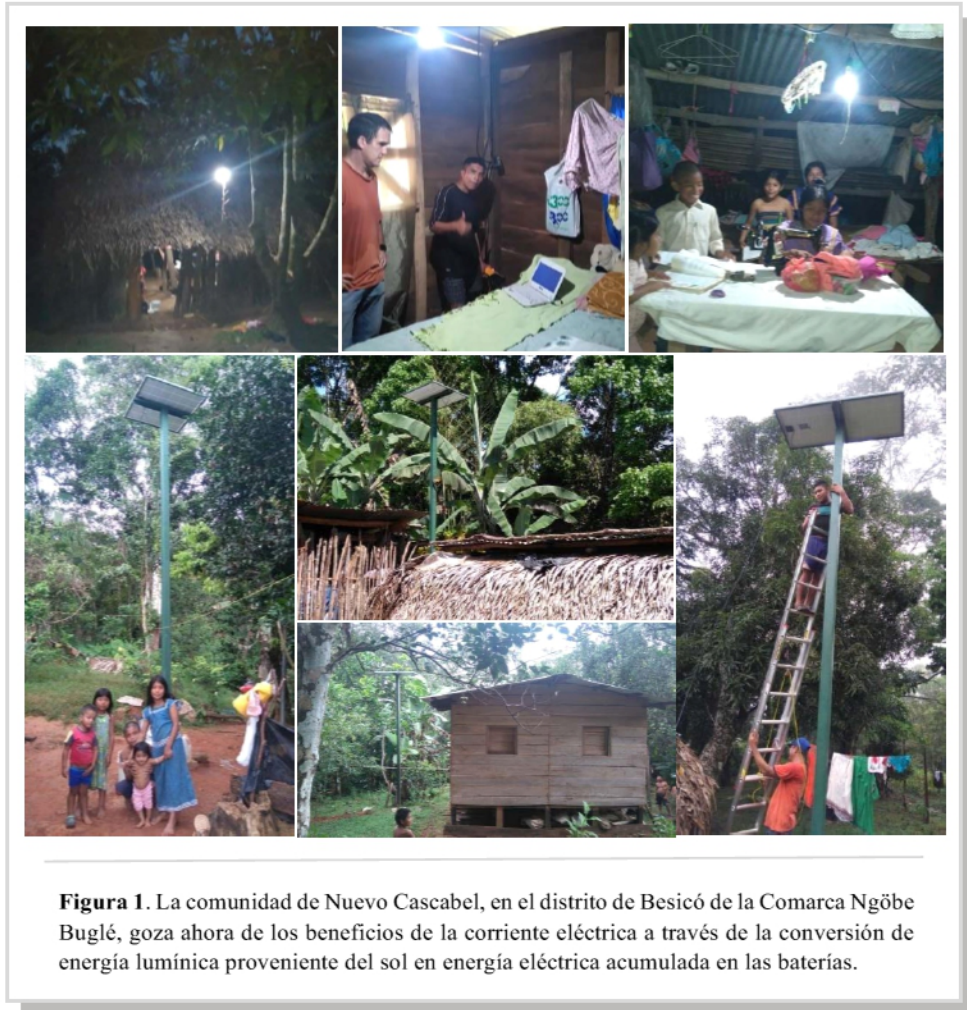
1.1 Proyecto de SENACYT CVP11-017

En el 2011 se participó en una convocatoria de la SENACYT denominada “Ciencia contra la pobreza”, cuyo objetivo era “Apoyar la generación o movilización de conocimiento, a fin de vincular el desarrollo de la ciencia y la tecnología a desafíos sociales a través de investigaciones aplicadas o desarrollos tecnológicos, que contribuyan a la identificación, análisis y solución de los problemas sociales, económicos y tecnológicos de un sector, región o grupo plenamente identificado como desfavorecido.”

Esta participación fue motivada por la enorme cantidad de proyectos de desarrollo social en que se había participado con la Misión Luterana de Panamá bajo la dirección del Reverendo Ernesto Weigandt en la construcción de acueductos rurales y proyectos de producción agrícola, que generaron una gran experiencia y permitieron tener los contactos necesarios con la comunidad objeto del proyecto en cuestión.

Se presentó y fue adjudicado el proyecto titulado “*Sistemas solares fotovoltaicos autónomos para la electrificación de una comunidad de Besigó en la Comarca Ngöbe Buglé*” el cual fue todo un éxito (Araúz, 2018). Nuestra estrella más cercana, el Sol, sirvió para otorgar a una comunidad aislada de esta Patria, la posibilidad de contar con energía

eléctrica doméstica y gozar de sus beneficios. El proyecto se comenzó a ejecutar varios años después finalizando formalmente en el 2018.



La parte principal del proyecto consistió en la instalación de 27 unidades autónomas fotovoltaicas en las viviendas de la comunidad de Nuevo Cascabel en la comarca Ngöbe Buglé que cuentan con estructuras que soportan los paneles fotovoltaicos con un poste resistente de hierro de casi 6 m de altura que garantiza la fortaleza y seguridad necesarias y evita en lo posible la tala de ramas y árboles. Y Equipamiento eléctrico de las viviendas con la unidad fotovoltaica: Cada equipamiento consta del controlador, la batería, el invertidor y los focos debidamente conectados con los cables apropiados. La instalación es resistente a las inclemencias del tiempo y apropiada para el tipo de viviendas. Por último, se instalaron también tres luminarias públicas similares a las anteriores.

Además del beneficio a la comunidad, esta iniciativa sirvió para generar otro proyecto de investigación cuyo tema es el “*Monitoreo de las condiciones climáticas y la eficiencia de un módulo pc-Si durante nueve meses (2018) ubicado en la Comunidad Nuevo Cascabel Comarca Ngäbe-Buglé, Panamá: Breve repaso sobre tecnología fotovoltaica*”, por parte de los físicos e investigadores Guillermo Araúz y Pablo Weigandt.



Figura 2. Pablo Martín Weigandt Beckmann en la presentación del proyecto en el taller internacional “Energías Renovables, Honduras 2016” de la Sociedad Mexicana de Física y el PMEXCAyC, como parte de sus actividades académicas en el marco del Doctorado en Física.

1.2 Tesis de maestría

Hace unos años se presentó la Tesis de Maestría “La entropía de Tsallis y los meteoros” en la Universidad Autónoma de Chiriquí con la asesoría del Dr. Oscar Sotolongo – Costa de la Universidad de la Habana de la hermana República de Cuba.

En esta investigación Pablo Weigandt expuso la formulación no extensiva de la Física Estadística usando la entropía de Tsallis que generaliza a la de Boltzmann – Gibbs, siendo ésta plausible para los casos en donde las correlaciones entre las partes del sistema físico juegan un papel importante y en donde las interacciones son de largo alcance.

Haciendo uso de esta formulación no extensiva y de datos astronómicos de las magnitudes luminosas de tres de las principales lluvias de meteoros, se formuló y corroboró un modelo que fortalece la teoría sobre el origen de los meteoros y cometas como producto de fragmentaciones violentas durante la edad primitiva de nuestro sistema solar (Sotolongo-Costa et al., Gámez, 2008). Se evidenció además que la formulación clásica para este caso, por medio de la Física Estadística extensiva, no concuerda con los datos y no funciona.

1.3 Dirección de investigación y asesoría de tesis

El autor de esta tesis doctoral tuvo la oportunidad de ser Director de Investigación y Documentación Científica de la Vicerrectoría de Investigación y Posgrado de la Universidad Autónoma de Chiriquí, y Director del Centro de Investigación de Física Aplicada, colaborando con el fomento y promoción de la investigación científica y, en particular, la Radioastronomía, dando lugar a la instalación de un radiotelescopio para la detección de las líneas espectrales del hidrógeno neutro. Como parte del trabajo de investigación y docencia, por parte de Pablo Weigandt, se asesoraron tesis de grado en el área (Villarreal, 2014), como un primer paso para desarrollar esta línea de investigación dentro de la universidad y a nivel nacional.



Figura 3. Radiotelescopio del Centro de Investigación de Física Aplicada en la Universidad Autónoma de Chiriquí.

2. La convocatoria

Ya para el 2013 se había concebido la posibilidad de crear un dispositivo de astronomía especializado que sirviera para propósitos específicos: observar objetos astronómicos que brillan muy poco, que están lejos en términos astronómicos, y con un amplio campo de visión. Esto está motivado por la teoría actualmente más aceptada (Modelo Lambda-CDM, s.f.) de la evolución del universo y formación de galaxias en donde debieran existir en abundancia escombros alrededor de las galaxias espirales y otros objetos como galaxias enanas, de muy bajo brillo superficial y que los telescopios convencionales no alcanzan a detectar o que detectan en poca cantidad. A esto se suma el contar con un campo de visión amplio que permita realizar una búsqueda eficiente y, a la vez, observar las estructuras ampliamente extendidas en las afueras de las galaxias. Otros esfuerzos similares se han venido dando en esta dirección (Martínez - Delgado et al., 2010; van Dokkum, Abraham & Merritt, 2014).

Con la experiencia de los investigadores del grupo se tomó la decisión de participar en el Programa de Fomento a la Investigación y Desarrollo (I+D) en la Convocatoria de Fomento a I+D (FID) 2016 de la SENACYT de Panamá.

El proyecto fue concebido por los investigadores Daniel Weigandt y Pablo Weigandt. Posteriormente se consiguió el apoyo de Omayra Pérez y Bernardo Fernández de la Estación RN50 de detección de radionucleidos del Sistema Internacional de Monitoreo del Tratado de Prohibición Completa de Ensayos Nucleares, que tiene su sede en la Universidad de Panamá. El Centro de Investigación de Física Aplicada (CIFA) de la Universidad Autónoma de Chiriquí (UNACHI), al cual pertenecen los creadores del proyecto, junto con la Universidad de Panamá, dieron su apoyo tanto para la propuesta como para la posterior ejecución del proyecto.

El proyecto participó con el título “Matriz de súper telefoto para la detección de corrientes de marea en galaxias cercanas” y fue uno de los 36 favorecidos entre al menos 275 proyectos presentados que pudimos contabilizar, adjudicándosele el código FID16-010. Esta convocatoria tenía como objetivo “Fortalecer la capacidad nacional de investigación

y desarrollo (I+D) en ciencia y/o tecnología para enfrentarnos con éxito a los grandes problemas del desarrollo nacional”.

El objetivo general planteado en nuestro proyecto fue el de contar en Panamá con un sistema óptico formado por una matriz de lentes súper telefoto para la detección de objetos de bajo brillo superficial. Se expuso la pertinencia que este proyecto tenía con el Plan Nacional de Ciencia, Tecnología e Innovación 2015-2019 (SENACYT, 2015), destacando puntos específicos, a saber: correspondencia con el objetivo de “Fortalecer la ciencia y la capacidad científica nacional”; el “derecho a participar de la ciencia, innovación y tecnología”; la misión de “desarrollar las potencialidades de la ciencia, la investigación, la tecnología y la innovación para hacer de Panamá uno de los más innovadores en la región”; el favorecimiento del Programa de “Desarrollo de la Ciencia y de las Capacidades Científicas” en particular, lo concerniente a los objetivos estratégicos de “Fortalecer la infraestructura para la generación de conocimiento” e “Incentivar la producción de conocimiento científico-tecnológico” ; y, finalmente, que se adapta perfectamente a las políticas nacionales de ciencia y tecnología y al plan nacional, tomando en consideración los desafíos nacionales planteados, en particular el de “desarrollar una mayor capacidad de generación, adaptación, difusión y utilización de conocimiento”.

3. Ejecución del proyecto

Al haber sido favorecidos por parte de la SENACYT con el código FID16-010, se procedió a investigar con más profundidad el equipo, herramientas y materiales necesarios más convenientes para concretar el sistema óptico que se propuso crear. Se adquirieron entonces lentes, cámaras, montura y demás materiales, muchos de ellos en el exterior y el equipo se puso manos a la obra.

El Sistema Novedoso de Óptica Refractiva Integrada (SNORRI) está ubicado en la comunidad de Churuquita Chiquita, Distrito de Penonomé, Provincia de Coclé en la República de Panamá en un área rural con poca contaminación lumínica y bien protegido.

SNORRI está constituido por una matriz de, por ahora, cuatro lentes súper telefotos con cuatro cámaras astronómicas, sujetas a una montura robótica, dentro de un domo de astronomía. A su vez, las cámaras están conectadas a computadoras que trabajan en red, y permiten el control de todo el sistema a través de una terminal.

Los lentes son los llamados “súper telefotos”, como los que se utilizan en eventos deportivos o para capturar la vida salvaje. Son de marca CANON, modelo EF 400mm f/2.8L IS II USM. Estos lentes permiten, junto a las características de las cámaras, tener un campo de visión amplio y capacidad de observación a “cielo profundo”. Este Sistema está diseñado, especialmente, para observaciones de objetos celestes lejanos y de bajo brillo superficial. El hecho que sean lentes “refractivos” y no “reflectantes” como los telescopios convencionales modernos, es porque son mejores para la observación de objetos de muy poco brillo.

Estos cuatro lentes, que forman una matriz, están sujetos a una estructura metálica creada en Churuquita. Acompañan recientemente a esta estructura unos anillos de ajuste para tener una perfecta alineación, de tal manera que todos los lentes estén paralelos y, como la distancia al lugar de observación es inmensa, apunten al mismo lugar en la bóveda celeste.

Las cámaras astronómicas son CCD de última generación, la SBIG Aluma 8300 de la empresa Diffraction Limited. Cuentan una CCD de alto rendimiento y un sistema de enfriamiento con una alta eficiencia.

Para este año 2021, además de la incorporación de los anillos de ajuste, se reemplazaron unos adaptadores que acoplan cada cámara con cada lente, y permiten una capacidad de enfoque más cómoda. Las cuatro cámaras están conectadas a cuatro computadoras que trabajan en red y se comunican con una terminal, que se encuentra cerca del sistema telescópico.

La estructura metálica junto con los lentes y las cámaras está sujeta a una montura robótica modelo Paramount MX, que a su vez está montada sobre un tubo cuadrado metálico firmemente adherido a la tierra con una base de cemento. La montura también está conectada a la red y se maneja a través de la terminal en el centro de control.

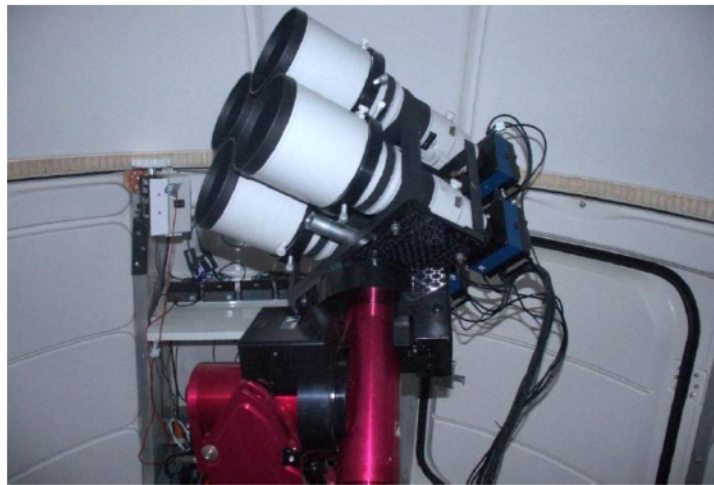


Figura 4. Sistema Novedoso de Óptica Refractiva Integrada (SNORRI): la montura, matriz de lentes súper telefoto y cámaras, conectadas a las computadoras.

Por último está el domo, con componentes de marca NexDome, que protege al equipo de las inclemencias del tiempo y sobre todo, del viento cuando se están realizando las mediciones. La abertura superior capaz de girar, permite mantener despejado el área de techo suficiente para realizar las observaciones. Este domo está apoyado sobre un piso hecho con carriolas aislado del sistema de soporte de la montura y que descansa sobre pilares que están alejados de la zona central. Todo esto para evitar en lo posible las vibraciones en el sistema.



Figura 5. SNORRI en Churuquita Chiquita.

El proceso de adquisición de la data como tal no es sencillo. Al principio es necesario ajustar y calibrar el equipo. Luego de decidir el objeto por observar, se requiere orientar el sistema telescópico a éste y darle seguimiento preciso con la montura robótica y la ayuda de una cámara guía, por períodos prolongados de tiempo, y tomar múltiples fotografías. Las imágenes que se adquieren requieren de un arduo trabajo de procesamiento para el cual se necesitan softwares especializados, tiempo y un alto grado de conocimientos en la materia.

Mencionamos también que se cuenta con un paquete de filtros astronómicos que permiten hacer diferentes mediciones y sirven para propósitos de investigación específicos.



Figura 6. Construcción del piso que soporta el domo. También se adaptaron ruedas a una tolda que protege el domo de las lluvias fuertes y es fácil de remover. En rojo, la base metálica antes de ser fijada al suelo y que ha de sostener a la montura robótica.



Figura 7. Las cuatro cámaras sometidas a un proceso de inspección y limpieza.

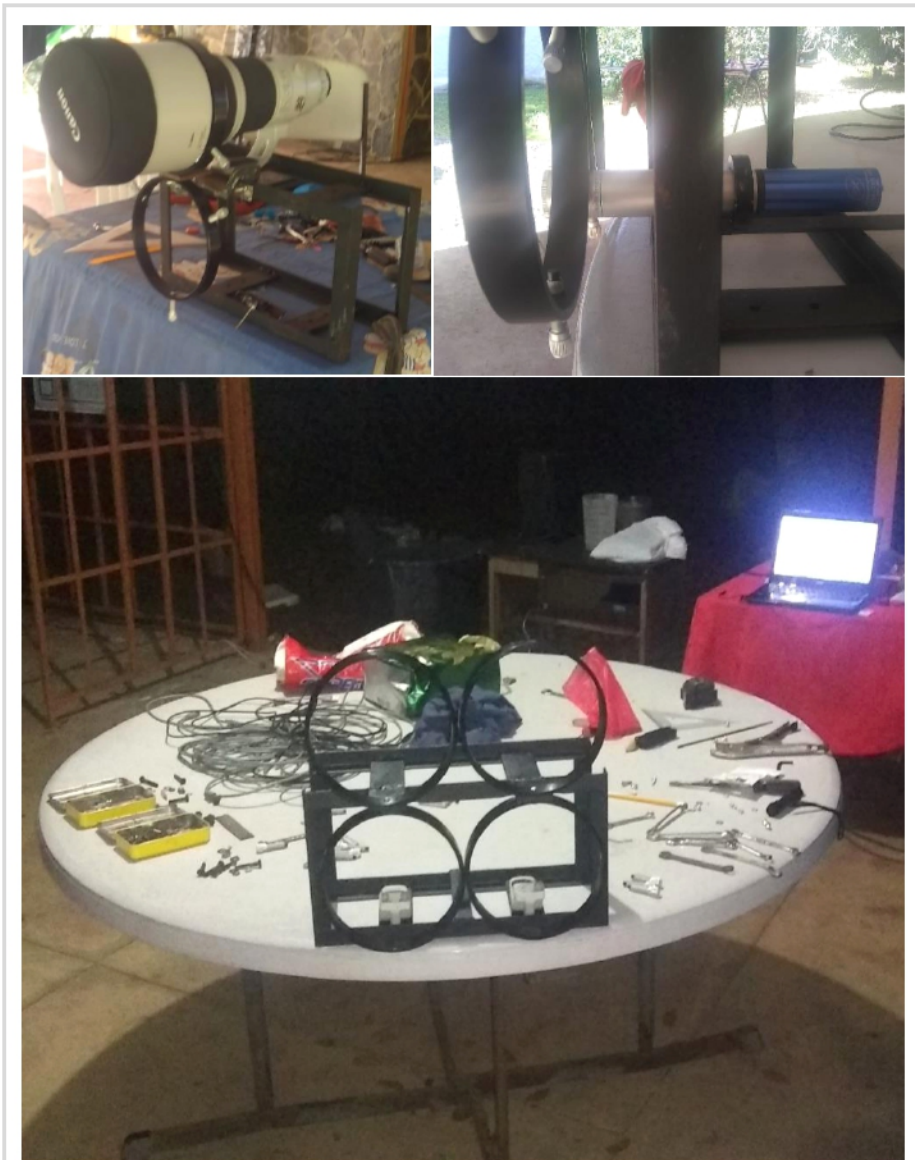


Figura 8. Se observan momentos de la construcción del soporte de la matriz de lentes súper telefoto. La base junto a los anillos sujetan los lentes. Arriba se ensaya con la base de la pequeña cámara guía y con uno de los anillos que sujetan cada lente.

Luego de tener todo debidamente instalado, se procedió a realizar pruebas de control y verificar que todo esté funcionando correctamente. De esta manera, SNORRI comenzó a trabajar y recolectar información valiosa para la investigación científica en astronomía.

El proyecto de SENACYT FID16-010 fue todo un éxito y su principal producto, SNORRI, llegó para quedarse, y realizar aportes a la investigación en Astrofísica por muchos años.

4. Divulgación

Destacamos aquí el impacto que ha tenido el proyecto y la divulgación que se le dio tanto en el ámbito académico y científico como en la comunidad en general.

4.1 Artículos científicos

El proyecto generó inmediatamente la publicación de los artículos titulados “Matriz de súper telefotos para la detección de objetos astronómicos de bajo brillo superficial en Churuquita Chiquita de Coclé, Panamá” en la revista *Tecnociencia* (Weigandt & Weigandt, 2020) y “SNORRI: Un proyecto de astrofísica en el seno de la comunidad coclesana” en la revista *Guacamaya* del Centro Regional Universitario de Coclé (Weigandt, Weigandt y Márquez, 2021).



Figura 9. El proyecto fue favorecido, participando con nuestras propias imágenes, en la portada de la Revista Guacamaya de la Universidad de Panamá, en su Volumen 5, Num. 2 de abril a septiembre del 2021.

4.2 Congresos, visitas y página web

- Participación de los investigadores como expositores de conferencias y sesión de pósters en la jornada de divulgación del Centro de Investigación de Física Aplicada de la Facultad de Ciencias Naturales y Exactas de la Universidad Autónoma de Chiriquí – agosto del 2019.



Figura 10. El coordinador del proyecto SNORRI dictando una conferencia y pesentando un póster en el Centro de Investigación de Física Aplicada de la UNACHI.

