

# Tarea 1 Astronomía Galáctica

## FIS2436

Gabriel Pichihueche  
Profesor: Nicolás Tejos

La reducción y análisis de datos de imágenes es de las áreas más importantes en astronomía observacional, ya que es el principal medio para el estudio de los objetos de la esfera celeste. En este trabajo aplicaremos esto utilizando imágenes obtenidas del “Dark Energy Camera Legacy Survey” (DECaLS) para detectar y separar fuentes, obtener datos valiosos de estas y estudiar sus distribuciones de brillo, color, magnitud y tamaño. Además se hará un ajuste de isofotas para algunas fuentes para comparar sus perfiles de brillo y sacar conclusiones a partir de estos.

El “Dark Energy Camera Legacy Survey” (DECaLS) es un gran survey hecho en el Observatorio Inter-Americano en el cerro Tololo, ubicado en la región de Coquimbo, este tenía como objetivo obtener un mapeo del cielo en alta calidad con varios filtros fotométricos (g, r y z) utilizando la “Dark Energy Camera” (DECam), este proyecto empezó en 2014 y terminó en 2019 con un total de 157 noches de exposición.

## Parte 1 Catálogos:

Para llevar a cabo esta tarea se eligió una región del cielo desde el viewer del DECaLS, de aquí se extrajo un fits el cual contiene las imágenes de este trozo en los 3 filtros disponibles (g, r y z), las coordenadas utilizadas son  $RA=132.8503^\circ$  y  $DEC=25.155^\circ$ , la imagen a continuación muestra como se ve el trozo elegido en blanco y negro:

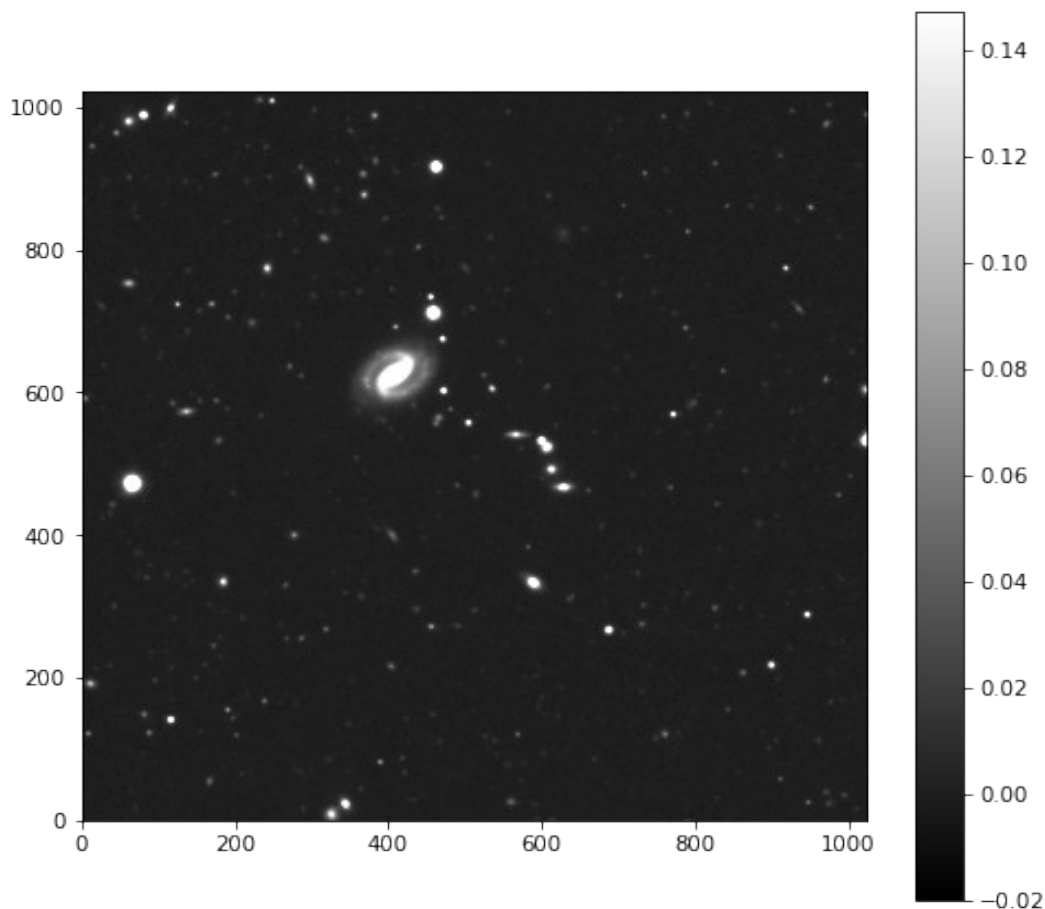


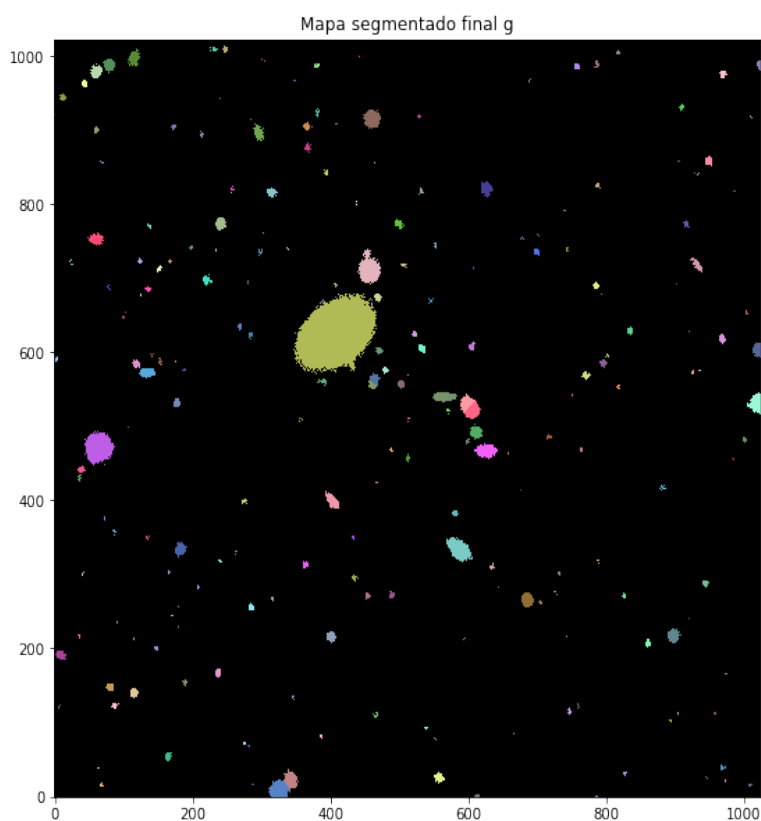
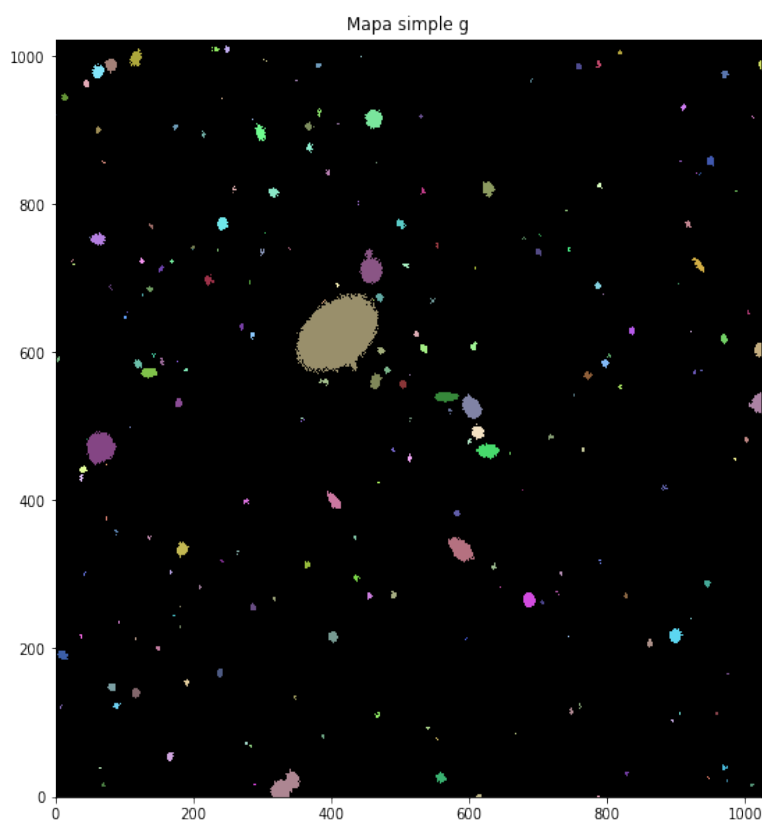
Imagen obtenida al promediar los flujos de los 3 filtros.

