

# IOP3 Pipeline

Instalación y operación

## Contenido

1. Requerimientos .....	2
2. Instrucciones de instalación .....	2
2.1. Instalación de Anaconda .....	2
2.2. Creación del entorno virtual .....	2
2.3. SExtractor .....	3
2.4. WCSTools .....	3
2.4.1. Catálogo 2MASS .....	4
2.5. MySQL .....	5
2.6. Generación de la base de datos .....	5
2.7. PHPmyAdmin (Opcional) .....	6
2.8. Entorno .....	6
2.8.1. Paquetes Python .....	6
2.9. Jupyter notebooks .....	7
3. Pipeline IOP <sup>3</sup> .....	7
3.1. Inicialización .....	8
3.1.1. Creación de la base de datos .....	8
3.1.2. Carga de datos sobre Blazars .....	8
3.2. Vista rápida .....	9
3.3. En detalle .....	10
3.3.1. Reducción de las imágenes de la noche .....	11
3.3.2. Calibración de las imágenes reducidas .....	15
3.3.3. Determinación de parámetros polarimétricos .....	23
3.3.4. Inserción de información en la base de datos .....	23

## 1. Requerimientos

- Instalación de Anaconda y entorno virtual
- SExtractor
- WCSTools
- MySQL
- Paquetes Python
- Jupyter notebooks
- Código de la pipeline IOP3

## 2. Instrucciones de instalación

### 2.1. Instalación de Anaconda

**Instalar Anaconda** en el servidor (atención a requerimientos hardware)

(referencia web: <https://phoenixnap.com/kb/install-anaconda-on-centos-8>)

Descarga del software

```
wget -P ~/Downloads https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh
```

Verificación de descarga correcta

```
cd ~/Downloads  
sha256sum Anaconda3.2020.02-Linux-x86_64.sh
```

(Compare that code to the SHA256 hash column value on the [Anaconda hashes web page](#).)

Instalación

```
bash Anaconda3-2021.05-Linux-x86_64.sh
```

### 2.2. Creación del entorno virtual

Crear entorno virtual Conda

```
conda create -n iop3
```

Verificación de entorno creado. Debe aparecer entre los entornos disponibles al ejecutar el comando

```
conda env list
```

Activación del entorno

```
conda activate iop3
```

Instalación de Python en el entorno virtual

```
conda install -c conda-forge python==3.8
```

(gracias Juan 😊)

## 2.3. SExtractor

Instalación de SExtractor

```
conda install -c conda-forge astromatic-source-extractor
```

Documentación recomendada:

- SExtractor for dummies (<https://arxiv.org/pdf/astro-ph/0512139.pdf>)

## 2.4. WCSTools

**Paquete software necesario para la calibración astrométrica**

URL base: <http://tdc-www.harvard.edu/wcstools/>

La descarga se realiza a través de la URL: <http://tdc-www.harvard.edu/software/wcstools/wcstools-3.9.6.tar.gz>

Descargar catálogo 2MASS de fuentes de referencia para la astrometría siguiendo las instrucciones del enlace <http://tdc-www.harvard.edu/software/catalogs/tmc.convert.html> . El catálogo para todo el cielo ocupa ¿? GB, en 92 ficheros. Los del hemisferio norte son los psc\_b\*.gz.

Adicionalmente, es posible descargar el catálogo de USNO-B1 a través de bittorrent (50 GB). El enlace está disponible en la siguiente URL: [https://www.ap-i.net/skychart/en/news/usno-b1.0\\_catalog](https://www.ap-i.net/skychart/en/news/usno-b1.0_catalog)

Instalar los componentes necesarios para la compilación de WCSTools en linux (Ubuntu)

Desde una consola de comandos Linux, ejecutar

```
sudo apt install build-essential
```

Descomprimir el fichero wcstools

```
tar zxvf wcstools-3.9.6.tar.gz
cd wcstools-3.9.6
make all
```

Hacer que los ejecutables del paquete sean accesibles a todos los usuarios del servidor blazar

```
sudo vim /etc/bashrc
```

Agregar en el fichero la instrucción

```
### WcsTools ###
export PATH=$PATH:<dir_base>/wcstools-3.9.6/bin
```

donde `<dir_base>` es el directorio donde hemos descomprimido las fuentes de wcstools. Evidentemente, podemos copiar el contenido de la carpeta `wcstools-3.9.6/bin` en cualquier ruta a la que apunte la variable de entorno `$PATH`, pero es más limpio y modular el procedimiento propuesto.

#### 2.4.1. Catálogo 2MASS

##### Necesario para la calibración astrométrica

Básicamente, sigo las instrucciones de la URL: <http://tdc-www.harvard.edu/software/catalogs/tmc.convert.html>

Editar el fichero `tmc.c`. Buscar la variable `root` y modificarla para que apunte al directorio donde hemos ubicado los ficheros `psc_*.gz`

A veces, conviene ampliar el número de caracteres de las variables `file` y `file0` de 256 a 512, si la variable `root` tiene una ruta larga.

Compilar el programa

```
gcc -Wall tmc.c -o tmc
```

Situarnos en el directorio donde están los `psc_*.gz` y ejecutar la siguiente instrucción

```
zcat psc_*.gz | tmc
```

zcat lista el contenido de los gz y se los pasa al programa tmcats para que este último clasifique las fuentes que contienen los gz en 180 directorios que los programas de wcstools gestionan eficientemente.

Descargo el fichero tmcats (<http://tdc-www.harvard.edu/software/catalogs/tmcats>), listo el contenido de ficheros 2MASS

```
ls [root]/*/* > filelist
```

(es importante listar con la ruta absoluta)

Doy permisos de ejecución al fichero tmcats y ejecuto

```
tmcats filelist
```

En libwcs/tmcread.c (del fichero descargado y descomprimido de wcstools), cambio tmcats al valor del "root" (ruta base donde se ubican los catálogos 2MASS).

Finalmente, recompilo WCSTools con

```
make all
```

## 2.5. MySQL

### Necesario para el registro de la información en la base de datos.

Se usará para registrar la actividad de reducción, calibración y características de las fuentes.

(Fuente: <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04-es>)

```
sudo apt install mysql-server
```

(para instalar)

```
sudo mysql_secure_installation
```

(para quitar usuario anónimo, validar contraseñas seguras y eliminar acceso remoto al root de MySQL)

NOTA: Si se pierde la información de la contraseña de root en MySQL, se puede recuperar siguiendo instrucciones similares a la dada por la siguiente referencia:

<https://linuxhint.com/change-mysql-root-password-ubuntu/>

## 2.6. Generación de la base de datos

### Acciones a realizar

1. Crear base de datos 'iop3db':

```
create database iop3db;
```

2. Crear usuario:

```
create user 'iop3admin' identified by 'IOP3_db_admin';
```

3. Dar a ese usuario con todos los permisos sobre esa base de datos:

```
grant all privileges on iop3db.* to 'iop3admin';
```

También debemos ejecutar la siguiente instrucción

```
SET GLOBAL local_infile=1;
```

para hacer posible la carga de datos desde un fichero CSV.

## 2.7. PHPMyAdmin (Opcional)

### Facilita la tarea de la gestión de información en la base de datos (opcional)

(Fuente: <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-20-04-es>)

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

Seguir instrucciones dadas en la página de referencia.