- Visita al proyecto de Miembros de la Unión Astronómica Internacional: el Profesor Eduardo Chung de la Facultad de Ciencias Naturales, Exactas y Tecnología de la

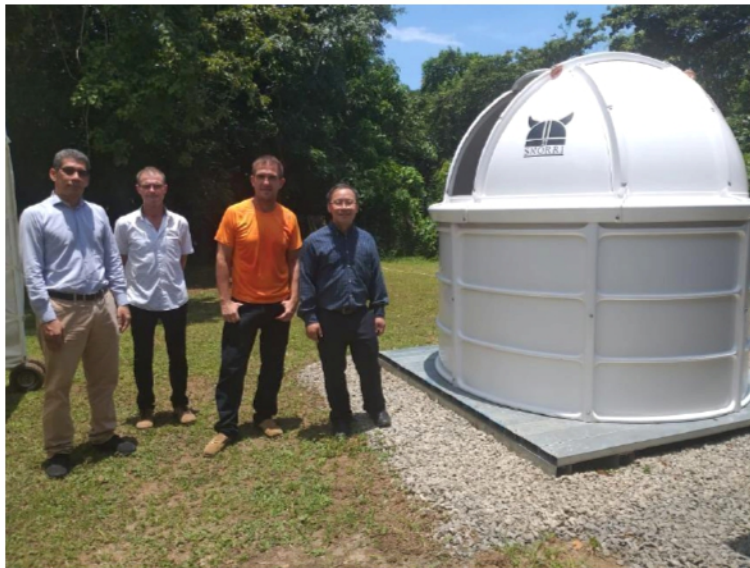
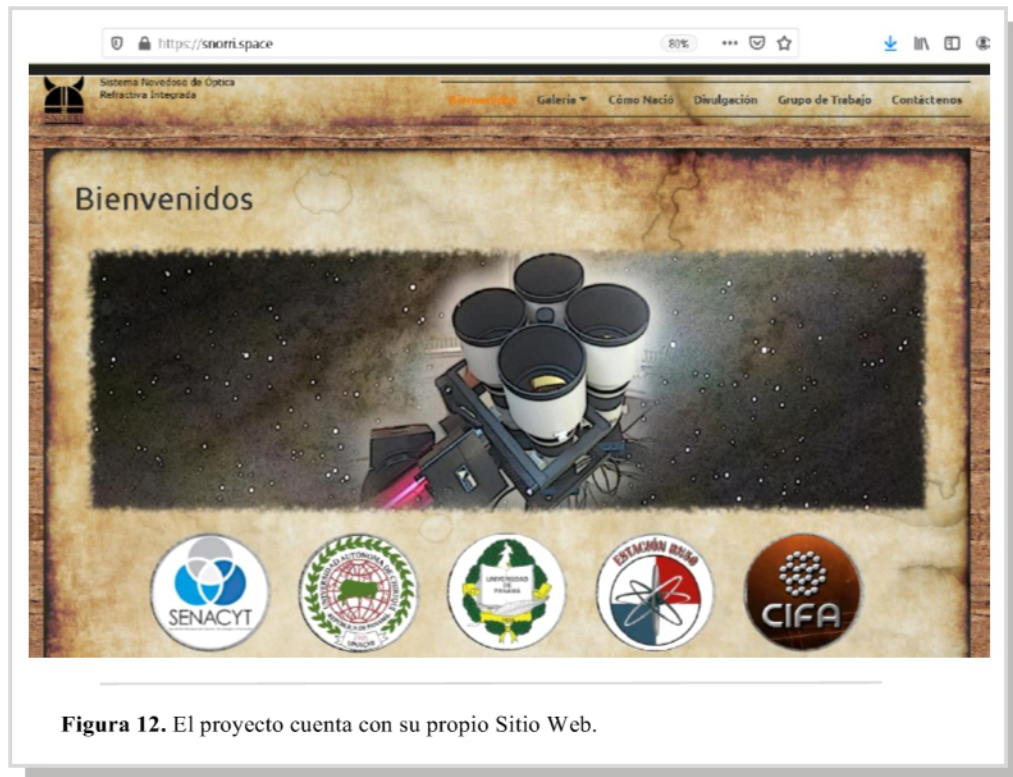


Figura 11. Visita a Churuquita por parte de miembros de la Unión Astronómica Internacional.

Universidad de Panamá y el Profesor y Director de Estudios Generales de la Universidad Marítima Internacional de Panamá.

- Sitio Web del proyecto. <https://snorri.space> (Weigandt, s.f).



4.3 Medios de comunicación masiva

Los medios se han manifestado acerca de SNORRI en forma clara y contundente:

- El Siglo: Este periódico de cobertura nacional publicó el 9 de septiembre del 2019, en la sección de PROVINCIAS, el artículo titulado “Astrofísicos panameños desarrollan un proyecto para estudiar las galaxias” donde destacan que se busca “recolectar datos y reforzar los conocimientos astronómicos en el país” (Saldaña, 2019).
- La Estrella de Panamá: En un artículo publicado el 9 de septiembre del 2019, este diario de circulación nacional publicó el artículo “Investigadores panameños trabajan para recolectar datos de las galaxias” donde puntualizan que “el equipo está diseñado

para detectar estructuras de bajo brillo superficial en el espacio exterior” (Ábrego, 2019).



Figura 13. El Siglo y La Estrella de Panamá fueron los primeros en informar al país lo que se está haciendo en Coclé en materia de investigación en Astrofísica.

- Telemetro: Este canal de televisión de difusión nacional, transmitió, el 8 de enero del 2020, en su programa Telemetro Reporta edición matutina y estelar, el segmento “Mentes Brillantes” con un reportaje en Churuquita Chiquita en donde se entrevista a los hermanos Weigandt y se presenta a SNORRI (Muñoz, 2020).



Figura 14. Participación en el segmento “Mentes Brillantes” del canal de televisión Telemetro en su edición matutina y estelar del 8 de enero del 2020.

- TVN: Esta televisora nacional presentó, el 21 de diciembre del 2020, en su programa matutino y en la edición estelar de TVN noticias, un reportaje con los investigadores y responsables de SNORRI, acerca del fenómeno denominado “estrella de belén” (Reyes, 2020).



Figura 15. SNORRI en la edición matutina y estelar de TVN noticias, el 21 de diciembre del 2020.

CAPÍTULO 2 - SNORRI: ESPECIFICACIONES TÉCNICAS

1. Construcción, armado y especificaciones técnicas del equipo

Primeramente se determinó el lugar propicio para la construcción de este sistema de observación astronómica que se ha dado a conocer como SNORRI (“Sistema Novedoso de Óptica Refractiva Integrada”). Luego de un análisis sobre la contaminación lumínica, la nubosidad, y seguridad del equipo, entre otros factores, se optó por instalarlo en la Comunidad de Churuquita Chiquita, en el Distrito de Penonomé, Provincia de Coclé, República de Panamá ($8^{\circ} 34' 12''$ latitud norte, $80^{\circ} 16' 48''$ longitud oeste).



Se procedió con el ensamblaje, armado de soporte, montaje de lentes y cámaras e instalación de software. Preliminarmente se armó el domo astronómico previa construcción de un piso y fundaciones adecuadas y se instaló la base para la montura robótica. Se hicieron todas las instalaciones de cableado con las computadoras, monitor y demás periféricos.

La matriz la forman cuatro lentes súper telefoto de marca Canon modelo EF 400 mm f/2.8L IS II USM (Canon Inc., 2011) siendo éstos extremadamente “rápidos” y contando con lo último en revestimiento antirreflectante (“super spectra coating”) lo que los hace ideales para el propósito del proyecto. Estos lentes se montaron en una estructura metálica de fabricación casera quedando alineados para observar una misma región en el firmamento.



Figura 17. Vista lateral de la matriz.

El domo observatorio es de la marca NexDome (NexDome, s.f) y las cámaras CCDs son modelo Aluma 8300, con un CCD KAF-8300 (ON Semiconductors, s.f.), de SBIG - Diffraction Limited de última generación (tercera) que salió recientemente al mercado (Diffraction Limited, 2019). Las cámaras se conectaron a una red de mini computadoras marca Gigabytes Brix, manejada a través de una terminal única.



Figura 18. Las computadoras, dentro del domo, que controlan cada una de las cámaras y el sistema de enfoque.



Figura 19. Vista trasera de las cuatro cámaras de la matriz.

La estructura metálica ya mencionada, soporta el corazón de SNORRI, que son los cuatro lentes súper telefoto con sus respectivas cámaras CCDs. Ésta es una única armazón de metal laminado de perfil “L” de 2,54 cm a la que se le incorporó posteriormente anillos de



Figura 20. Matriz completa: lentes, cámaras, montura con el contrapeso, y la base que no toca el piso.

ajuste (ADN, s.f), y a su vez está unida a la montura robótica Paramount MX+ que supera con creces la capacidad para manipular los aproximadamente 20kg de la Matriz (Software Bisque, 2015) y que al mismo tiempo está sujeta a la base fija en tierra. El sistema computacional está armado con el hardware y software más apropiado. Todo lo anterior está dentro del domo astronómico debidamente instalado.

Cada uno de los cuatro lentes se fijó en la distancia focal máxima de 400mm y cuentan con una apertura de 143mm lo que ofrece en conjunto una relación focal de f/1.4. El campo de visión es de $2,6^\circ \times 1,9^\circ$.

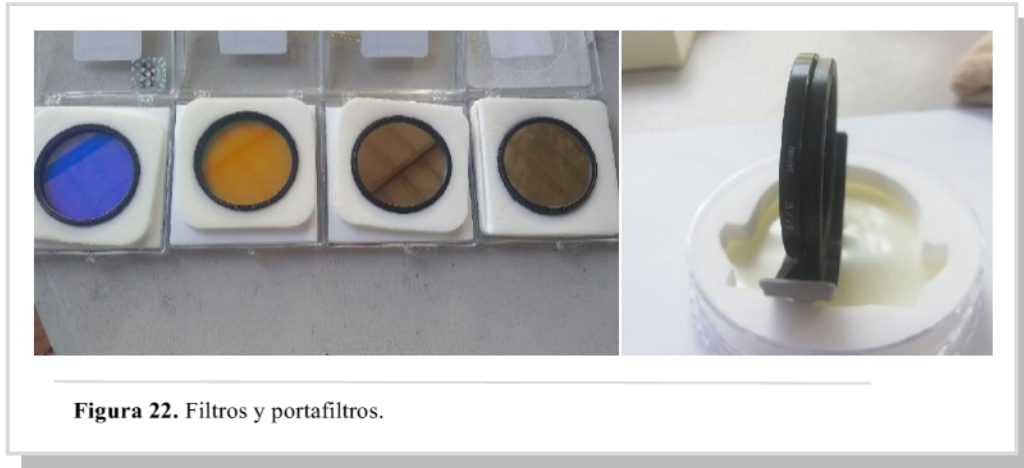
En el centro de la parte inferior del soporte metálico de la matriz de lentes se ubicada una pequeña cámara guía modelo Ultrastar ICX825AL de la marca Starlight Xpress acoplado a un tubo guía QHYCCD (Telescope.net, s.f.).



Figura 21. Imagen frontal. En la parte inferior está la cámara guía. Los lentes cuentan con un “parasol” para evitar la contaminación lumínica.

Entre cada lente súper telefoto y su cámara se colocó un controlador modelo Mark II (Astromechanics, s.f.) para ajustar periódicamente el foco del lente a través de la computadora en el comando central de operaciones.

Se cuenta además con un conjunto de filtros sloan de fotometría (Astrodon, s.f), de la segunda y más reciente generación, adaptados para ser usados con el portafiltros del lente súper telefoto CANON.



Las tablas a continuación resumen las principales características de los componentes más importantes de SNORRI:

Tabla 1. <i>Lente súper telefoto CANON EF400mm f/2,8L IS II USM</i>	
Longitud focal, relación focal	400mm, f/2,8
Construcción del objetivo	16 elementos en 12 grupos
Abertura mínima	f/32
Ángulo de visión	Diag.: 6° 10', Vert.: 3° 30', Hor.: 5° 10'
Distancia de enfoque mínima	2,7 m
Amplitud máxima	0,17×
Campo de visión	Aprox. 139 × 208 mm (a 2,7 m)
Filtro	Que se encastre en la serie 52(WII)
Diámetro de longitud máx.	163 × 343 mm
Peso	Aproximadamente 3850 g
Parasol	ET-155(WII)
Tapa de objetivo	E-180D
Estuche	Estuche de objetivo 400C

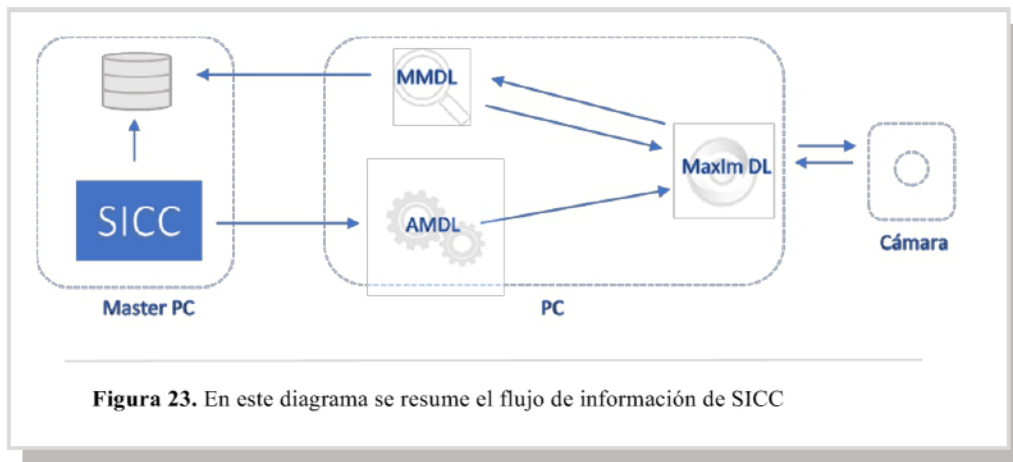
Tabla 2. Cámara Aluma 8300	
Sensor de imagen	KAF-8300
Eficiencia cuántica máxima	56%
Tamaño del sensor	17,96 mm x 13,52 mm
matriz de pixeles de las imágenes	3326 x 2504 pixeles
Tamaño del pixel	5,4 μ m (cuadrado)
Tasa de digitalización de los pixeles	10 MPix/sec
Número total de pixeles	8.3 millones de pixeles
Carga máxima del pixel ante de la saturación	25,000 e-
Ruido de lectura	10 e- típico
Corriente de oscuridad e/p/s a 0° C	0.15 e-/p/s típica
Antideslumbramiento	Sí
Convertidor A/D	16-bit
Tiempo mínimo de exposición	0.1 sec
Tiempo de descarga del fotograma completo	~1 sec vía USB / ~7 sec vía WiFi
Alimentación	12 VDC, 5A max.
Obturador	Mecánico con iluminación uniforme
Regulación de temperatura	Sí
Enfriamiento	De -40° a -45° C
Dimensiones	4.5" x 4.5" x 4" (con manijas)
Peso	2.2 lbs
Interface para la PC	USB2 y WiFi 802.11 b/g/n
Sistemas Operativos compatibles	Windows de 32 y 64 bit, Macintosh, Linux, iOS, Android
Tabla 3. Montura Paramount MX+	
Pecisión al apuntar (All-sky)	30 arcosegundos o menos
Holgura o "backlash"	Despreciable
Desempeño del seguimiento y error periódico.	Siete arcosegundos o menos de pico a pico en el error periódico antes de la corrección.
Diseño	GEM (Montura ecuatorial alemana)
Armazón y engranajes	Aluminio 6061 o 6063
Tornillo sin fin	Latón
Contrapeso	Acero inoxidable
Electrónica del sistema de control	Sistema de control de movimiento de doble eje MKS 5000
Motores	Servomotores sin escobillas
Alineación polar	Escalas Vernier calibradas para la altitud y el acimut
Tipo de montaje para el OTA	Placa de montaje "Versa-Plate"
Capacidad de soporte de peso de la montura	Capacidad total para los instrumentos: 100 lb (45 kg). Capacidad total incluyendo el contra-peso: 200 lb (90 kg)
Mount Weight Peso de la montura	50 lb (23 kg)
Peso del contrapeso	Dos pesas 20 lb. (9 kg) más el tubo de 16" (41 cm) y una extensión extra que se necesitó usar de 7" (17,8 cm)
Rango de inclinación ecuatorial para el ajuste de elevación del eje polar	10° a 65° (Para ajustarlo a 8° se tuvo que conseguir una ligera inclinación inicial)
Seguimiento pasado el meridiano	Dos horas
Gears Engranajes	Engranajes de RA y declinación de 7,5" (19 cm) con dientes de 375 dientes de latón
Balineras	Balineras de bolas de contacto de 6" (15 cm)

2. Comando central de operaciones

En cuanto a la terminal de manejo, se desarrolló una interface denominada “SNORRI Interface Control Center” (SICC) con el fin de tener el control del sistema desde un solo lugar.

SICC es el programa que centraliza las operaciones de SNORRI que se dan a través de la red de computadoras y cámaras que constituyen el sistema informático. Éste consta actualmente de cuatro cámaras y cuatro computadoras conectadas en red LAN con un Switch y que se comunican con SICC a través de TCP. Cada computadora tiene instalado un sistema operativo Windows 7 Professional y Maxim DL 6.x (Diffraction Limited, 2019).

SICC es un conjunto de tres programas escritos en C#: SICC, MMDL (Monitor de Maxim DL) y AMDL (Acciones para Maxim DL). El primero está instalado en la llamada “computadora maestra” o “Master PC” que maneja el operador, y los otros dos, en cada una de las computadoras.

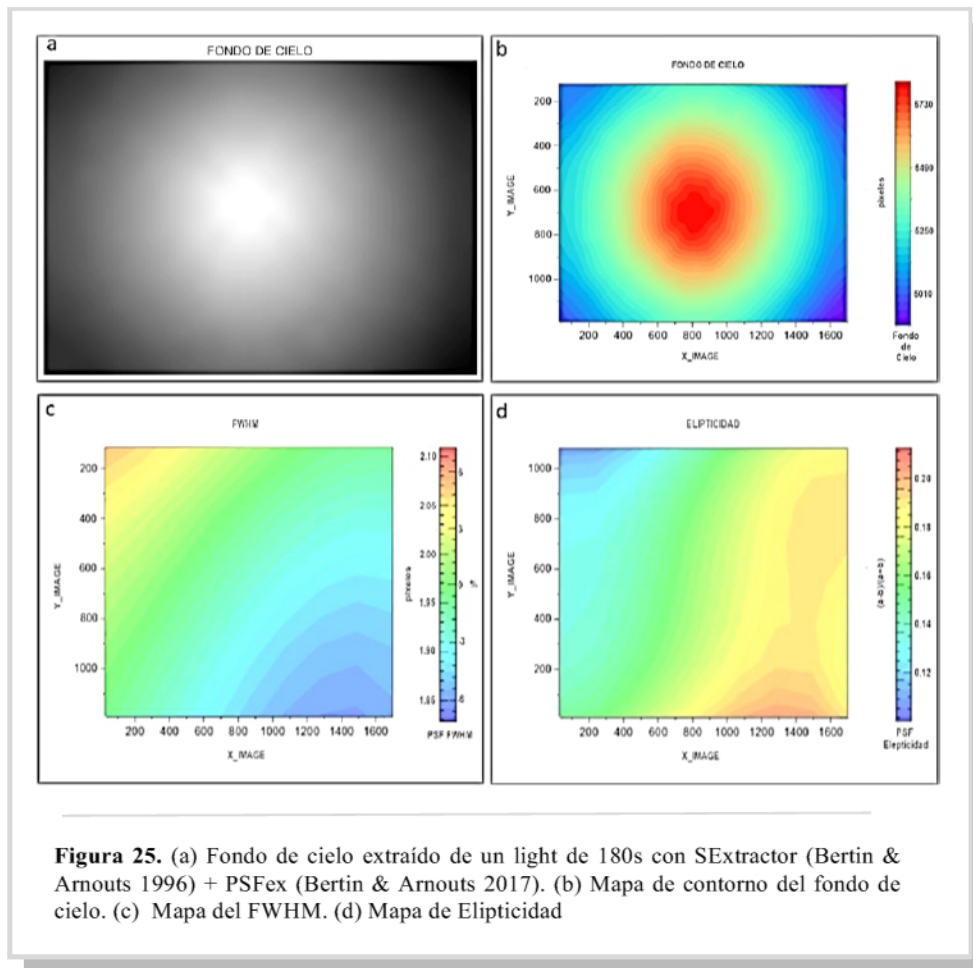


El SICC en la Master PC envía órdenes de acciones a AMDL y ésta las ejecuta en Maxim DL que es el programa que se comunica directamente con la cámara. Por otro lado, MMDL monitorea los eventos de Maxim DL y el estatus de la cámara. Esta información es enviada

Se presentan los resultados correspondientes a las pruebas de linealidad (“linearity”), ganancia (“gain”) y ruido de lectura (“read-out noise”) de una de las cámaras a temperatura de -20°C .

En la inspección de calidad de la homogeneidad de los “masterbias” se pudo observar, usando Iraf (Tody, 1986), que el histograma se aproxima bastante a una gaussiana tendiendo a lo ideal. En la transformada de Fourier de la imagen o imagen FFT, se observa que existen patrones no aleatorios, aunque pequeños, que se perciben por la crucecita en el medio. Para las otras cámaras los resultados son similares.

Se realizó un análisis de la calidad de la imagen basada en un “Light” sin calibrar con una exposición de 180 segundos. Los resultados se resumen en las siguientes imágenes.



Como se puede ver en la imagen fondo de cielo (figura 25a), el patrón de iluminación está bien centrado en el chip, lo que indica una buena colimación general de los componentes ópticos en el lente. En el mapa de contorno de la derecha se puede observar que la iluminación cae aproximadamente un 15% desde el centro hasta el borde (figura 25b).

Hay un pequeño gradiente en los valores de la función de anchura a media altura (FWHM) (figura 25c). Las variaciones en FWHM son de aproximadamente 10%, a ese nivel se ven comúnmente, y generalmente encuentran su origen en la precisión mecánica de la interfaz entre los lentes y las cámaras, así como una combinación de aberraciones intrínsecas en el diseño óptico de la lente e inclinación leve en el plano focal. El promedio de FWHM está por debajo de los 2 píxeles a lo largo de casi toda la imagen con una degradación máxima de ~ 0.3 píxeles. La elipticidad está por debajo del 18% a lo largo de casi toda la imagen (figura 25d).