Vemos que hay un agran cantidad de objetos de varios tamaños y formas en la imagen, es ideal para lo que se quiere lograr con la tarea.

El siguiente paso es detectar y separar fuentes en la imagen utilizando herramientas computacionales, en este caso se escribió un código en Python para llevar a cabo esto, principalmente usando la librería photutils, la cual provee varias funciones útiles para el trabajo.

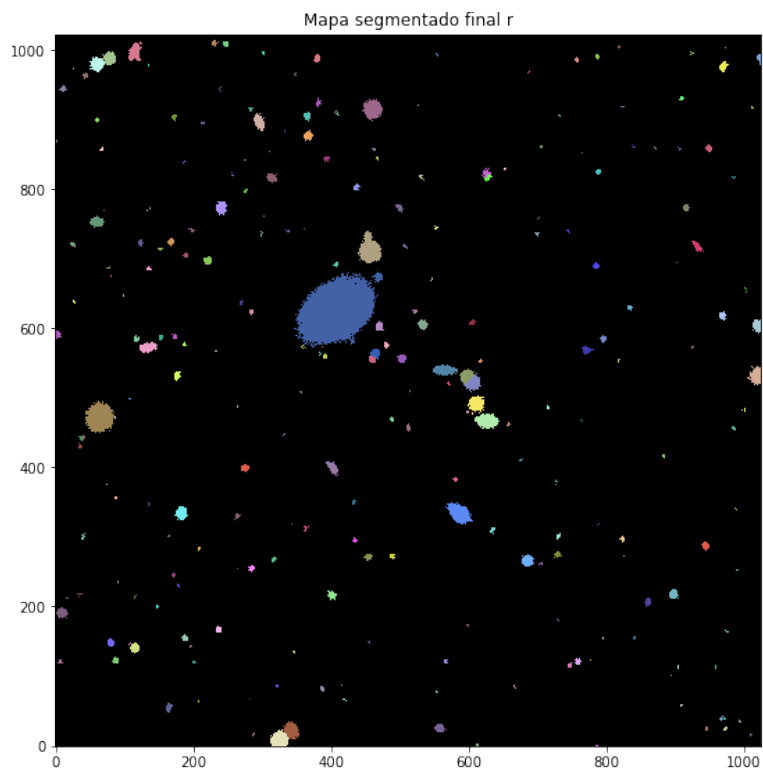
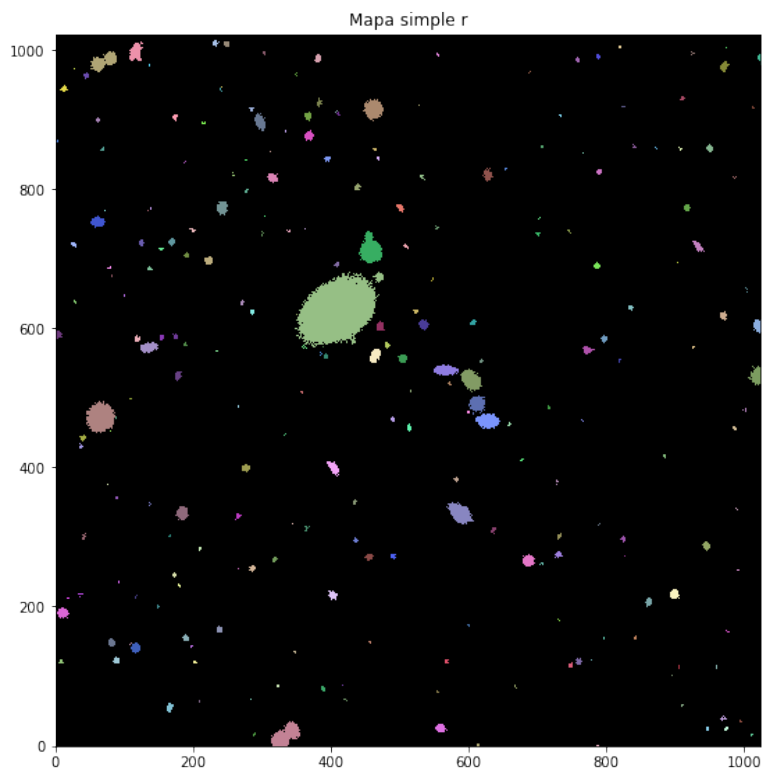
Usando las funciones de `detect_sources` para detectar fuentes y `deblend_sources` para separar fuentes que estén pegadas, se determinó que cualquier conjunto de al menos 5 píxeles con flujos mayores a 3 veces la desviación estándar del fondo se tomaran como fuentes (estos valores salieron principalmente por ensayo y error). Se obtuvieron los siguientes mapas de fuentes para cada filtro:

-Para el filtro g:



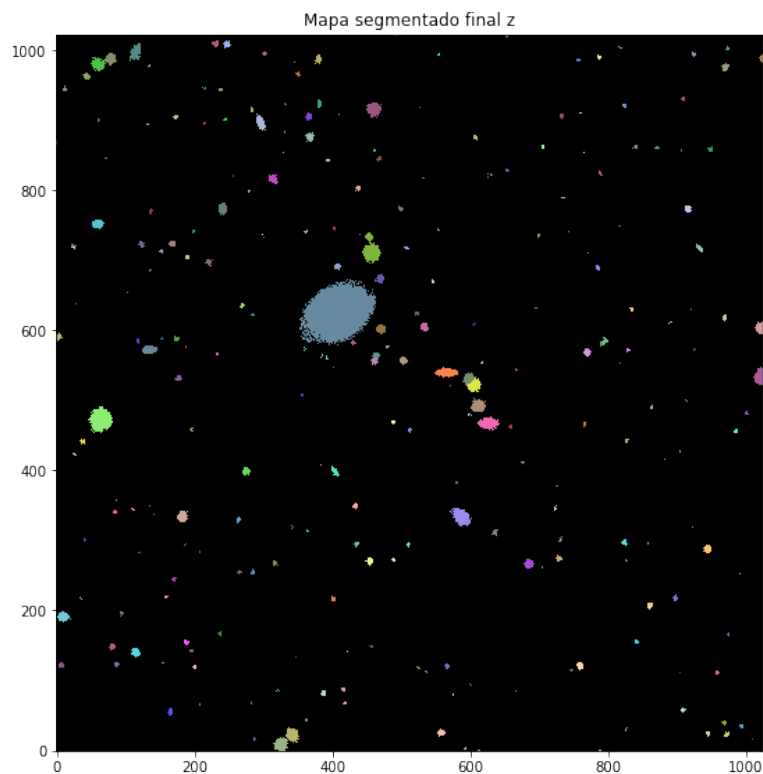
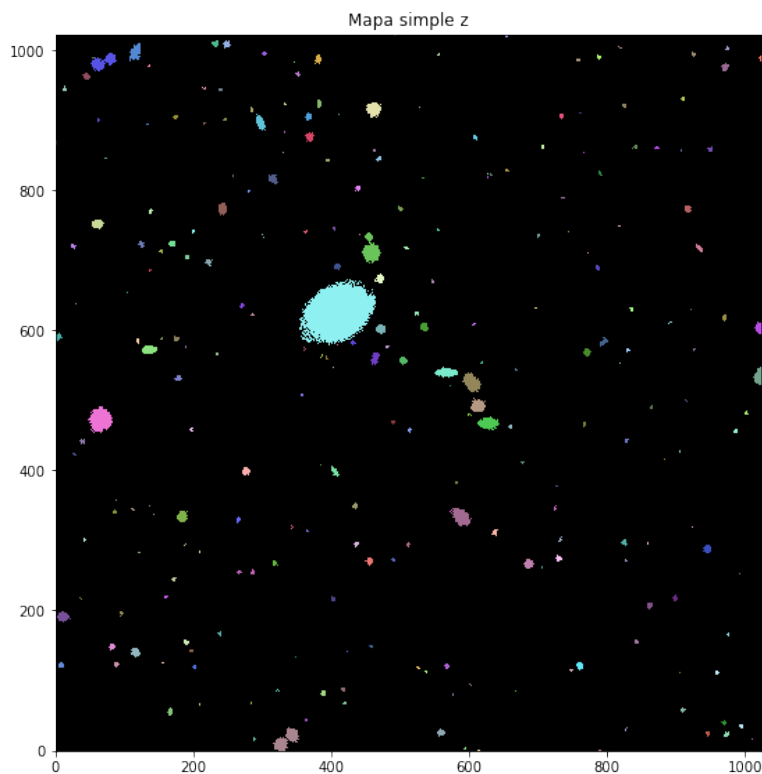
**Mapa simple y mapa con deblend filtro g (226 y 232 fuentes detectadas)**

