

Python e interfaz gráfica para calcular el *Seeing* en el Observatorio Astronómico Nacional

Python and graphical interface to calculate the Seeing at the National Astronomical Observatory

Jaidary Mejía ¹
Giuliat Navas  ²

Universidad Nacional Abierta, Trujillo, Venezuela¹

Fundación Centro de Investigaciones de Astronomía Francisco J. Duarte, Mérida, Venezuela^{1,2}

jaidarymejia30@gmail.com¹
giuliatnavas@gmail.com²

Fecha de recepción: 26/01/2025

Fecha de aceptación: 23/05/2025

Pág: 37 – 59

DOI: 10.5281/zenodo.18163310

Resumen

La observación astronómica requiere recolectar la luz proveniente de los astros. Sin embargo, cuando estas observaciones se realizan desde la superficie terrestre, la atmósfera actúa como un factor distorsionador de la luz, ya que está compuesta por capas de aire con diferentes densidades y temperaturas. Cuando la luz de un astro atraviesa estas capas, se dobla y dispersa, resultando en imágenes distorsionadas. Este efecto distorsionador de la atmósfera sobre los astros se conoce como *Seeing*, y su valor representa la calidad de la visibilidad atmosférica en un momento y lugar determinado. Este trabajo propone un código en Python, junto con su respectiva interfaz gráfica, para calcular el valor del *Seeing*. Se usa Python como herramienta gracias a su alto nivel en el procesamiento de datos y sus importantes bibliotecas especializadas en Astronomía. El método del FWHM se utiliza para determinar el *Seeing* en la escala de Pickering y se aplica a observaciones de abril del año 2024, debido al volumen de datos tomados en las diferentes fases lunares durante el mes. Gracias a la interfaz propuesta, los operadores de los telescopios podrían emplearla como herramienta de trabajo diario, permitiéndoles un conocimiento más profundo y cuantitativo de la calidad del cielo.



Esta obra está bajo licencia CC BY-NC-SA 4.0.

Este código permitiría evaluar y analizar en tiempo real la calidad de las observaciones astronómicas, facilitando la toma de decisiones sobre la viabilidad y ejecución de los proyectos planificados, además de cuantificar el valor del *Seeing* en las imágenes astronómicas.

Palabras clave: astronomía con Python, atmósfera, cálculo del *Seeing*, interfaz gráfica, refracción.

Abstract

Astronomical observation requires collecting light from celestial objects. However, when these observations are made from the Earth's surface, the atmosphere acts as a distorting factor for the light, as it is composed of layers of air with varying densities and temperatures. When light from a celestial object passes through these layers, it bends and scatters, resulting in distorted images. This distorting effect of the atmosphere on celestial objects is known as *Seeing*, and its value represents the quality of atmospheric visibility at a given time and location. This work proposes a Python code, along with its graphical interface, to calculate the *Seeing* value. Python is used as a tool due to its high capability in data processing and its important specialized libraries for Astronomy. The *Full Width at Half Maximum* (FWHM) method is employed to determine the *Seeing* on the Pickering scale and is applied to observations from April 2024, given the volume of data collected during different lunar phases that month. Thanks to the proposed interface, telescope operators could use it as a daily work tool, enabling a deeper and more quantitative understanding of sky quality. This code would allow real-time evaluation and analysis of the quality of astronomical observations, facilitating decision-making regarding the feasibility and execution of planned projects, as well as quantifying the *Seeing* value in astronomical images.

Keywords: astronomy with Python, atmosphere, calculation of *Seeing*, graphical interface, refraction.

Introducción

La astronomía, como disciplina observacional, se dedica al estudio de todos los objetos celestes más allá de los límites de nuestro planeta. Su fase inicial consiste en la recolección de la mayor cantidad de luz emitida por objetos luminosos, la cual es captada mediante telescopios y cámaras CCD (CCD, Charge Coupled device). Estos objetos luminosos pueden emitir o reflejar luz por diversas razones, dependiendo de su naturaleza. Algunos emiten luz propia debido a reacciones de fusión nuclear que ocurren en su núcleo (Enciclopedia Concepto, 2022). Otros, aunque no tienen luz propia, reflejan la luz de las estrellas a

las cuales orbitan. Algunos de estos objetos también pueden emitir o dispersar la luz bajo circunstancias particulares, como la sublimación de sus materiales o la ionización causada por estrellas cercanas (Zuñiga, 2019). Este complejo espectro de fenómenos convierte la luz en el principal vehículo para entender la composición, estructura y dinámica del universo.

La observación astronómica requiere recolectar esta luz proveniente de los astros, mediante receptores ubicados generalmente en la superficie terrestre. Este proceso abarca desde nuestros ojos y telescopios que captan la radiación electromagnética en el rango visible (longitudes de onda entre 400 y 700 nanómetros), hasta antenas que detectan frecuencias de radio (longitudes de onda superiores a 1 centímetro y de varios kilómetros de largo). Esta radiación proveniente de cualquier astro debe atravesar la atmósfera terrestre antes de llegar al instrumento o al ojo del observador en la superficie terrestre. Debido a que la luz cruza la atmósfera, el camino óptico de esta señal se modifica por el fenómeno de la refracción (Fernández et al., 2023, p. 197).

La refracción es el cambio en la dirección de propagación de una onda electromagnética al atravesar dos medios homogéneos con distintos índices de refracción (Malacara, 2015, p. 34; Miczaika y Sinton, 1967, p. 15). Dado que la densidad del aire disminuye a medida que se incrementa la altura respecto a la superficie terrestre, es usual tratar a la atmósfera como un conjunto de capas esféricas, donde cada capa posee sus propias características físicas (Fernández et al., 2023, p. 43; Smart, 1962, p. 60), de manera que, la refracción depende, entre otras cosas, de las características físicas de la atmósfera y de la altitud sobre el horizonte o distancia cenital del cuerpo observado siendo ésta más pronunciada cuando los objetos a observar están cerca del horizonte terrestre, porque el rayo de luz atraviesa una mayor cantidad de atmósfera (Abad et al., 2002).

Una noche nublada puede bloquear por completo la visión de los astros desde la superficie terrestre. Las nubes no solo actúan como obstáculos visuales, sino que también dispersan la luz, especialmente la proveniente de fuentes artificiales (contaminación lumínica), lo cual puede alterar la visibilidad del cielo y dificultar la observación. Las partículas presentes en la atmósfera, como el polvo y el vapor de agua, dispersan la luz emitida por las ciudades, incrementando la luminosidad del cielo nocturno y dificultando la observación de astros tenues (Innovations Foresight, 2016). Estos efectos combinados con la refracción atmosférica, la turbulencia atmosférica, la dispersión de la luz y la absorción atmosférica (Pastrana, 2013) pueden alterar el brillo, el color y la posición aparente de los objetos celestes en el cielo observados desde telescopios terrestres, interferencias que no experimentan los telescopios ubicados fuera de la atmósfera terrestre, tal como el Telescopio Espacial Hubble, James Webb, Gaia, entre otros.

La turbulencia en la atmósfera es causada por el movimiento del aire y las variaciones de la temperatura. La atmósfera de la Tierra está compuesta de capas de aire con diferentes densidades y temperaturas, que están en constante movimiento debido a factores como el

viento y las corrientes de aire. Cuando la luz de un cuerpo celeste atraviesa estas capas turbulentas, su trayectoria se desvía de manera irregular, causando variaciones en la posición del cuerpo y en su brillo. Desde la superficie de la Tierra, estas desviaciones producen un efecto visual en el que los cuerpos celestes parecen titilar. Este efecto visual varía según el tipo de cuerpo celeste. Las estrellas, al ser puntos de luz muy distantes y pequeños en el cielo, su luz es más susceptible a las variaciones causadas por la atmósfera terrestre, lo que las hace ver como titilar. En cambio, los planetas y los asteroides, al tener un tamaño aparente mayor a las estrellas, no se ven titilar tanto como las estrellas, la luz que reflejan atraviesa la atmósfera en un área más amplia, lo que promedia las desviaciones y reduce el efecto de titilación. Cuando la luz pasa a través de diferentes capas de aire con distintas temperaturas y densidades, hace que la luz se doble y se disperse, resultando imágenes borrosas y distorsionadas (Sentinelmission, 2024). Este efecto distorsionador de la atmósfera sobre los objetos celestes es lo que llamamos el *Seeing*, y su valor representa la calidad de visión o visibilidad atmosférica en un instante o en un lugar determinado (Astropractica, 2023).