La tabla 4 resume la estadística de los datos obtenidos de calidad de imagen.

Tabla 4. Calidad de imagen

	Objetos Cargados	Objetos Aceptados	Half-Light diameter (pix)	FWHM (pix)	Elipticidad	Residuos	Asimetría	Área Equivalente de ruido
Promedio	1106	1046	2.01	1.91	0.15	0.02	0.06	0.04
Rango	1106 - 1106	1046 - 1046	2.01 - 2.01	1.86 - 2.07	0.11 - 0.20	0.02 - 0.04	0.04 - 0.09	0.03 - 0.07

Resultados comparables se han conseguido con equipo óptico similar (Abraham & Van Dokkum, 2014), donde se obtuvo una variación en el decaimiento de la iluminación en la imagen fondo de cielo mayor, una degradación máxima en la FWHM levemente mayor, y una elipticidad menor. Los valores promedios menores en FWHM son de esperar, puesto que la calidad del cielo en ese lugar es mucho mejor que en zonas tropicales. La diferencia en los valores de elipticidad pueden ser atribuidos a desajustes en los sistemas de autoguía.

CAPÍTULO 3 - LOS DATOS: ADQUISICIÓN, CANALIZACIÓN Y CARACTERÍSTICAS

1. Ajuste y configuración del equipo

Con el fin de lograr los más altos niveles de fiabilidad, productividad y rendimiento se consideró adquirir el mejor equipo y software posibles que el mercado pueda ofrecer.

Antes de realizar las mediciones como tales, se hizo uso de procedimientos sistemáticos con los cuales se puede cuantificar, identificar y aislar los problemas, de tal suerte que se pueda afinar los sistemas de seguimiento, apuntado y autoenfoco, entre otros. Estas tareas se repiten dependiendo de la manipulación que se haga en el equipo debido a cambios de configuración físico del sistema o mantenimiento de éste.

Primero se analiza la configuración física del *sistema de imágenes*. Se examina todos los componentes mecánicos y la interfaz de montaje, esto incluye la alineación polar preliminar y el equilibrado de la montura para reducir errores en RA.

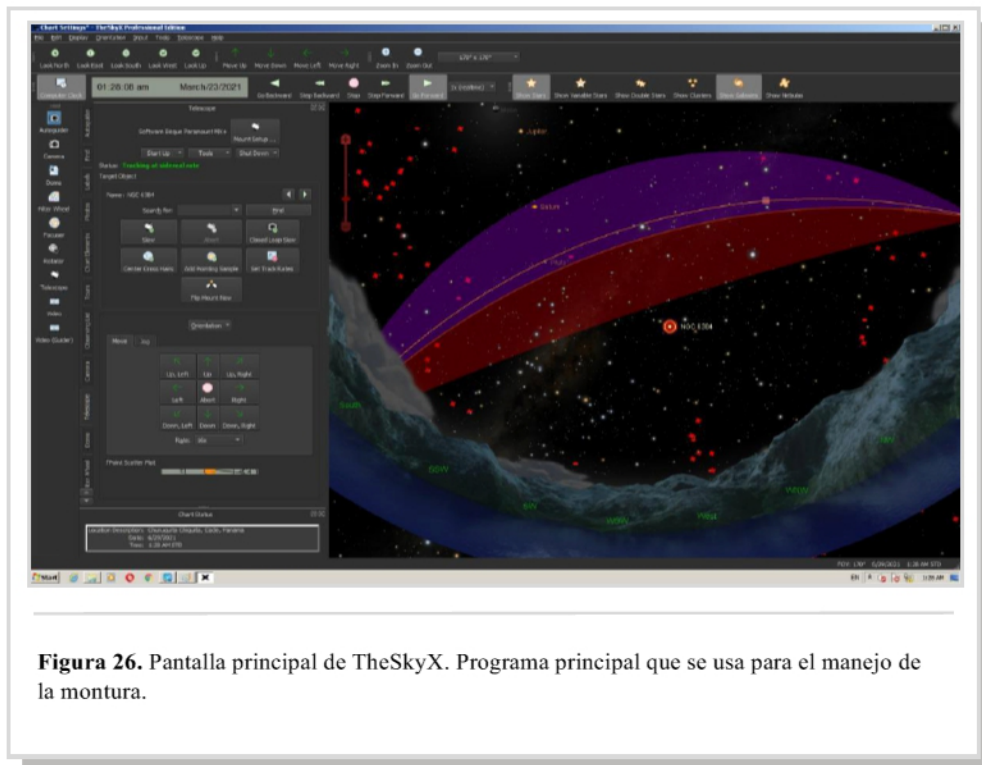


Figura 26. Pantalla principal de TheSkyX. Programa principal que se usa para el manejo de la montura.

TheSkyX Pro (Software Bisque, s.f.) se está usando como software principal para la calibración y el manejo de la montura (Figura 26). Se hizo y aplicó un modelo Tpoint de ~300 puntos incluyendo ambos meridianos (Figura 27a). También se generó y cargó la curva de corrección de errores periódicos (PEC) más óptima para la montura (Figura 27b). Con el modelo Tpoint se hizo un ajuste fino de la alineación polar.

Para la autoguía se está usando tanto TSX como PHD2 (Stark Labs, 2021). Ambos resultan ser convenientes pero PHD2 parece ser más estable al retomar su funcionamiento luego de presentarse episodios de nubosidad.

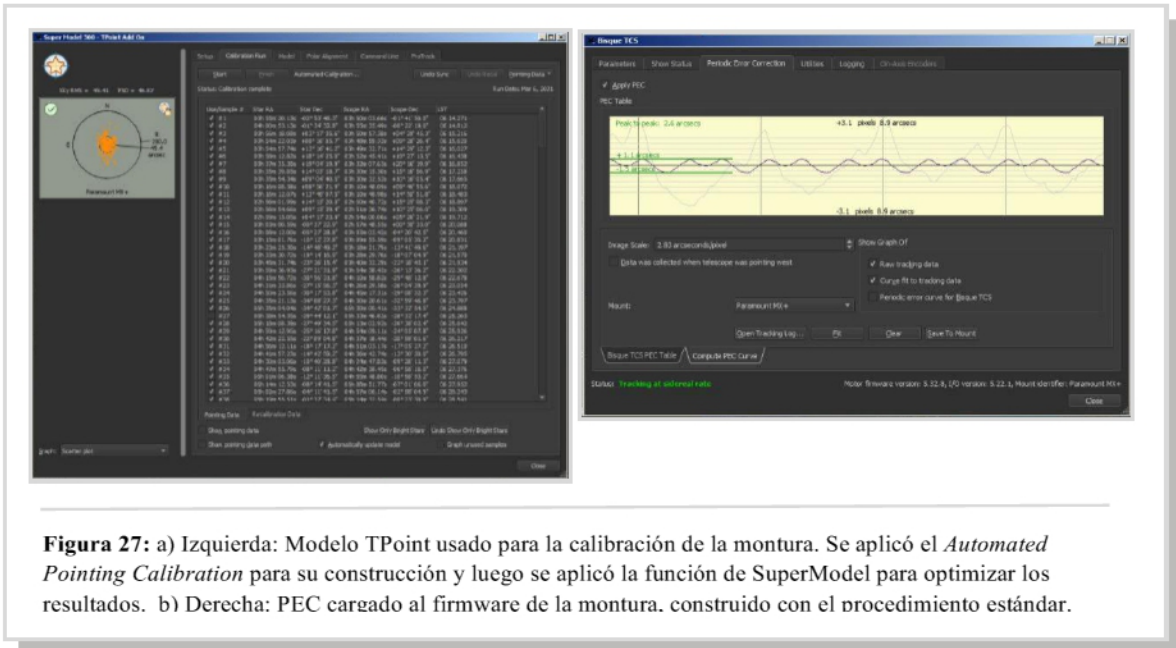


Figura 27: a) Izquierda: Modelo TPoint usado para la calibración de la montura. Se aplicó el *Automated Pointing Calibration* para su construcción y luego se aplicó la función de SuperModel para optimizar los resultados. b) Derecha: PEC cargado al firmware de la montura. construido con el procedimiento estándar.

Para manejar el equipo motorizado de enfoque y mantener un enfoque óptimo durante toda las sesiones de medición, se está utilizando el software FocusMax (CCDWare, s.f.). Es fácil de integrar en SICC debido a que se pueden incorporar scripts de automatización luego que un intervalo de exposiciones (Slot) de MaximDL es completada. Para cada lente se caracterizó el sistema generando las *V-curves* y mediciones de HFD suficientes para que el programa alcance el mejor enfoque dentro de la zona de enfoque crítico de los lentes. Existen procedimientos al respecto como los descritos por Morten la Cour (2018).

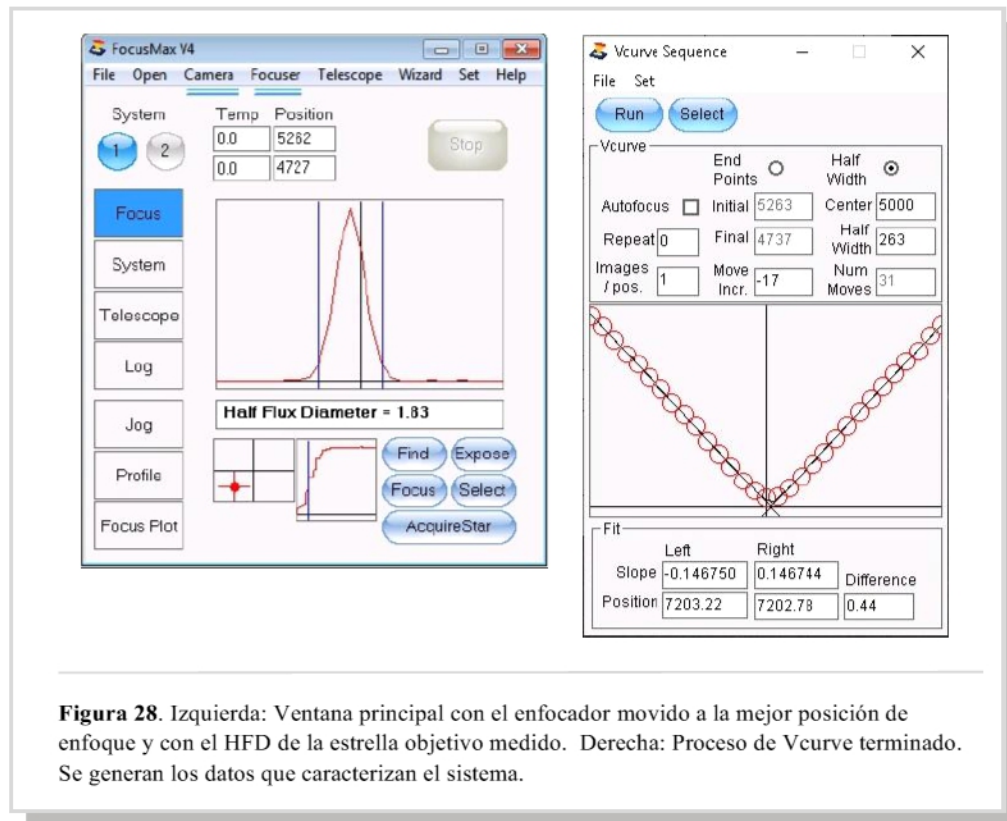


Figura 28. Izquierda: Ventana principal con el enfocador movido a la mejor posición de enfoque y con el HFD de la estrella objetivo medido. Derecha: Proceso de Vcurve terminado. Se generan los datos que caracterizan el sistema.

2. Adquisición y reducción de la data

2.1 Adquisición

En el verano del 2021, entre los meses de enero y abril, comenzaron las observaciones astronómicas buscando galaxias como objetivos principales de cada campo de $2.6^\circ \times 1.9^\circ$ que se ajusten mejor a nuestro entorno de observación y que sean visibles en nuestra posición geográfica, de esta manera, poner en marcha la tarea de demostrar la sensibilidad y la funcionalidad de nuestro equipo.

Debido a situaciones climatológicas adversas, se trabajó únicamente en 2 campos con tiempos acumulados de ~ 9 h netas aproximadamente, y otros 3 campos de menos tiempo

con fines relacionados a la adquisición de material para la divulgación. Para la adquisición de las imágenes se usaron filtros Sloan Astrodon u, r, i.

2.2 Proceso general de reducción

La reducción de la data se hizo siguiendo procedimientos estándares para la corrección de bias y flat-fielding. Cada noche de observación se tomaron ~100 bias y ~30 darks con tiempo de exposición y temperatura de la cámara igual al de las lightframes. También se tomaron una serie de ~80 skyflats tanto al anochecer como al amanecer apuntando dentro del círculo solar cerca del cenit, desplazada en la dirección contraria al sol en 15 grados con el fin de disminuir los gradientes (Chromey & Hasselbacher,1996). Este proceso se hizo automatizado con la ayuda de un software para compensar el cambio de la luminosidad

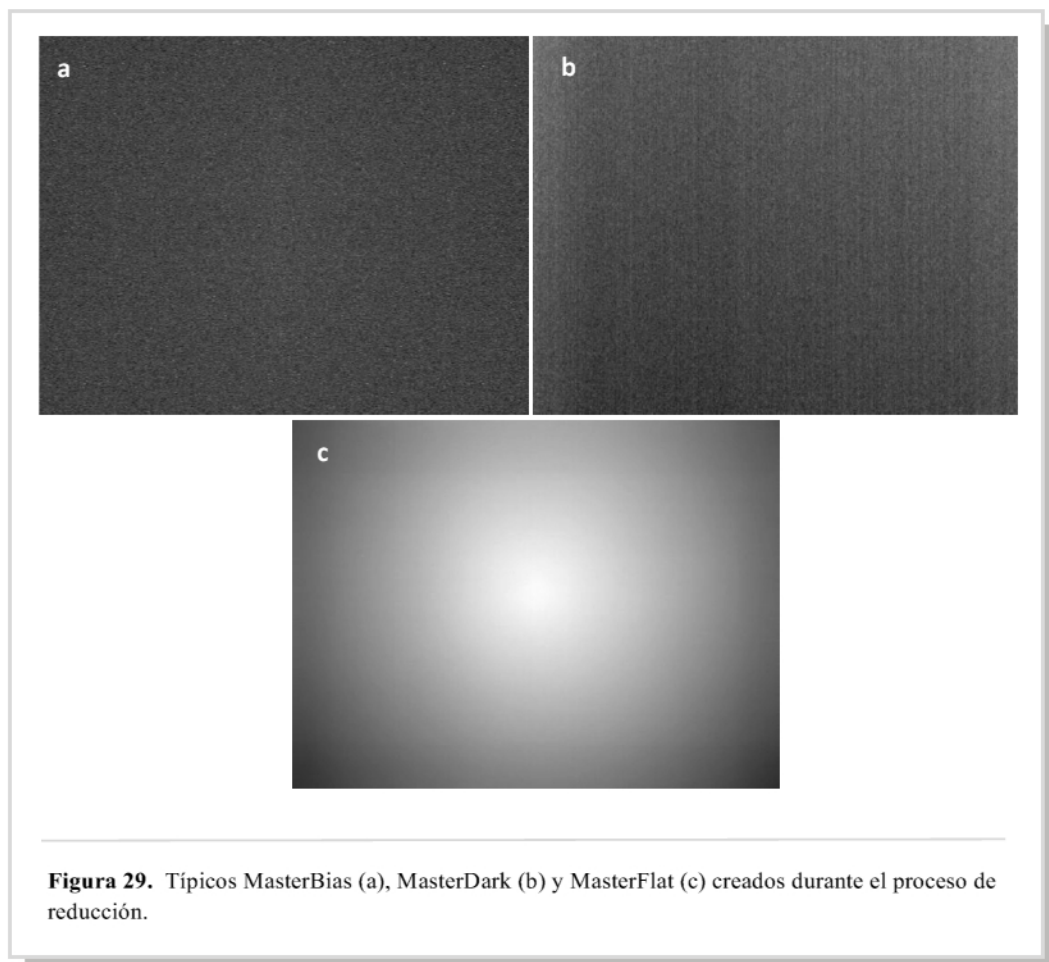


Figura 29. Típicos MasterBias (a), MasterDark (b) y MasterFlat (c) creados durante el proceso de reducción.

del cielo, sobre todo en esta latitud, y tomando como límite de ADU ~33% de la saturación de la cámara. Se seleccionó ~65% de los flats, con tiempos de exposición entre 1 y 10 segundos. Para cada sesión, se realizó el proceso de calibración con los respectivos MasterBias, MasterDarks, MasterFlats y Lights, luego el conjunto de imágenes de todas las sesiones fueron combinadas para lograr una única imagen representativa del tiempo total de exposición. Cabe destacar que alrededor del 45% de las exposiciones son descartadas por diversos motivos como desenfoco, abertura el domo no alineada, contaminación lumínica, viento, árboles, inconvenientes con el seguimiento, etc. pero el más recurrente es la nubosidad.

2.3 Selección inicial de las imágenes

El proceso de selección luego de la calibración de las imágenes fue el siguiente:

- Medición de características tales como: cantidad de objetos detectables para cada imagen, FWHM promedio y elipticidad promedio.
- De ser necesario, guardar los valores en el encabezado de cada archivo para su posterior uso.
- Determinar las condiciones de aprobación o rechazo.
- Guardar las imágenes aprobadas y rechazadas en carpetas separadas.
- Realizar una inspección visual de las imágenes aprobadas y rechazadas para validar el proceso y especialmente para rechazar aquellas que, a pesar de todo lo anterior, fueron aprobadas teniendo defectos o sin cumplir con las expectativas.

En general, se encontró que alrededor del 2% de las imágenes defectuosas pasaron por el filtro de aprobación inicialmente. En la mayoría de los casos se debió a la existencia de nubes altas y bien delgadas, puesto que en ocasiones son interpretadas como señal. Prácticamente ninguna de las imágenes rechazadas tenía característica de aprobación antes de la inspección visual.

Para la reducción de imágenes con fines científicos se escogieron los softwares especializados como IRAF (Tody, 1993) y SExtractor (Bertin & Arnouts, 1996), usando como programa de visualización principal a SAOImage DS9 (Joye & Mandel, 2003). Básicamente con paquetes de Python tales como Astropy, Pyraf y PyQT, se desarrollaron algunos programas sencillos para la automatización de procesos en lo concerniente a la calibración, registro y apilado, como también para el posterior análisis de los resultados.

Por otro lado, para la reducción de imágenes con fines de divulgación se utilizó Pixinsight (Pleiades Astrophoto, 2021), en donde las prácticas arbitrarias de manipulación no tienen cabida. Es importante mantener la fiabilidad de la información y comunicarla con responsabilidad a través de las imágenes resultantes.

2.4 Solución astrométrica de las imágenes como modo de filtrado

Este procedimiento ayuda a filtrar exposiciones que no corresponden al campo en cuestión o con algunas características que hacen que no se pueda encontrar la solución WCS. Son muy pocas las exposiciones que no pasan esta prueba pero vale la pena hacerlo. En la mayoría de los casos el motivo es el efecto de “doble estrella”. El escenario de observación donde las exposiciones científicas muestran este fenómeno es cuando el telescopio está observando cerca del meridiano. La causa más probable es el leve movimiento de un componente que permite la función de estabilización de imagen del lente (Danieli et al., 2020).

Para la prueba de solución astrométrica se codificó un script denominado `astap.py`. Sin uso de servidores externos, realiza la solución astrométrica de imágenes y mueve las exposiciones que no tienen solución a una carpeta para su posterior verificación visual. Hace uso de la versión local de línea de comando del software ASTAP (Han, k, s.f.), con la base de datos de estrellas H17.

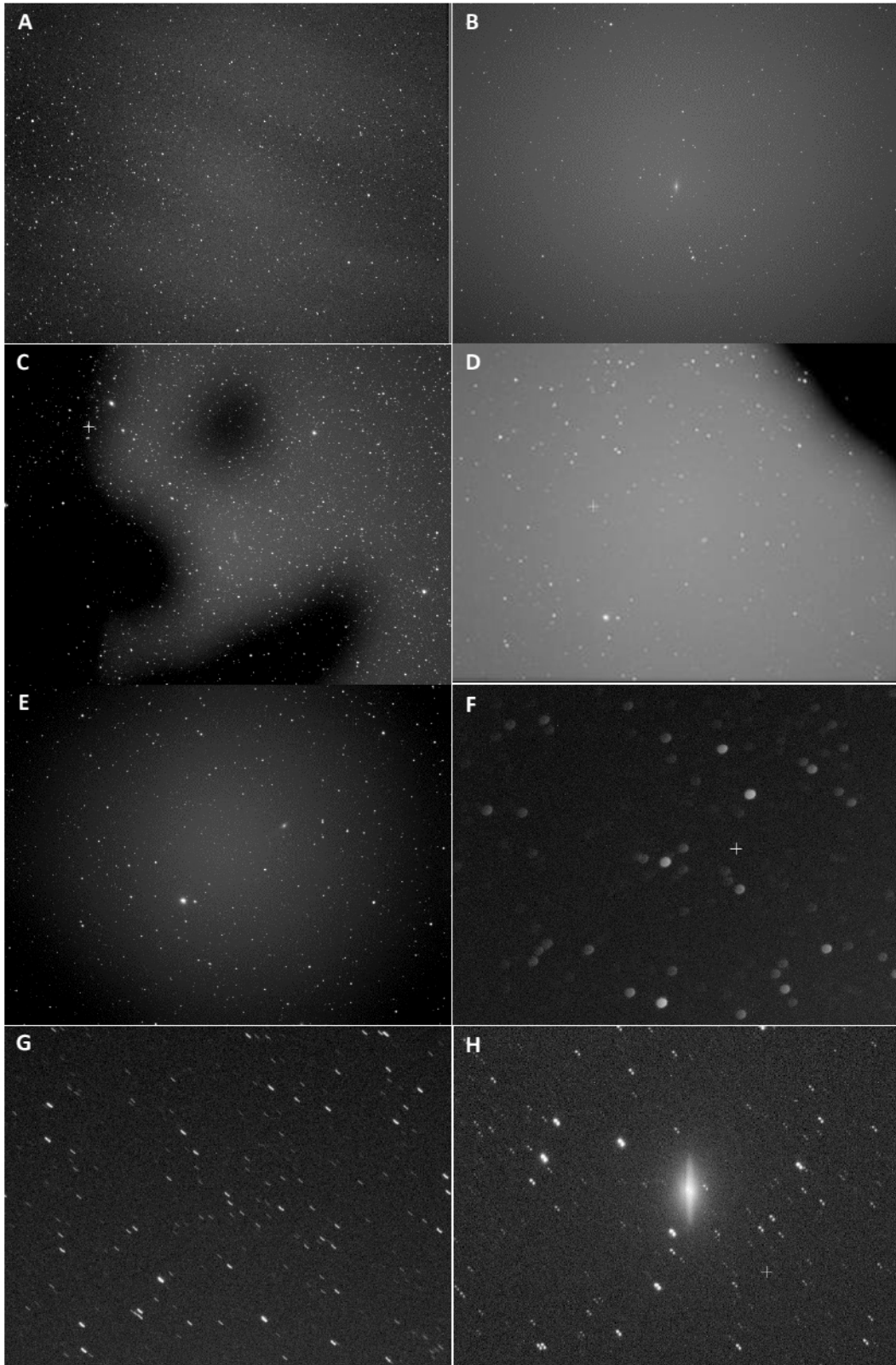


Figura 30. Ejemplo de exposiciones con problemas. **A.** Presencia de nubes finas. **B.** Paso de múltiples nubes durante la exposición. **C.** Exposición bloqueada parcialmente por la presencia de árboles. **D.** Exposición bloqueada parcialmente por la cúpula del domo. **E.** Presencia de contaminación lumínica debido al inicio del amanecer. **F.** Lente fuera de foco. **G.** Exposición con problemas de seguimiento. **H.** Exposición con efecto de “doble estrella” (ver sección 2.4).