-Para el filtro r:



Mapa simple y mapa con deblend filtro r (243 y 252 fuentes detectadas)

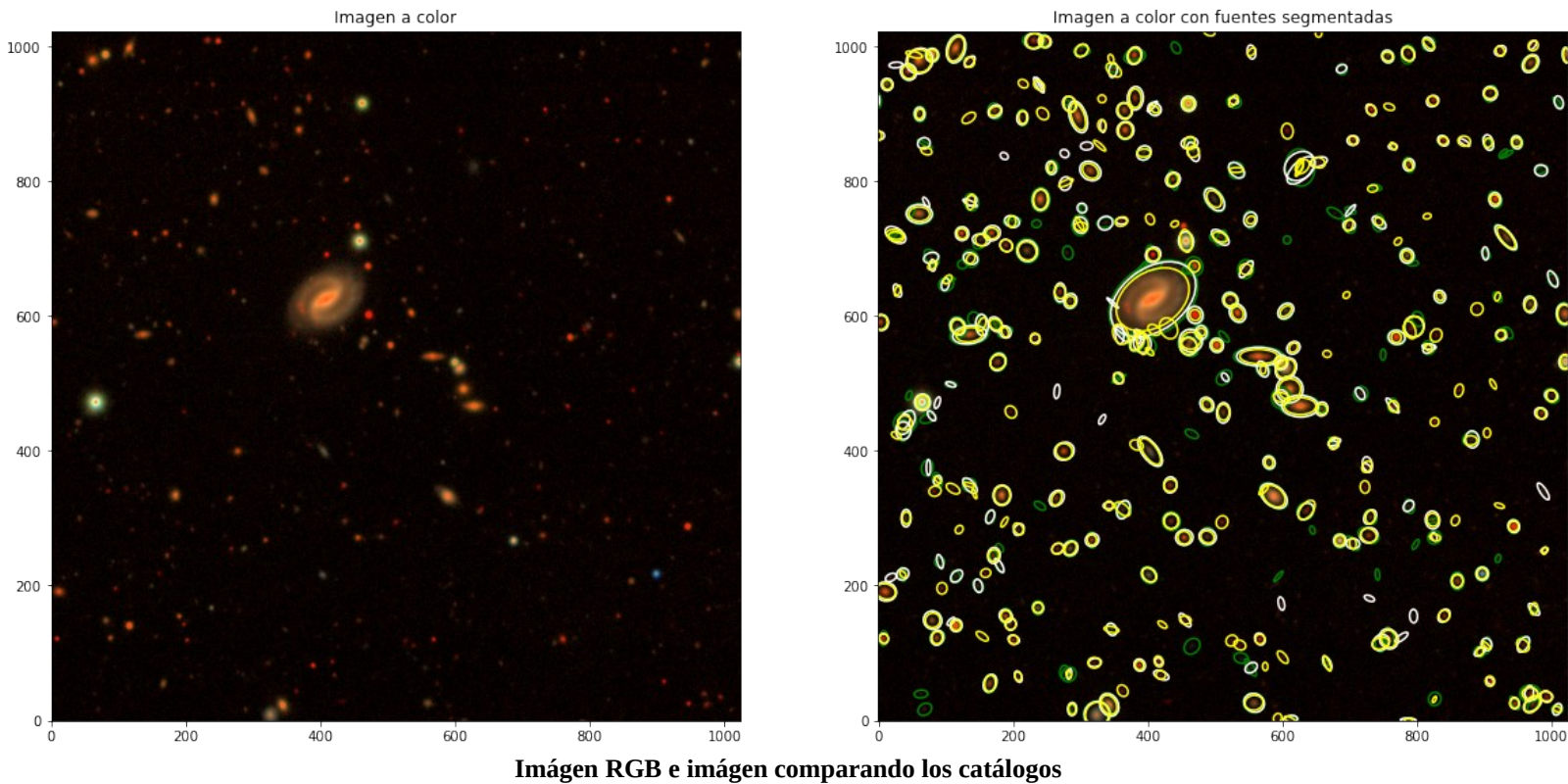
-Para el filtro z:



Mapa simple y mapa con deblend filtro z (254 y 261 fuentes detectadas)

Notamos como la imagen con deblend (la de la derecha) efectivamente separa algunas fuentes que aparecen como si fueran una sola en el mapa simple (el de la izquierda), ya teniendo estos mapas podemos generar un catálogo para cada filtro, el cual contiene la información de cada fuente, su ubicación, tamaño, flujo, área, etc (En el pdf del código adjunto se pueden ver los catálogos como tablas)

Ahora generamos una imagen a color usando la función `make_lupton_rgb` de `astropy`, tomando el filtro `g` como filtro azul, el filtro `r` como filtro verde y `z` como el filtro rojo, esta función toma las 3 imágenes y las junta en una sola a color usando un stretch en `arcsinh` (la escala de la separación entre colores) entre las imágenes, el cual se puede ajustar a mano para obtener mejores resultados (en este caso se usó un stretch de 0.1), además se graficaron las inclinaciones y tamaños de las fuentes dadas por los catálogos para comparar, la siguiente imagen muestra la imagen en RGB (izquierda) obtenida y la misma imagen con las fuentes detectadas graficadas encima (derecha):



Cabe aclarar que en la imagen de la derecha, los centroides verdes corresponden al filtro `g`, los blancos al `r` y los amarillos al `z`. Ya desde aquí se pueden notar varias cosas interesantes, primero, al graficar los 3 catálogos se ve que no detectan exactamente las mismas fuentes, hay fuentes que solo se detectaron en uno o dos filtros, y también objetos muy débiles que no pudieron ser detectados por ningún filtro (esto último se nota mejor en los plots del código). Lo otro extraño es que en el borde de la galaxia espiral barrada (el objeto más grande de la foto) se detectaron muchas fuentes pequeñas, la mayoría de estas no se pueden ver a simple vista, y