En términos prácticos, el *Seeing* se ve reflejado en el titilar de las estrellas. En buenas condiciones de *Seeing*, las estrellas no deberían titilar. Por el contrario, en una mala noche, el titilar de las estrellas puede ser muy violento e incluso afectar su tonalidad. A través de los telescopios terrestres, el grado de turbulencia es todavía más evidente. Las imágenes de las estrellas se mostrarán temblorosas y será difícil enfocarlas en noches con un valor de *Seeing* mayor. De manera que, el valor del *Seeing* nos proporciona información sobre la calidad de las observaciones.

Conocer el valor del *Seeing* en una noche de observación en los telescopios del Observatorio Astronómico Nacional (OAN) es crucial para determinar si las condiciones son adecuadas para continuar con las observaciones. Este conocimiento permite a los investigadores evaluar la calidad del cielo y, por lo tanto, la calidad de los datos que se pueden obtener. Mantener un registro detallado del *Seeing* a lo largo del año es una herramienta invaluable para la planificación de proyectos y observaciones a largo plazo en el observatorio, información de la que no disponemos actualmente, ya que en el OAN no contamos con un método para calcular el *Seeing*, lo que impide registrar sus valores correspondientes. De allí radica la importancia de este estudio en la que se propone un código en Python con su respectiva interfaz gráfica para calcular el *Seeing* como herramienta de trabajo para ser empleada por los asistentes científicos en el OAN (operadores de los telescopios). Esta herramienta facilitaría, primeramente, el cálculo del valor del *Seeing* en cualquier momento de la observación y en cualquier proyecto, permitiéndoles monitorear sus variaciones en el lugar, asegurando así, que los proyectos se desarrollen en las mejores condiciones atmosféricas posibles y contribuyendo significativamente a la calidad de los datos recolectados.

Desarrollo

El *Seeing*

El *Seeing* determina la mejor resolución angular posible en unas condiciones dadas. En observatorios a gran altitud (superiores a 4000 msnm), como Mauna Kea en Hawái o la Palma en España, pueden alcanzar valores de 0,4'' (segundos de arco), mientras en observatorios a baja altitud (inferiores a 2000 msnm) es común que el *Seeing* supere valores de 1''. Actualmente, gracias a este estudio se registró en el OAN un valor del *Seeing* entre 1,9'' y 3,8'', pero hace 30 años atrás, el valor del *Seeing* en el OAN llegó a valores inferiores a 1''. Otros observatorios, como el Instituto de Astrofísica de Canarias tiene un *Seeing* promedio de 0,7'' (IAC, 2024), y en la República Checa de 5'' (Kolář, 2015). De manera que, conocer el valor del *Seeing* es fundamental para los astrónomos, ya que pueden seleccionar las mejores noches para realizar observaciones detalladas, por ejemplo para estudios de fotometría, y las noches con mayor valor de *Seeing* destinarlos a proyectos menos sensibles, tales como al estudio de las posiciones astrométricas de los astros, astrofotografía, entre otros.

El requisito principal para las observaciones astronómicas de alta resolución es operar bajo una atmósfera lo más tranquila y estable ópticamente posible, caracterizada por una turbulencia (óptica) débil o de pequeña escala (Parada et al., 2020). En consecuencia, se determina que calcular el *Seeing* en los observatorios es crucial, puesto que afecta directamente la nitidez de las imágenes astronómicas. Un mal *Seeing* hace que las imágenes sean menos claras y más difíciles de enfocar, lo que puede eliminar detalles sutiles en planetas, asteroides y en estrellas dobles.

Actualmente, existen dos escalas para medir la calidad del *Seeing* en una noche, la escala de Antoniadi, diseñada por el astrónomo francés Antoniadi y la escala de Pickering ideada por William H. Pickering. Ambas son similares y se usan indistintamente. La escala de Antoniadi divide la visión en cinco grados, y va del I (perfecto valor del *Seeing*) al V (péssimo valor del *Seeing*), y la escala de Pickering divide la visión en diez grados, y va del 1 (buen valor) al 10 (péssimo valor) (Sarazin y Roddier, 1990).

Tomando en cuenta los valores del *Seeing*, es crucial seleccionar sitios idóneos para la instalación de telescopios ópticos terrestres teniendo en cuenta el *Seeing* natural del lugar y una baja contaminación lumínica. Los observatorios de Mauna Kea en Hawái (Racine, 1996), el Observatorio del Roque de los Muchachos en La Palma, Islas Canarias (Radu et al., 2011) y el Observatorio Paranal en Chile (Lombardi et al., 2009) son ejemplos de ubicaciones con excelente *Seeing* natural. La estratificación de la atmósfera y su efecto en el *Seeing* son generalmente bien conocidos, aunque cada sitio presenta ligeras diferencias debido a la topografía natural (o construida) y clima particular. Esto ha impulsado la práctica de realizar campañas de prueba de sitios antes de comisionar un telescopio en un determinado lugar, con el fin de determinar primeramente las estadísticas de turbulencia atmosférica en el sitio.

(Skidmore et al., 2009).

El sector de Llano del Hato en Mérida, donde se encuentra el OAN fue considerado por el Dr. Jurgen Stock (Fundador del OAN) como uno de los mejores lugares en Venezuela para comisionar los telescopios que están actualmente, debido a varias razones. Su ubicación a una altitud de 3600 metros sobre el nivel del mar (m.s.n.m), reduce la cantidad de atmósfera que la luz debe atravesar antes de llegar a los telescopios, mejorando así la calidad de las observaciones. Además, cuenta con un clima favorable, muchas noches despejadas durante el año, baja humedad y baja contaminación lumínica, lo cual es crucial para obtener imágenes claras y precisas.

Cálculo del *Seeing*

Cuando se analiza una estrella en una imagen astronómica, se observa que está representada por un grupo de píxeles, algunos más brillantes cerca del centro y otros de menor intensidad a su alrededor. Las imágenes ideales de fuentes puntuales obtenidas por un instrumento óptico tienen un patrón de intensidad llamado disco de Airy (American Association of Variable Star Observers, 2010). Sin embargo, la luz de una estrella al pasar por la atmósfera da lugar a una distribución de intensidad no uniforme. Para medir la intensidad de una imagen que no tiene los bordes bien definidos, se emplea el FWHM, o “anchura a media altura” (*Full Width at Half Maximum*, por sus siglas en inglés), para describir la anchura de una señal en el punto donde su amplitud es la mitad de su valor máximo (Figura 1).

En astronomía, el FWHM es relevante para describir la calidad de una imagen astronómica, y es el método que aplicaremos en este estudio para calcular el valor del *Seeing* en la escala de Pickering. El FWHM se refiere a la anchura de la estrella en el punto donde la intensidad de la luz cae a la mitad de su valor máximo. Un menor valor de FWHM indica que las estrellas son más puntuales y la imagen es más nítida y, por lo tanto mejor calidad de observación (Astropractica, 2023). En otras palabras, el *Seeing* se mide como la mejor resolución angular posible, y se obtiene al expresar en segundos de arco el valor de la anchura a media altura (FWHM) en píxeles. Este concepto es crucial y es el que usaremos en este trabajo para cuantificar la estabilidad y claridad del cielo observado.

En el marco de este trabajo, las observaciones utilizadas para probar el código y la interfaz del cálculo del *Seeing* fueron tomadas del Telescopio Reflector de 1 metro de diámetro, ubicado en el OAN ($8^{\circ}47'11''N$, $70^{\circ}52'18,8''O$, 3,600 msnm). Las imágenes usadas fueron suministradas por la CATO (Comité de Asignación de Tiempo de Observación) respetando la propiedad de las observaciones y el derecho moral de los investigadores principales de los proyectos (Navas, 2022). Para esas observaciones, al telescopio se le acopló una cámara CCD, modelo FLI PL4240 256 en su configuración F/5,2 permitiéndole a la luz recorrer una distancia focal de 5.150 mm, con una anchura del pixel de 13,5 μm y cubriendo un campo de visión de $19' \times 19'$ (minutos de

arco).