MySQL 8.0 y superiores introduce variaciones en el almacenamiento de contraseñas que hace imposible conectarse al servidor a través de phpmyadmin. Afortunadamente, esta web nos enseña cómo solucionar este problema

<https://devanswers.co/phpmyadmin-access-denied-for-user-root-localhost/>

(Acceso de root a través de phpmyadmin para MySQL 8.0+)

## 2.8. Entorno

### Ambiente aislado para instalar los paquetes necesarios para la ejecución de Python

#### 2.8.1. Paquetes Python

Primero, se activa el entorno conda

```
conda activate iop3
```

Luego se instalan los paquetes necesarios para ejecutar la pipeline

```
conda install -c conda-forge numpy scipy matplotlib pandas astroquery
```

```
conda install -c astropy aplpy
```

```
conda install -c conda-forge mysql-connector-python
```

En la última versión de `aplpy` la función `blockreduce` cambia de módulo. Es por ello que hay que editar el fichero

```
[root_del_entorno_conda]/lib/python3.x/site-packages/aplpy/core.py
```

y cambiar la línea

```
from astropy.nddata.utils import block_reduce
```

Por

```
from astropy.nddata.blocks import block_reduce
```

## 2.9. Jupyter notebooks

### **Estupendo entorno para la ejecución y visualización de resultados (opcional).**

Activación del entorno

```
conda activate /home/IOP3/env/iop3
```

Instalación de paquetes

```
conda install -c conda-forge jupyterlab
```

```
conda install -c anaconda nbconvert
```

## 3. Pipeline IOP<sup>3</sup>

Todo lo necesario se proporcionará en a través de GitHub. El paquete es descargable a través de un zip usando la siguiente URL

<https://github.com/cesarhusrod/iop3/archive/refs/heads/main.zip>

El repositorio se podría descargar mediante git (si fuera público) de la siguiente forma

```
cd <directorio_codigo>
```

```
git clone https://github.com/cesarhusrod/iop3.git
```

Dado que es privado, si es miembro de desarrollo del paquete puede descargarlo a través de ese enlace escribiendo (cuando se lo pida) su nombre de usuario y su contraseña (que en mi caso es un token generado por el sistema).

El código incluye los siguientes ficheros:

iop3

- Carpeta iop3\_code
  - iop3\_create\_database.py
  - iop3\_pipeline.py
  - iop3\_reduction.py
  - iop3\_astrometric\_calibration.py
  - iop3\_photometric\_calibration.py
  - iop3\_polarimetry.py
  - iop3\_add\_db\_info.py
  - mcFits.py
  - mcReduction.py
- Carpeta conf
  - blazar\_photo\_calib\_last.csv
  - Otros ficheros de configuración de SExtractor

## 3.1. Inicialización

### 3.1.1. Creación de la base de datos

Ejecutar el comando

```
python iop3_create_database.py --db_user=iop3admin --  
db_password=IOP3_db_admin iop3db.sql
```

Para crear la base de datos en los que se registrarán los resultados de la reducción, calibración, extracción de parámetros de fuentes problema y registro de medidas de grado y ángulo de polarización.

### 3.1.2. Carga de datos sobre Blazars

Lo siguiente es la carga de datos sobre blazars. Esta tarea se realiza sólo una vez, por eso se ha separado del trabajo de inserción/actualización de resultados de reducción/calibración/fotometría. El fichero que contiene la información se llama

blazar\_photo\_calib\_bulk.csv

Y está disponible en el directorio de configuración.



Para insertar la información de los blazars en la base de datos, basta con ejecutar el comando

```
python iop3_add_blazar_to_db.py [ruta_conf]/blazar_photo_calib_bulk.csv
```

donde [ruta\_conf] es el path al directorio de configuración de la pipeline.

Se puede ejecutar a posteriori si hemos agregado nueva información al fichero blazar\_photo\_calib\_bulk.csv. El sistema comprueba el campo 'name\_mc'. Si no lo encuentra ya registrado, entiende que se trata de un nuevo blazar o calibrador y lo inserta.

### 3.2. Vista rápida

La pipeline trabaja procesando runs (noches) independientes. Cada run viene dado por un directorio que contiene ficheros de tipo FITS (\*.fits). Estos ficheros "raw" (crudos) incluyen bias, flats, observaciones de estrellas de calibración (estrellas HD) y observaciones de los objetos que se desea estudiar (blazars).

El nombre del directorio que contiene cada run debe cumplir el siguiente patrón: MAPCAT\_YYYY-MM-DD

La estructura de directorios que se propone es esta

# INPUT run directories

# [root\_data]/data/raw/MAPCAT/YMMMDD (1)

# OUTPUT reduction

# [root\_data]/data/reduction/MAPCAT/YMMMDD (2)

# OUTPUT calibration

# [root\_data]/data/calibration/MAPCAT/YMMMDD/YYYYMMDD -hhmmss (3)

# OUTPUT polarization

# [root\_data]/data/final/MAPCAT/YMMMDD (4)

(1) Imágenes descargadas del observatorio (raw) de tipo bias, flats y science para cada run.

(2) Imágenes obtenidas tras el proceso de reducción: superbias, superflats y de ciencia.

(3) Imágenes calibradas astrométrica y fotométricamente. Hay **un subdirectorio para cada una de las imágenes** procesadas en ese run. Contiene PNGs, catálogos SExtractor y ficheros con medidas de flujo, magnitud y error para fuentes problema (CSVs).

(4) Fichero CSV con la medida de polarización, error de polarización, ángulo y error de polarización (además de otros parámetros) para todos los objetos observados en la correspondiente noche. Combina las mediciones obtenidas en el paso anterior de los 4 últimos ficheros correspondientes a cada ángulo de la rejilla para cada blazar/HD de calibración en estudio.

Donde `[root_data]` es el directorio de procesamiento raíz para todo el proceso de tratamiento de imágenes FITS. Como explicación adicional a los patrones empleados:

- `YYYY`, se refiere al año de observación del run, con 4 dígitos.
- `YY`, se refiere al año de observación del run, con 2 dígitos.
- `MM`, al mes de observación del run, con 2 dígitos.
- `DD`, al día de observación del run, con 2 dígitos.
- `hh`, a la hora de captura de una imagen del run, con 2 dígitos.
- `mm`, al minuto de captura de una imagen del run, con 2 dígitos.
- `ss`, al segundo de captura de una imagen del run, con 2 dígitos.

Para ejecutar la tarea sobre cada run, basta con ejecutar la instrucción

```
python iop3_pipeline.py --border_image=15 directorio-de-configuración  
[root_data]/data/raw/MAPCAT/YMMMDD
```

A modo de ejemplo:

```
python iop3_pipeline.py --border_image=15  
/home/cesar/desarrollos/Ivan_Agudo/conf/  
/home/cesar/desarrollos/Ivan_Agudo/data/raw/MAPCAT/160604/
```

La pipeline ejecutará automáticamente todo el proceso: reducción, calibración, cómputo de polarimetrías y registro en la base de datos (`iop3db`), para la carpeta con las imágenes de la noche (run). Este proceso genera mucha información útil para resolver posibles errores y algunos ficheros CSV que se usarán en las partes finales de la pipeline para insertar la información en la base de datos. Independientemente de esto, es cierto que la información generada es abundante. En el futuro, cuando la pipeline sea eficiente y estable, se podrá modificar el código para minimizar el número de ficheros generados.

### 3.3. En detalle

`iop3_pipeline.py`

Este script realiza llamadas consecutivas a otros 4 scripts

1. `iop3_reduction.py`
2. `iop3_astrometric_calibration.py`
3. `iop3_photometric_calibration.py`
4. `iop3_polarimetry.py`
5. `iop3_add_db_info.py`

Como parámetros necesarios, requiere del directorio de configuración (donde está el fichero de los blazars y las configuraciones de *SExtractor*) y el directorio de imágenes (raw) de la noche.

**NOTA:** Es posible que haya que tocar los ficheros de configuración de *SExtractor* (`daofind.sex` y `sex.conf`) porque hace uso de filtros y ficheros cuya ruta debe establecerse de forma absoluta.