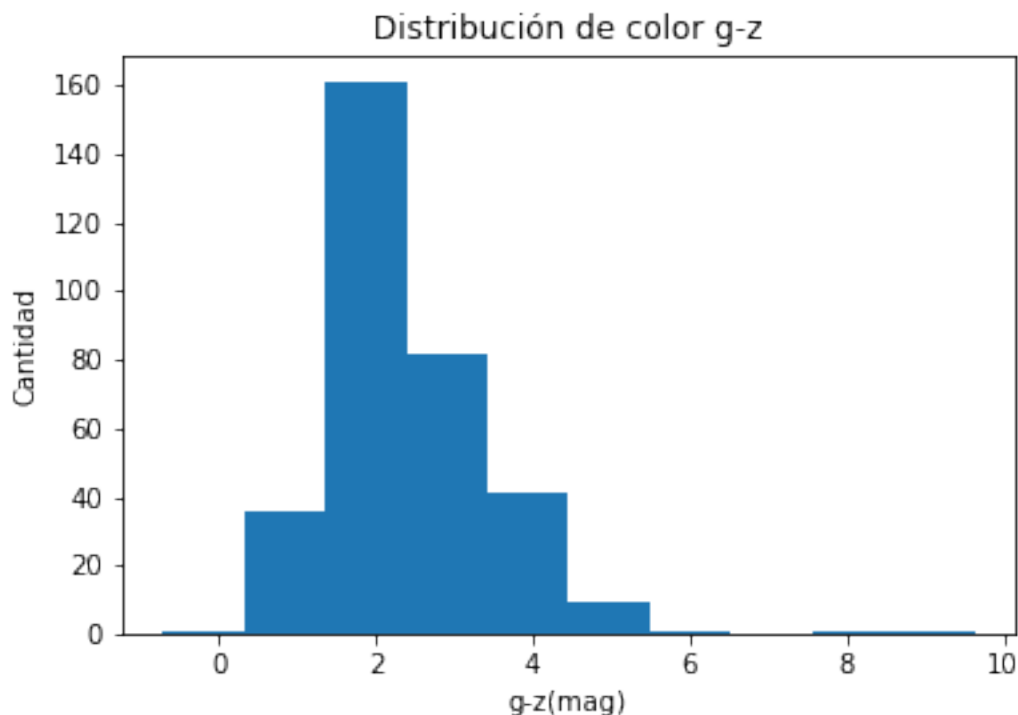
otras de lleno no se ven en la imagen, esto se debe probablemente a que el sistema para detectar fuentes tiene problemas a la hora de detectar fuentes cerca de objetos grandes como la galaxia en cuestión, al cambiar los parámetros para la detección (principalmente la cantidad de píxeles mínima y la cantidad de desviaciones estándar sobre el fondo) estas fuentes extra pueden desaparecer, pero al hacer esto se pierden varias fuentes que si se ve que están ahí, así perdiendo información que necesitamos. Esto se podría solucionar quizás agregando métodos extra para la detección de fuentes para objetos demasiado grandes (con una cantidad de píxeles mucho mayor al mínimo que se le da a la función) para aplicarlos sobre el mapa simple así como la función de deblending que ya tiene photutils para separar fuentes.

Luego se esto, se generaron las gráficas con las distribuciones de color, magnitud y tamaño para los catálogos, para esto se calculó la magnitud aparente para las fuentes con la fórmula:

$$m = -2.5 \log(F)$$

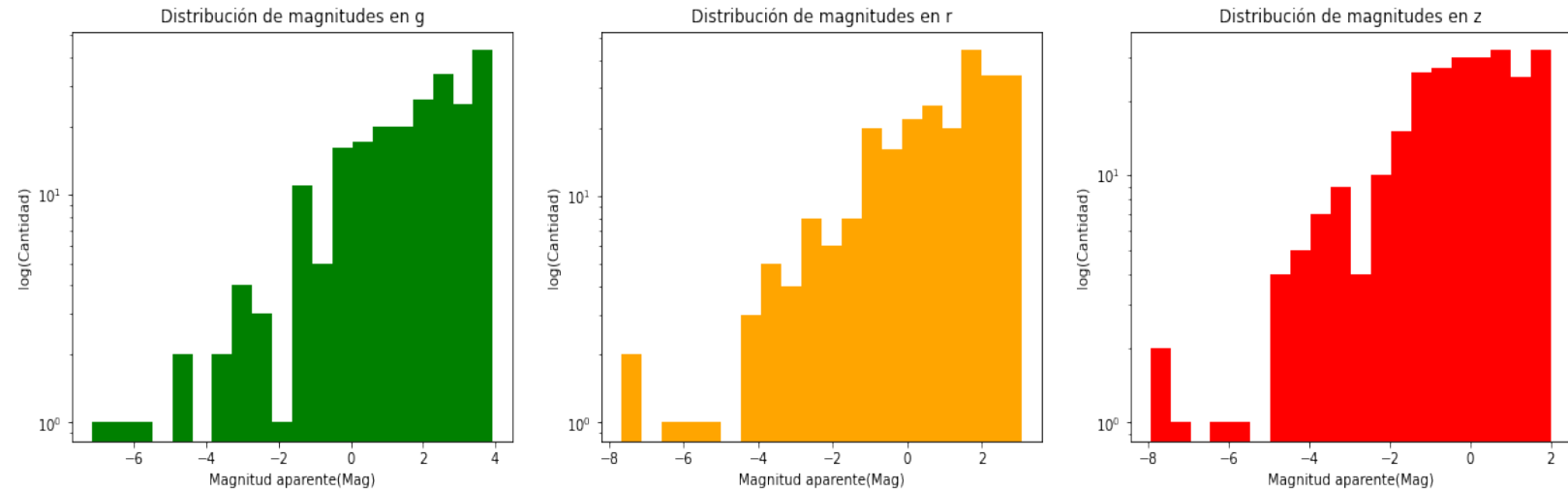
Siendo  $m$  la magnitud aparente y  $F$  el flujo medido, el cual se puede obtener a partir de los catálogos generados.

En el caso de la distribución de color se calculó un índice de color simple restando las magnitudes de los objetos en el filtro  $g$  con los del filtro  $z$  (la misma lógica del índice  $B-V$ ), se obtuvo la siguiente distribución:



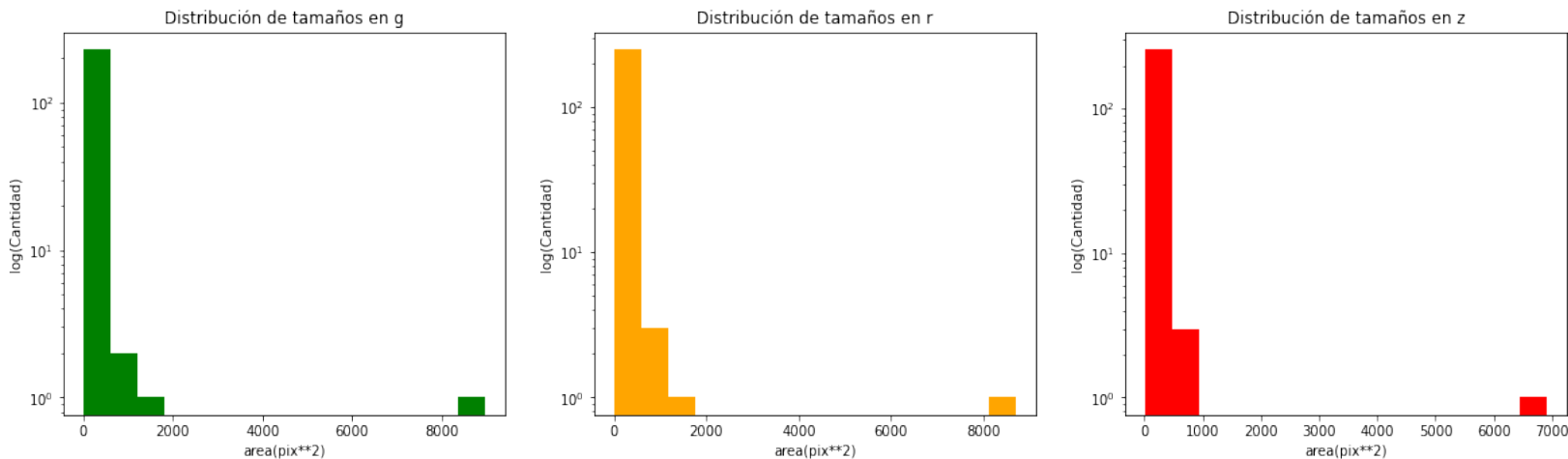
Para generar este histograma se restaron las magnitudes de 333 fuentes (para esto se generaron catálogos nuevos con la imagen promedio para no tener discrepancias en la cantidad de fuentes entre filtros), vemos que la mayoría tiene magnitudes sobre 0, lo cual significa que la mayoría de los objetos son más rojizos, esto coincide con el color de la mayoría de los objetos en la imagen RGB, con una minoría de los objetos mostrando un brillo más azulado, esto último también se ve reflejado en la tabla, hay una cantidad muy bajas de objetos con magnitudes en 0 o menor a 0 en comparación a los objetos más rojos.

Para la distribución de magnitudes se obtuvo lo siguiente:



De aquí se ve que la gran mayoría de las fuentes tienen magnitudes aparentes positivas, siendo estas la gran cantidad de estrellas y galaxias pequeñas que se ven en la imagen a color, los objetos de magnitudes negativas deben ser la galaxia espiral barrada y las pocas estrellas brillantes presentes en la imagen. Además se puede apreciar que los brillos tienden a ser mayores en el filtro z, lo cual concuerda con la distribución de color ya mencionada.