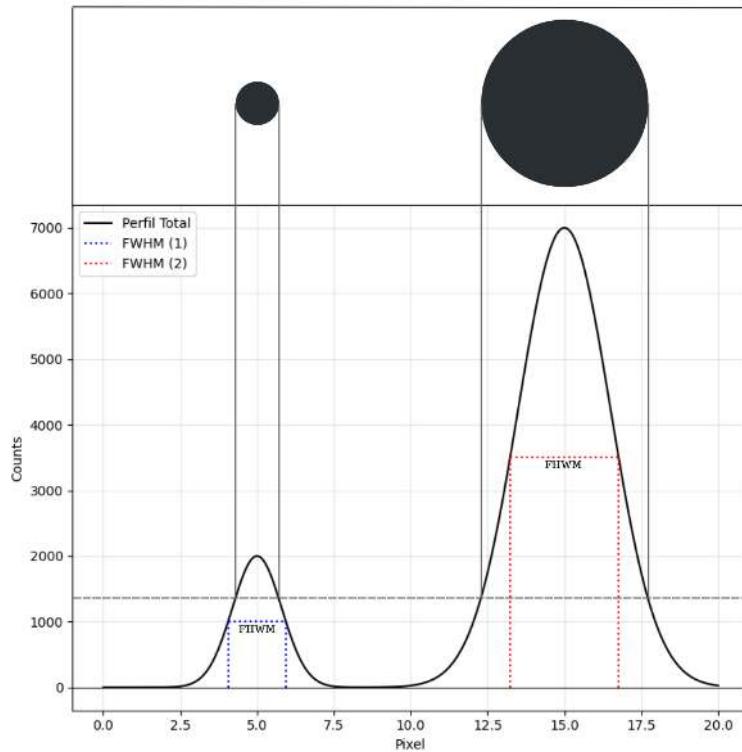


Figura 1: Esquema que representa el FWHM, se observa la función donde el eje horizontal se expresa en pixeles, mientras el eje vertical se expresa en unidades de cuentas. Se puede apreciar que la estrella 1 es de menor tamaño que la estrella 2 y sus FWHM son diferentes.

Fuente: Elaboración propia (2025)

Metodología para calcular el *Seeing*

La metodología que se propone en este trabajo automatiza el procesamiento de imágenes astronómicas en formato FITS (*Flexible Image Transport System*), permitiendo identificar estrellas en el campo observado, calcular los contornos de las estrellas de acuerdo a las cuentas (valores ADU) y ajustar los perfiles gaussianos a través del FWHM para estimar el valor del *Seeing*.

Python fue elegido como la herramienta principal debido a su flexibilidad, extensas bibliotecas científicas y capacidad para manejar datos complejos. Las bibliotecas utilizadas en el desarrollo del código incluyen: Astropy: para manejar archivos FITS y unidades astronómicas, Numpy: para cálculos numéricos y manipulación de matrices, Matplotlib: para la visualización de datos y resultados, Scipy: para realizar los ajustes gaussianos y el análisis funcional, Photutils: para la fotometría y la detección de objetos astronómicos. Estas bibliotecas permiten

calcular con precisión las curvas de luz necesarias para determinar el valor del *Seeing*. El método desarrollado en este trabajo comprende las siguientes etapas implementadas en las secciones del código descrito en la Tabla 1.

Definición de la ubicación y lectura de los datos

El análisis comienza con la definición de la ruta en donde se encuentran almacenados las imágenes astronómicas y la manera de leerlos, tal como se detalla en Mejía y Navas (2024). Estos archivos fits contienen una matriz bidimensional que representa los valores de cuentas (en ADU), junto con una cabecera (header) que incluye información clave sobre la configuración del telescopio. Esta cabecera o encabezado proporciona datos como la escala de los píxeles (PIXSCAL), la fecha de observación, las coordenadas ecuatoriales del campo observado (RA, DEC), el tiempo de exposición y el binning de la cámara (CCDSUM), entre otros parámetros. Cada archivo fits se procesa en Python empleando la biblioteca (Astropy Project, [s.f.-b](#)), que facilita la manipulación de imágenes astronómicas y sus metadatos.

Pre-procesamiento

El valor de la escala de píxeles se ajusta dinámicamente en función del binning del sensor CCD (el cual se lee en la cabecera de los datos), multiplicando la escala base por los factores de binning aplicados en las direcciones horizontal y vertical del sensor. Las coordenadas de las estrellas a analizar se ingresan manualmente, después de haber sido identificadas previamente mediante herramientas de visualización como SAOImage DS9 (Harvard-Smithsonian Center for Astrophysics, [s.f.](#)). Estas coordenadas a introducir corresponde a la posición central (x,y) de cada estrella en píxeles.

Estimación del radio óptimo de apertura

Se estima un radio de apertura óptimo para cada estrella en el campo de observación. Este radio maximiza la desviación estándar del flujo dentro de una región circular, identificando la zona donde el flujo estelar es más significativo. El rango explorado para el radio en este trabajo abarca de 3 a 25 píxeles, y se detiene automáticamente cuando la desviación estándar comienza a decrecer.

Cálculo del FWHM

Para cada estrella, el código define una Región de Interés (ROI) centrada en sus coordenadas centrales (x,y) y ajustada según el radio óptimo. Posteriormente, modela el perfil de flujo utilizando una función Gaussiana 1D ajustada con el método de mínimos cuadrados de Levenberg-Marquardt (LevMarLSQFitter) (Astropy Project, [s.f.-a](#)). A partir del ajuste Gaussiano modelado, se calcula el ancho total a media altura (FWHM) en píxeles, el cual se convierte a segundos de arco utilizando la escala de píxeles ajustada.

Análisis de los resultados

El análisis de los resultados en el código, se basa en el estudio de los parámetros calculados para cada estrella identificada de forma individual, tomando en cuenta las estadísticas fundamentales como el flujo total, la media y la desviación estándar del flujo. Como resultado, el código muestra el radio óptimo de apertura de cada estrella seleccionada y el valor del *Seeing* estimado en segundos de arco, además, calcula y muestra el valor promedio del *Seeing* para la imagen astronómica estudiada a partir de los valores individuales de *Seeing* estimados de todas las estrellas analizadas en la imagen.

Generación de gráficos y reportes

El análisis visual mediante gráficos generados por el código cumple un rol crucial en la interpretación y validación de los resultados obtenidos en el cálculo del *Seeing*. Los perfiles de flujo y los ajustes gaussianos permiten evaluar la distribución de la luz emitida por cada estrella y verificar la precisión del modelo gaussiano aplicado en las coordenadas ingresadas. Este análisis gráfico es fundamental para identificar desviaciones significativas o anomalías en el perfil de flujo, que podrían afectar la medición del *Seeing* al no estar la estrella debidamente centrada dentro del ROI. Además, proporcionan una referencia visual inmediata que ayuda a validar el ajuste y asegura que los datos sean representativos de la calidad óptica y atmosférica en el momento de la observación.

Los mapas de intensidad de las regiones de interés (ROI) ofrecen una representación espacial del flujo en torno a cada estrella. Estos gráficos son especialmente útiles para identificar problemas como la contaminación de fuentes cercanas, ruido de fondo o características específicas del sensor que puedan influir en los resultados. Al visualizar directamente la distribución de luz en el área analizada, el investigador puede detectar patrones inesperados y corregir posibles errores en los cálculos o en la selección de parámetros.

Finalmente, los contornos isofotales aportan una perspectiva más detallada de la simetría y distribución del flujo alrededor de las estrellas. En conjunto, estos gráficos no solo complementan los datos numéricos, sino que también ofrecen una herramienta poderosa para validar y enriquecer el análisis, garantizando conclusiones más precisas y confiables.

Almacenamiento de resultados

El código permite guardar en un archivo de texto los resultados, detallando el número total de estrellas analizadas, los parámetros calculados para cada estrella y el promedio estimado del valor del *Seeing* para dicha observación. Esto facilita al asistente científico del OAN el registro de los valores del *Seeing* durante las noches de observación.