### 3.3.1. Reducción de las imágenes de la noche

Tarea llevada a cabo por el script `iop3_reduction.py`

En esta fase

- se clasifican las imágenes (*flats*, *bias*, *ciencia*),
- se combinan los *bias* entre sí para obtener **un único superbias**,
- se combinan los *flats* entre sí atendiendo al ángulo de la rejilla de difracción (4 *superflats*),
- se reducen las imágenes de ciencia (estrellas HD y blazars) haciendo uso del *superbias* general y del *superflat* correspondiente al ángulo de polarización de la imagen considerada.

Como parte importante de la salida, se generan los ficheros FITS reducidos y, además, estos 5 ficheros CSV:

- **input\_data\_raw.csv**, con información sobre las imágenes del observatorio (dimensiones, objeto observado, valores de la matriz de la imagen...).
- **masterbias\_data.csv**, con información al respecto del masterbias generado (FITS usados para su generación, estadística de la imagen...).
- **masterflats\_data.csv**, con información sobre los masterflats generados a partir de combinación de flats agrupados por ángulo de polarización, además de estadísticas de cada imagen.
- **output\_data\_red.csv**, con información sobre la imagen reducida y estadísticas de la matriz de la imagen.

Estos ficheros serán empleados posteriormente para insertar información en la base de datos (en el paso de la llamada al script `iop3_add_db_info.py`). Dado el formato de dichos ficheros, pueden revisarse fácilmente haciendo uso de *MS Excel* o con *OpenOffice* para verificar la validez de los ficheros procesados.

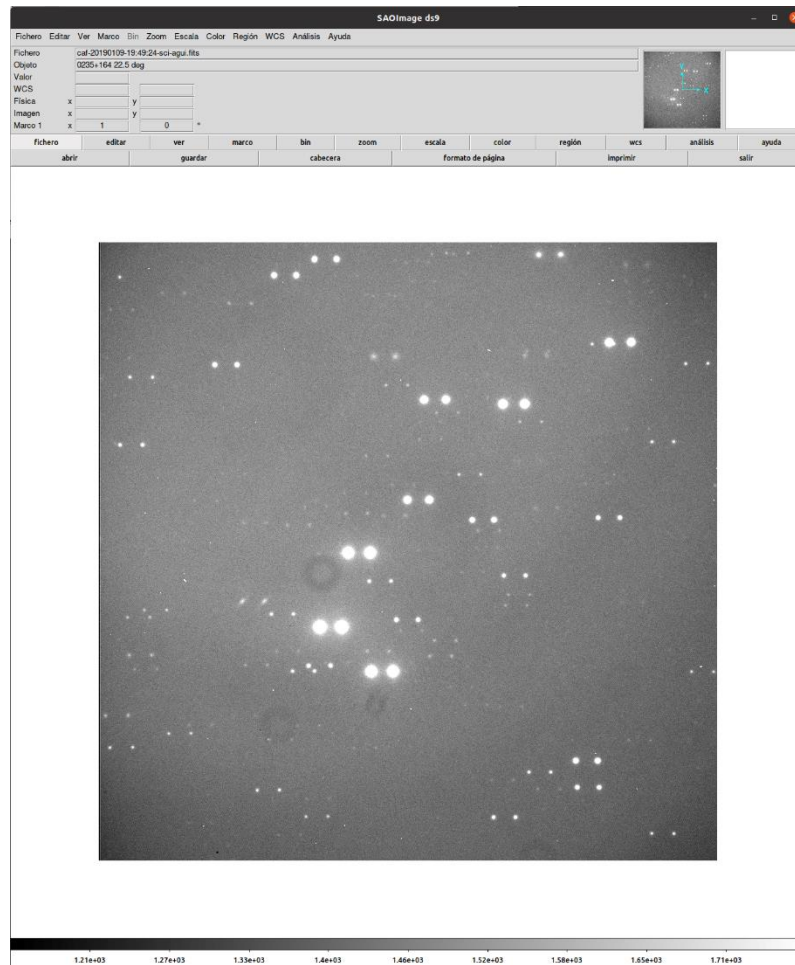


Figura 1. Imagen obtenida por el telescopio.

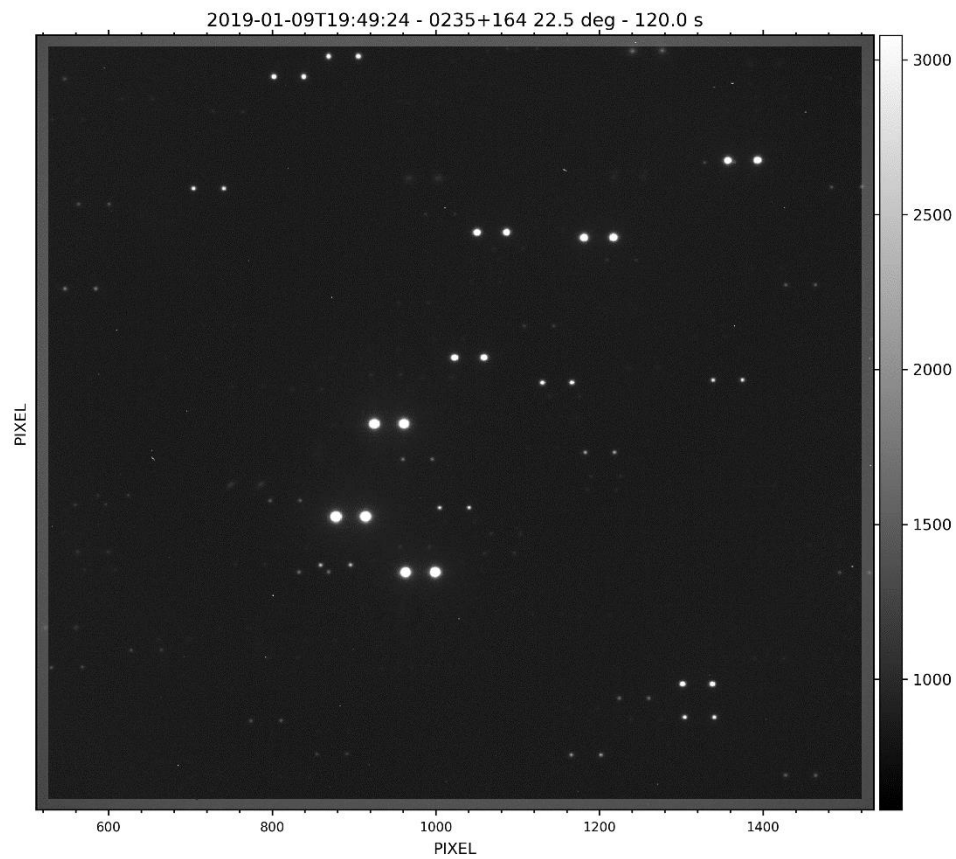


Figura 2. Imagen anterior reducida.