Finalmente para la distribución de tamaños se utilizó el área de los centroides, el cuál se extrajo directamente de los catálogos, obteniendo una distribución como se muestra a continuación:



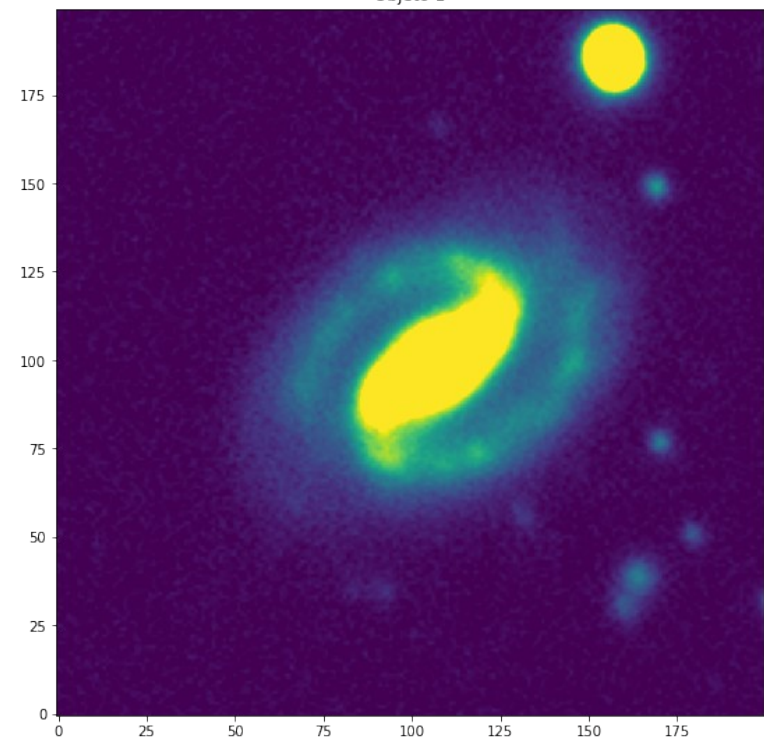
Al igual que las primeras dos gráficas, esto coincide perfectamente con lo observado en la imagen a color, la gran mayoría de los objetos son muy pequeños, con un grupo más pequeño siendo ligeramente más grande, y un solo objeto con un tamaño mucho mayor al resto, este corresponde claramente a la galaxia espiral barrada.

## Parte 2 Isofotas:

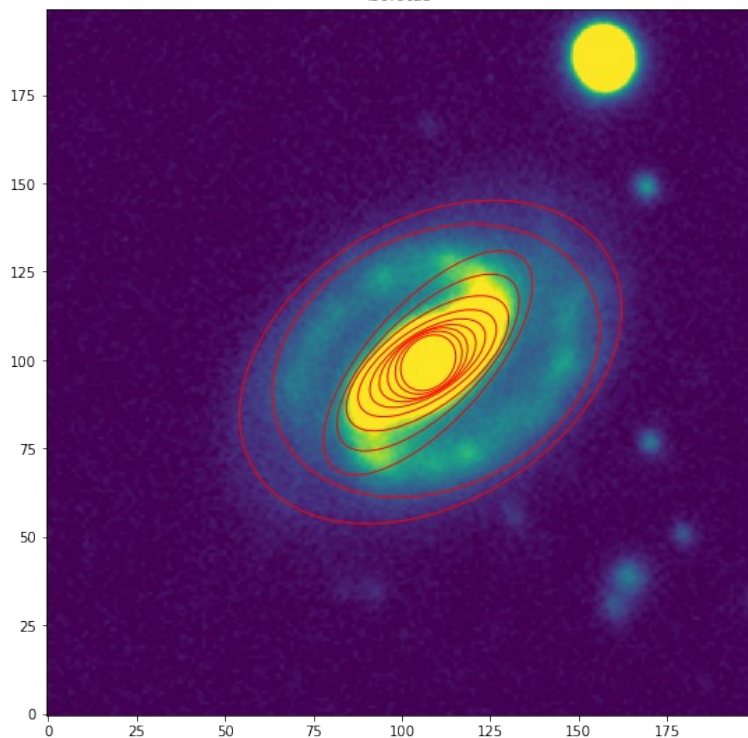
Para hacer el ajuste de isofotas se tomaron 9 de los objetos más grandes en los filtros en los que no estuvieran ni tan saturados ni tan tenues. Usando los métodos de photutils se obtuvo una lista de isofotas para cada objeto siguiendo un proceso de 3 pasos, primero se corta la imagen para aislar el objeto en cuestión y se prueba como se ve en los 3 filtros para elegir el mejor para el ajuste, luego de esto se dibuja una elipse de prueba como un objeto de geometría de photutils, usando como referencias los datos de los centroides en los catálogos de cada filtro, y finalmente usando la función `image_fit` se obtiene una lista de isofotas la cual se puede graficar, a continuación se muestra la imagen de cada objeto (izquierda) junto con la imagen con algunas de las isofotas de la lista (derecha):

### -Objeto 1 (Filtro g):

Objeto 1



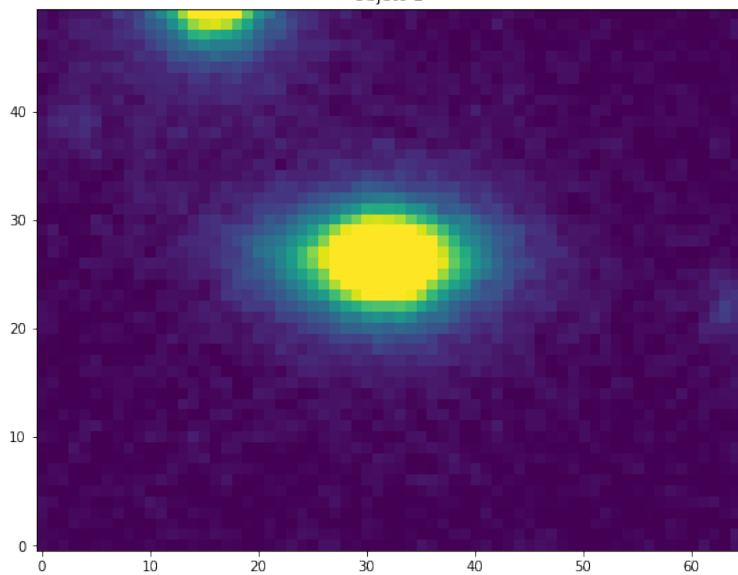
Isofotas



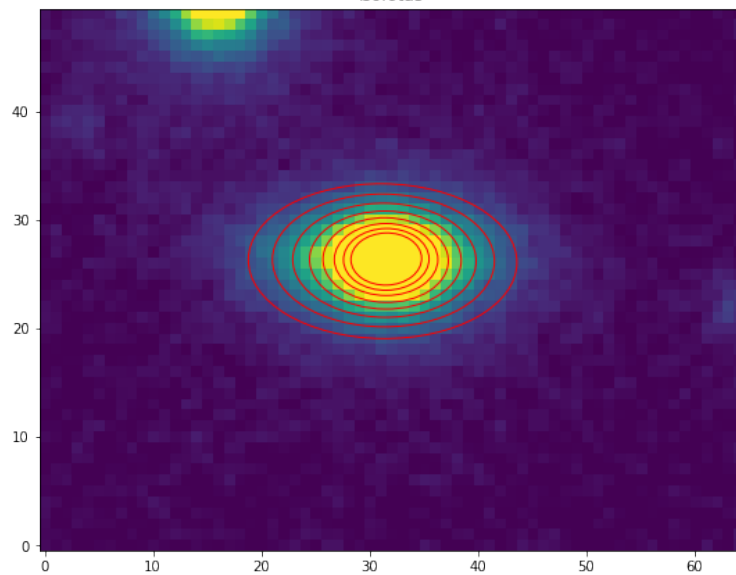


-Objeto 2 (Filtro r):