2.5 Sustracción del fondo de cielo

Luego del proceso de selección y descarte de las imágenes se sustrae el fondo de cielo de cada una de ellas. Este procedimiento está inspirado en parte por los métodos de sustracción de fondo de cielo en la canalización de datos de NGVS (CADC, s.f.) y Astro-WISE (OmegaCEN, 2020), entre otros.

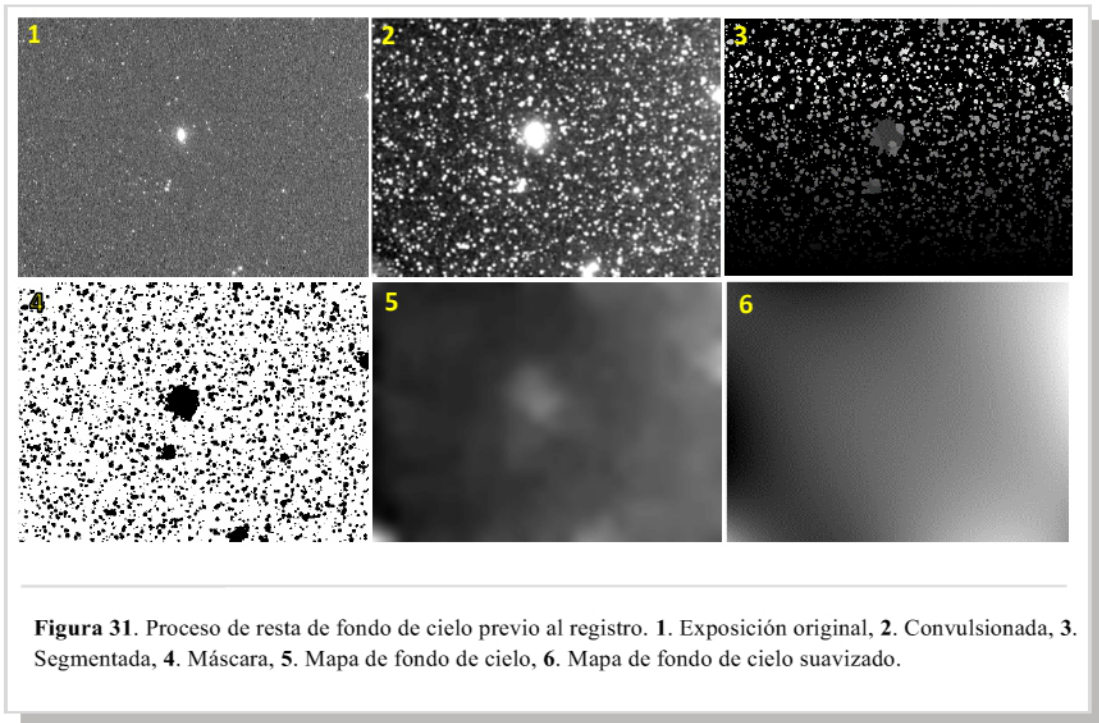
Para ello se desarrolló un script llamado `bkgr_subtract.py` que realiza tarea siguiendo estos pasos:

- Convulsionar la imagen con la tarea *gauss* de IRAF usando un sigma de 7. El suavizado va a ser que la segmentación realizada por SExtractor en el paso siguiente incluya los halos alrededor de las estrellas brillantes y luz residual débil de las fuentes en general.
- Crear una imagen segmentada (`img_seg`) de la imagen del paso anterior con SExtractor usando un `BACK_SIZE` de 128.
- Crear una mascara haciendo $mask = 1 - img_seg / img_seg$ para que dicha máscara tenga un valor de 0 en las fuentes detectadas y 1 donde no.
- Crear el mapa de fondo de cielo (`background`) corriendo SExtractor con: `WEIGHT_TYPE: MAP_WEIGHT`, `WEIGHT_IMAGE: mask` y `BACK_SIZE: 128`.
- Suavizar el mapa de `background` ajustándolo con un polinomio de cuarto grado usando la tarea *imsurfit* de IRAF.
- Restar el mapa suavizado con la imagen en cuestión usando la tares *imarith* de IRAF.

2.6 Registro de las imágenes

Todas las exposiciones que han pasado por las etapas previas, deben ser registradas a una cuadrícula común. Para ello se utilizó un conjunto de programas de Astromatic.net: SExtractor, SCAMP (Bertin, 2006) y SWARP (Bertin et al., 2002).

Con SExtractor se extraen las fuentes y sus posiciones en las imágenes. SCAMP hace uso de esta información y calcula una solución astrométrica comparando estas posiciones de



origen con las de un catálogo de referencia. Esta solución se almacena en un archivo de texto con una extensión de archivo .head en formato WCS estándar. Por su parte, SWARP toma los archivos .head y remuestrea las imágenes de entrada a una cuadrícula común.

Para realizar esta tarea se hizo un script de python llamado registration.py. Se generan todos los catálogos con SExtractor en formato FITS_LDAC y se guarda en una carpeta. Los parámetros de los catálogos, son los recomendados por la documentación de SCAMP a saber:

XWIN_IMAGE, YWIN_IMAGE, ERRAWIN_IMAGE, ERRBWIN_IMAGE,
 ERRTHETAWIN_IMAGE, FLUX_AUTO, FLUXERR_AUTO, FLAGS,
 FLAGS_WEIGHT, FLUX_RADIUS y ELONGATION.

Se hace correr SCAMP con uno de los catálogos usando el parámetro ASTREF_CATALOG: SDSS-R9 (Ahn et al., 2012) para crear el correspondiente catálogo de referencia local para este campo. Esto se hace para que el proceso siguiente no demore tanto. Luego se hace correr SCAMP nuevamente con todos los catálogos creados por SExtractor como datos de entrada para generar los correspondientes archivos .head con la

solución astrométrica y usando como catálogo de referencia, el SDSS-R9 local creado anteriormente.

Por último, se usa SWARP para crear las imágenes registradas con un archivo de configuración `snorri.swarp` cuyos parámetros a destacar son:

Tabla 5. <i>Parámetros destacados de <code>snorri.swarp</code> para registrar</i>		
RESAMPLE	Y	# Remuestras las imágenes de entrada (Y/N)?
COMBINE	N	# Combinar las imágenes remuestreadas (Y/N)?
RESAMPLING_TYPE	LANCZOS3	# Método de remuestreo de imágenes
SUBTRACT_BACK	N	# Restar el fondo de cielo (Y/N)?
DELETE_TMPFILES	N	# Eliminar los archivos FITS remuestreados (Y/N)?
PIXELSCALE_TYPE	MANUAL	# Cómo ajustar el tamaño del pixel .
PIXEL_SCALE	2.832	# Tamaño del pixel en arcosegundos.
IMAGE_SIZE	0	# Tamaño de la imagen. 0 = automático.

Luego de terminado el proceso se pueden eliminar los mapas de peso, en este paso no se van a utilizar.

3. Apilado

3.1 Calibración fotométrica

Antes de la combinación o apilado de las imágenes, éstas deben ser calibradas a un mismo nivel de flujo, para ello usamos el zeropoint de cada imagen. El zeropoint de cada imagen registrada es calculado comparando las mediciones de la fotometría de las fuentes puntuales de ésta con un catálogo de referencia como lo son GSC-2.3, APASS, SDSS-DR12, USNO-A2.0 etc.

A continuación se resume el procedimiento que se está usando actualmente:

Para varios de los pasos se codificaron scripts en python, los nombre aparece entre paréntesis.

Verificación de los encabezados:

Existen tareas que exigen cierta información de los encabezados de las imágenes que a veces no está presente. Por lo tanto, hay que agregarla ya sea en el momento de cada tarea o al inicio de todas ellas. En particular, SWARP elimina las claves OBS-DATE (fecha y hora de la observación) y EXPTIME (duración de la exposición) que son necesarias para los procesos posteriores, por lo tanto, se agregan dichas claves y sus valores correspondientes para todas las imágenes usando el script `insert_missingkeys.py`.

Masa de Aire (`calculate_airmass.py`):

Primero se calculó la masa de aire para cada imagen y se guardó dicha información en el encabezado respectivo bajo el nombre clave AIRMASS. Para ello se usó la tarea *setairmass* de IRAF.

Generación de catálogos (`create_zcat.py`):

Para cada imagen se generan catálogos con SExtractor de las estrellas no saturadas con parámetros tales como coordenadas estelares, magnitud instrumental, error de magnitud instrumental, índice de estelaridad y bandera de calidad (ALPHA_SKY, DELTA_SKY, MAG_AUTO, MAGERR_AUTO, CLASS_STAR, FLAGS).

Coeficiente de Extinción (vstack_zcat.py):

Se unieron todos los catálogos creados en el paso anterior y se adjuntaron las columnas con la información del airmass y la fecha de exposición. Esta gran tabla se guardó en una hoja de cálculos y se procedió a calcular el coeficiente de extinción para cada sesión de observación.

Por último, se agregó el valor de este coeficiente en el encabezado de las imágenes bajo en nombre clave de EXTINCT.

Cálculo del zeropoint (calculate_zeropoint.py):

Para cada imagen se procedió a realizar una relación cruzada entre su catálogo y la información suministrada por el catálogo de referencia SDSS-R12 (Alam et al., 2015) en VizieR. (xmatch.py). El resultado fue filtrado por parámetros tales como magnitud de la estrella, índice de estelaridad (CLASS_STAR), error de magnitud instrumental (MAGERR_AUTO) y bandera de calidad (FLAG) para evitar fuentes con fotometría potencialmente dañada.

La información de este nuevo catálogo ampliado y la información del encabezado de la imagen de AIRMASS y EXTINCT, se calculó el zeropoint para cada fuente usando relación: $ZP = m_std - (m_int - kX)$, donde m_std es la magnitud de la estrella estándar en el catálogo de referencia, m_int es la magnitud instrumental medida por SExtractor de la misma estrella, k es el coeficiente de extinción la noche de la observación y X es la masa de aire de la imagen.

El zeropoint de la imagen va a ser el valor de la mediana de los valores de zeropoint calculados para cada fuente luego de aplicar un esquema de rechazo iterativo de 2 sigmas.

Esta información se guardó en el encabezado de las imágenes bajo en nombre clave de MAGZPT.

Nivel de Flujo:

Este paso forma parte del mismo script que el anterior. De todos los zeropoint de cada imagen se calcula la media luego de aplicar nuevamente un esquema de rechazo iterativo de 2 sigmas. Toda imagen con zeropoint rechazado va a ser descartada.

Con estos datos, se calcula el nivel de flujo para cada imagen usando la relación:

$$10^{((ZP - ZP_average) / (-2.5))},$$

donde ZP es el valor del zeropoint de cada imagen y ZP_average es la media de todas los ZP.

El valor calculado se guarda en el encabezado de las imágenes bajo el nombre clave: FLXSCALE .

Ponderación:

Previa a la combinación, las imágenes van a ser ponderadas. Para calcular cada valor se corre nuevamente SWARP pero esta vez con las imágenes registradas. Al ejecutar SWARP se van a crear nuevas imágenes registradas y los correspondientes mapas de peso. Cada mapa va a tener un valor constante, éste va a ser guardado en el encabezado del archivo correspondiente de la imagen registrada bajo el nombre clave: WEIGHT. Para ello se utiliza el script insert_weight.py. Este script también incorpora el valor de MAGZPT para futuras referencias.

Este proceso es necesario si se van a apilar las imágenes usando *imcombine* de IRAF. Es importante destacar que los valores correspondientes de FLXSCALE van a ser heredados siempre y cuando se mantenga el valor predeterminado de FSCALE_KEYWORD en el archivo de configuración de SWARP.

3.2 Combinación

Con las exposiciones científicas debidamente calibradas fotométricamente y registradas, se procede a la combinación o apilado de éstas. Esto se está haciendo de dos maneras:

1. Utilizando la tarea *imcombine* de IRAF con la siguiente configuración: (los parámetros en **negrita** son los más importantes, los demás son valores predeterminados)

Tabla 6. Parametros destacados en la configuración de imcombine		
(combine = average)		Type of combine operation
(reject = avsigclip)		Type of rejection
(offsets = wsc)		Input image offsets
(scale = !FLXSCALE)		Image scaling
(weight = !WEIGHT)		Image weights
(lsigma = 3.)		Lower sigma clipping factor
(hsigma = 3.)		Higher sigma clipping factor

2. Usando SWarp, con los parámetros:

Tabla 7. Parametros destacados de snorri.swarp para combinar		
RESAMPLE	N	# Remuestras las imágenes de entrada (Y/N)?
COMBINE	Y	# Combinar las imágenes remuestreadas (Y/N)?
COMBINE_TYPE	AVERAGE	# Tipo de combinación
FSCALE_KEYWORD	FLXSCALE	# Nombre clave en el archivo FITS para el factor # multiplicativo aplicada a cada imagen

Ambos métodos producen buenos resultados y son lo suficientemente precisos para los trabajos de investigación que se están desarrollando hasta este momento. Para realizar un apilamiento promedio con rechazo de pixeles y así lidiar con artificios tales como rayos cósmicos, marcas por satélites y asteroides e imperfecciones en el CCD se están usando los algoritmos de rechazo de IRAF como *avsigclip* y *crreject*. Para tener otras opciones en este sentido, se está pensando implementar en el futuro el nuevo tipo de combinación *CLIPPED* de SWarp y el procedimiento descrito en Gruen et al (2014).

3.3 Resumen del proceso

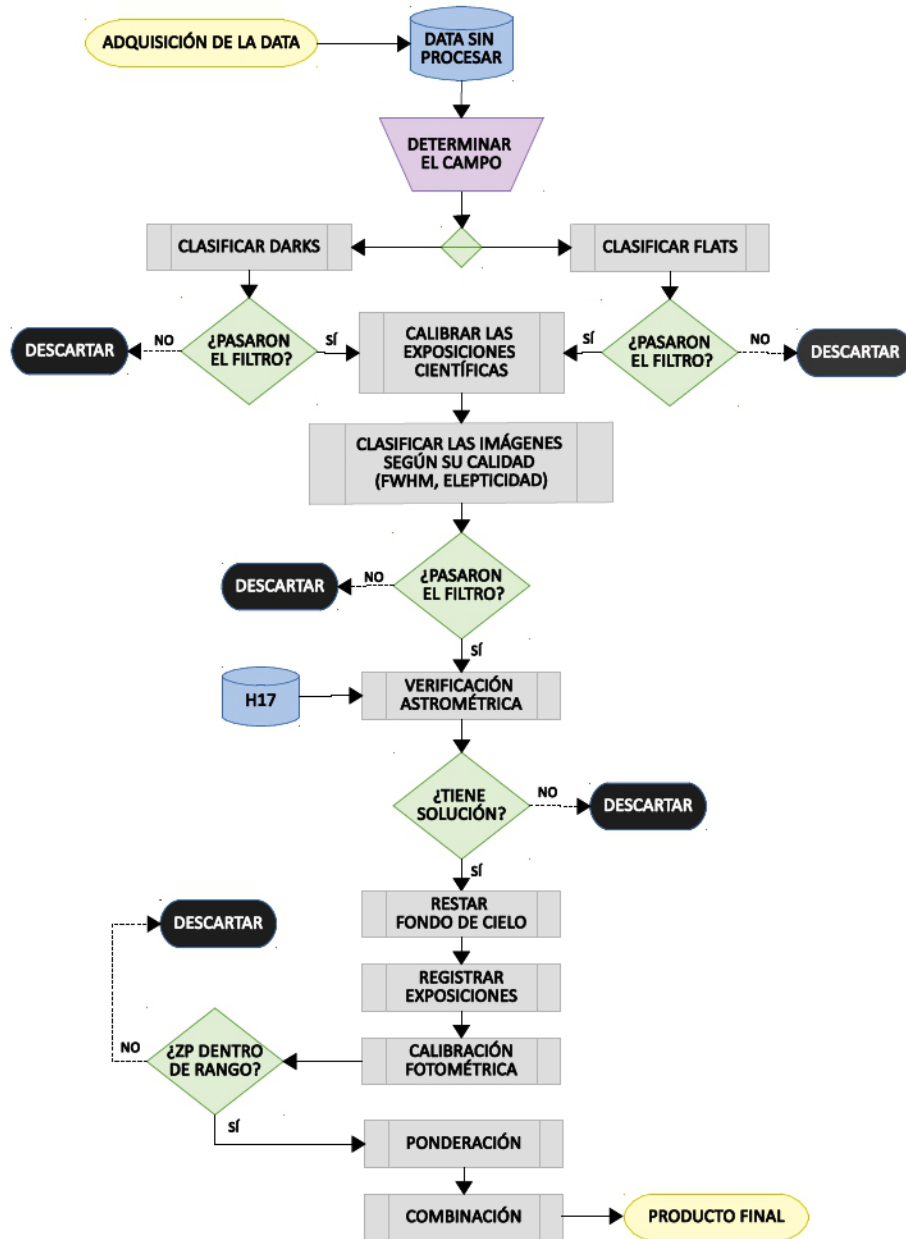


Figura 32. Diagrama de flujo del proceso general que se realiza para el tratamiento de las exposiciones científicas.

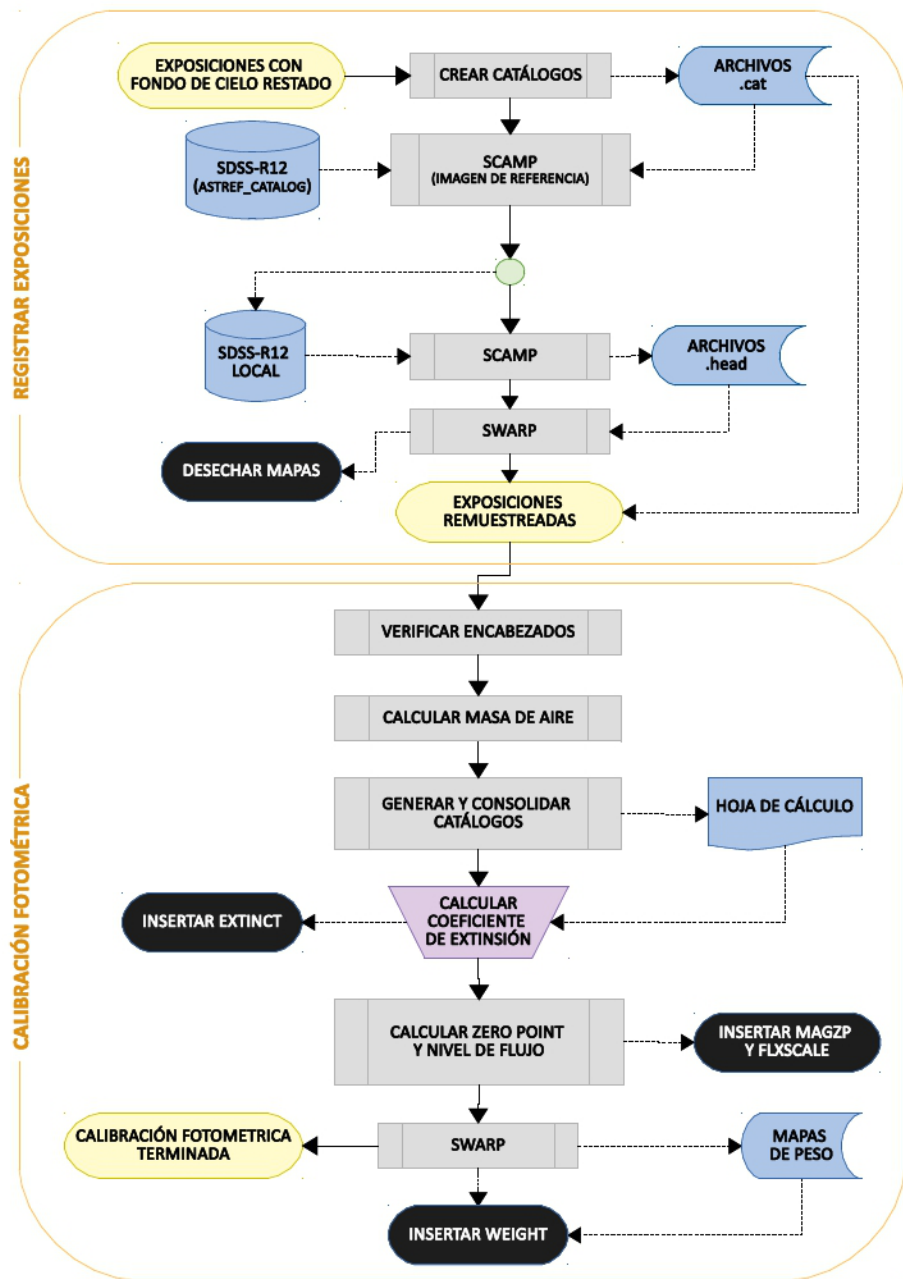


Figura 33. Diagrama de flujo de los procesos de Registro de Exposiciones y Calibración Fotométrica.