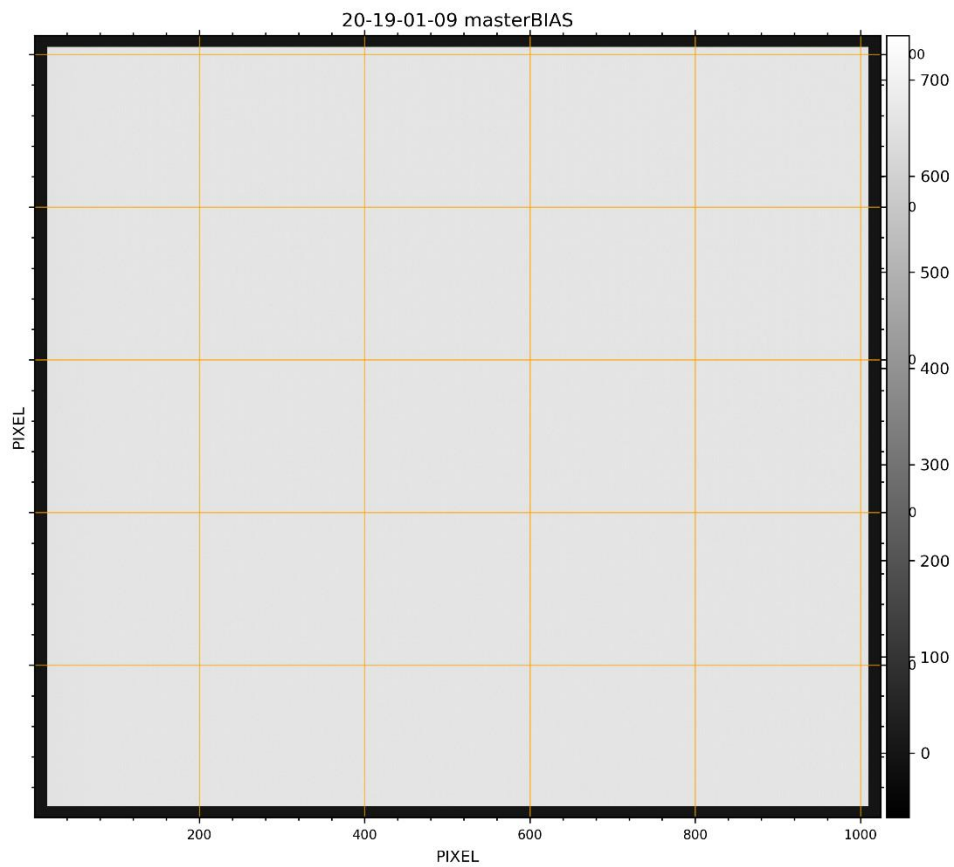


Figura 3. MasterBIAS de la noche.

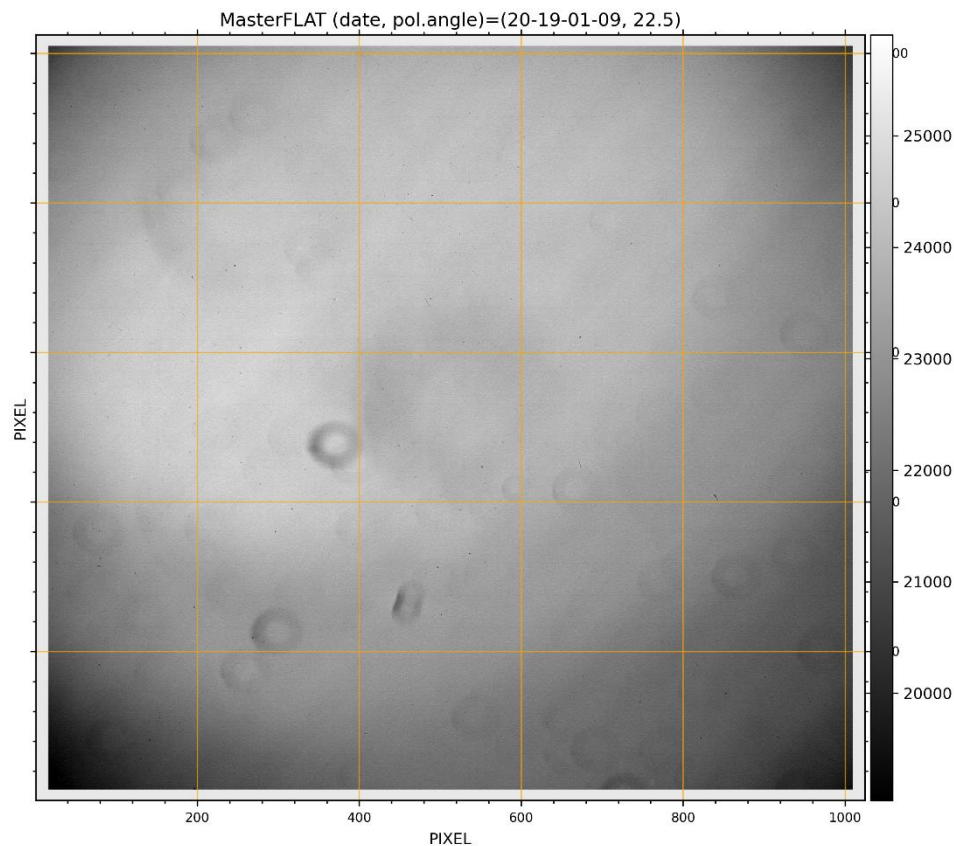


Figura 4. MasterFlat para el ángulo de rotación igual a 22.5°. Se generan 4 por noche (uno por cada ángulo del polarizador).

Este masterflat se normaliza y divide píxel a píxel la imagen original tomada en este ángulo, previa resta del masterbias

$$(\text{reducción} = (\text{raw} - \text{masterbias}) / \text{flat\_normalizado}).$$

**NOTA:** Se generan en el directorio de reducción, cuatro ficheros html a modo de log gráfico:

- masterbias\_fits.html
- masterflat\_fits.html
- raw\_fits.html
- reduced\_fits.html

Los nombres expresan claramente a qué se refiere su contenido.

### 3.3.2. Calibración de las imágenes reducidas

En la fase dos se toman las imágenes de ciencia reducidas y se calibran primero astrométrica (`iop3_astrometric_calibration.py`) y después fotométricamente (`iop3_photometric_calibration.py`).

- La calibración astrométrica consiste en el procedimiento que consigue asignar a cada pixel (posición en x e y) de la imagen una coordenada celeste (ascensión recta y declinación). Para hacer esto, se necesitan extraer las coordenadas (en unidades de píxeles en x e y) de las fuentes en la imagen, y buscar semejanzas con la misma zona de cielo para fuentes cuyas coordenadas vienen dadas en grados (ra, dec). Se usa el software de WCSTools que hemos instalado para realizar este proceso.
- La calibración astrométrica permite determinar para un flujo de apertura medido en cualquier fuente, cual es la magnitud AB que tiene. Este proceso también implica una comparación entre magnitudes AB de objetos reales medidos en el filtro R del catálogo de MAPCAT ([https://www.iaa.csic.es/~iagudo/research/MAPCAT/source\\_list/list.html](https://www.iaa.csic.es/~iagudo/research/MAPCAT/source_list/list.html)) y las medidas de flujo (cuentas en los píxeles de la fuente detectada en la imagen) para las mismas fuentes usadas como patrón (calibradores fotométricos del objeto de estudio). Nuestras magnitudes vienen dadas en el filtro R. Esto permite establecer un punto cero (ordenada en el origen de una recta de pendiente igual a 1) que es usado por SExtractor (software de detección y medida de fuentes en imágenes FITS) para determinar la magnitud y el error asociado a la magnitud de cada objeto en la imagen. Para este proceso, es necesario la calibración astrométrica del punto anterior, que permite la discriminación precisa de los calibradores asociados a cada fuente.