Objeto 2

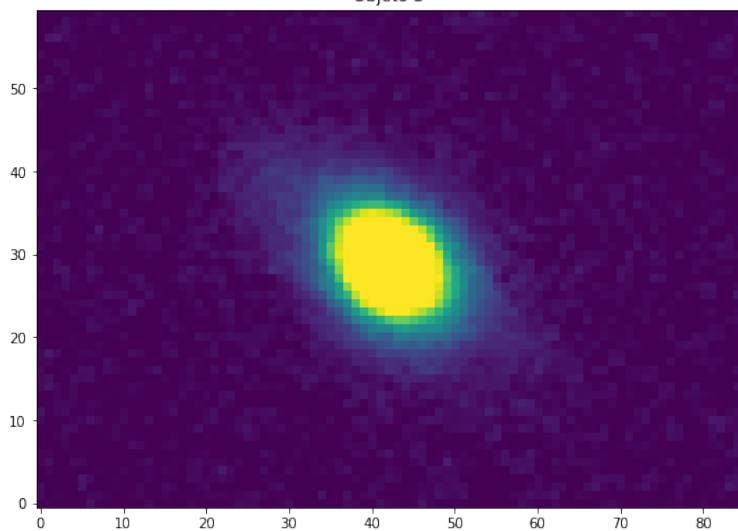


Isofotas

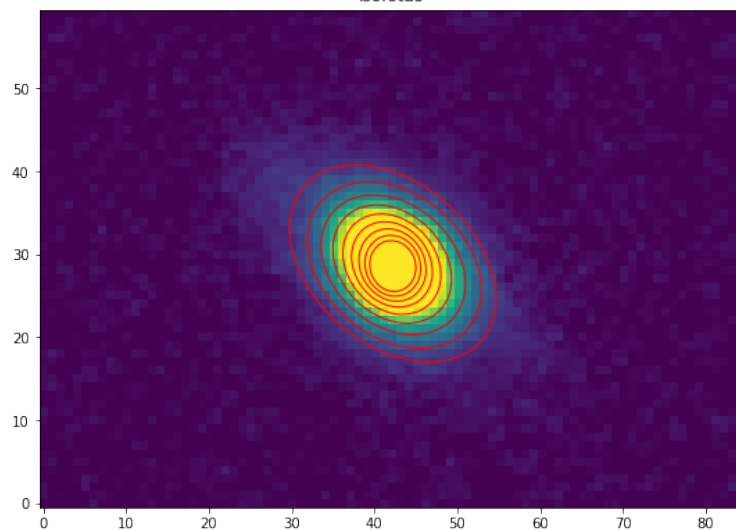


-Objeto 3 (Filtro r):

Objeto 3

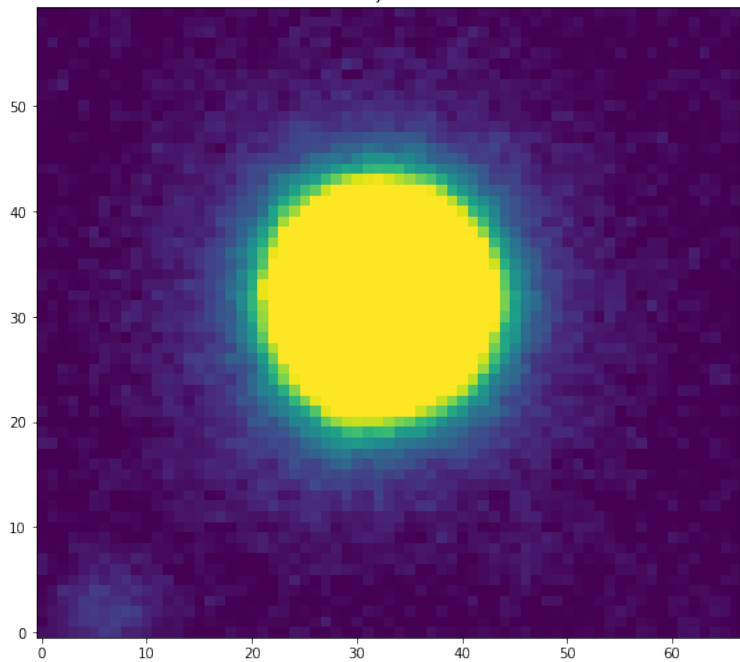


Isofotas

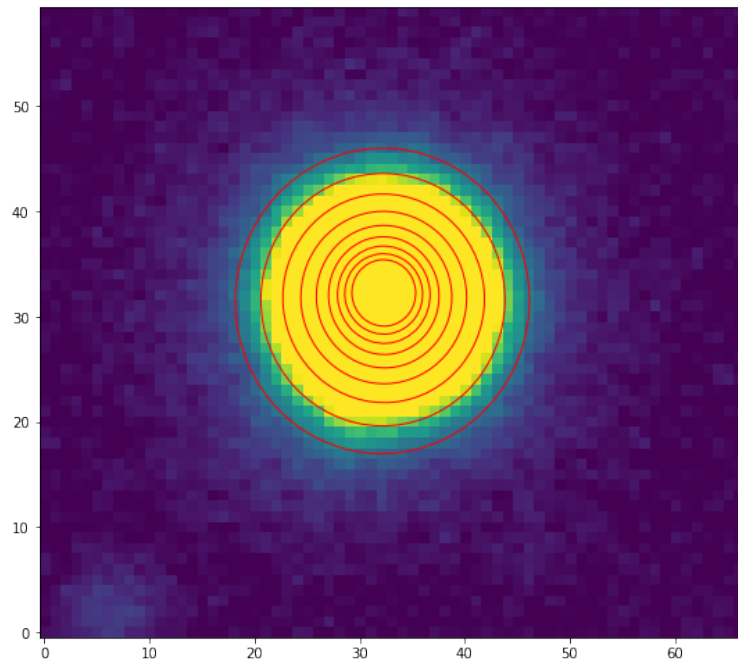


-Objeto 4 (Filtro r):

Objeto 4

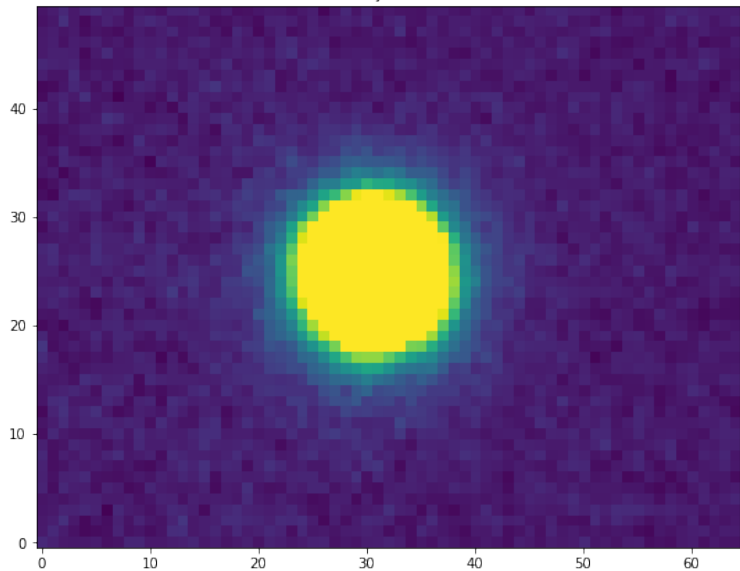


Isofotas

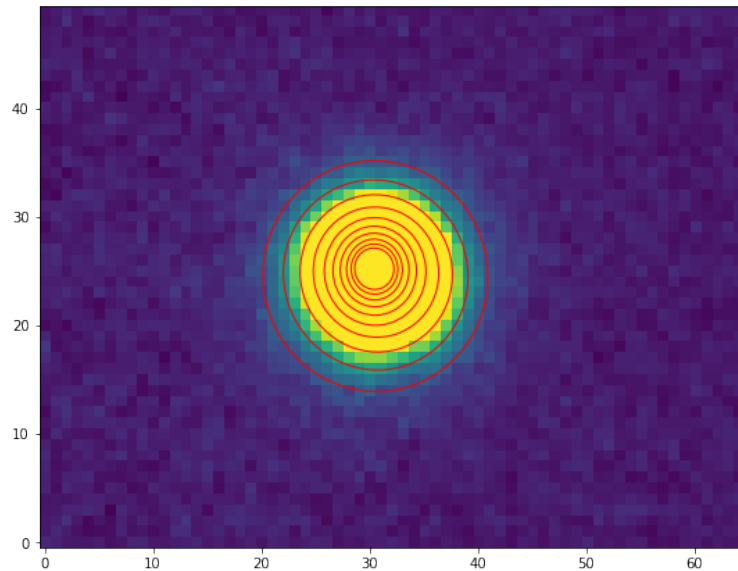


-Objeto 5 (Filtro r):