Tabla 8. *Scripts de Python desarrollados para el procesamiento de imágenes*

NOMBRE	FUNCIÓN
copyfilefromlist.py	Copia las exposiciones inicialmente seleccionadas como aceptables al directorio "lights". Se reenumeran automáticamente si tienen el mismo nombre.
reducprocess.py	Hace la reducción básica. Las imágenes calibradas van al directorio "lights_cal"
astap.py	Hace la solución astrométrica para cada imagen. Crea un archivo log de las imágenes sin solución y las mueve al directorio "sin solución"
bkgr_subtract.py	Hace la sustracción del background de las imágenes calibradas. Copia al directorio "lights_cal_b".
registration.py	Realiza el proceso de registro usando SExtractor y SCAMP conectados a través del paquete <i>astromatic_wrapper</i> .
insert_missingkeys.py	Inserta las claves necesarias en el encabezado de los archivos de las imágenes antes del cálculo de la masa de aire.
calculate_airmass.py	Calcula la masa de aire y guarda dicho valor en el encabezado de cada archivo para cada sesión de observación.
create_zcat.py	Crea los catálogos que se van a usar para el cálculo del coeficiente de extinción y el zero point.
xmatch.py	Realiza la relación cruzada entre el catálogo de entrada y el catálogo de referencia. Se usa en varios procesos.
vstack_zcat.py	Genera una hoja de cálculos con todos los catálogos creados por create_cat.py y agrega la información de airmass y fecha para cada registro.
insert_extinct.py	Inserta en el encabezado de los archivos el valor del coeficiente de extinción.
calculate_zeropoint.py	Calcula el zeropoint para cada imagen, dado el respectivo catálogo. Obtiene el airmass y coeficiente de extinción del encabezado. Realiza procesos de eliminación de valores atípicos. Calcula también el nivel de flujo relativo y lo guarda en el encabezado.
insert_weight.py	Para cada imagen, obtiene el valor de ponderación del mapa de peso y lo guarda en el encabezado.
astrometry_accuracy.py	Calcula las diferencias astrométicas (arcsec) entre un catálogo generado por SExtractor y un catálogo de referencia. Alternativamente, calcula la distancia (arcsec) entre una fuente y el centro de la imagen.
create_psfexcat.py	Dada solamente la imagen, hace todo el proceso para generar un catálogo con el estimador SPREAD_MODEL. Hace una relación cruzada con dicho catálogo y uno de referencia y lo guarda en una hoja de cálculo.

4. Resultados y análisis

Todas las exposiciones científicas individuales hasta el momento tienen una duración de 90s. Por ensayo y error, se ha determinado que es el mayor tiempo posible de exposición que nos permite eficientemente lidiar con la nubosidad. Se obtienen exposiciones libres de nubes y se desechan las contaminadas. El lado negativo de esto es la gran cantidad de imágenes que se deben procesar causando, en cierta medida, un problema de espacio en



Figura 34. Arriba: Muestra de una sección de exposición individual de 90s típica. Abajo: La misma sección pero de una exposición combinada con el procedimiento descrito en las secciones anteriores con un tiempo acumulado de ~8 horas.

discos y la necesidad de poseer poder de procesamiento, así como el tiempo consumido para ello.

4.1 Características astrométricas

Las distorsiones astrométricas en las imágenes son corregidas durante el proceso de registro. La Figura 35 muestra los errores de ubicación entre las fuentes detectadas por

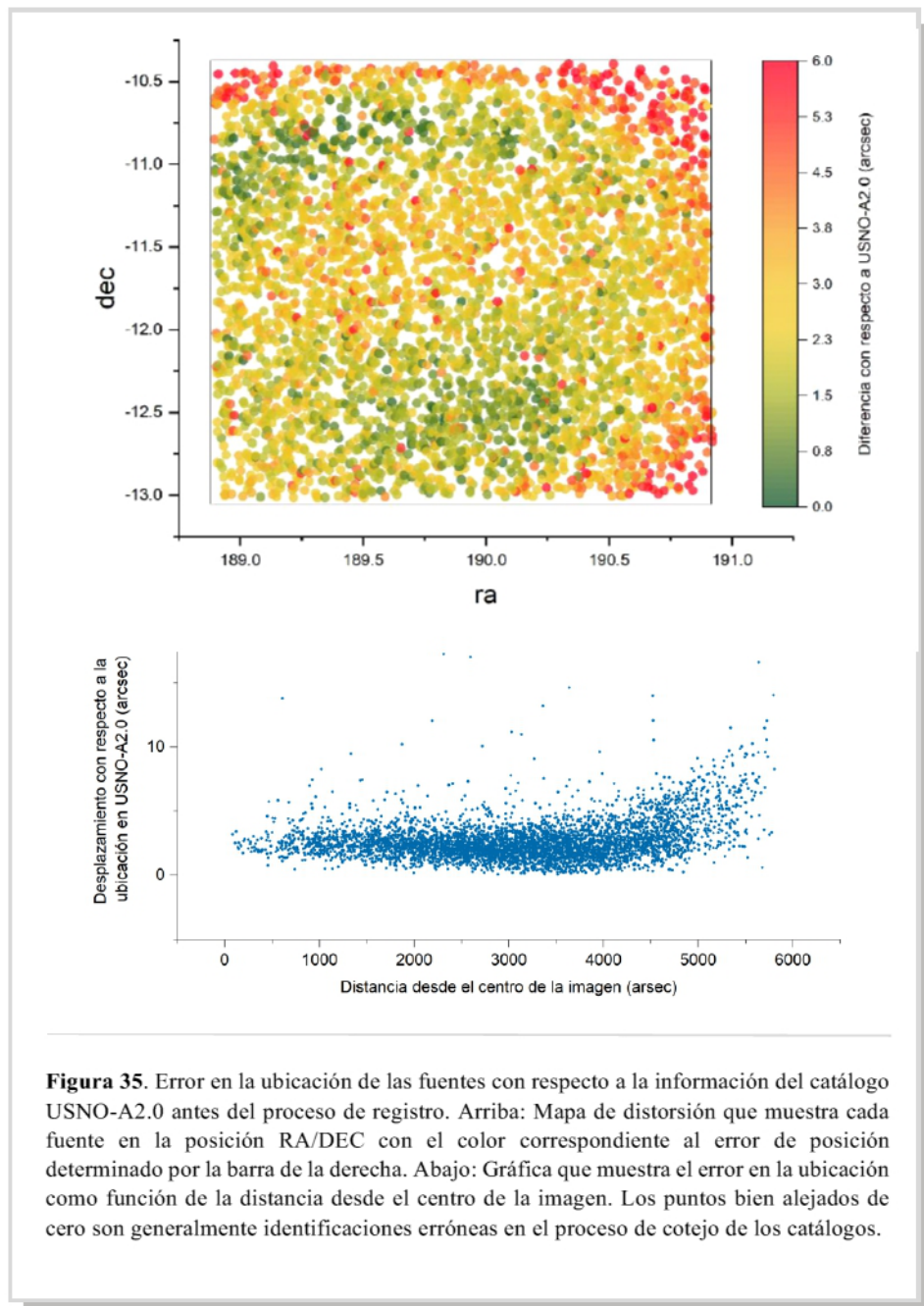


Figura 35. Error en la ubicación de las fuentes con respecto a la información del catálogo USNO-A2.0 antes del proceso de registro. Arriba: Mapa de distorsión que muestra cada fuente en la posición RA/DEC con el color correspondiente al error de posición determinado por la barra de la derecha. Abajo: Gráfica que muestra el error en la ubicación como función de la distancia desde el centro de la imagen. Los puntos bien alejados de cero son generalmente identificaciones erróneas en el proceso de cotejo de los catálogos.

SExtractor y las correspondientes posiciones de las fuentes del catálogo USNO-A2.0 (Monet et al., 2021) más cercanas en RA/DEC para una exposición individual típica con el fondo de cielo restado antes de ser registrada. Las diferencias están en el nivel de varios segundos de arco. Antes del registro se encontró que alrededor del 95% está por debajo de 5.66" (2 pixeles) y el 5% por debajo de 1". Tanto en la gráfica de arriba como en la de abajo de la Figura 35 puede verse como los errores de posición aumentan cerca de los

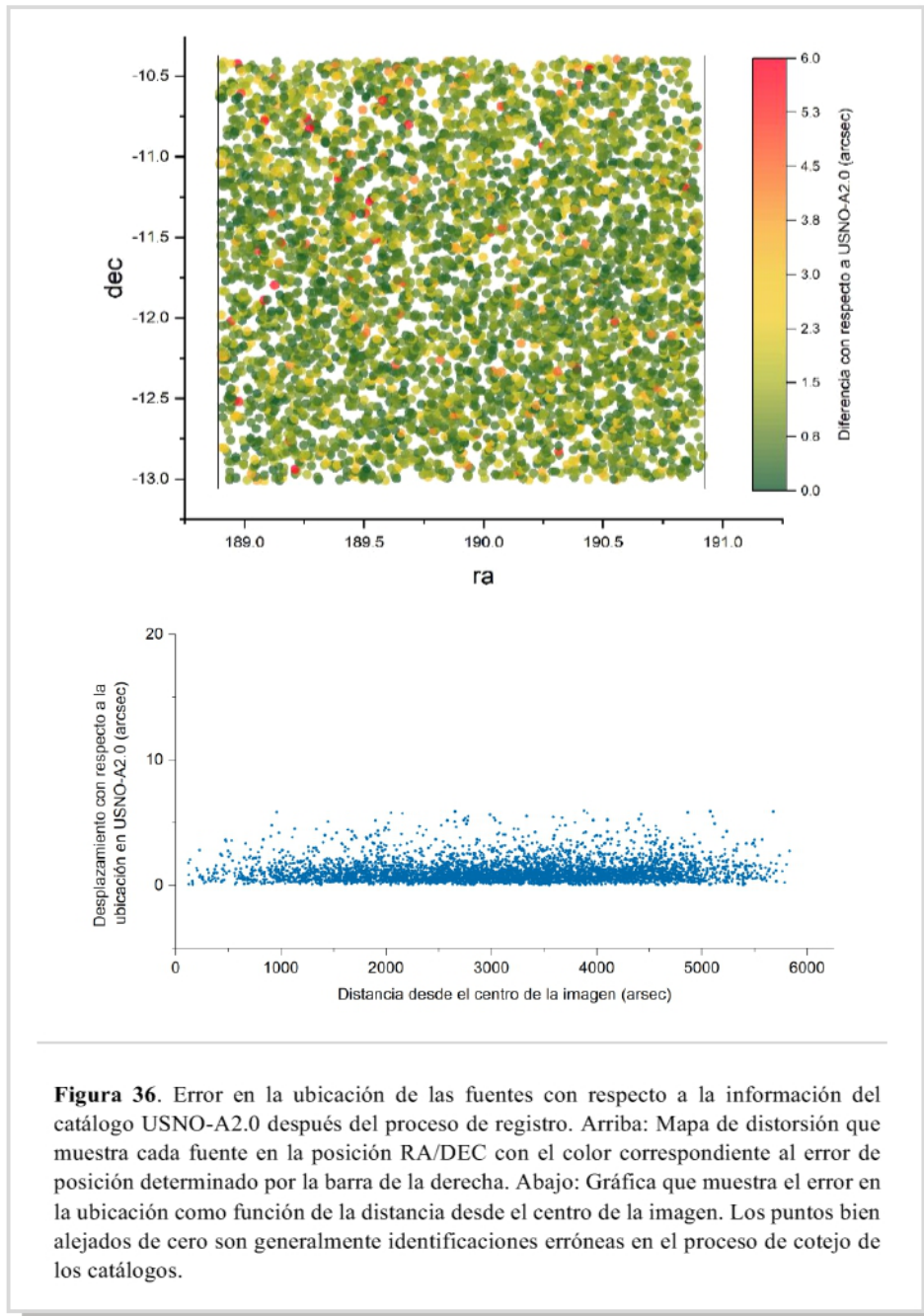


Figura 36. Error en la ubicación de las fuentes con respecto a la información del catálogo USNO-A2.0 después del proceso de registro. Arriba: Mapa de distorsión que muestra cada fuente en la posición RA/DEC con el color correspondiente al error de posición determinado por la barra de la derecha. Abajo: Gráfica que muestra el error en la ubicación como función de la distancia desde el centro de la imagen. Los puntos bien alejados de cero son generalmente identificaciones erróneas en el proceso de cotejo de los catálogos.

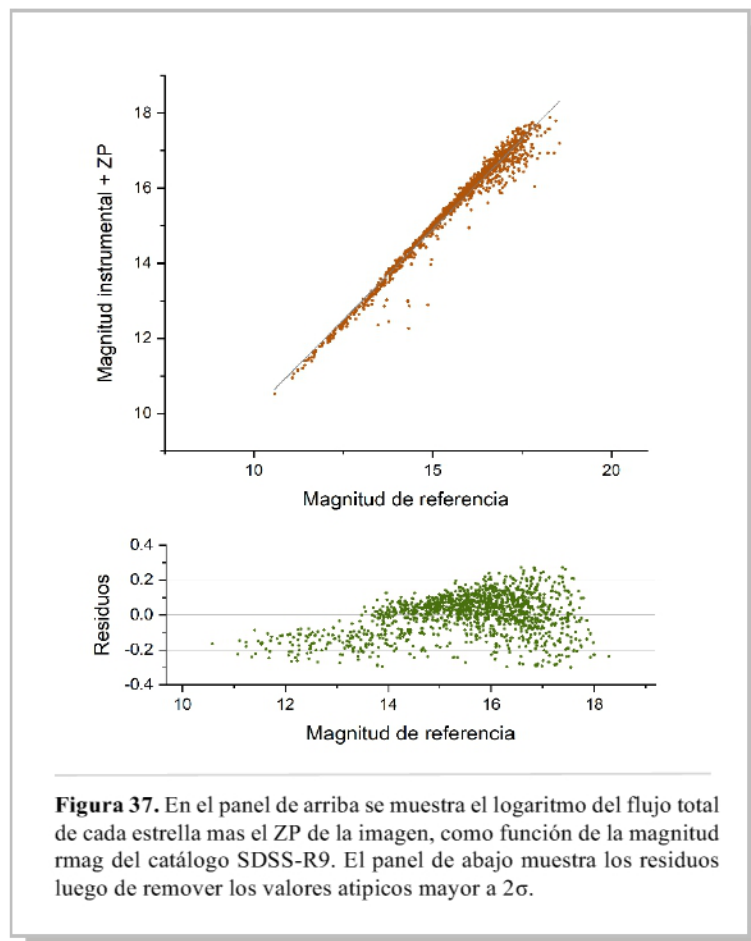
bordes del campo de visión. Esto generalmente se debe a distorsiones en la óptica de equipo (Laycock, et al. (2008).

Después del remuestreo hecho por SCAMP, más del 98% de las distorsiones están por debajo 2.83" (1 píxel) y más del 80% por debajo de 1" como puede verse en la Figura 36.

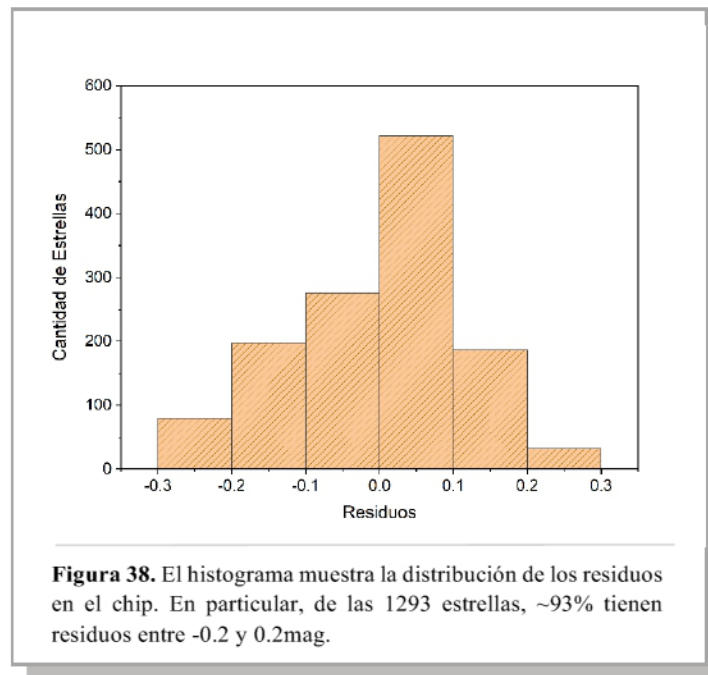
4.2 Características Fotométricas

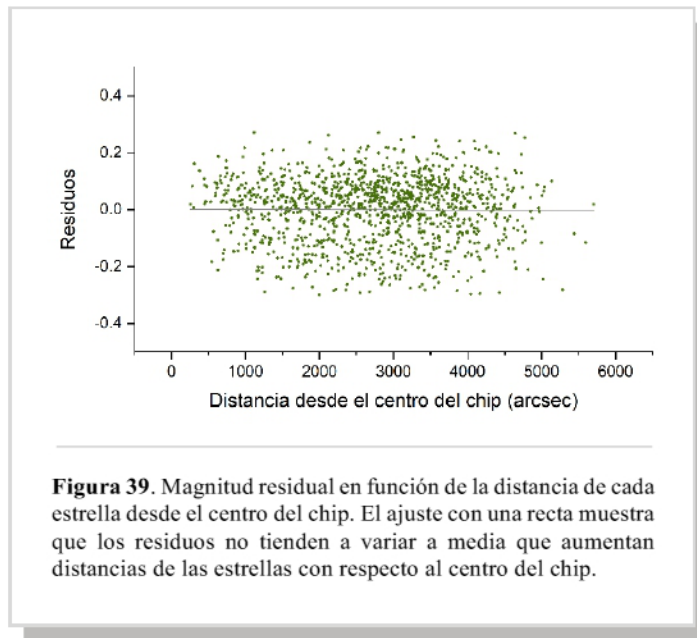
Precisión fotométrica:

Para verificar la precisión de la calibración fotométrica de una exposición calibrada y registrada, cruzamos las magnitudes de las estrellas con un catálogo de referencia. La



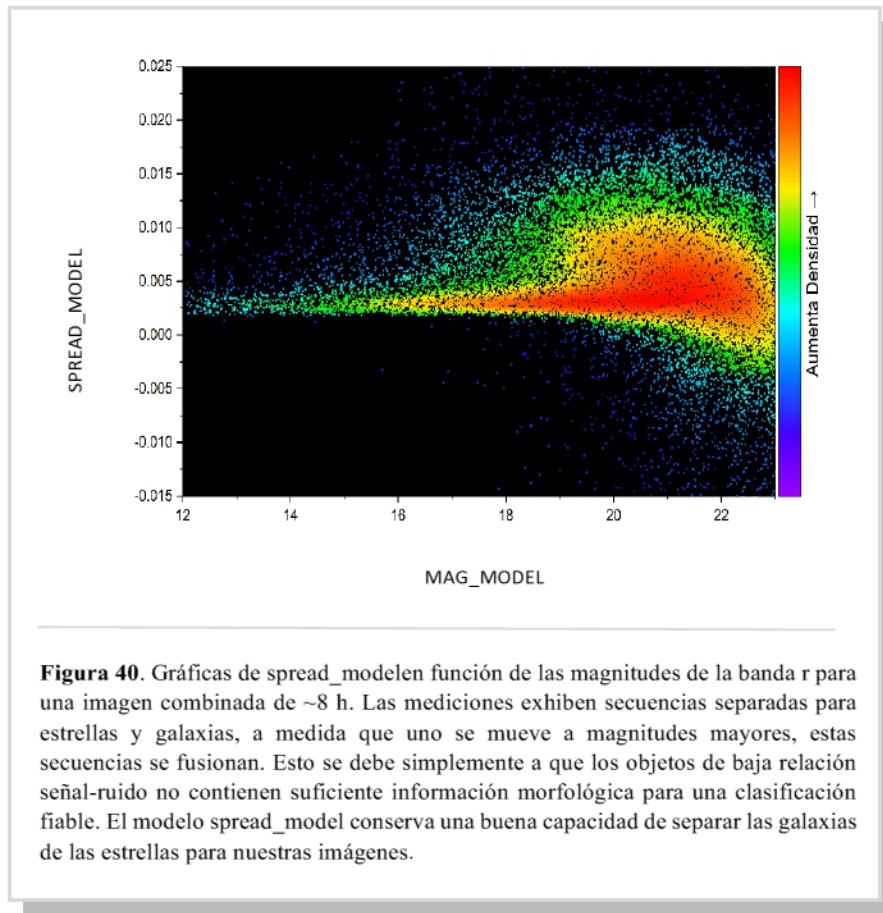
gráfica superior en la Figura 37 muestra un ejemplo de una relación entre el flujo total de estrellas y las magnitudes $rmag$ de SDSS-R9. Un polinomio de primer orden se ajusta a la relación y los residuos de la gráfica inferior, se forman restando el ajuste de las magnitudes $MODEL_MAG$ obtenidas con SExtractor . Las estrellas fueron seleccionadas de la relación cruzada filtrando los resultados con: $rmag < 20$ para evitar falsas detecciones, $-.003 < SPEAD_MODEL < 0.003$ para seleccionar evitar incluir fuentes extendidas (Mercurio, 2015) , $FLAG = 0$ para garantizar una fotometría confiable, $e_rmag < 0.01$ y $MAGERR_MODEL < 1.0$ (Erben, 2009) para descartar objetos con fotometría problemática. La desviación estándar calculada de los residuos es de $\sim 0.23\text{mag}$ el cual es usado como medida de precisión fotométrica (Lehtinen et al., 2018). La Figura 38 muestra la distribución de los residuos y la Figura 39 muestra los residuos en función de la distancia de la estrella desde el centro del chip. Se puede ver que la precisión fotométrica es prácticamente homogénea a lo largo del chip.