Tabla 1: Código propuesto para calcular el *Seeing* en una imagen .fit.

```

#LIBRERIAS A USAR
from astropy.io import fits
import numpy as np
import matplotlib.pyplot as plt
from photutils.aperture import CircularAperture, ApertureStats
from astropy.modeling import models, fitting
from scipy.signal import find_peaks, peak_widths
from matplotlib.colors import LogNorm
from scipy.ndimage import gaussian_filter
from datetime import datetime
# 1. DEFINICION DE LA RUTA DE LA IMAGEN ASTRONÓMICA
file = fits.open('/content/C2021S3_2X2_V_30s_007.fit')
imag = file[0].data
header = file[0].header
# 2. PROCESO PARA AJUSTAR PIXSCAL AL CCD DEL TELESCOPIO
ccdsu = header['CCDSUM']                                # Obtener el valor del binning
binning_x, binning_y = map(int, ccdsu.split())          # Convertir a enteros8
pixscale = header['PIXSCAL'] * binning_x                # Ajustar escala de píxeles
# Obtener la fecha del estudio (en formato YYYY-MM-DD)
fecha_estudio = header.get('DATE-OBS', datetime.now().strftime('%Y-%m-%d'))
# 3. PROCESO PARA INGRESAR LAS COORDENADAS (X,Y) DEL CENTRO DE LAS ESTRELLAS
def ingresar_coordenadas_estrellas():
    num_estrellas = int(input("¿Cuántas estrellas desea ingresar? "))
    print("Escribe las coordenadas de las estrellas previamente identificadas en SAOImage:")
    coordenadas = [(float(input(f"X de estrella {i + 1}: ")), float(input(f"Y de estrella {i + 1}: "))) for i in range(num_estrellas)]
    return coordenadas
coordenadas_estrellas = ingresar_coordenadas_estrellas()
# 4. PROCESO PARA DEFINIR EL RADIO DE APERTURA SEGÚN LA DESVIACIÓN ESTÁNDAR DEL FLUJO
min_radius, max_radius = 3, 25
def calcular_radio_optimo(x_star, y_star, imag):
    fluxes = []
    max_std_flux, optimal_radius = -float('inf'), None
    for radius in range(min_radius, max_radius + 1):
        aper = CircularAperture([x_star, y_star], r=radius)
        stats = ApertureStats(imag, aper)
        fluxes[radius] = {'flux': stats.sum, 'mean_flux': stats.mean, 'std_flux': stats.std}
        if stats.std > max_std_flux:
            max_std_flux = stats.std
            optimal_radius = radius
        elif stats.std < max_std_flux:                                         # Si comienza a decrecer, detener
            break
    return optimal_radius, fluxes
# 5. CÁLCULO DEL MÉTODO FWHM
def calcular_Seeing(x_star, y_star, optimal_radius, imag):
    x_min, x_max = int(x_star - optimal_radius), int(x_star + optimal_radius)
    y_min, y_max = int(y_star - optimal_radius), int(y_star + optimal_radius)
    # Extraer la región de la estrella
    star_region = imag[y_min:y_max, x_min:x_max]
    # Calcular la fila central de la estrella (pasando por y_star)
    fila_central = int(y_star - y_min)                                     # Coordenada relativa a la ROI
    x = np.arange(star_region.shape[1])                                      # Posiciones en X dentro de la ROI
    y = star_region[fila_central, :]                                         # Flujo en la fila central (perfil 1D)

```

Continúa en la siguiente página...

... Viene de la página anterior

Tabla 2: Código propuesto para calcular el *Seeing* en una imagen .fit.

```

# Ajuste Gaussiano inicial
g_init = models.Gaussian1D(amplitude=np.max(y) - np.min(y), mean=np.mean(x), stddev=np.std(x),
                            bounds={'amplitude': (0, np.max(y) - np.min(y)), 'stddev': (0.5, 1.6)})
fitter = fitting.LevMarLSQFitter()
g = fitter(g_init, x, y - np.min(y))
# Escalar el ajuste gaussiano para igualar el máximo de los datos reales
peaks, _ = find_peaks(g(x))
results_half = peak_widths(g(x), peaks, rel_height=0.5)
fwhm_pixels = results_half[0][0]
fwhm_arcsec = fwhm_pixels * pixscale
g.level = np.min(y)
return fwhm_arcsec, star_region, x, y, lambda x: g(x) + np.min(y)

# 6. PROCESAMIENTO DE CADA ESTRELLA
resultados = {}
for x_star, y_star in coordenadas_estrellas:
    optimal_radius, fluxes = calcular_radio_optimo(x_star, y_star, imag)
    if optimal_radius:
        # Calculamos el Seeing y obtenemos la región de la estrella
        fwhm_arcsec, star_region, x, y, g = calcular_Seeing(x_star, y_star, optimal_radius, imag)
        # Ajuste dinámico del tamaño de la ROI en función de optimal_radius
        roi_size = int(optimal_radius * 1.5)          # Multiplicamos el radio óptimo por 1.5 (o el valor que consideres adecuado)
        # Definir los límites de la ROI
        x_min, x_max = int(x_star - roi_size), int(x_star + roi_size)
        y_min, y_max = int(y_star - roi_size), int(y_star + roi_size)
        # Extraer la región de la estrella con el nuevo tamaño dinámico de la ROI
        star_region = imag[y_min:y_max, x_min:x_max]
        # Almacenar los resultados de esta estrella
        resultados[(x_star, y_star)] = {
            'optimal_radius': optimal_radius,
            'flux': fluxes[optimal_radius]['flux'],
            'mean_flux': fluxes[optimal_radius]['mean_flux'],
            'std_flux': fluxes[optimal_radius]['std_flux'],
            'fwhm_arcsec': fwhm_arcsec,
            'star_region': star_region,
            'x': x,
            'y': y,
            'g': g
        }
    }

# 7. IMPRESIÓN DEL VALOR DEL Seeing DE CADA ESTRELLA Y EL VALOR PROMEDIO DE LA IMAGEN
print("\nSeeing de cada estrella:")
for (x_star, y_star), stats in resultados.items():
    print(f"Estrella ({x_star:.2f}, {y_star:.2f}): Seeing: {stats['fwhm_arcsec']:.2f} arcsec")
promedio_Seeing = np.mean([res['fwhm_arcsec'] for res in resultados.values()])
print(f"\nPromedio del Seeing de la imagen: {promedio_Seeing:.2f} arcsec")
nombre_imagen = header['FILENAME']

# 8. PROCESO PARA GUARDAR LOS RESULTADOS EN UN ARCHIVO TXT
def guardar_resultados_txt(resultados, promedio_Seeing, nombre_imagen):
    nombre_archivo = f"Seeing_{nombre_imagen}.txt"
    with open(nombre_archivo, 'w') as f:
        f.write(f"Número de estrellas: len(resultados)\n\n")

```

Continúa en la siguiente página...

... Viene de la página anterior

Tabla 3: Código propuesto para calcular el *Seeing* en una imagen .fit.

```

for i, ((x_star, y_star), stats) in enumerate(resultados.items(), 1):
    f.write(f"Estrella {i}:\n")
    f.write(f"Coordenadas: ({x_star:.2f}, {y_star:.2f})\n")
    f.write(f"Radio óptimo: {stats['optimal_radius']} píxeles\n")
    f.write(f"Flujo: {stats['flux']:.2f}, Media: {stats['mean_flux']:.2f}, Desviación estándar: {stats['std_flux']:.2f}\n")
    f.write(f"Seeing: {stats['fwhm_arcsec']:.2f} arcsec\n\n")
f.write(f"Promedio del Seeing de la imagen: {promedio_Seeing:.2f} arcsec\n")
print(f"\nResultados guardados en {nombre_archivo}")
# Llamada para guardar los resultados
guardar_resultados_txt(resultados, promedio_Seeing, nombre_imagen)
# 9. OPCIÓN PARA VER DETALLES COMPLETOS DE LOS RESULTADOS
ver_info_completa = input("\nDesea ver la información completa de cada estrella? (s/n): ").strip().lower()
if ver_info_completa == 's':
    for (x_star, y_star), stats in resultados.items():
        print(f"\nEstrella ({x_star:.2f}, {y_star:.2f}):")
        print(f"Radio óptimo: {stats['optimal_radius']} píxeles")
        print(f"Flujo: {stats['flux']:.2f}, Media: {stats['mean_flux']:.2f}, Desviación estándar: {stats['std_flux']:.2f}")
        # Mostrar la región de interés y el ajuste gaussiano
        plt.figure(figsize=(16, 6))
        # Región de la estrella
        plt.subplot(1, 2, 1)
        plt.imshow(stats['star_region'], cmap='gray', origin='lower')
        plt.colorbar()
        plt.title('Región de la Estrella')
        # Perfil de flujo y ajuste
        plt.subplot(1, 2, 2)
        plt.plot(stats['x'], stats['y'], 'b.', label='Datos reales')
        plt.plot(stats['x'], stats['g'](stats['x']), 'r-', label='Ajuste Gaussiano')
        plt.title('Perfil de Flujo y Ajuste Gaussiano')
        plt.xlabel('Píxel')
        plt.ylabel('Flujo')
        plt.legend()
        plt.show()
    # Función para imprimir niveles y valores de contornos ISO
def imprimir_niveles_contornos(contours):
    print("\nNiveles de Contornos ISO y Valores:")
    for i, level in enumerate(contours.levels):
        # Extraer los valores de los contornos para el nivel actual
        print(f"Nivel {i + 1} - Valor: {level:.2f}")
    # Crear una región de interés (ROI) alrededor de la estrella
    roi_size = 15 # Ajustar según el tamaño de la estrella (pixeles)
    x_min, x_max = int(x_star - roi_size), int(x_star + roi_size)
    y_min, y_max = int(y_star - roi_size), int(y_star + roi_size)
    star_region = imag[y_min:y_max, x_min:x_max]

```

Continúa en la siguiente página...