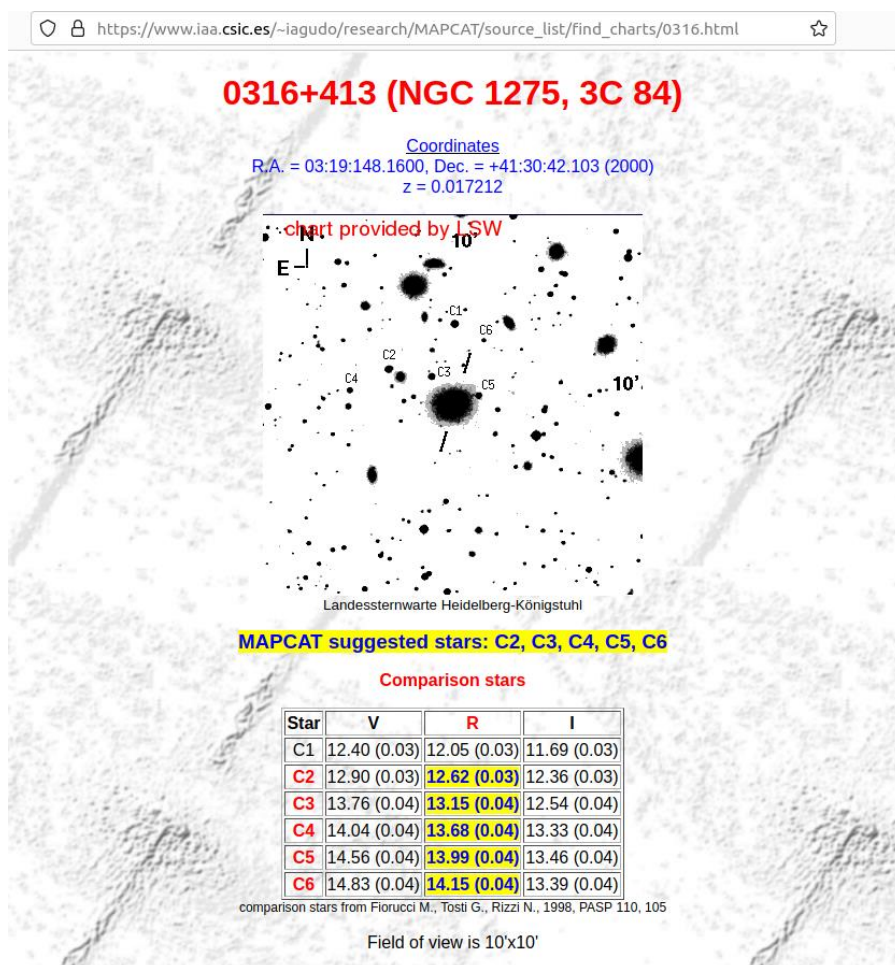


Figura 5. Campo asociado al blazar 0316+413

([https://www.iaa.csic.es/~iagudo/research/MAPCAT/source\\_list/find\\_charts/0316.html](https://www.iaa.csic.es/~iagudo/research/MAPCAT/source_list/find_charts/0316.html))

El procedimiento es diferente según se trate de imágenes de corta exposición (estrellas HD usadas como validadores del proceso) o de larga (blazars). Las de primer tipo se caracterizan por ser bastante ruidosas (tiempo de exposición menor o igual que 2 s), lo que interfiere en el proceso de detección de fuentes reales necesarias para realizar la calibración astrométrica. En el tratamiento de éstas, se implementa un proceso adicional de filtrado de la imagen (*filtro de mediana* o *de paso alto*) que minimiza el ruido de cielo de la imagen original.

En la calibración astrométrica los ficheros reducidos se agrupan por objeto (clave de cabecera del FITS llamada OBJECT) y se ordenan por fecha de observación (clave de cabecera DATE-OBS). De esta forma se agrupan por observaciones del mismo objeto, fecha (aproximada) y para los cuatro ángulos del polarizador (0, 22.5, 45 y 67.5 grados). Si se consigue calibrar astrométricamente un fichero del grupo, esa información se copia en las cabeceras de aquellos que en los que no se consiguió. Los motivos pueden ser varios, pero normalmente se refiere a imágenes con poco contraste o ruidosas, en las que no se han conseguido suficientes fuentes o las localizadas no se corresponden con fuentes reales (ruido).

Para cada imagen reducida, deben identificarse y eliminarse las fuentes más al sur de cada pareja de fuentes (porque la original se divide en dos), se debe realizar una rotación de 90° (introducida por el

instrumento), y deben detectarse y filtrarse las fuentes restantes de interés para que las rutinas de WCSTools sean capaces de calibrar astrométricamente usando las fuentes del catálogo 2MASS.

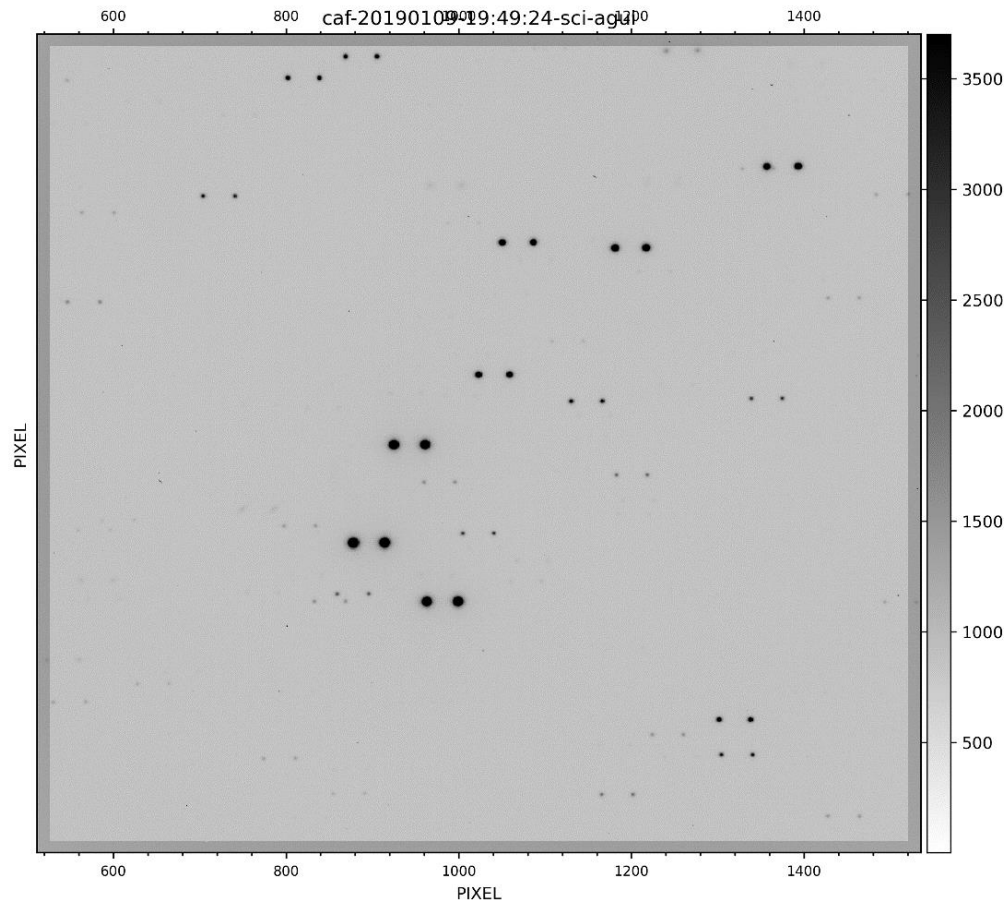


Figura 6. Imagen reducida.