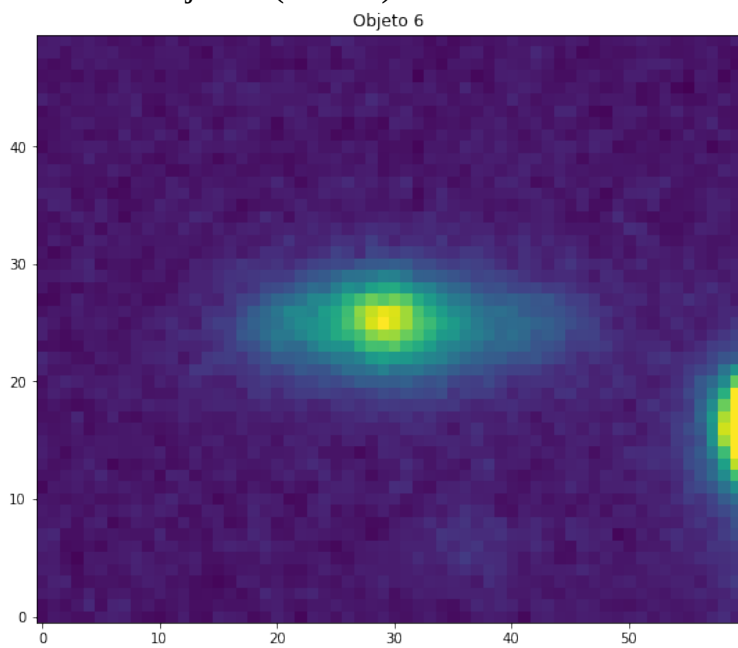
Objeto 5



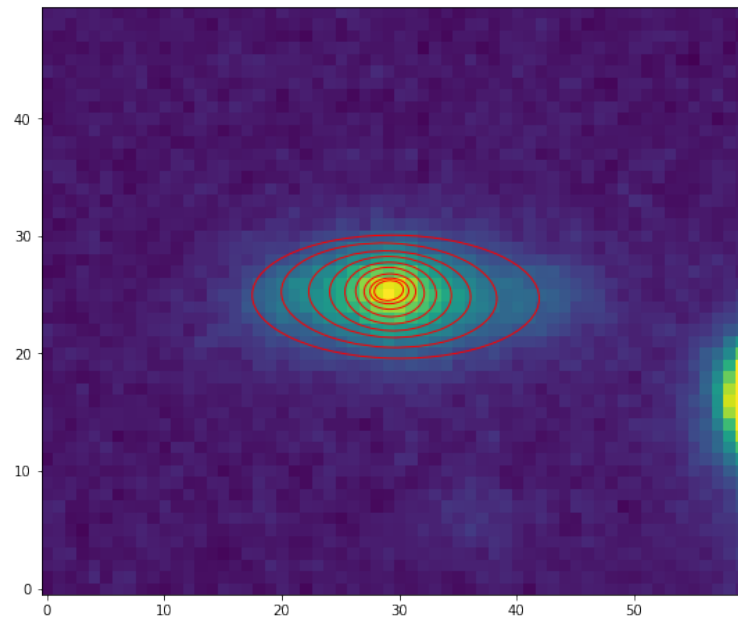
Isofotas



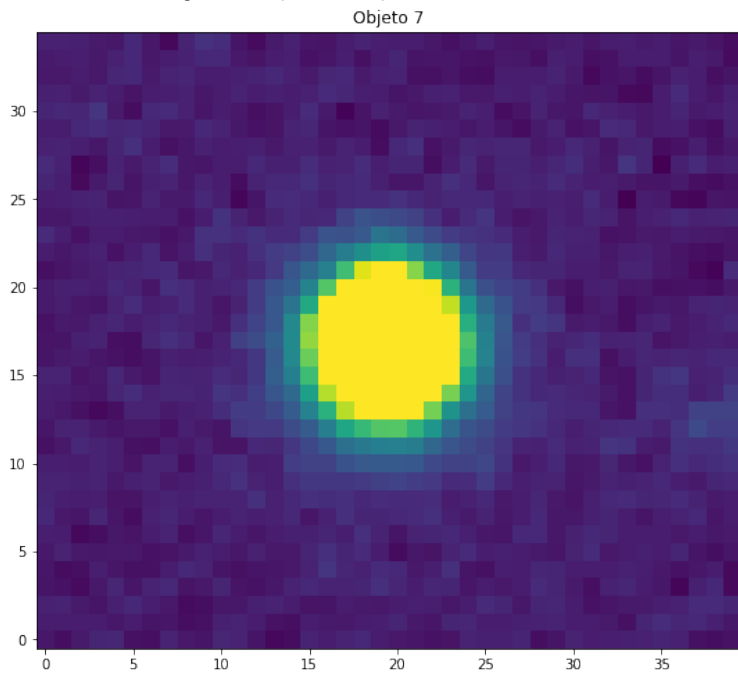
-Objeto 6 (Filtro r):



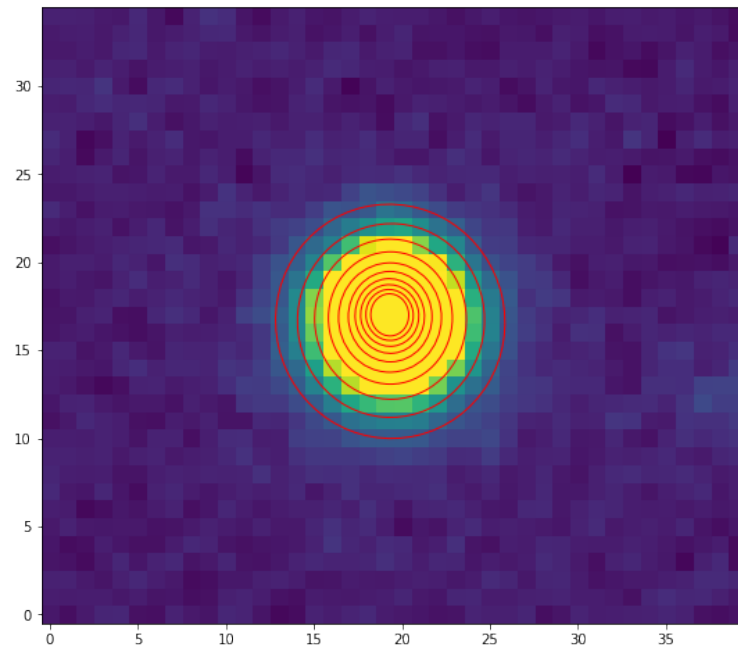
Isofotas



-Objeto 7 (Filtro r):