... Viene de la página anterior

Tabla 4: Código propuesto para calcular el *Seeing* en una imagen .fit.

```
# Suavizar la imagen para mejor detección de isofotas
smoothed_image = gaussian_filter(star_region, sigma=2.0)
# Crear una figura para la visualización
plt.figure(figsize=(8, 8))
plt.imshow(star_region, cmap='gray', norm=LogNorm(), origin='lower')
# Calcular los contornos ISO (isofotas)
contour_levels = np.linspace(np.min(smoothed_image), np.max(smoothed_image), 8)
contours = plt.contour(smoothed_image, levels=contour_levels, colors='red', linewidths=1, origin='lower')
# Añadir título y etiquetas
plt.clabel(contours, inline=True, fontsize=7, fmt='%.1f')
plt.title(f'Contornos ISO de la estrella: {x_star, y_star} ')
plt.xlabel('X (pixeles)')
plt.ylabel('Y (pixeles)')
# Mostrar la imagen con los contornos ISO
plt.show()
# Llamar a la función para imprimir los niveles y valores de los contornos
imprimir_niveles_contornos(contours)
else:
    print("Fin del programa.")
```

Fuente: Elaboración propia (2025).

Se pueden mencionar algunas consideraciones técnicas en el código a tener en cuenta, tal como es la resolución dinámica, donde los tamaños de las Regiones de Interés (ROI) y los radios de apertura se ajustan automáticamente para optimizar los cálculos según las características específicas de cada estrella. Además, se aplica un filtro Gaussiano para mejorar la detección de contornos isofotales, lo que facilita una mejor identificación de los detalles en las imágenes. El código también ofrece flexibilidad al permitir ajustes dinámicos en los parámetros de entrada, como la cantidad de estrellas y los radios explorados, proporcionando una mayor adaptabilidad a diferentes escenarios.

El código incluye un módulo adicional para registrar y analizar promedios de *Seeing* durante períodos de meses. Este módulo permite ingresar valores manuales y generar gráficos de tendencias, facilitando estudios estadísticos a largo plazo. El código está descrito en la Tabla 5.

Tabla 5: Código para calcular el valor del *Seeing* por mes y registrar los resultados.

```

import numpy as np
import matplotlib.pyplot as plt
# Solicitar la cantidad de meses
num_meses = int(input("¿Cuántos meses va a estudiar? "))
# Solicitar la cantidad de datos por mes
num_datos = int(input("¿Cuántos datos de Seeing desea ingresar por mes? "))
# Inicializar una lista para almacenar los promedios de cada mes
promedios_Seeing = []
# Bucle para cada mes
for mes in range(1, num_meses + 1):
    # Inicializar una lista para almacenar los datos de Seeing
    datos_Seeing = []
    # Bucle para ingresar los datos de Seeing
    for dato in range(1, num_datos + 1):
        Seeing = float(input(f"Ingrese el valor del Seeing {dato} para el mes {mes}: "))
        datos_Seeing.append(Seeing)
    # Calcular el promedio de Seeing para el mes actual
    promedio = np.mean(datos_Seeing)
    promedios_Seeing.append(promedio)
    # Imprimir los promedios de Seeing para cada mes
    for mes, promedio in enumerate(promedios_Seeing, start=1):
        print(f"\nEl promedio de Seeing para el mes {mes} es: {promedio:.2f}")
    # Crear una gráfica con los resultados
    meses = list(range(1, num_meses + 1))
    plt.figure(figsize=(16, 6))
    plt.plot(meses, promedios_Seeing, marker='o', linestyle='-', color='b')
    plt.xlabel('Mes')
    plt.ylabel('Promedio de Seeing')
    plt.title('Promedio de Seeing por Mes')
    plt.xticks(meses) # Asegurar que solo se muestren números enteros en el eje x
    plt.grid(True)
    plt.show()

```

Fuente: Elaboración propia (2025).

Interfaz gráfica para usar el código

Para la aplicación práctica de este código y con el fin de ser usado por los asistentes científicos del OAN, se ha desarrollado una interfaz gráfica mediante una aplicación web usando Python. Esta interfaz facilita el análisis de datos para los usuarios sin experiencia previa en programación. Las instrucciones para ejecutar el código son las siguientes:

1. Descargar la carpeta “Interfaz _ *Seeing*” y guárdalo en la computadora.
2. Asegurarse de tener las imágenes .fits a analizar en una carpeta de fácil acceso.

3. Abrir el ejecutable “Calculadora_de_Seeing.exe”. Se despliega la interfaz de usuario.
4. Clickear en el botón “Cargar imagen” y seleccionar la imagen .fits a analizar
5. Introducir el número de estrellas que deseas tomar en cuenta para el cálculo del *Seeing*.
6. Proporcionar las coordenadas (x,y) centrales de cada estrella a analizar de la imagen astronómica, las cuales deben previamente identificar con *SaoImageDS9*.
7. Una vez que los datos estén completos, puedes optar por visualizar los resultados detallados para cada estrella o cerrar la aplicación.

Los datos se almacenan en el archivo resultados_Seeing.db, ubicado en la misma carpeta que el ejecutable, lo que permite su acceso posterior. Para consultar el código de la interfaz gráfica, se ha proporcionado un enlace a una carpeta compartida en Google Drive. Dentro de esta carpeta, es necesario descargar el archivo index.py y ejecutarlo en Visual Studio Code (VSCode). Se recomienda crear un entorno virtual nuevo e instalar en él las librerías requeridas para la ejecución del código. Además, dentro de la carpeta se incluye un script que detalla, paso a paso, los comandos necesarios para ejecutar el código en VSCode en sistemas operativos Windows. El enlace de acceso es el siguiente: <https://drive.google.com/drive/folders/1vHx-KJd1yEFYV4eRLciRwyqTMWTym1X6?usp=sharing>

Aplicación del código

Tras presentar el método para calcular el *Seeing*, procedemos aplicarlo a diversas imágenes astronómicas captadas en el OAN durante Abril de 2024. Este mes fue elegido para evaluar el método en un conjunto de datos, ya que, a pesar de sus condiciones atmosféricas desfavorables, se logró recopilar un volumen significativo de imágenes que abarcan distintas fases lunares lo que lo hizo interesante para este estudio. En futuras investigaciones será esencial estudiar la variabilidad del *Seeing* en otros meses para determinar sus valores con mayor precisión.

Las observaciones empleadas en este trabajo son de diversas noches, considerando diferentes fases lunares y utilizando varios filtros fotométricos. Las imágenes usadas fueron suministradas por la CATO (Comité de Asignación de Tiempo de Observación) respetando la propiedad de las observaciones y el derecho moral de los investigadores de los proyectos (Navas, 2022). El procesamiento se llevó a cabo utilizando herramientas especializadas como SAOimageDS9 y entornos interactivos como Google Colab Notebooks donde se desarrolló el código.

La Tabla 3 presenta un resumen de las observaciones realizadas en abril del año 2024, detallando parámetros esenciales como la fecha de observación, el nombre de las imágenes analizadas, las coordenadas centrales (x,y) de las estrellas seleccionadas, el valor del *Seeing* promedio de cada imagen en segundos de arco según nuestro código, y el *Seeing* de referencia

obtenido a través del software Astrométrica (Raab, 2025). También se incluyen las condiciones lunares, y los filtros utilizados. Este análisis aborda imágenes con presencia de tanto cometas como asteroides, utilizando los filtros R o V y técnicas específicas según las características empleadas por el investigador. La inclusión de las observaciones con diferentes fases lunares y filtros permite contextualizar los valores de *Seeing* obtenidos, asegurando el análisis de diversas condiciones atmosféricas que deben ser consideradas en la interpretación de los resultados.

Tabla 6: Cálculo del *Seeing* en observaciones del mes de abril del año 2024 del OAN.