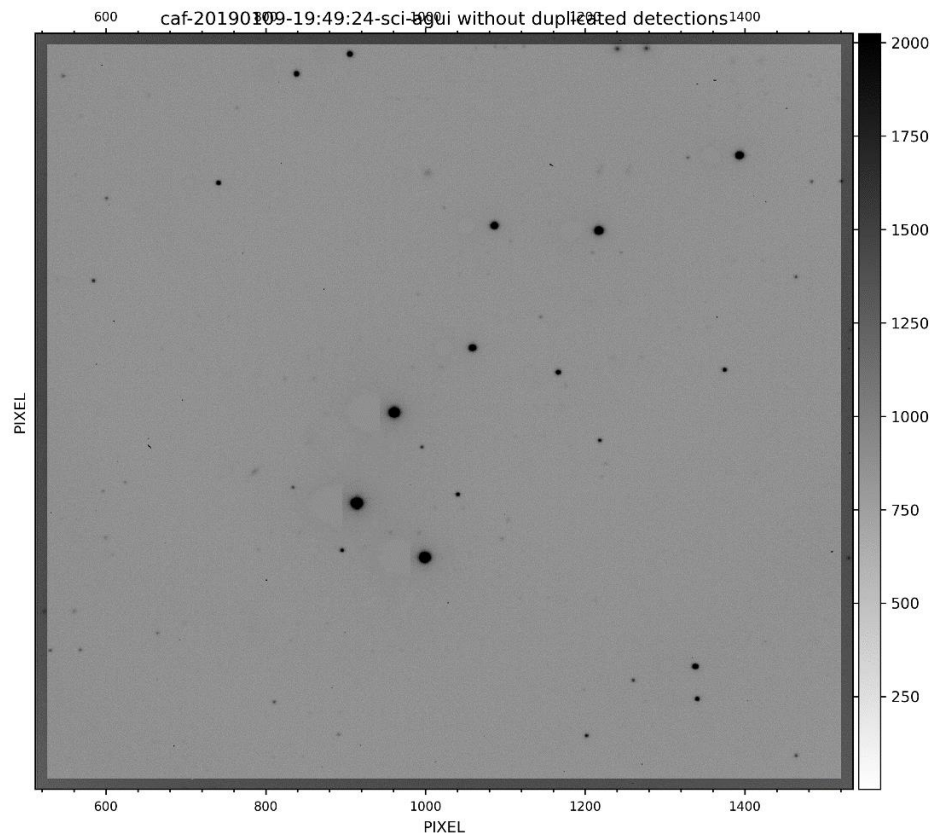


Figura 7. Imagen sin fuentes duplicadas.

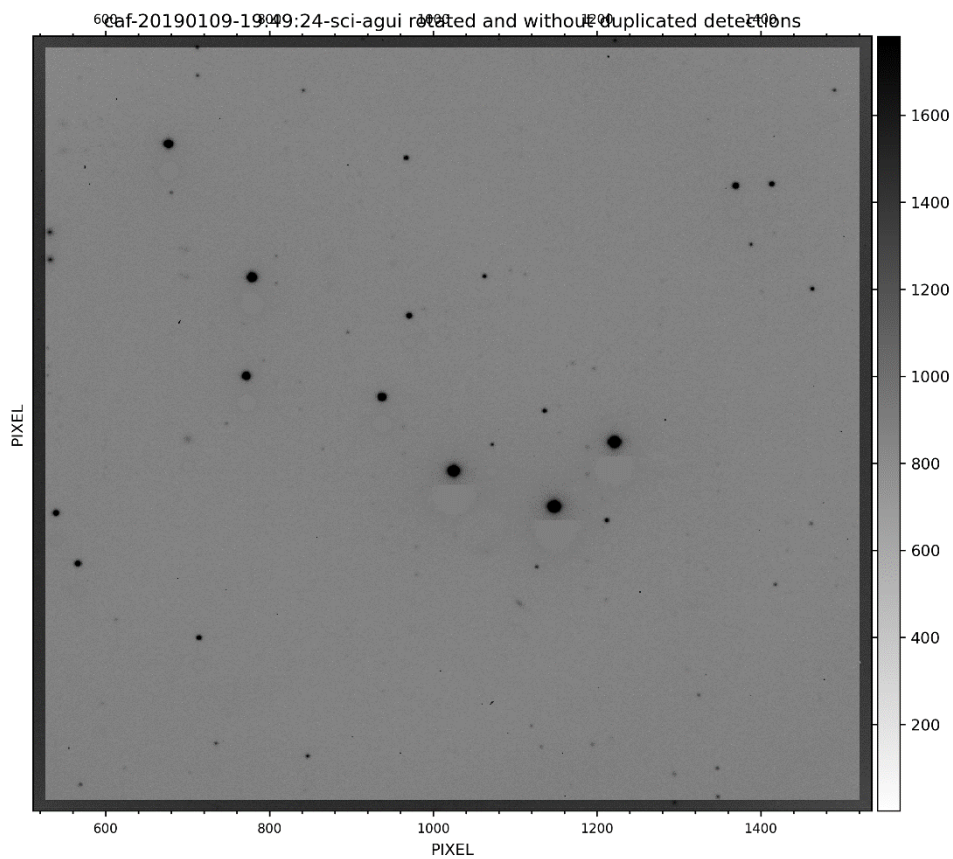


Figura 8. Imagen sin fuentes duplicadas y rotada 90° en sentido antihorario.



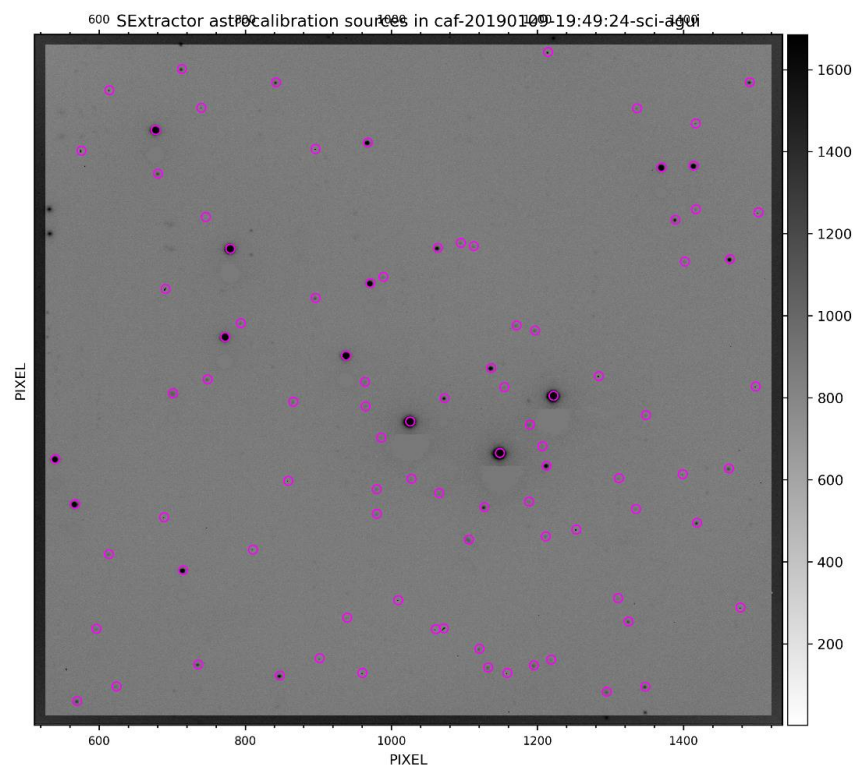


Figura 9. Detección de fuentes que serán usadas para la calibración astrométrica. Se seleccionan fuera del borde de la imagen y las más brillantes en número que nunca supera las 75 (por agilizar el proceso).