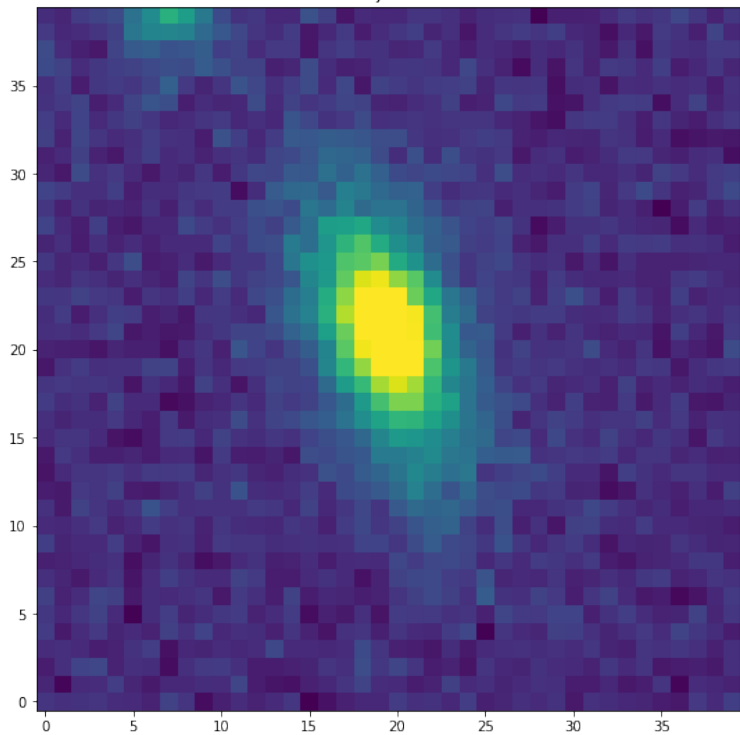


Isofotas

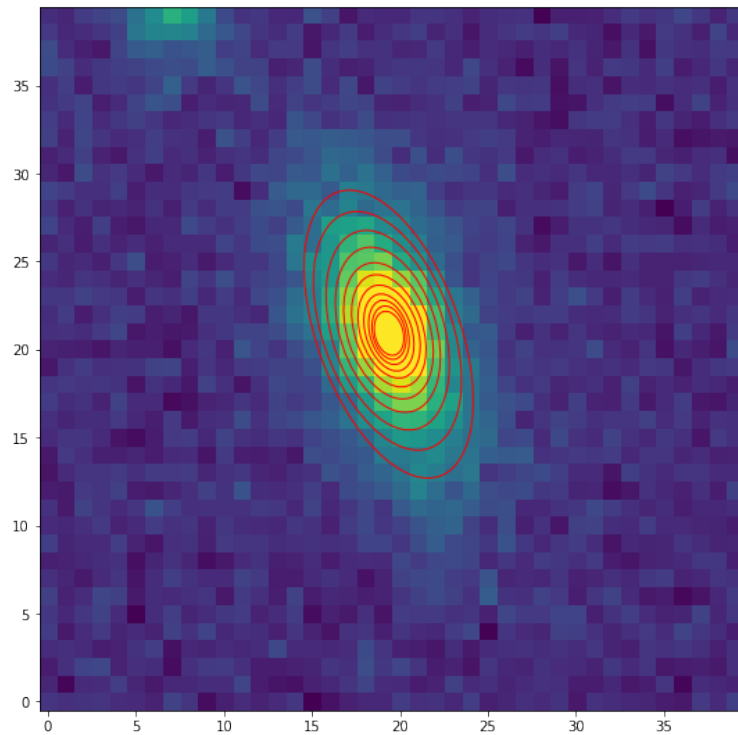


-Objeto 8 (Filtro z):

Objeto 8

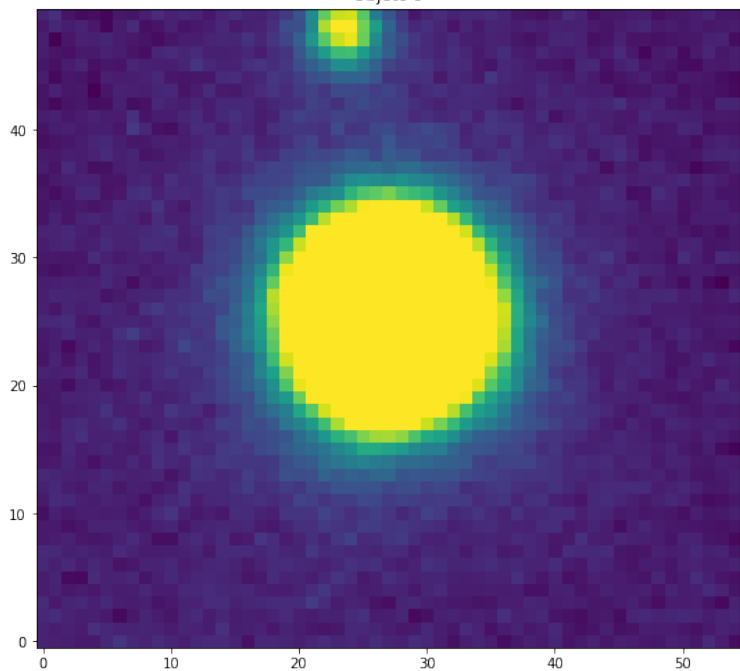


Isofotas

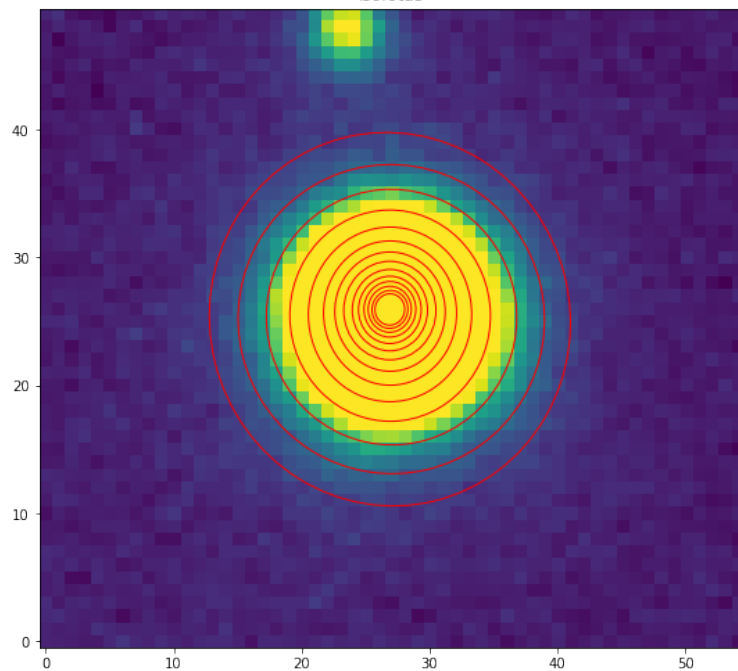


-Objeto 9 (Filtro r):

Objeto 9



Isofotas



Vemos que el método de photutils es bastante efectivo, las isofotas coinciden muy bien con la forma que tienen los objetos, ya sean más elípticos o estrellas más redondas, los perfiles de luz se ve que coinciden bien.

El objeto que más llama la atención es (claramente) la galaxia espiral barrada, la cual es la única que muestra isofotas que cambian de inclinación, y se ve además que estas siguen perfectamente la forma que tiene la espiral, con figuras más circulares en el centro del bulbo, luego pasando a elipses más prominentes y terminando en elipses menos excéntricas cuando llegamos a los brazos espirales, este es el perfil de luz que se esperaría de estos objetos con formas más complejas (principalmente galaxias espirales barradas que estén de cara hacia nosotros). El resto de objetos tienen perfiles de luz más parejos, especialmente los que son más redondos, estos simplemente muestran figuras más circulares centrados en el centro, una distribución que se esperaría de estrellas principalmente, finalmente los objetos con formas más elípticas muestran también perfiles más parejos de brillo, pero en este caso las isofotas son todas elípticas centradas en el centro del objeto con la misma inclinación, algo que uno esperaría de galaxias elípticas que están inclinada respecto a nosotros, probablemente si estuvieran más cerca se podrían apreciar perfiles más complejos de brillo.

Para concluir, con este trabajo se logró efectivamente separar y estudiar las fuentes en los diferentes filtros y ajustar isofotas a algunas de estas. Aplicando tanto temas teóricos como de programación para la manipulación y análisis de los datos de cada imagen, se lograron obtener varios resultados en forma de tablas, gráficas e imágenes, las cuáles, en su mayoría, coincidieron bien entre ellas y con lo esperado en base al conocimiento en la materia, esto sirvió tanto para reforzar la parte teórica, como para practicar y aprender en la parte de programación para su aplicación en astronomía.

## **Bibliografía:**

DECaLS: -Información: <https://www.legacysurvey.org/decamls/>

-Viewer: <https://www.legacysurvey.org/viewer#NGC%205551>

Photutils: -Detección de fuentes: <https://photutils.readthedocs.io/en/stable/segmentation.html>

-Isofotas: <https://photutils.readthedocs.io/en/stable/isophote.html>

Lupton RGB: -Documentación:

[https://docs.astropy.org/en/stable/api/astropy.visualization.make\\_lupton\\_rgb.html](https://docs.astropy.org/en/stable/api/astropy.visualization.make_lupton_rgb.html)

-Información creación de imágenes a color:

<https://docs.astropy.org/en/stable/visualization/rgb.html#astropy-visualization-rgb>

Repositorio de github con el código utilizado:

<https://github.com/Gabopic/TareasFIS2436/blob/main/Tarea1Galact.ipynb>