Clasificación de Objetos:

Con el fin de probar la precisión en la clasificación de objetos (galaxia/estrella) y el rango de magnitudes hasta el cual estas medidas se pueden utilizar de manera confiable, se trazó el comportamiento del modelo de propagación de SExtractor *spread_model* en la banda r en función de MAG_MODEL para una imagen combinada que representa ~8h de exposición. *Spread_model* utiliza el modelo PSF local para cuantificar las diferencias entre los objetos similares a PSF y los objetos resueltos (Bertin, 2017). En la figura 40, está claro que hay una fuerte secuencia estelar alrededor del valor 0.0, y que las galaxias exhiben valores mayores. Los valores negativos menores al modelo PSF representan interferencias electrónicas y rayos cósmicos entre otros. La secuencia estelar estrecha y la secuencia de galaxias anchas comienzan a fusionarse un poco antes de $r = 20$, pero hay una separación diferenciable en las dos distribuciones de puntos hasta $r = 20.5$. Así pues, es posible definir

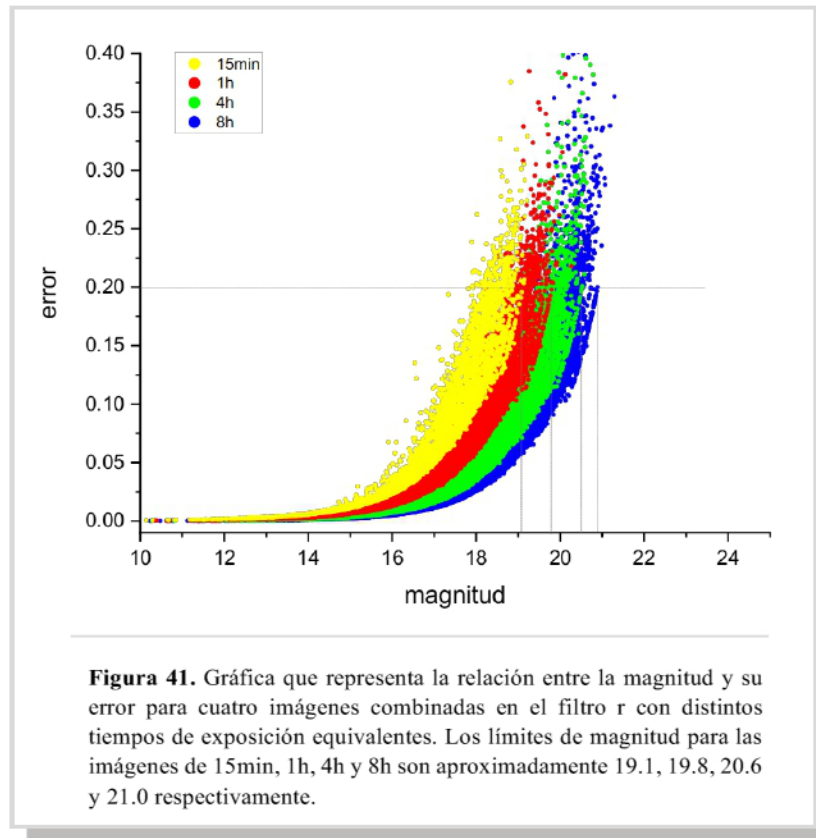


una muestra de objetos que se encuentran fuera de la secuencia estelar de una manera estadísticamente significativa. Para las imágenes combinadas como ésta, un buen corte para separar las estrellas sería $\text{spread_model} \leq 0.003$.

Magnitud Límite:

Para tener una idea de la profundidad de las imágenes combinadas se estimó la magnitud límite a 5σ . Para ello se graficó el error de magnitud estimado de cada fuente como función de su magnitud (Ngeow, 2017). En este caso, se traza la magnitud de MAG_ERRAUTO con respecto a cada MAG_AUTO calculada usando SExtractor. Un enfoque más refinado sería usar SPREAD_MODEL pero con las magnitudes de aberturas automáticas es suficiente para el propósito del presente documento.

En términos generales, los objetos extendidos serán más ruidosos para una magnitud dada que las fuentes puntuales debido a que tienen más cielo en sus aperturas. Esto significa que, para un error de magnitud fija dado, una fuente puntual será más débil que una fuente



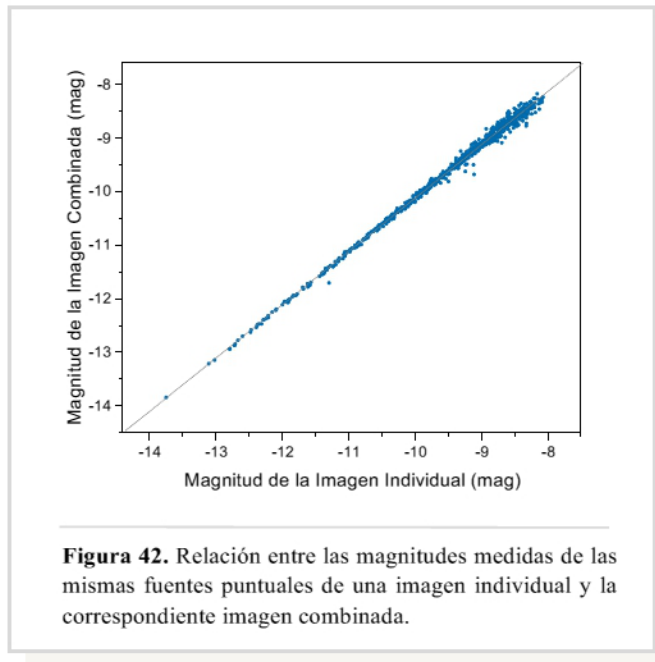
extendida. Una detección de 5σ corresponde a una S/N de 5 o, equivalentemente, a un error de magnitud de 0,198 magnitudes. Por lo tanto, para encontrar el límite de detección de fuente puntual de 5σ , se encuentra la fuente más débil cuyo error de magnitud es de 0.198 magnitudes o menos. Como debe ser una fuente puntual, normalmente se denomina límite de detección de fuente puntual de 5σ .

En la figura 41 pueden observarse dichos límites de detección para imágenes combinadas con distintos tiempos de exposición equivalentes. Obsérvese como a medida que aumenta la cantidad de exposiciones combinadas, aumenta la magnitud límite. Es decir, las imágenes se hacen cada vez más “profundas”. Es por eso que se tiene la idea de seguir trabajando en la adquisición de data y aumentar así la capacidad de detección.

Comparación de Magnitudes y FWHM:

El objetivo principal del apilamiento de imágenes, usando la presente canalización de datos, es combinar de manera óptima todas las mediciones disponibles en una mejor representación del cielo dadas las características instrumentales.

Para ver de qué manera afecta el proceso de apilado a las medidas de magnitud y FWHM se procede a comparar los valores de estos parámetros en las mismas fuentes puntuales de una exposición original y su correspondiente imagen combinada. Para tal fin se usó SExtractor con el proceso de ASSOC activado.



La Figura 42 muestra claramente que el procesamiento de los datos no ha afectado la medida de las magnitudes, éstas permanecen prácticamente constantes. De la Figura 43, se puede deducir que los valores de FWHM han disminuido con el apilado en términos generales, como es de esperarse.

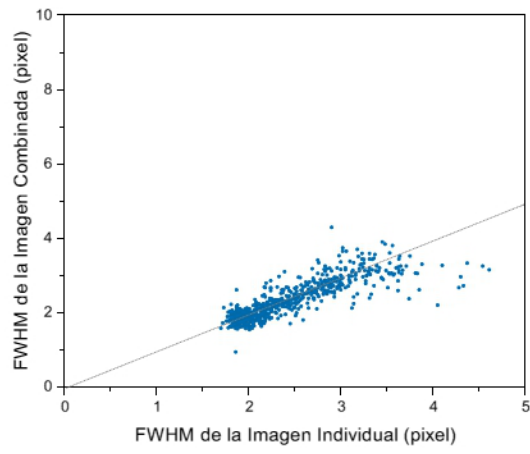


Figura 43. Relación entre las FWHM de las mismas fuentes puntuales medidas de una imagen individual y su correspondiente imagen combinada.

CONCLUSIONES

La gestión y ejecución de este proyecto de astrofísica ha permitido cumplir con los objetivos de este doctorado que es promover y fortalecer la gestión en investigación en los países de la región. Gracias a las diversas actividades académicas y de formación profesional en las que he participado en el transcurso del desarrollo del doctorado (ver ANEXO 2), así como con la realización de este proyecto en particular, me llevo un cúmulo de conocimientos que han hecho de mí un profesional más completo y capacitado para seguir trabajando en pro de la investigación científica.

El Sistema Novedoso de Óptica Refractiva Integrada (SNORRI) admite mejoras y, sobre todo, puede ser ampliado fácilmente sin una enorme cantidad de inversión. Esto permitirá incrementar la cantidad y mejorar la calidad de la data que se ha de adquirir en la época de la estación seca en Panamá, cuando es posible realizar las observaciones astronómicas.

Como se puede observar tiene una gran componente tecnológica especializada en optometría, en imágenes, en computación y en ciencias: Física (óptica y astronomía) y Matemática.

Los hardware y softwares utilizados son de una calidad tal que han permitido logrado, hasta el momento, altos niveles de fiabilidad, productividad y rendimiento. Para el manejo, calibración y seguimiento de la montura, el sistema de autoenfoco y la secuenciación de exposiciones de las cámaras se usó un conjunto de softwares de propietario hechos para tales fines y que tienen una curva de aprendizaje relativamente larga.

Para lograr los diferentes procesos en torno a la canalización del procesamiento de imágenes como los son: monitoreo y filtrado de imágenes según calidad en FWHM, elipticidad y precisión astrométrica entre otros, proceso de reducción básica de la data, estimación y resta de cielo, calibración astrométrica y registro, cálculos referentes a la masa de aire, coeficiente de extinción y zero point, normalización y apilado, etc., se creó una gama de scripts de python los cuales hacen uso en gran medida de softwares especializados en distintas tareas específicas. En el futuro se pretende enlazar todos los códigos en una

sola plataforma para lograr una mayor automatización. Al analizar algunas imágenes se pudo determinar que el proceso de apilado no afectó la medida de las magnitudes y que los valores de FWHM han disminuido.

Al hacer un análisis general de algunas imágenes resultantes se determinó una buena precisión fotométrica y astrométrica. Prácticamente toda la totalidad de las distorsiones astrométricas están por debajo de un pixel y los residuos fotométricos son bajos y están distribuidos uniformemente a lo largo del chip. Por otro lado, se concluyó que es posible definir una muestra de objetos que se encuentran fuera de la secuencia estelar de una manera estadísticamente significativa con exposiciones equivalentes a 8h o más. Con nuestro equipo la relación existente entre magnitud límite y tiempo de exposición nos permite proyectar un aumento significativo en la capacidad de detección.

REFERENCIAS

- Abraham, R. & Van Dokkum, P. (2014). Ultra-Low Surface Brightness Imaging with the Dragonfly Telephoto Array. *Publ. Astron. Soc. Pac.*, 126(935), 55-69.
- Ábrego, V. (9 de septiembre de 2019). Investigadores panameños trabajan para recolectar datos de las galaxias. La Estrella de Panamá. <https://www.laestrella.com.pa/nacional/190909/datos-galaxias-panamenos-recolectar-investigadores>
- ADM. (Sin fecha). Adjustable Rings with Delrin Tipped Thumb Screws. <https://www.admaccessories.com/product/r175-175-mm-adjustable-rings-delrin-tipped-thumb-screws/>
- Ahn, C. P., Alexandroff, R., Allende Prieto, C. et al. (2012). The Ninth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the SDSS-III Baryon Oscillation Spectroscopic Survey. *Astrophys. J., Suppl. Ser.*, 203, 21.
- Alam, S., Albareti, F. D., Allende Prieto, C. et al. (2015). The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-II. *ApJS*, 219, 12.
- Araúz, M. (2016). De la oscuridad a la luz. *IMAGiNA: Revisa de ciencia, tecnología e innovación*, 2, 9.
- Astrodon. (Sin fecha). Astrodon pemium filters for astronomical imaging and research. <https://astrodon.com/>
- Astromechanics. (Sin fecha). ASCOM Canon Lens Controller Mark II. <https://astromechanics.org/ascom.html>
- Bertin et al. (2002). The TERAPIX Pipeline, in ADASS XI. *ASP Conference Ser.*, 281, 228.
- Bertin, E. & Arnouts, S. (1996). SExtractor: Software for source extraction. *A&AS*, 317, 393.
- Bertin, E. (2006). Automatic Astrometric and Photometric Calibration with SCAMP, in ADASS XV. *ASP Conference Ser.*, 351, 112
- Bertin, E. (2017). SExtractor User Manual - Model fitting. <https://sextractor.readthedocs.io/en/latest/Model.html>
- CADC. (sin fecha). The NGVS data pipeline: background subtraction. <http://www.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/en/community/ngvs/docs/megaback.html>
- Canon Inc. (2011). Canon EF Lens EF 400mm f/2.8L IS II USM: instruction manual. <https://gdlp01.c-wss.com/gds/6/0300005626/03/ef300-400f28lisisium-im3-eng.pdf>

CCDWare. (sin fecha). FocusMax. <https://www.ccdware.com/products/focusmax/>

Chromey, F. & Hasselbacher, D. (1996). The Flat Sky: Calibration and Background Uniformity in Wide Field Astronomical Images. *PASP*, 108, 944 – 949.

Diffraction Limited. (2019). Aluma: User’s Manual. <http://diffractionlimited.com/downloads/AlumaUsersManual.pdf>

Diffraction Limited. (2019). Cyanogen Imaging ® MaxIm DL: User’s Manual. <http://diffractionlimited.com/help/maximdl/MaxIm-DL.htm>

Erben, T. et al. (2009). CARS: the CFHTLS-Archive-Research Survey. I. Five-band multi-colour data from 37 sq. deg. CFHTLS-wide observations. *A&A*, 493, 1197.

Gruen, D., Seitz, y Bernstein, G. M. (2014). Implementation of Robust Image Artifact Removal in SWarp through Clipped Mean Stacking. *PASP*, 126, 158-169.

Han, k. (sin fecha). The Astrometric STacking Program. <http://www.hnsky.org/astap.htm>

Joye W. A., Mandel E. (2003). Features of SAOImage DS9, in ADASS XII. *ASP Conference Ser.*, 295, 489.

La Cour, M. (2018). Statistical view on FocusMax. <http://docplayer.net/20132843-Statistical-view-on-focusmax.html>

Laycock, S. et al. (2008). Digital access to a sky century at Harvard. II: Initial photometry and astrometry. <https://arxiv.org/pdf/0811.2005.pdf>

Lehtinen, K., Prusti, T. et al. (2018). Digitization and astrometric calibration of Carte du Ciel photographic plates with Gaia DR1. *A&A*, 616 – 185.

Martínez Delgado, D. et al. (2010). Stellar tidal streams in spiral galaxies of the local volume: a pilot survey with modest aperture telescopes, *AJ*, 140, 962-967.

Mercurio, A. , Merluzzi, P. et al. (2015). Shapley Supercluster Survey: construction of the photometric catalogues and i-band data release. *MNRAS*, 453(4), 3685–3698.

Modelo Lambda-CDM (s.f.). En Wikipedia. https://es.wikipedia.org/wiki/Modelo_Lambda-CDM

Monet, D. et al. (2021). USNO-A2.0: A Catalog of Astrometric Standards. <http://tdc-www.harvard.edu/catalogs/ua2.html>

Muñoz, M. (8 de enero de 2020). Mentas Brillantes: Pablo y Daniel, hermanos científicos. Telemetro Reporta. <https://www.telemetro.com/reportajes/2020/01/08/mentas-brillantes-pablo-y-daniel-hermanos-cientificos/2416288.html>

NexDome. (Sin fecha). 2.2 Meter dome. https://dd3add39-c382-40f5-bb8f-d4c4da4473fb.filesusr.com/ugd/8a866a_faf0f9fea82c4cf78615b165971bb83f.pdf

- Ngeow, C-C y Luo, S-C. (2017). Limiting magnitudes and night sky brightness at the Langkawi National Observatory based on observations of standard stars. *IOP Conf. Ser. J. Phys*, 852, 012002.
- OmegaCEN. (2020). Astro-WISE: HOW-TO Subtract sky background. http://doc.astro-wise.org/man_howto_sky.html
- ON Semiconductors. (sin fecha). KAF-8300: 3326 (H) x 2504 (V) FullFrame CCD Image Sensor. <https://www.onsemi.com/pdf/datasheet/kaf-8300-d.pdf>
- Pleiades Astrophoto. (2021). Pixinsight. <http://pixinsight.com/>
- Reyes, N. (21 de diciembre de 2020). Este lunes se observará el fenómeno de la 'estrella de Belén'. TVN Noticias. https://www.tvn-2.com/videos/noticias/lunes-observara-fenomeno-estrella-Belen_2_5744445504.html
- Saldaña, I. (9 de septiembre de 2019). Investigadores desarrollan un proyecto para estudiar las galaxias. El Siglo. <http://elsiglo.com.pa/panama/astrofisicos-panamenos-desarrollan-proyecto-para-estudiar-galaxias/24138400>
- SENACYT. (2015). Política nacional de ciencia, tecnología e innovación de Panamá y plan nacional 2015 – 2019. https://www.senacyt.gob.pa/wp-content/uploads/2017/03/PENCYT-2015_2019.pdf
- Shany Danieli et al.(2020). The Dragonfly Wide Field Survey. I. Telescope, Survey Design, and Data Characterization. *ApJ*, 894, 119.
- Software Bisque. (sin fecha). TheSky Professional. <https://www.bisque.com/product/theskyx-pro/>
- Sotolongo Costa, O., Gámez, R., Luzón, F., Posadas, A., Weigandt, P. (2008). Non extensivity in meteor showers. *Apeiron*. 15(2), 187 - 201.
- Stark Labs. (2021). PHD2 Guiding User Manual. https://openphdguiding.org/PHD2_User_Guide.pdf
- Telescopes.net. (Sin fecha). QHYCCD - Guide Scope f/4 CS Mount with Helical Focuser and Mounting. <https://telescopes.net/qhyccd-guide-scope-130mm-f-4-cs-mount-with-helical-focuser-and-mounting.html>
- Telescopes.net. (Sin fecha). Starlight Xpress - Ultrastar ICX825AL Sony Exview Interline CCD Camera – Monochrome. <https://telescopes.net/starlight-xpress-ultrastar-icx825al-sony-exview-interline-ccd-camera.html>
- Tody, D. (1986). The IRAF data reduction and analysis system. In Proc. SPIE. Instrumentation and Astronomy VI, Ed. D.L. Crawford, 627 – 733.
- Tody, D. (1993). IRAF in the Nineties, in ADASS II. *ASP Conference Ser.*, 52, 173.

Van Dokkum, P. G., Abraham, R., & Merritt, A. (2014). The Discovery of Seven Extremely Low Surface Brightness Galaxies in the Field of the Nearby Spiral Galaxy M101. *ApJL*, 782, L24.

VILLARREAL, D. (2014). Determinación del estado de funcionamiento del radiotelescopio dana mediante el análisis de la línea espectral del H neutro de 21cm y la observación del flujo solar. Tesis. UNACHI, David, Chiriquí, Panamá. 105 págs.

Weigandt, D. (Sin fecha). SNORRI: Sistema novedoso de óptica refractiva integrada. [Sitio Web]. <https://snorri.space/>

Weigandt, P. & Weigandt, D. (2020). Matriz de súper telefotos para la detección de objetos astronómicos de bajo brillo superficial en Churuquita Chiquita de Coclé, Panamá. *Tecnociencia*, 22(1), 37-47.