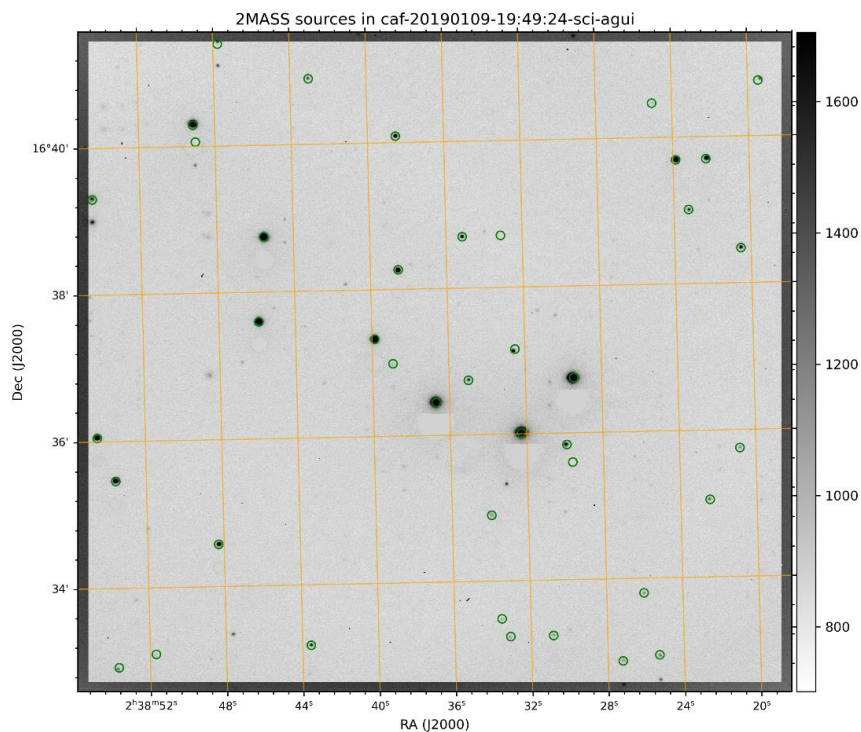


Figura 10. Tras calibrar, se superponen las coordenadas celestes de fuentes de un catálogo consolidado con 2MASS para verificar que la asociación entre coordenadas de pixel y celeste es correcta.

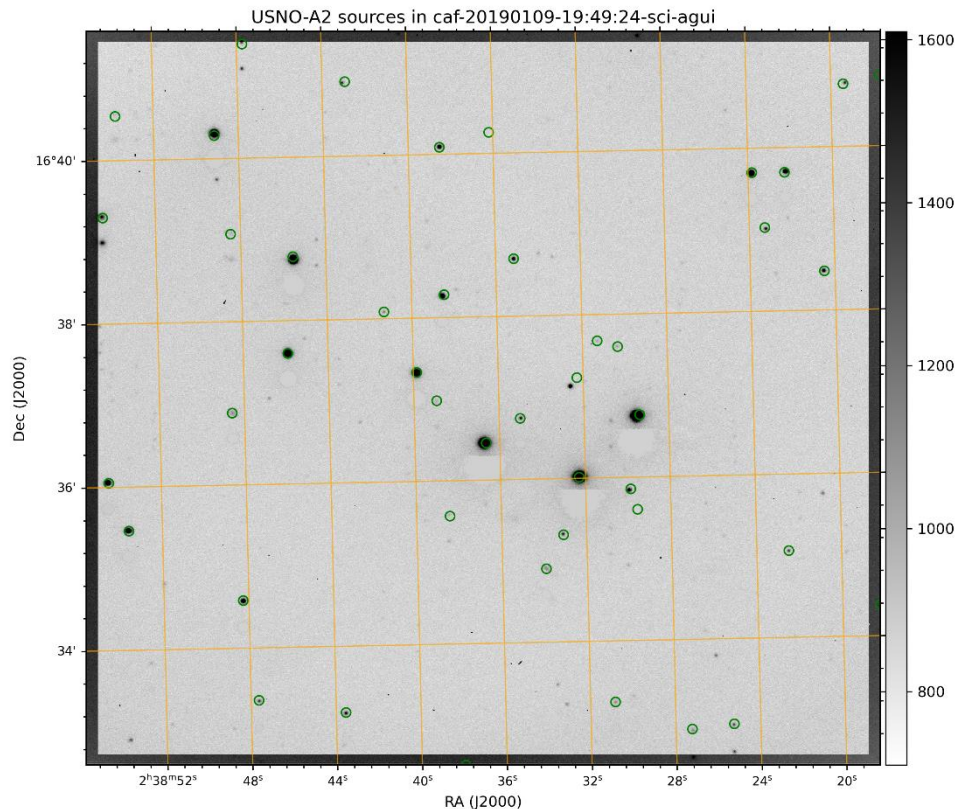


Figura 11. Dado que 2MASS es el catálogo usado para calibrar astrométricamente, conviene visualizar las fuentes de otros catálogos para comprobar el ajuste. Aquí se muestra USNO-A2, pero también se generan imágenes para SDSS-DR9 (cuando hay datos) y NOMAD.

En la calibración fotométrica se intenta eliminar como referencias aquellos calibradores cuya medida de flujo está saturada (en su fuente ordinaria o extraordinaria).

El resultado del proceso es una imagen FITS con nuevos elementos en su cabecera que permiten establecer la relación unívoca entre coordenadas de pixel y coordenadas celestes (calibración astrométrica), así como una relación entre el flujo asociado a una fuente (suma del valor de sus píxeles) y la magnitud (AB) asociada (calibración fotométrica).

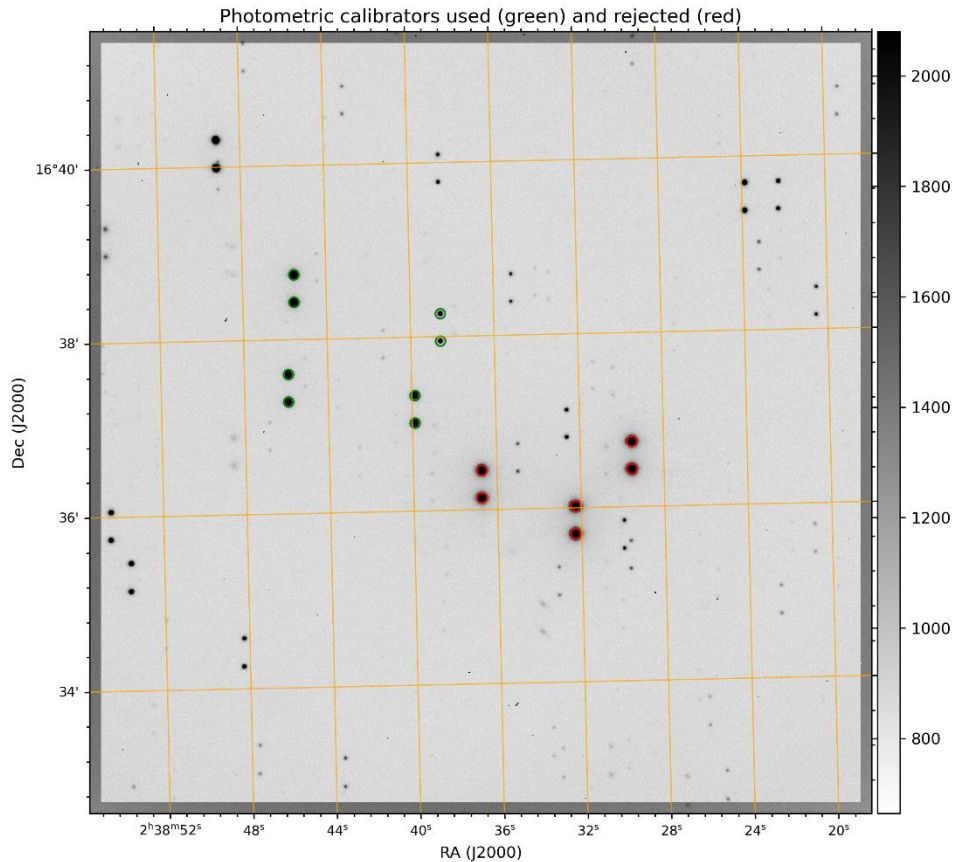


Figura 12. Calibradores aceptados (verde) y rechazados (rojo) para la determinación del punto cero fotométrico. Es uno de los ficheros generados por la pipeline con sufijo `_sources_mc_calib.png`, que se menciona posteriormente.