Fecha	Imagen	Estrella 1	Estrella2	Estrella3	Seeing	SeeingR	Fase Lunar	Filtro
06-04-2024	62P_R_120s_001	(1064,1203)	(399,1621)	(1822,765)	2,07	2,075	Menguante 9.9 % de iluminación	R
06-04-2024	62P_R_120s_007	(1051,1185)	(386,1603)	(1809,747)	2,06	2,075	Menguante 9.9 % de iluminación	R
06-04-2024	62P_R_120s_014	(1044,1174)	(379,1592)	(1801,736)	2,09	2,075	Menguante 9.9 % de iluminación	R
07-04-2024	25869_R_120s_006_A	(1186,1005)	(302,1593)	(1925,849)	2,10	2,15	Menguante 3.8 % de iluminación	R
07-04-2024	25869_R_120s_011_A	(1175,988)	(290,1577)	(1914,833)	2,08	2,075	Menguante 3.8 % de iluminación	R
12-04-2024	C2023A3_2X2_V_60s_001	(138,336)	(897,480)	(598,707)	3,61	3,62	Creciente 8.1 % de iluminación	V
12-04-2024	C2023A3_2X2_V_60s_015	(138,336)	(897,480)	(598,707)	3,77	3,8	Creciente 8.1 % de iluminación	V
12-04-2024	C2023A3_2X2_V_60s_035	(138,336)	(897,480)	(598,707)	3,64	3,7	Creciente 8.1 % de iluminación	V
13-04-2024	25869_R_120s_011_B1	(917,1084)	(100,809)	(1344,1590)	2,08	2,1	Creciente 24.1 % de iluminación	R
13-04-2024	25869_R_120s_011_B3	(907,1097)	(91,822)	(1335,1603)	2,11	2,125	Creciente 24.1 % de iluminación	R
15-04-2024	25869_R_120s_002_C	(1170,1084)	(1935,1692)	(376,600)	2,10	2,125	Creciente 43.6 % de iluminación	R
15-04-2024	25869_R_120s_006_C	(1177,1078)	(1942,1686)	(383,595)	2,08	2,1	Creciente 43.6 % de iluminación	R
15-04-2024	25869_R_120s_010_C	(1187,1073)	(1951,1680)	(393,589)	2,08	2,1	Creciente 43.6 % de iluminación	R

Fuente: Elaboración propia (2025).

Los resultados obtenidos sobre el cálculo del *Seeing* mediante el programa desarrollado en este trabajo demuestran una notable concordancia con los valores proporcionados por el software Astrométrica (Raab, 2025), tal como se evidencia en la Tabla 6. Durante las observaciones realizadas en abril de 2024, los valores calculados por el programa muestran una pequeña desviación respecto al *Seeing* de referencia de Astrométrica, con diferencias que oscilan entre +0,02 y +0,06 segundos de arco. Esta proximidad en los resultados no solo refuerza la veracidad y confiabilidad del programa que se propone en este trabajo, sino que también valida su capacidad para reproducir mediciones precisas y consistentes en diferentes condiciones astronómicas, posicionándolo como una herramienta confiable y gratuita a ser utilizada para monitorear el cielo en observaciones científicas, educativas o divulgativas.

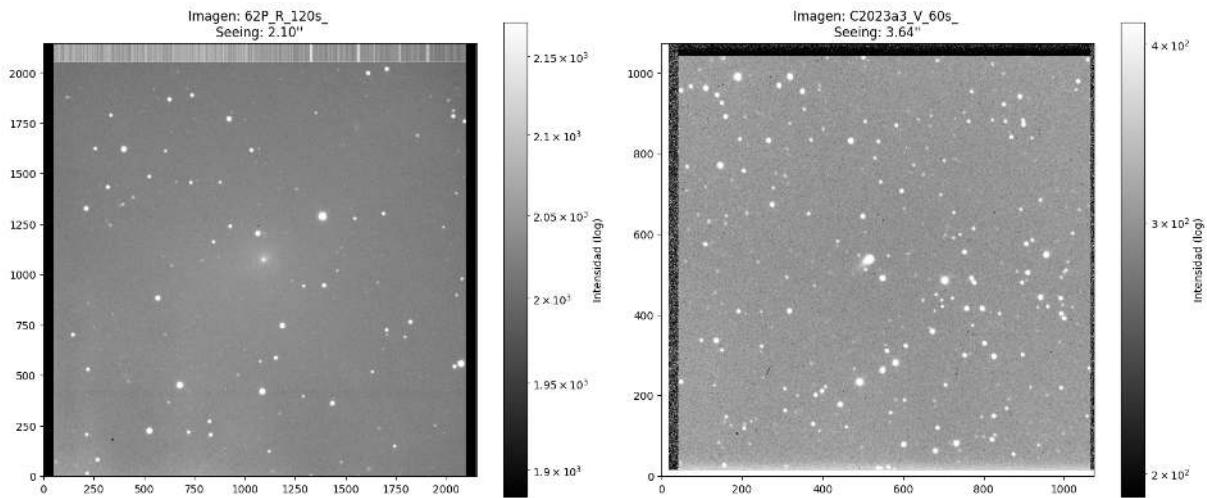


Figura 2: Izquierda: imagen fit del cometa 62P con un *Seeing* de $2,10''$. Derecha: imagen .fit del cometa C2023 A3 con un *Seeing* de $3,64''$.

Fuente: Elaboración propia (2025)

A partir de los datos obtenidos en la Tabla 6, se puede concluir que los valores promedios de *Seeing* durante el mes de abril en las imágenes del filtro V son de $3,67''$ y las del filtro R son de $2,08''$. Los altos valores de *Seeing* en ambos filtros se deben a las variaciones en las condiciones atmosféricas registradas en el OAN durante el mes de abril, que es mayoritariamente nublado y lluvioso. Además, estos valores de *Seeing* revelan que las estrellas, asteroides y/o cometas presentes en las imágenes analizadas están un poco distorsionadas tal como se puede ver en la Figura 2. En donde la calidad de los astros se ven mejor, más puntuales y más nítidos en la imagen con menor valor de *Seeing* que en la imagen con un *Seeing* de $3,64$ segundos de arco. Un menor valor de FWHM indica que los astros en las imágenes son más puntuales, la imagen es más nítida y, por lo tanto, la observación es de mejor calidad. El valor óptimo y esperado para observar los astros es con un *Seeing* inferior a 1 segundo de arco, sin embargo, ese valor es difícil alcanzarlo durante el mes de abril. Por otra parte, aunque la iluminación lunar afecta todas las observaciones en la Tabla 6, el filtro R (rojo) es generalmente menos sensible a los efectos de la atmósfera y a la contaminación lumínica, en comparación con el filtro V (visual). Esto convierte al filtro rojo en una opción a considerar en presencia de condiciones atmosféricas variables, como alta iluminación lunar o alta nubosidad.

La sensibilidad a las diferentes longitudes de onda (y, por lo tanto, a los filtros) varía según la interacción de las ondas de luz con la atmósfera. El filtro R cubre la parte roja del espectro visible, que son longitudes de onda más largas, entre 600 a 700 nanómetros (nm). Estas longitudes de onda son menos susceptibles a la dispersión y absorción de la luz por la atmósfera en comparación con longitudes de onda más cortas. Esto significa que, en general, las imágenes obtenidas con el filtro R son menos afectadas por la turbulencia atmosférica, lo que reduce el impacto del *Seeing* en las imágenes (LibreTexts, s.f.). Por otro lado, el filtro V, que abarca

la banda verde del espectro visible, aproximadamente entre 500 a 600 nm, es más sensible a la dispersión de la luz y turbulencia atmosférica lo que puede causar un mayor desenfoque de las estrellas en las imágenes astronómicas y dificultar la obtención de mediciones precisas. Por lo tanto, si el asistente científico requiere tomar imágenes con el filtro V en un proyecto determinado, debe verificar que el valor del *Seeing* en esa noche sea lo suficientemente bueno para lograr resultados científicos interesantes. De lo contrario, recomendarle al investigador el uso del filtro R para mejorar los valores del *Seeing* en las observaciones, o cambiar de proyecto.

La aplicación de este código permite un análisis sistemático y reproducible del cálculo del *Seeing* a cualquier imagen .fit a lo largo del tiempo, contribuyendo así a la evaluación y al monitoreo de la calidad del cielo para las planificaciones de observaciones astronómicas durante el año. La Figura 3 muestra uno de los gráficos visuales generados por este código, como parte de las metodologías aplicadas en el análisis de los resultados obtenidos para la estrella 1 en la imagen 62P_R_120s_001.fit. Se observan dos paneles que analizan el flujo de luz de la estrella 1 en una imagen 62P_R_120s_001.fit. En el panel izquierdo se observa una representación en escala de grises de la región de interés alrededor de la estrella 1. El brillo más intenso (en el centro) indica la ubicación de la estrella 1, donde la mayor parte de la luz está concentrada, mientras que las zonas más oscuras reflejan áreas con menor intensidad del flujo de luz. La barra de color al lado derecho del panel muestra la escala de valores del flujo, donde los tonos claros representan valores altos y los oscuros valores bajos. En el panel derecho se ilustra el perfil de flujo y su correspondiente ajuste gaussiano a la estrella 1. Los puntos azules representan los datos reales del flujo de luz de la estrella a lo largo del eje de las x. La línea roja corresponde al ajuste gaussiano, que modela matemáticamente cómo se dispersa la luz de la estrella 1. Esta curva sigue una forma de campana característica que refleja la distribución del flujo, afectada por las condiciones atmosféricas y las propiedades ópticas del telescopio.