Weigandt, P., Weigandt, D. & Márquez, A. (2021). SNORRI: Un proyecto de astrofísica en el seno de la comunidad coclesana. *Revista Científica Guacamaya*, 5(2), 19-29. <https://revistas.up.ac.pa/index.php/guacamaya/article/view/2053>

ANEXO 1

Scripts principales hechos en Python desarrollados para el procesamiento de imágenes. Los programas con GUI se deben correr en python2.7, los demás en python3.5+. A pesar de que son totalmente funcionales, hay que hacer la salvedad que este conjunto de scripts está en una etapa de desarrollo, no se puede ver como un producto terminado.

copyfilesfromlist.py

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
'''
Copia los archivos de una lista al directorio "lights" (puede cambiarse en el código si se
quiere otro nombre).
La lista debe incluir la ruta completa y estar ubicada en el directorio raíz del proyecto.
Se reenumeran automáticamente si tienen el mismo nombre.
'''
import os
import shutil
import sys

from PyQt4.QtGui import QDialog, QApplication, QWidget

class filedialog(object):
    def __init__(self, parent=None):
        self.fdialog()
        self.copyfiles()

    def fdialog(self):
        self.filePath = QDialog.getOpenFileName()
        print(self.filePath)
        self.destination = os.path.dirname(str(self.filePath)) + '/lights'

    def copyfiles(self):
        filesToFind = []
        with open(self.filePath, "r") as fh:
            for row in fh:
                filesToFind.append(row.strip())

        for filename in filesToFind:
            self.safe_copy(filename, self.destination)
            print(filename)

    def safe_copy(self, file_path, out_dir, dst=None):
        name = dst or os.path.basename(file_path)
        if not os.path.exists(os.path.join(out_dir, name)):
            shutil.copy(file_path, os.path.join(out_dir, name))
```

```

else:
    base, extension = os.path.splitext(name)
    i = 1
    while os.path.exists(os.path.join(out_dir, '{}_{}'.format(base, i,
                                                                    extension))):
        i += 1
    shutil.copy(file_path, os.path.join(out_dir, '{}_{}'.format(base, i,
                                                                    extension)))

if __name__ == '__main__':
    app = QApplication(sys.argv)
    myapp = filedialog()
    sys.exit(app.exec_())

```

gui_reducprocess.py (gui para el script de reducprocess.py)

```

#!/usr/bin/env python
# -*- coding: UTF-8 -*-
from PyQt4.QtGui import *
from PyQt4.QtCore import QThread, pyqtSignal
import time
from PyQt4 import uic
import sys
import reducprocess

class TaskThread(QThread):
    taskFinished = pyqtSignal()

    def run(self):
        reducprocess.go([myapp.ui.lw_lights.query(True),
                        myapp.ui.lw_flats.query(True),
                        myapp.ui.lw_bias.query(True),
                        myapp.ui.lw_darks.query(True),
                        ])

        self.taskFinished.emit()

class ReductionForm(QWidget):
    def __init__(self):
        QWidget.__init__(self)
        self.ui = uic.loadUi('oap-reduction.ui', self)
        self.btn_of = [self.ui.btn_lights, self.ui.btn_flats, self.ui.btn_bias,
self.ui.btn_darks]
        for button in self.btn_of:
            button.clicked.connect(lambda: self.openfiles())
        self.btn_go.clicked.connect(lambda: self.onStart())
        self.progressBar.setRange(0, 100)
        self.progressBar.setValue(0)
        self.myLongTask = TaskThread()

```

```

        self.myLongTask.taskFinished.connect(self.onFinished)

    def onStart(self):
        self.myLongTask.start()

    def onFinished(self):
        self.progressBar.setValue(100)

    def openfiles(self):
        sending_button = self.sender()
        print (sending_button.objectName())
        if sending_button.objectName() == "btn_lights":
            objle = self.ui.lne_lights
            objlw = self.ui.lw_lights
        if sending_button.objectName() == "btn_flats":
            objle = self.ui.lne_flats
            objlw = self.ui.lw_flats
        if sending_button.objectName() == "btn_bias":
            objle = self.ui.lne_bias
            objlw = self.ui.lw_bias
        if sending_button.objectName() == "btn_darks":
            objle = self.ui.lne_darks
            objlw = self.ui.lw_darks
        f = QFileDialog.getOpenFileNames(self.ui, 'Abrir Archivo', "", "**.* *.*")
        objlw.list = f
        objlw.fill(True) # True es solo archivos sin ruta
        objle.setText(objlw.dirpath())

if __name__ == '__main__':
    app = QApplication(sys.argv)
    myapp = ReductionForm()
    myapp.show()
    sys.exit(app.exec_())

```

reducprocess.py

```

from pyraf import iraf
from constants import *
import os
import filetools as ft

def task(name=None, inputlist=None, combine='average', scale='none',
        reject='none', nlow=0, nhigh=1, lsigma=3, hsigma=3,
        fields=None, op=None, operand2=None, output=None):
    if os.path.exists(TEMPLIST):
        os.remove(TEMPLIST)
    tempfile = open(TEMPLIST, 'w')
    for item in inputlist:
        tempfile.write("%s\n" % item)
    tempfile.close()

```

```

if name == 'imcombine':
    iraf.imcombine(input='@' + TEMPLIST, combine=combine, scale=scale, reject=reject,
                  nlow=nlow, nhigh=nhigh, lsigma=lsigma, hsigma=hsigma, output=output)
if name == 'imarith':
    iraf.imarith(operand1='@' + TEMPLIST, op=op,
                operand2=operand2, result='@' + TEMPLIST)
if name == 'imstat':
    if os.path.exists(MODELIST):
        os.remove(MODELIST)
    iraf.imstat(images='@' + TEMPLIST, fields=fields, format='no', Stdout=MODELIST)

def go(frames):
    iraf.reset(clobber="no")
    listpath = [PATHLIGHTS, PATHFLATS, PATHBIAS, PATHDARKS]
    # Borrar posibles archivos en el directorio masters
    print('borrando archivos en Masters')
    if os.path.exists(MASTERBIAS):
        os.remove(MASTERBIAS)
    if os.path.exists(MASTERFLAT):
        os.remove(MASTERFLAT)
    if os.path.exists(MASTERDARK):
        os.remove(MASTERDARK)

    print('copiando archivos a directorios temporales...')
    for f, lp in zip(frames, listpath):
        ft.copypath(f, lp)

    filelists = [[] for i in range(4)]

    # cambio el path original en frames al path nuevo (lights_cal, flats_cal, etc.)
    for lp in listpath:
        for file in os.listdir(lp):
            filelists[listpath.index(lp)].append(lp + file)
            print lp + file

    # Genera el MasterBias
    print ('generando el Masterbias...')
    task(name='imcombine', inputlist=filelists[2], combine='average',
         reject='minmax', nlow=0, nhigh=1, scale='none', output=MASTERBIAS)

    # Flats - MasterBias
    print ('Flats - Masterbias ...')
    task(name='imarith', inputlist=filelists[1], op='-', operand2=MASTERBIAS)

    print ('Darks - Masterbias ...')
    # Darks - MasterBias
    task(name='imarith', inputlist=filelists[3], op='-', operand2=MASTERBIAS)

    # Genera el MasterDark (para un tiempo de exposicion especifico)
    print ('generando el MasterDark...')
    task(name='imcombine', inputlist=filelists[3], combine='average',
         reject='minmax', nlow=0, nhigh=1, scale='exposure', output=MASTERDARK)

    # Normalizar los Flats-MasterBias
    print ('normalizando el Flats-Masterbias...')

```

```

task(name='imstat', inputlist=filelists[1], fields='mode')
task(name='imarith', inputlist=filelists[1], op='/', operand2='@' + MODELIST)

# Genera el MasterFlat de los Flats-MasterBias normalizados
print ('generando el MasterFlat...')
task(name='imcombine', inputlist=filelists[1], combine='average',
      scale='mode', reject='avsigclip', lsigma=3, hsigma=3, output=MASTERFLAT)

# Lights - MasterBias
print ('Lights - Masterbias...')
task(name='imarith', inputlist=filelists[0], op='-', operand2=MASTERBIAS)
print "se resto el masterbias"

# (Lights - MasterBias) - MasterDark
print ('(Lights - MasterBias) - MasterDark')
task(name='imarith', inputlist=filelists[0], op='-', operand2=MASTERDARK)

print "dividiendo por el MasterFlat"
# (Lights - MasterBias) - MasterDark / MasterFlat
task(name='imarith', inputlist=filelists[0], op='/', operand2=MASTERFLAT)

# Registro
print ('borrando el los archivos temporales de Bias, Darks y Flats ...')
ft.removefolder(PATHBIAS)
ft.removefolder(PATHDARKS)
ft.removefolder(PATHFLATS)
print "***** Fin *****"

```

astap.py

```

#!/usr/bin/env python
# coding=utf-8
import os
import sys
import argparse
from astropy.io.fits import getdata, getheader, writeto, Header
import shutil

```

```

parser = argparse.ArgumentParser(description='Realiza la solución '+
                                     'astrométrica usando ASTAP ')
parser.add_argument("-dir", action='store', default='',
                    help=" nombre de la carpeta que contiene las imágenes." +
                    "Esta carpeta debe estar en el directorio raíz del proyecto")
parser.add_argument("-ra", type=float, default='',
                    help=" RA del centro del campo que se procesa")
parser.add_argument("-dec", type=float, default='',
                    help=" DEC del centro del campo que se procesa ")
parser.add_argument("-r", type=int, default='30',
                    help=" Radio de búsqueda en grados")
parser.add_argument("-fov", type=float, default='1.97',
                    help=" altura del campo en grados ")

def get_solution(args):
    folder = './' + args.dir + '/'

    logfile = 'astap_log'
    if os.path.exists(logfile):
        os.remove(logfile)
    logfile = open(logfile, 'a')
    logfile.write('No tiene solución :(')

    for image in os.listdir(folder):
        filename = os.path.join(folder, image)
        print(filename)
        wcs_filename = os.path.splitext(filename)[0] + '.wcs'
        ini_filename = os.path.splitext(filename)[0] + '.ini'
        ra = args.ra
        dec = args.dec
        spd = dec + 90
        r = args.r
        fov = args.fov
        result = os.system('astap -f ' + filename + ' -ra ' + str(ra) + ' -spd ' + str(spd)
+ ' -r ' + str(r) + ' -fov ' + str(fov) + ' -wcs')
        if result:
            print('num_error: ' + str(result) + '*****')
            logfile.write('\n')
            logfile.write(filename)
            shutil.move(filename, os.path.join('./sin_solucion', image))
        else:
            data = getdata(filename)
            hdr = getheader(wcs_filename, ignore_missing_end=True)
            writeto(filename, data, hdr, overwrite=True, output_verify='warn')

            if os.path.exists(wcs_filename):
                os.remove(wcs_filename)
            if os.path.exists(ini_filename):
                os.remove(ini_filename)

    logfile.close()

```

```

def main(arguments=None):
    if arguments == None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    get_solution(args)

if __name__ == "__main__":
    main()

```

bkg_subtract.py

```

#!/usr/bin/env python
# coding=utf-8
import os.path
import sys
from constants import *
import astromatic_wrapper as aw
import argparse
import os
from pyraf import iraf

parser = argparse.ArgumentParser(description='Resta el background de las imagenes. '
                                           'Procedimiento: ' +
'1. Convulsionamos la imagen con iraf.gauss sigma 7, ' +
'2. Creamos una imagen segmentada de 1 con sextractor, ' +
'3. Creamos una mascara haciendo mask = 1 - seg/seg, ' +
'4. Creamos el mapa de background corriendo sextractor con: '+
    "'WEIGHT_TYPE': 'MAP_WEIGHT', "+
    "'WEIGHT_IMAGE': mask, " +
'5. Restamos el background de la imagen.')

parser.add_argument("-inpath", action='store', default=PATHLIGHTS_WCS,
                    help=" Directorio donde residen las imagenes." )
parser.add_argument("-outpath", action='store', default=PATHLIGHTS_BKG,
                    help=" Directorio de destino de las imagenes 'sin background'.")

def get_background(args):
    inpath = args.inpath
    outpath = args.outpath
    temp = "temp.fits"
    seg = "seg.fits"
    mask = "mask.fits"
    bkg = "bkg.fits"
    bkgfitted = "bkg-fitted.fits"

    for img in os.listdir(inpath):
        if os.path.exists(temp):
            os.remove(temp)

```

```

if os.path.exists(seg):
    os.remove(seg)
if os.path.exists(mask):
    os.remove(mask)
if os.path.exists(bkg):
    os.remove(bkg)
if os.path.exists(bkgfitted):
    os.remove(bkgfitted)

image = inpath + img
iraf.gauss(image, temp, 7.0)

files = {
    'temp': temp,
}
kwargs = {
    'code': 'SExtractor',
    'cmd': 'sextractor',
    'config': {
        'CATALOG_TYPE': 'NONE',
        'FILTER_NAME': 'SEXCONV',
        'BACK_SIZE': '128',
        'CHECKIMAGE_TYPE': 'SEGMENTATION',
        'CHECKIMAGE_NAME': 'seg',
    },
    'temp_path': '.',
    'params': ['NUMBER', 'EXT_NUMBER', 'XWIN_WORLD', 'YWIN_WORLD', 'MAG_AUTO'],
}

sextractor = aw.api.Astromatic(**kwargs)
sextractor.run_frames(files['temp'], frames=[1])

iraf.imarith(seg, '/', seg, mask)
iraf.imarith(1, '-', mask, mask)

kwargs = {
    'code': 'SExtractor',
    'cmd': 'sextractor',
    'config': {
        'CATALOG_TYPE': 'NONE',
        'FILTER_NAME': 'SEXCONV',
        'WEIGHT_TYPE': 'MAP_WEIGHT',
        'WEIGHT_IMAGE': mask,
        'BACK_SIZE': '128',
        'CHECKIMAGE_TYPE': 'BACKGROUND',
        'CHECKIMAGE_NAME': 'bkg',
    },
    'temp_path': '.',
    'params': ['NUMBER', 'EXT_NUMBER', 'XWIN_WORLD', 'YWIN_WORLD', 'MAG_AUTO'],
}

files = {
    'image': image,
}

```

```

sextractor = aw.api.Astromatic(**kwargs)
sextractor.run_frames(files['image'], frames=[1])

iraf.imsurfit(bkg, bkgfitted, 4, 4)
img_out = img.split('.')[0] + '_b.' + img.split('.')[1]
iraf.imarith(image, '-', bkgfitted, os.path.join(outpath, img_out))

print ('*** FIN ***')

def main(arguments=None):
    if arguments == None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    get_background(args)

if __name__ == "__main__":
    main()

```

registration.py

```

#!/usr/bin/env python
# coding=utf-8
import argparse
import os.path
import sys
import astromatic_wrapper as aw
from constants import *
import os
import shutil

parser = argparse.ArgumentParser(description='Registra las imagenes en una' +
                                         'cuadrícula comun usando SCAMP')
parser.add_argument("-refcatfile", action='store', default='',
                    help=" nombre de la carpeta que contiene los catalogos de ." +
                    "Vezier creados corriendo este script haciendo los cambios: " +
                    "'ASTREF_CATALOG': 'SDSS-R9', " +
                    "'SAVE_REFCATALOG': 'Y', " +
                    "'REFOUT_CATPATH': REFCATPATH,")

def go_registration(args):

    if os.path.exists(TEMPCAT):
        shutil.rmtree(TEMPCAT)
    os.mkdir(TEMPCAT)

    for image in os.listdir(PATHLIGHTS_BKG):
        catalogo = os.path.splitext(image)[0] + '.cat'

```

```

files = {
    'image': os.path.join(PATHLIGHTS_BKG, image),
}
kwargs = {
    'code': 'SExtractor',
    'cmd': 'sextractor',
    'config': {
        'CATALOG_TYPE': 'FITS_LDAC',
        'CATALOG_NAME': os.path.join(TEMPCAT, catalogo),
        'FILTER_NAME': SEXCONV,
    },
    'temp_path': '.',
    'params': ['XWIN_IMAGE', 'YWIN_IMAGE', 'ERRAWIN_IMAGE', 'ERRBWIN_IMAGE',
               'ERRTHETAWIN_IMAGE', 'FLUX_AUTO', 'FLUXERR_AUTO', 'FLAGS',
               'FLAGS_WEIGHT', 'FLUX_RADIUS', 'ELONGATION'],
}

sextractor = aw.api.Astromatic(**kwargs)
sextractor.run_frames(files['image'], frames=[1])

print ('*** FIN CREACION DE CATALOGOS ***')

for cat in os.listdir(TEMPCAT):
    files = {
        'cat': TEMPCAT + cat,
    }
    kwargs = {
        'code': 'SCAMP',
        'cmd': 'scamp',
        'config': {
            'ASTREF_CATALOG': 'FILE',
            'ASTREFCAT_NAME': os.path.join(REFCATPATH, args.refcatfile),
            'CHECKPLOT_DEV': 'NULL'
        },
        'temp_path': '.',
    }

    scamp = aw.api.Astromatic(**kwargs)
    scamp.run(files['cat'])

print ('*** FIN ***')

def main(arguments=None):
    if arguments == None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    go_registration(args)

if __name__ == "__main__":
    main()

```

insert_missingkeys.py

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import argparse
import os
import astropy.io.fits as fits
import sys
from constants import *

parser = argparse.ArgumentParser(description='Toma la clave dada del header de los' +
                                         'archivos del directorio de origen y los inserta' +
                                         ' en el header de los archivos del directorio de ' +
                                         'destino. Los nombres de los archivos remuestreados +'
                                         'deben tener el formato: nombre_archivo.resamp.fits')
parser.add_argument("-dirorig", action='store', default=PATHLIGHTS_BKG,
                    help=" Directorio de los archivos de imagenes de donde se van a" +
                    "leer las claves.")
parser.add_argument("-dirdest", action='store', default=RESAMPLED_EXT,
                    help=" Directorio con las imagenes remuestreadas .")
parser.add_argument("-keys", nargs='+', default='',
                    help=" Claves para insertar")
parser.add_argument("-af", nargs='+', default='DATE',
                    help=" Las claves se van a insertar arriba de esta clave.")

def insert_key(args):
    img_list_in = args.dirorig
    img_list_out = args.dirdest
    keys = args.keys
    af = args.af
    for image_in in os.listdir(img_list_in):
        image_out = os.path.join(img_list_out, os.path.splitext(image_in)[0] +
        '.resamp.fits')
        image_in = os.path.join(img_list_in, image_in)
        print (image_in)
        print (image_out)
        im_in = fits.open(image_in, mode="update")
        im_out = fits.open(image_out, mode="update")
        hdr_in = im_in[0].header
        hdr_out = im_out[0].header

        # Lee las claves del argumento keys de los archivos en dirorig y los inserta en
        # el header de los archivos de las imagenes remuestreadas de dirdest.
        for k in keys:
            KEY = hdr_in[k]
            if k not in hdr_out:
                print (KEY)
                hdr_out.insert(af, (k, KEY, " "))

    im_in.flush()
    im_in.close()
    im_out.flush()
```

```

        im_out.close()

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    insert_key(args)

if __name__ == "__main__":
    main()

```

calculate_airmass.py

```

#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import argparse
import os
import astropy.io.fits as fits
import sys
import datetime
from pyraf import iraf
from constants import *

parser = argparse.ArgumentParser(description='Calcula el airmass para cada imagen')
parser.add_argument("-dir", action='store', default=RESAMPLED_EXT,
                    help="Directorio donde residen las imagenes")
parser.add_argument("-obs", action='store', default='snorri',
                    help="Nombre del observatory (IRAF)")
parser.add_argument("-utk", action='store', default='DATE-OBS',
                    help="Fecha y hora UTM de la observacion")
parser.add_argument("-eq", action='store', default='EQUINOX',
                    help="Nombre clave para el valor del equinox")

def local_to_sidereal_time(date, time, observatory):
    year, month, day = date.split("-")
    hour, minute, second = time.split(":")
    time_hours = float(hour) + float(minute)/60. + float(second)/3600.
    iraf.asttimes(observatory=observatory, year=year, month=month, day=day,
                 time=time_hours)
    sidereal_time = iraf.asttimes.lmst
    return str(sidereal_time)

```

```

def get_airmass(args):
    img_path = args.dir
    img_list = os.listdir(img_path)
    observatory = args.obs
    raKey = 'CRVAL1'
    decKey = 'CRVAL2'
    UTKey = args.utk
    equinoxKey = args.eq

    for image in img_list :
        image = os.path.join(img_path, image)
        im = fits.open(image, mode="update")
        hdr = im[0].header

        # Lee el UTM del header
        UT = hdr[UTKey]
        print (UT)

        # Calcular hora local
        LTKey = 'LT'
        if LTKey not in hdr:
            LT = datetime.datetime.strptime(UT, "%Y-%m-%dT%H:%M:%S.%f") -
datetime.timedelta(hours=5)
            time = LT.time().isoformat()
            date = LT.date().isoformat()
            LT = LT.strftime("%Y-%m-%dT%H:%M:%S.%f")[:-3]
            hdr.set(LTKey, LT, " Local date-time ")
            print (LT)

        # Calcular la hora sidereal local con la hora local y poscion del observatorio.
        STKey = 'ST'
        if STKey not in hdr:
            ST = local_to_sidereal_time(date, time, observatory)
            hdr.set(STKey, ST, " Local sidereal time ")
            print (ST)

        if equinoxKey not in hdr:
            hdr.set(equinoxKey, 2000.0)

        im.flush()
        im.close()

        # Estimar el airmass:
        if 'AIRMASS' not in hdr:
            iraf.module.setairmass(images=image, observa=observatory, ra=raKey,
                                  dec=decKey, st=STKey, ut=UTKey, equinox=equinoxKey)

    return None

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    get_airmass(args)

```

```
if __name__ == "__main__":
    main()
```

create_zcat.py

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import argparse
import sys
import astromatic_wrapper as aw
from constants import *
from astropy.io import ascii
import os
import shutil
import xmatch as xm

parser = argparse.ArgumentParser(description='Genera los catalogos de las imagenes')
parser.add_argument("-dir", action='store', default=RESAMPLED_EXT,
                    help=" Directorio donde residen las imagenes")
parser.add_argument("-dirzp", action='store', default=TEMPZPCAT,
                    help=" Directorio donde van a recibir los catalogos")

def create_cat(args):
    dirzp = args.dirzp
    im_path = args.dir

    # Cuidado!!! Elimina el directorio de catalogos de existir.
    if os.path.exists(dirzp):
        shutil.rmtree(dirzp)
    os.mkdir(dirzp)

    for image in os.listdir(im_path):
        catalogo = os.path.splitext(image)[0] + '.cat'
        print (catalogo)
        files = {
            'image': im_path + image,
        }
        kwargs = {
            'code': 'SExtractor',
            'cmd': 'sextractor',
            'config': {
                # Cambiar los parametros de ser necesario.
                'CATALOG_TYPE': 'ASCII_HEAD',
                'CATALOG_NAME': os.path.join(dirzp, catalogo),
                'FILTER_NAME': SEXCONV,
                'MAG_ZEROPOINT': '0',
                'PIXEL_SCALE': '2.83',
```

```

        'SATUR_LEVEL': '65000.0',
        'GAIN_KEY': 'EGAIN',
        'SEEING_FWHM': '6.52',
        'DETECT_THRESH': '2',
        'ANALYSIS_THRESH': '2',
        'DETECT_MINAREA': '3',
        'CLEAN': 'Y',
        'CLEAN_PARAM': '1.0',
        'BACKPHOTO_TYPE': 'LOCAL',
    },
    'temp_path': '.',
    'params': ['NUMBER', 'ALPHA_J2000', 'DELTA_J2000', 'MAG_AUTO', 'MAGERR_AUTO',
               'FLUX_AUTO', 'FLUXERR_AUTO', 'FLAGS', 'CLASS_STAR']
}

sextractor = aw.api.Astromatic(**kwargs)
sextractor.run_frames(files['image'], frames=[1])
print ('*** FIN CREACION DE CATALOGOS ***')

'''
Referencia cruzada y filtrado de catalogos. Generacion de las hojas de calculo que se
van a unir en vsack_zcat.py para realizar el calculo del coeficiente de extinsion por
noche, en forma "manual" por ahora.
'''

for cat in os.listdir(dirzp):
    cat = os.path.join(dirzp, cat)
    print(cat)
    table_filtered = xm.xmatch(cat)
    ascii.write(table_filtered, os.path.splitext(cat)[0] + '.csv',
                overwrite=True, format='csv')

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    create_cat(args)

if __name__ == "__main__":
    main()

```

xmatch.py

```

#!/usr/bin/env python
# -*- coding: UTF-8 -*-
from astropy.io import ascii
from astropy import units as u

```

```

from astroquery.xmatch import XMatch

def xmatch(cat='test.cat'):
    data = ascii.read(cat, format='sextractor')