Como se ha mencionado anteriormente, se genera un subdirectorio para cada una de las imágenes reducidas que se quiere calibrar. Los ficheros de interés son:

- `*_final.fits`, que corresponde al fichero calibrado astrométricamente y fotométricamente.
- `*_astrocal_process_info.csv`, que contiene información de la cabecera del fichero calibrado astrométricamente.
- `*_photocal_process_info.csv`, que contiene parte de la información del fichero anterior más nuevas keywords sobre calibración fotométrica y gráficos generados.
- `*_photocal_res.csv`, con información sobre la fuente de estudio en su versión ordinaria y extraordinaria. Estas medidas se emplean para estimar los parámetros de polarimetría.
- `*_sources_mc_calib.png`, que muestra las fuentes empleadas en la calibración y si están saturadas (rojo) o no (verde). Las imágenes de fuentes de calibración de tipo HD siempre están saturadas.

donde '\*' se corresponde con el nombre del fichero reducido sin la extensión `.fits`.

Existen, además, varios ficheros FITS que se generan durante el proceso de filtrado de mediana, eliminación de fuentes duplicadas y rotación, así como las correspondientes figuras en formato PNG

que permiten su visualización rápida. Otros ficheros corresponden a catálogos de detecciones de fuentes localizadas por *SExtractor*.

### 3.3.3. Determinación de parámetros polarimétricos

Dado que se han obtenido las coordenadas, flujos y demás parámetros identificados como de utilidad de las fuentes problema (ordinaria y extraordinaria) para cada imagen obtenida en el paso anterior (ficheros \*\_photocal\_res.csv), lo que se hace en esta fase es agrupar aquellas que corresponden a cada blazar/estrella HD en los cuatro ángulos de polarización. Se realizan las correspondientes transformaciones y operaciones (explicadas en el artículo de *Zapatero-Osorio, 2015*) y se obtiene un fichero a la salida con información sobre la polarización y su error y el ángulo de polarización y su error para cada una de las fuentes problema observadas. Se trata de un único fichero por noche, aunque la información final es accesible a través de dos ficheros de texto con extensiones .res y .csv.

Si hay fuentes de calibración polarimétrica (estrellas HD), se pueden comprobar los valores esperados con los obtenidos a fin de verificar el buen funcionamiento de la pipeline. Estas fuentes tienen medidas de polarimetría y ángulo de polarización definidas en el fichero de fuentes.

El fichero de interés es:

- `[root_data]/final/MAPCAT/YMMDD/MAPCAT_polR_YYYY-MM-DD.csv`, que contiene entre otras las mediciones de la polarización, el ángulo, la magnitud R y los errores de las anteriores magnitudes para cada una de las fuentes observadas en la noche.

### 3.3.4. Inserción de información en la base de datos

Con todos y cada uno de los ficheros de interés de formato CSV mencionados anteriormente, la última parte de la pipeline se encarga de insertar los nuevos datos de forma ordenada en las correspondientes tablas de la base de datos.

El registro de **nueva información** se realiza según la siguiente secuencia:

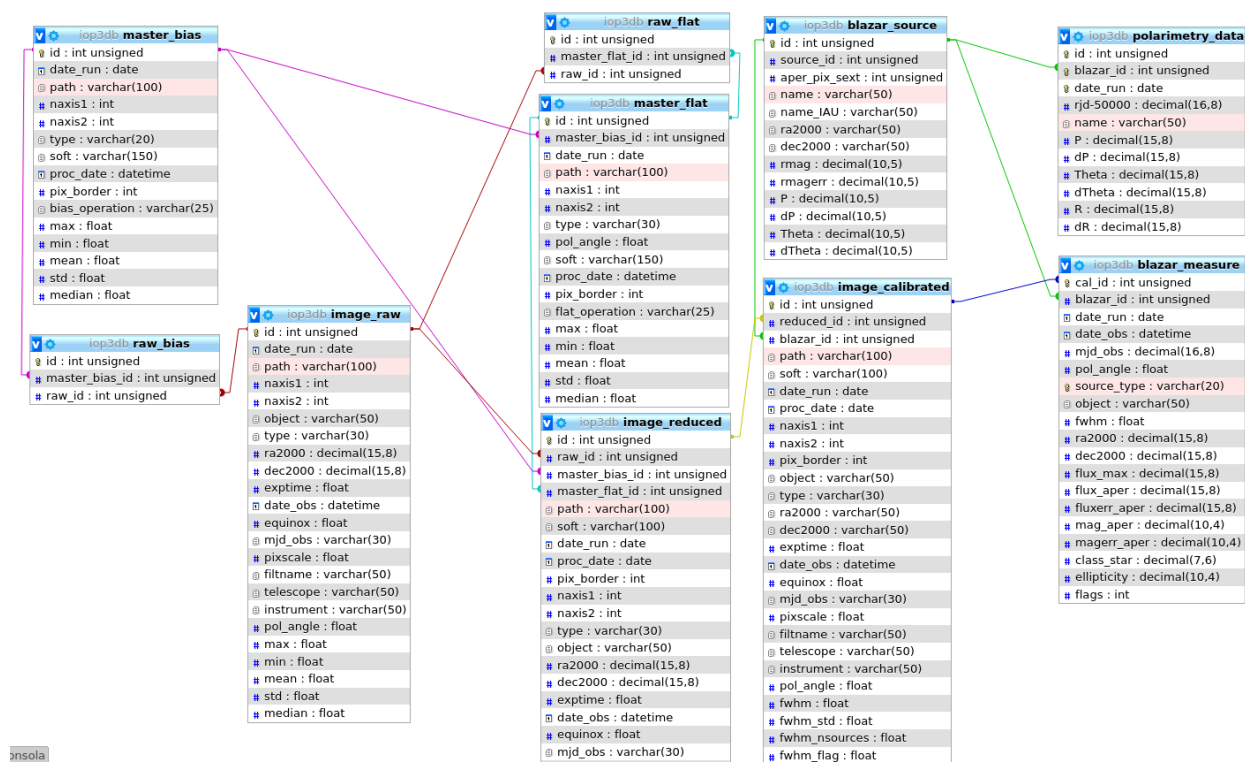
1. Imágenes RAW
2. MasterBIAS generados
3. Relación entre ficheros RAW y MasterBIAS
4. Master FLATS
5. Relación entre ficheros RAW y MasterFLATs
6. Imágenes reducidas
7. Imágenes calibradas astrométrica y fotométricamente.
8. Medidas asociadas a cada Blazar o Estrella HD de calibración
9. Medidas de polarización, ángulo de polarización, magnitudes y los correspondientes errores.



**Nótese** que puede darse el caso de que haya más imágenes RAW (las obtenidas en el telescopio) de tipo “ciencia” (blazars y HDs) que imágenes calibradas. Eso se debe a que la calidad de la imagen es mala o el apuntado (centro de la imagen) no está suficientemente próximo a una de las fuentes problema que figuran en uno de los ficheros de configuración de la pipeline.

**Nótese** que puede haber más ficheros calibrados de los necesarios para determinar la polarimetría. Esto puede deberse a que se han repetido observaciones en una noche. Siempre se tomarán las últimas imágenes para determinar la polarimetría.

La estructura de la base de datos se muestra en la figura adjunta.



Hay *relaciones redundantes* (diseño “no normalizado” en la jerga de base de datos). Es decir, que la misma información (mismo campo) está en varias tablas. Se ha optado por esta vía

- para **simplificar las consultas**. De no ser así, habría que incluir varias tablas para obtener la información deseada.
- También porque se ha pensado en que sólo se van a realizar **modificaciones de la base de datos usando scripts** convenientemente testados, eliminando así cualquier posibilidad de generar información incongruente entre tablas.

Junto a este documento, se adjunta el informe generado por *phpMyAdmin* al respecto de la estructura de la base de datos (tablas, relaciones, claves principales y foráneas, disparadores de acciones automáticas...), así como información sobre cada uno de los campos (tipo, descripción...).