Discusiones

En este contexto, el estudio propuesto resalta la importancia de contar con herramientas eficientes para medir el *Seeing* y su variabilidad, ya que estos parámetros pueden influir en la elección de filtros o técnicas de observación, así como en el diseño de experimentos astronómicos. Conocer el *Seeing* permite a los astrónomos optimizar los tiempos de observación y seleccionar las mejores condiciones para la toma de datos, mejorando la precisión de las mediciones y facilitando la identificación de eventos astronómicos relevantes.

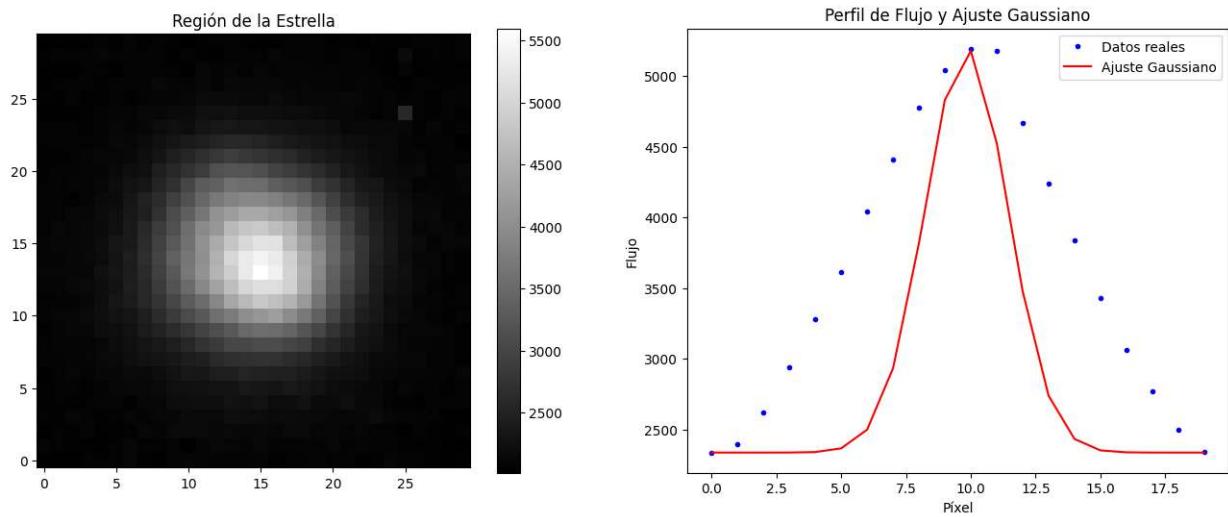


Figura 3: Descripción gráfica de la Región de Interés (ROI) (Izquierda) y ajuste gaussiano (derecha) de la estrella 1 de la imagen 62P_R_120s_001.fit. Estos gráficos, generados mediante las bibliotecas de Python, sirven como herramientas clave para el análisis y la visualización de resultados, los cuales forman parte del enfoque propuesto en este trabajo para presentar la información, a través del uso de la interfaz gráfica.

Fuente: Elaboración propia (2025)

El método propuesto en este estudio, basado en el análisis de perfiles de flujo y la medición del FWHM (*Full Width at Half Maximum*) mostró valores muy similares al del programa Astrométrica y, ha demostrado ser eficiente para calcular el *Seeing* de manera automatizada. Este enfoque no solo proporciona una medición precisa, sino que también permite su implementación en observatorios que operan con grandes volúmenes de datos. Al utilizar Python como plataforma para este análisis, se facilita la integración con otros sistemas de análisis de imágenes y se simplifica el proceso de cálculo, reduciendo la posibilidad de errores humanos.

Un aspecto que a menudo se pasa por alto, pero que puede afectar significativamente la calidad de las observaciones, es el *Seeing* de la cúpula del observatorio. Recientemente, (Munro et al., 2023) sugirieron que el *Seeing* de la cúpula es un factor determinante que debe ser monitorizado, ya que puede contribuir a la degradación de la calidad de las imágenes, especialmente en observaciones de alta resolución. Las condiciones dentro de la cúpula pueden ser influidas por factores como la temperatura interna, la humedad y las vibraciones estructurales, en el caso del OAN, no se emplea aire acondicionado en la cúpula, de manera que este tipo de fluctuaciones térmicas no afectan las observaciones de nuestros valores de *Seeing*. Por otra parte, este estudio no ha considerado explícitamente el impacto del *Seeing* de la cúpula, es relevante mencionar que las vibraciones de la estructura de la cúpula pudiesen tener efectos negativos sobre la calidad de las observaciones, al generar pequeñas perturbaciones

en la imagen que afectan la medición del *Seeing*. Este es un aspecto que podría ser objeto de futuras investigaciones, para determinar si las vibraciones de la cúpula del OAN están relacionadas con fluctuaciones en la calidad del *Seeing* en las imágenes obtenidas. En este sentido, sería útil explorar posibles métodos para mitigar este efecto, como la estabilización de la estructura de la cúpula o el uso de técnicas avanzadas de filtrado para reducir la influencia de las vibraciones en las mediciones astronómicas.

El uso de Python como herramienta para el análisis del *Seeing* ofrece múltiples ventajas, tanto en términos de eficiencia como de adaptabilidad. Este enfoque no solo es útil para mejorar la calidad de las observaciones en el OAN, sino que también puede ser adaptado a otros observatorios, incluso aquellos con diferentes configuraciones instrumentales o condiciones atmosféricas. Además, la capacidad de realizar cálculos automáticos y repetitivos es fundamental para observatorios que necesitan procesar grandes volúmenes de datos de manera rápida y precisa. Una posible extensión de este trabajo sería integrar el análisis del *Seeing* con otros parámetros de calidad del cielo, como la transparencia atmosférica o la contaminación lumínica, lo que permitiría obtener una evaluación más completa de las condiciones de observación. También sería interesante realizar un análisis de variabilidad de los valores del *Seeing* durante el año y así determinar los mejores meses para programar las observaciones.

Python es un lenguaje versátil con bibliotecas especializadas que optimizan el análisis de datos astronómicos, facilitando cálculos preciso como el *Seeing*. Su comunidad activa impulsa mejoras constantes, asegurando acceso a técnicas avanzadas. A diferencia de otros lenguajes de programación, su sintaxis clara permite desarrollar código libremente, sin restricciones en la manipulación de imágenes. Además, su capacidad para analizar datos, graficarlos y generar interfaces interactivas lo hacen una interesante opción para estudios astronómicos.

Conclusiones

El presente trabajo desarrolló un método automatizado para el cálculo del *Seeing* en imágenes astronómicas en formato fits, integrando herramientas avanzadas de procesamiento de datos mediante Python. Este enfoque permitió realizar mediciones precisas y sistemáticas, contribuyendo al monitoreo de la calidad del cielo en diferentes contextos de observación. Los resultados obtenidos destacan la eficiencia del método propuesto, ya que automatiza el análisis de las imágenes astronómicas mediante la identificación de estrellas, el cálculo de contornos isofotales y el ajuste de perfiles gaussianos para determinar el ancho total a media altura (FWHM). Estos procesos, implementados de manera reproducible, eliminan la subjetividad de métodos manuales tradicionales.

La adaptabilidad del método frente a variaciones en las condiciones de observación se evidenció mediante el ajuste dinámico de parámetros, como el radio de apertura óptimo y la

escala de píxeles. Estos ajustes garantizan la robustez del análisis ante distintas configuraciones instrumentales y condiciones atmosféricas. Asimismo, el análisis del impacto de los filtros fotométricos confirmó que el filtro R (rojo) es menos sensible a los efectos de la dispersión atmosférica, lo que lo posiciona como una opción confiable en condiciones de iluminación lunar elevada.