    # FILTRO PARA EL ZERO POINT
    data = data[(data['FLAGS'] == 0) & (data['CLASS_STAR'] >= 0.75) &
                (data['MAGERR_AUTO'] <= 0.1)]

    table = XMatch.query(cat1=data,
                        cat2='vizier:V/147/sdss12',
                        max_distance=5 * u.arcsec, colRA1='ALPHA_J2000',
                        colDec1='DELTA_J2000', selection='best')

    return table

```

vstack_zcat.py

```

#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import argparse
import sys
from constants import *
from astropy.io import ascii, fits
from astropy.table import vstack, Column
import os

parser = argparse.ArgumentParser(description='Genera los catalogos de las imagenes')
parser.add_argument("-dir", action='store', default=RESAMPLED_EXT,
                    help=" Directorio donde residen las imagenes")
parser.add_argument("-dirzp", action='store', default=TEMPZPCAT,
                    help=" Directorio donde residen los catalogos")
parser.add_argument("-obskey", action='store', default='DATE-OBS',
                    help=" Clave de la fecha y hora de observacion")
parser.add_argument("-xkey", action='store', default='AIRMASS',
                    help=" Clave del airmass")

def consolidar(args):
    dirzp = args.dirzp
    im_path = args.dir
    table_list = []
    date_obsKey = args.obskey
    XKey = args.xkey

    i = 0
    for cat in os.listdir(dirzp):
        airmass_value = []
        date_value = []

```

```

image = os.path.splitext(os.path.join(im_path, cat))[0] + '.fits'
im = fits.open(image)
hdr = im[0].header

if cat.endswith(".csv"):
    i = + 1
    cat = os.path.join(dirzp, cat)
    print ('CATALOGO: ' + cat)
    tableCat = ascii.read(cat, format='csv')
    for r in tableCat:
        airmass_value.append(hdr[XKey])
        date = hdr[date_obsKey].split('T')[0]
        date_value.append(date)
    col_a = Column(airmass_value, name=XKey)
    col_d = Column(date_value, name=date_obsKey)
    tableCat.add_column(col_a)
    tableCat.add_column(col_d)
    table_list.append(tableCat)

...
Se juntan una debajo de la otra todas las tablas (catalogos con formato csv)
del directorio catalogos_zp. El valor del airmass y la fecha para cada catalogo fue
agregado en el ciclo de arriba
...
table_full = vstack(table_list)

ascii.write(table_full, 'vstack_zcat_f' + '.csv', overwrite=True, format='csv')

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    consolidar(args)

if __name__ == "__main__":
    main()

```

insert_extinction.py

```

import argparse
import os
import sys
from astropy.io import fits, ascii
from constants import *

parser = argparse.ArgumentParser(description='Inserta en el encabezado de los archivos '
+
                                     'el valor del coeficiente de extinción de los archivos.')

```

```

parser.add_argument("-dir", action='store', default=RESAMPLED_EXT,
                    help=" Directorio donde residen las imagenes")
parser.add_argument("-extkey", action='store', default='EXTINCT',
                    help=" Clave del coeficiente de extincion")
parser.add_argument("-obskey", action='store', default='DATE-OBS',
                    help=" Clave del coeficiente de extincion")

def insert_ext(args):
    im_path = args.dir
    extKey = args.extkey
    date_obsKey = args.obskey

    """
    Ejemplo:
    La tabla de extinction_data.csv es asi (este es un ejemplo):
    date          ext_coeff
    2021-03-10   0.1245
    2021-03-11   0.1487
    2021-03-12   0.1032
    2021-03-13   0.1245
    2021-03-14   0.1487
    2021-03-15   0.1032
    2021-03-16   0.1245
    2021-03-17   0.1487
    """

    data = ascii.read('extinction_data.csv', format='csv')
    print(data)
    for image in os.listdir(im_path):
        image = os.path.join(im_path, image)
        im = fits.open(image, mode="update")
        hdr = im[0].header
        date_obs = hdr[date_obsKey]
        date = date_obs.split('T')[0]
        print ('-----')
        print(date)
        print (image)
        flag = 0
        for d in data:
            if date == d['date']:
                k = d['ext_coeff']
                flag = 1
                hdr.set(extKey, k, " Extinction coefficient (mags) ")
                im.flush()
                im.close()
                print (k)
                print ('-----')
                print
                break

    if flag is 0:
        print('Esta imagen no corresponde a ninguna fecha del archivo de extinciones')

```

```

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    insert_ext(args)

if __name__ == "__main__":
    main()
calculate_zeropoint.py

# !/usr/bin/env python
# -*- coding: UTF-8 -*-
import argparse
import sys
from constants import *
from astropy.io import ascii, fits
import numpy
import os
import shutil
import xmatch as xm

parser = argparse.ArgumentParser(description='Calcula el ZP para cada imagen y el ZP' +
                                         'promedio, la escala de flujo para cada imagen y guarda'
                                         +
                                         'los valores calculados en el header de los archivos.')
parser.add_argument("-dir", action='store', default=RESAMPLED_EXT,
                    help=" Directorio donde residen las imagenes")
parser.add_argument("-dirzp", action='store', default=TEMPZPCAT,
                    help=" Directorio donde residen los catalogos")
parser.add_argument("-extkey", action='store', default='EXTINCT',
                    help=" Clave para el coeficiente de extincion")
parser.add_argument("-xkey", action='store', default='AIRMASS',
                    help=" Clave del airmass")
parser.add_argument("-zpkey", action='store', default='MAGZPT',
                    help=" Clave para el zero point")
parser.add_argument("-fskey", action='store', default='FLXSCALE',
                    help=" Clave para la escala de flujo")
parser.add_argument("-fmag", action='store', default='rmag',
                    help=" Nombre del campo de la magnitud del filtro correspondiente" +
                    "del catalogo de referencia.")

def calculate_zp(args):
    im_path = args.dir
    cat_path = args.dirzp
    extKey = args.extkey
    XKey = args.xkey
    zpKey = args.zpkey
    fsKey = args.fskey
    f_mag = args.fmag

    ZPlist=[]
    Imlist = []

```

```

for cat in os.listdir(cat_path):

    if cat.split('.')[2] == 'cat':

        image = im_path + cat.split('.')[0] + '.resamp.fits'
        cat = cat_path + cat

        im = fits.open(image, mode="update")
        hdr = im[0].header
        if XKey in hdr:
            k = hdr[extKey]
            X = hdr[XKey]

        table = xm.xmatch(cat)
        table_filtered = table[table[f_mag] >= 12]

        # La formula es con extinction es: ZP = m_std - (m_int - kX)
        # ZP_star = table_filtered[f_mag] - (table_filtered['MAG_APER'] - k*X)

        ZP_star = table_filtered[f_mag] - (table_filtered['MAG_AUTO'])

        # eliminacion de outliers de zp de estrellas para cada imagen
        # con metodo de "2sigmas from the mean"
        elements = numpy.array(ZP_star)
        mean = numpy.mean(elements, axis=0)
        std = numpy.std(elements, axis=0)

        # final_list: lista de todos los zp por estrella
        final_list = [x for x in ZP_star if (x > mean - 2 * std)]
        final_list = [x for x in final_list if (x < mean + 2 * std)]

        ZP = numpy.median(final_list) # zp de la imagen

        if zpKey not in hdr:
            hdr.set(zpKey, ZP, " Photometric ZP (mags) for default extinction ")

        ZPlist.append(ZP) # guarda el zp de la imagen en la lista que va a contener
                        # todos los zps ( uno por imagen)
        Imlist.append(image)

        im.flush()
        im.close()

elements = numpy.array(ZPlist)
mean = numpy.mean(elements, axis=0)
sd = numpy.std(elements, axis=0)

# ZP_final_list: lista de todos los zp por imagen
ZP_final_list, Im_final_list = zip(*[(x, y) for (x, y) in zip(ZPlist, Imlist) if
                                     (x > mean - 2 * sd)])

ZP_final_list, Im_final_list = zip(*[(x, y) for (x, y) in
                                     zip(ZP_final_list, Im_final_list) if
                                     (x < mean + 2 * sd)])

```

```

elements = numpy.array(ZP_final_list)
ZP_average = numpy.mean(elements, axis=0)
ZP_median = numpy.median(elements, axis=0)

# se van a eliminar todas aquellas imagenes cuyo zp sea un outlier. Se mueven al
# directorio ZP_TRASH
for im in os.listdir(im_path):
    if os.path.join(im_path, im) not in Im_final_list:
        print(im)
        shutil.move(os.path.join(im_path, im), os.path.join(ZP_TRASH, im))

for image, ZP in zip(Im_final_list, ZP_final_list):
    im = fits.open(image, mode="update")
    hdr = im[0].header
    FS = 10 ** ((ZP - ZP_average) / (-2.5))
    hdr.set('FS', FS, "Relative flux scaling from ZP ")
    print('FS = ' + str(FS))
    im.flush()
    im.close()

print('{}: {}'.format('Los ZPs', ZP_final_list))
print('{}: {}'.format('ZP promedio', ZP_average))
print('{}: {}'.format('ZP mediana', ZP_median))
print("""
1. Se calculo el zeropoint y se guardo en el header
2. Se calculo el flux-scale y se guardo en el header
""")

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    calculate_zp(args)

if __name__ == "__main__":
    main()

```

insert_wieght.py

```

#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import argparse
import os
import sys
from pyraf import iraf
from astropy.io import fits, ascii
from constants import *

```

```

parser = argparse.ArgumentParser(description='Para cada imagen, obtiene el valor de ' +
'ponderación del mapa de peso y lo guarda en el encabezado. ')
parser.add_argument("-dir", action='store', default=RESAMPLED2,
                    help=" Directorio donde residen las imagenes y sus pesos luego" +
                    "de la segunda corrida con SWARP.")
parser.add_argument("-dirzp", action='store', default=RESAMPLED_EXT,
                    help=" Directorio donde residen las imagenes con el zero point")
parser.add_argument("-wkey", action='store', default='WEIGHT',
                    help=" Clave asignada para ponderacion de la imagen")
parser.add_argument("-zkey", action='store', default='MAGZPT',
                    help=" Clave del zero point")

def insert_w(args):
    im_path = args.dir
    im_z_path = args.dirzp
    wKey = args.wkey
    zKey = args.zkey
    for image_w in os.listdir(im_path):
        if image_w.split('.')[2] == 'weight':
            image_d = os.path.join(im_path, image_w.split('.')[0] + '.'
                                   + image_w.split('.')[1] + '.' + image_w.split('.')[3])
            image_z = os.path.join(im_z_path, image_w.split('.')[0] + '.resamp.fits')
            image_w = os.path.join(im_path, image_w)
            print (image_z)
            weight = iraf.imstat(images=image_w, fields='mean', format='no', Stdout=1)
            w = float(weight[0]) * 1000

            im_d = fits.open(image_d, mode="update")
            hdr = im_d[0].header

            im_z = fits.open(image_z)
            hdrz = im_z[0].header
            z = hdrz[zKey]

            hdr.set(wKey, w, "Image Weight for coadding")
            hdr.set(zKey, z, "Zero Point for coadding")

            im_d.flush()
            im_d.close()

    print ('***** FIN *****')

def main(arguments=None):
    if arguments is None:
        arguments = sys.argv[1:]
    args = parser.parse_args(arguments)
    insert_w(args)

if __name__ == "__main__":
    main()

```

constants.py

```
MASTERDARK = './masters/masterdark.fit'  
MASTERBIAS = './masters/masterbias.fit'  
MASTERFLAT = './masters/masterflat.fit'  
PATHFLATS = './flats_cal/'  
PATHBIAS = './bias_cal/'  
PATHDARKS = './darks_cal/'  
PATHLIGHTS = './lights_cal/'  
PATHLIGHTS_WCS = './lights_cal_wcs/'  
PATHLIGHTS_BKG = './lights_cal_wcs_b/'  
TEMPCAT = './catalogos/'  
TEMPZPCAT = './catalogos_zp/'  
TEMPLIST = 'temp.list'  
MODELIST = 'mode.list'  
DATABASES = './databases/'  
SEXCONV = 'filters/default.conv'  
REFCATPATH = './catalogos_ref/'  
RESAMPLED = './resampled/'  
RESAMPLED2 = './resampled2/'  
RESAMPLED_EXT = './resampled_extinction/'  
ZP_TRASH = './zp_trash/'
```

ANEXO 2

Logros dentro del Doctorado:

1. Logré conseguir financiamiento de tres proyectos de investigación, de los cuales por lo menos dos tienen gran impacto directo en la comunidad. Hemos desarrollado tres proyectos al respecto: Proyecto “*Sistemas solares fotovoltaicos autónomos para la electrificación de una comunidad de Besigó en la Comarca Ngöbe Buglé*”, Panamá: Breve repaso sobre tecnología fotovoltaica”, Proyecto “*Monitoreo de las condiciones climáticas y la eficiencia de un módulo pc-Si durante nueve meses (2018) ubicado en la Comunidad Nuevo Cascabel Comarca Ngäbe-Buglé*” y Proyecto, “*Sistema Novedoso de Óptica Refractiva Integrada (SNORRI)*”.

2. Como candidato al doctorado participé en actividades científicas (Física) en más de una Universidad Nacional y en más de una regional de Centroamérica y el Caribe. Soy profesor de la UNACHI /director de investigación) y estudiante de doctorado de la Universidad de Panamá. Además, realicé la maestría en colaboración con Cuba en un tema de astrofísica de gran impacto (y novedoso) porque se trata de un fenómeno del tipo caos determinista: “La entropía de Tsallis y los meteoros” con la asesoría del Dr. Oscar Sotolongo – Costa de la Universidad de la Habana, Cuba.

3. Participé en pasantías científicas y didácticas y congresos nacionales y regionales y realicé publicaciones en revistas nacionales y regionales indexadas:

Weigandt, P. & Weigandt, D. (2020). Matriz de súper telefotos para la detección de objetos astronómicos de bajo brillo superficial en Churuquita Chiquita de Coclé, Panamá. *Tecnociencia*, 22(1), 37-47,

Weigandt, P., Weigandt, D. & Márquez, A. (2021). SNORRI: Un proyecto de astrofísica en el seno de la comunidad coclesana. *Revista Científica Guacamaya*, 5(2), (2021), 19-29. <https://revistas.up.ac.pa/index.php/guacamaya/article/view/2053>

Sotolongo Costa, O., Gámez, R., Luzón, F., Posadas, A., Weigandt, P. (2008). Non extensivity in meteor showers. *Apeiron*. 15(2), 187 - 201.

Samudio, D., Weigandt, P., Toral B, J., & Sotolongo-Costa, O. (2014). Non-extensivity in seismic catalogs of Panamá. *Nova Scientia*, 7(13), 149–169. <http://novascientia.delasalle.edu.mx/ojs/index.php/Nova/article/view/119>

4. Este esfuerzo de investigación independiente en la Astronomía y la Astrofísica en Panamá permite contar con una matriz telescópica novedosa, para desarrollar investigación científica en el País.

Actividades realizadas para el doctorado como prerrequisito:

1. Títulos académicos

a- Licenciado en Matemática y Licenciado en Física de la Universidad de Panamá.

b-Maestría en Física. Magister en Física. Título de la tesis de maestría: “La entropía de Tsallis y los meteoros”.

2. Créditos de maestría y doctorado

-Todos los cursos de la Maestría en Física de la Universidad Autónoma de Chiriquí, créditos del Doctorado en Física de la Universidad de Panamá, créditos de cursos de la Universidad de La Habana.

3. Proyectos de la SENACYT y UNACHI

Investigador principal y coordinador de la UNACHI:

- Matriz de súper telefotos para la detección de corrientes de marea en galaxias cercanas (FID16-010) PROGRAMA DE FOMENTO A LA INVESTIGACIÓN Y DESARROLLO. UNACHI

- Sistemas solares fotovoltaicos autónomos para la electrificación de una comunidad de Besigó en la Comarca Ngbe Buglé (CVP11-017)

- Proyecto “*Monitoreo de las condiciones climáticas y la eficiencia de un módulo pc-Si durante nueve meses (2018) ubicado en la Comunidad Nuevo Cascabel Comarca Ngäbe-Buglé*”

4. Participación en escuelas, congresos, talleres y estadías, a nivel nacional y regional durante el doctorado.

- Estadía científica en UP-Estación RN50.

- Actividades de divulgación realizadas en el Centro de Investigación de Física Aplicada, agosto 06 de 2019. Matriz de súper telefoto para la detección de corrientes de marea en galaxias cercanas. Tipo de Participación: Orador. Sistema Novedoso de Óptica Refractiva Integrada. Tipo de Participación: Afiche.

- XXVIII Congreso Científico Nacional Santiago de Veraguas, octubre 02 a octubre 06 de 2017. Aplicaciones civiles y científicas de la Estación RN50. Tipo de participación: Afiche y Sistema de electrificación solar en una comunidad de difícil acceso Tipo de participación: Orador.

- THIRD ANDEAN SCHOOL ON NUCLEAR PHYSICS Bogotá, Colombia, julio 24 de 2017 – julio 28 de 2017. The RN50 station in Panama of the Comprehensive Nuclear-Test-Ban Treaty (CTBT). Tipo de participación: Afiche

- Congreso Nacional de Ciencias Espaciales. Santiago de Veraguas, mayo 13 de 2016 - mayo 14 de 2016. Física no extensiva en la lluvia de meteoros. Tipo de participación: Orador.

- Taller de Energías Renovables Tegucigalpa, Honduras, enero 21 de 2016 - enero 22 de 2016. Sistemas solares fotovoltaicos autónomos para la electrificación de unam, comunidad de Besigó en la comarca Ngöbe Buglé. Tipo de participación: Orador y participante.

- Taller de Enseñanza de la Física Tegucigalpa, Honduras, enero 18 de 2016 - enero 19 de 2016. Sistemas solares fotovoltaicos autónomos para la electrificación de una comunidad de Besigó en la comarca Ngöbe Buglé Tipo de participación: Orador y participante.

- VI School on Cosmic Rays and Astrophysics, Science for Development ICTP-UNACH-UNESCO Seminar Chiapas, México, noviembre 17 de 2015 - noviembre 25 de 2015. Tipo de participación: participante

- Primera Escuela de Física Fundamental Centroamericana Guatemala, Guatemala, octubre 26 de 2015 - octubre 30 de 2015. Tipo de participación: Participante

- FOURTH INTERNATIONAL DATA CENTRE TRAINING COURSE FOR NATIONAL DATA CENTRE TECHNICAL STAFF. Vienna, Austria. 2002. Participante.

5. Extensión y divulgación del Sistema Novedoso de Óptica Refractiva Integrada (SNORRI)

a. Sitio Web del proyecto SNORRI. (Weigandt, s.f). <https://snorri.space>

b. Imagen de portada del proyecto SNORRI en la Revista Guacamaya de la Universidad de Panamá, en su Volumen 5, Num. 2 de abril a septiembre del 2021. <https://revistas.up.ac.pa/index.php/guacamaya/issue/view/310>

c. Visita al proyecto de Miembros de la Unión Astronómica Internacional: el Profesor Eduardo Chung de la Facultad de Ciencias Naturales, Exactas y Tecnología de la Universidad de Panamá y el Profesor y director de Estudios Generales de la Universidad Marítima Internacional de Panamá.

d. El Siglo: Periódico de cobertura nacional publicó el 9 de septiembre del 2019, en la sección de PROVINCIAS, el artículo titulado “Astrofísicos panameños desarrollan un proyecto para estudiar las galaxias” donde destacan que se busca “recolectar datos y reforzar los conocimientos astronómicos en el país” (Saldaña, 2019).

<http://elsiglo.com.pa/panama/astrofisicos-panamenos-desarrollan-proyecto-para-estudiar-galaxias/24138400>

e. La Estrella de Panamá: diario de circulación nacional publicó el 9 de septiembre del 2019, el artículo “Investigadores panameños trabajan para recolectar datos de las galaxias” donde puntualizan que “el equipo está diseñado para detectar estructuras de bajo brillo superficial en el espacio exterior” (Ábrego, 2019).

<https://www.laestrella.com.pa/nacional/190909/datos-galaxias-panamenos-recolectar-investigadores>

f. Telemetro: canal de televisión de difusión nacional, transmitió, el 8 de enero del 2020, en su programa Telemetro Reporta edición matutina y estelar, el segmento “Mentes Brillantes” con un reportaje en Churuquita Chiquita en donde se entrevista a los hermanos Weigandt y se presenta a SNORRI (Muñoz, 2020).

<https://www.telemetro.com/reportajes/2020/01/08/mentes-brillantes-pablo-y-daniel-hermanos-cientificos/2416288.html>

g. TVN: Televisora nacional presentó, el 21 de diciembre del 2020, en su programa matutino y en la edición estelar de TVN noticias, un reportaje con los investigadores y responsables de SNORRI, acerca del fenómeno denominado “estrella de belén” (Reyes, 2020). https://www.tvn-2.com/videos/noticias/lunes-observara-fenomeno-estrella-Belen_2_5744445504.html

6. Publicaciones científicas

a. Weigandt, P., Weigandt, D. & Márquez, A. (2021). SNORRI: Un proyecto de astrofísica en el seno de la comunidad coclesana. *Revista Científica Guacamaya*, 5(2), 19-29. <https://revistas.up.ac.pa/index.php/guacamaya/article/view/2053>

b. Weigandt, P. & Weigandt, D. (2020). Matriz de súper telefotos para la detección de objetos astronómicos de bajo brillo superficial en Churuquita Chiquita de Coclé, Panamá. *Tecnociencia*. 22(1), 37-47.

<https://revistas.up.ac.pa/index.php/tecnociencia/article/view/1087>

c. Sotolongo Costa, O., Gámez, R., Luzón, F., Posadas, A., Weigandt, P. (2008). Non extensivity in meteor showers. *Apeiron*. 15(2), 187 - 201.

<http://redshift.vif.com/JournalFiles/V15NO2PDF/V15N2SOT.pdf>

d. Samudio, D., Weigandt, P., Toral B, J., & Sotolongo-Costa, O. (2014). Non-extensivity in seismic catalogs of Panamá. *Nova Scientia*, 7(13), 149–169.

<http://novascientia.delasalle.edu.mx/ojs/index.php/Nova/article/view/119>

7. Asesoría de tesis sobre un radiotelescopio y experiencia como Director de Investigación y Documentación Científica en la Vicerrectoría de Investigación y Posgrado de la UNACHI.