El uso de representaciones gráficas, como mapas de intensidad y ajustes gaussianos, fue esencial para la validación de los resultados. Estas herramientas visuales no solo complementaron los datos numéricos, sino que también facilitaron la identificación de anomalías, mejorando la precisión y confiabilidad de las conclusiones. En relación con los filtros empleados, los resultados mostraron que las imágenes procesadas con el filtro rojo presentaron valores de *Seeing* más consistentes, resaltando su ventaja frente al filtro verde (V), que mostró mayor sensibilidad a las condiciones atmosféricas y de iluminación lunar.

El método propuesto demostró su relevancia en la observación de cuerpos menores del Sistema Solar, como asteroides y cometas. La capacidad para adaptar las configuraciones instrumentales según las propiedades de brillo y movimiento de estos objetos permitió obtener datos precisos y representativos. Además, el análisis del *Seeing* en diferentes fechas y fases lunares estableció una base sólida para comprender las variaciones en la calidad del cielo nocturno a lo largo del tiempo.

El uso de Python como herramienta principal en el desarrollo del método destacó por su flexibilidad, amplio ecosistema de bibliotecas científicas y su capacidad para manejar datos complejos de manera eficiente. Comparado con otros lenguajes, Python ofrece una integración más fluida entre diferentes etapas del análisis, desde el manejo de archivos .fits con Astropy hasta los ajustes gaussianos precisos utilizando Scipy. Además, su comunidad activa y su enfoque en la accesibilidad permiten desarrollar métodos avanzados sin requerir un nivel extremo de especialización en programación. La combinación de estas características posiciona a Python como una opción ideal para aplicaciones astronómicas, ofreciendo una ventaja competitiva frente a otros lenguajes más rígidos o menos especializados en tareas científicas. En este contexto, su utilidad se evidencia en la capacidad de implementar un flujo de trabajo automatizado y reproducible, facilitando análisis complejos y fortaleciendo el alcance de los estudios astronómicos modernos.

Referencias

- Abad, A., Docobo, J., y Elipe, A. (2002). *Curso de Astronomía* (1era. ed.) Prensas Universitarias de Zaragoza.

- American Association of Variable Star Observers. (2010). *Guía de fotometría CCD: Capítulo 3.* https://www.aavso.org/sites/default/files/publications_files/ccd_photometry_guide/CCDPhotometryGuide-Spanish/SpanishPhotometryGuide-Chapter3.pdf
- Astropractica. (2023). *SEEING y FWHM: Escala para condiciones de visibilidad atmosférica y procedimiento para medirla.* Astropractica. <https://www.astropractica.org/tem3/seeing/seeing.htm>
- Astropy Project. (s.f.-a). *Astropy modeling: Introduction.* Astropy Documentation. <https://docs.astropy.org/en/stable/modeling/>
- Astropy Project. (s.f.-b). *FITS File Handling (astropy.io.fits).* *Astropy v5.0.* Astropy Documentation. <https://docs.astropy.org/en/stable/io/fits/>
- Enciclopedia Concepto. (2022). *Cuerpos celestes.* <https://concepto.de/cuerpos-celestes/>
- Fernández, L., De Biasi, M., y De Elía, G. (2023). *Astronomía Esférica. Fundamentos de Astrometría. Libros de Cátedra.* Editorial de la UNLP. Universidad Nacional de la Plata.
- Harvard-Smithsonian Center for Astrophysics. (s.f.). *SAOImage DS9 Documentation.* LibreTexts. <https://sites.google.com/cfa.harvard.edu/saoimageds9/documentation>
- IAC. (2024). *Seeing.* Instituto de Astrofísica de Canarias. <https://www.iac.es/en/observatorios-de-canarias/sky-quality/sky-quality-parameters/seeing>
- Innovations Foresight. (2016). *Introduction to astronomical seeing.* Innovations Foresight. <https://www.innovationsforesight.com/education/astronomical-seeing-tutorial/>
- Kolář, J. (2015). *How to measure seeing.* European Association for Astronomy Education (EAAE). https://eaae-astronomy.org/images/projects/catch-a-star/2015/18_How_to_measure_seeing.pdf
- LibreTexts. (s.f.). *Astronomical spectra, filters, and magnitudes.* LibreTexts. https://phys.libretexts.org/Bookshelves/Astronomy__Cosmology/Supplemental_Modules_%28Astronomy_and_Cosmology%29/Cosmology/Astrophysics_%28Richmond%29/07:_Astronomical_Spectra,_Filters_and_Magnitudes?form=MG0AV3
- Lombardi, G., Zitelli, V., y Ortolani, S. (2009). The astroclimatological comparison of the Paranal Observatory and El Roque de Los Muchachos Observatory. *Monthly Notices of the Royal Astronomical Society*, 399(2), 783-793. <https://doi.org/10.1111/j.1365-2966.2009.15309.x>
- Malacara, D. (2015). *Óptica básica.* Fondo de Cultura Económica (FCE). <https://content.e-bookshelf.de/media/reading/L-8009399-8b7fa80700.pdf>
- Mejía, J., y Navas, G. (2024). Python como herramienta para el análisis de las imágenes astronómicas del Telescopio Reflector: Python as a tool for the analysis of astronomical images from the Reflector Telescope. *Conocimiento Libre y Licenciamiento (CLIC)*, (29), 23-49. <https://convite.cenditel.gob.ve/publicaciones/revistaclic/article/view/1260>
- Miczaika, G., y Sinton, W. (1967). *Las herramientas del astrónomo.* Eudeba.
- Munro, J., Hansen, J., Travouillon, T., Grosse, D., y Tokovinin, A. (2023). Dome seeing analysis of the Anglo-Australian Telescope. *Journal of Astronomical Telescopes, Instruments and Systems*, 9(1). <https://doi.org/10.1117/1.JATIS.9.1.017004>

- Navas, G. (2022). Implicaciones de la propiedad intelectual en la investigación astronómica venezolana, la ética y el reconocimiento de los derechos morales en los descubrimientos. *Propiedad Intelectual*, 23(3). <https://doi.org/10.53766/PI/2022.23.03>
- Parada, R., Rueda, S., y Monzo, C. (2020). Local seeing measurement for increasing astrophysical observatory quality images using an autonomous wireless sensor network. *Sensors*, 20(13), 3792. <https://doi.org/10.3390/s20133792>
- Pastrana, R. (2013). Medición de Seeing en el Observatorio Astronómico Centroamericano de Suyapa por medio de Técnicas Fotométricas. *Ciencias Espaciales*, 3(1), 54-65. <https://doi.org/10.5377/ce.v3i1.1082>
- Raab, H. (2025). *Software Astrometrica*. Software Astrometrica. <http://www.astrometrica.at/>
- Racine, R. (1996). Temporal Fluctuations of Atmospheric Seeing. *Publications of the Astronomical Society of the Pacific*, 108(722), 372-374. <http://www.jstor.org/stable/40680729>
- Radu, A., Angelescu, T., Curtef, V., Felea, D., Hasegan, D., Lucaschi, B., Manea, A., Popa, V., y Ralita, I. (2011). An astroclimatological study of candidate sites to host an imaging atmospheric Cherenkov Telescope in Romania. *Monthly Notices of the Royal Astronomical Society*, 422(3), 2262-2273. <https://doi.org/10.1111/j.1365-2966.2012.20771.x>
- Sarazin, M., y Roddier, F. (1990). The ESO differential image motion monitor. *Astronomy and Astrophysics*, 227(1), 294-300. <https://ui.adsabs.harvard.edu/abs/1990A&A...227..294S>
- Sentinelmission. (2024). *Astronomical Seeing - Definition & Detailed Explanation - Astronomical Units & Measurements Glossary*. Sentinelmission. <https://sentinelmission.org /astronomical-units-measurements-glossary/astronomical-seeing/#:~:text=Astronomical%20seeing%20is%20typically%20measured%20using%20a%20scale,FWHM%20value%2C%20the%20better%20the%20seeing%20conditions%20are>
- Skidmore, W., Els, S., Travouillon, T., Riddle, R., Schock, M., Bustos, E., Seguel, J., y Walker, D. (2009). Thirty meter telescope site testing V: seeing and isoplanatic angle. *The Astronomical Society of the Pacific*, 121(884), 1151-1166. <https://doi.org/10.1086/644758>
- Smart, W. (1962). *Text-Book on Spherical Astronomy*. Quinta edición. Cambridge University Press.
- Zuñiga, J. (2019). Identificando Cuerpos Celestes en el Universo Observable. *Con-Ciencia y Técnica*, 3(1), 63-69. <https://revistas.sena.edu.co/index.php/conciencia/article/view/2406>