

Manual para reducción de datos, procesamiento de imágenes y análisis fotométrico de apertura sintética con el uso de DS9 y IRAF.

Carlos Gajardo Silva
carlosanga2016@udec.cl

20 de diciembre de 2018

Índice

1. Introducción	4
2. Método	4
2.1. Ordenamiento de los archivos	4
2.1.1. Carpeta Maestra	4
2.1.2. Creación de listas de imágenes	4
2.2. Reducción/Limpieza de imágenes	6
2.2.1. Ejecución de IRAF	6
2.2.2. Corrección por OVERSCAN de todas nuestras imágenes y obtención del DATASEC/TRIMSEC	7
2.2.3. Determinación de la zona de OVERSCAN (BIASSEC) y de la zona de datos útiles (DATASEC/TRIMSEC)	11
2.2.4. Modelación de OVERSCAN y separación TRIMSEC-BIASSEC	19
2.2.5. Combinación de imágenes de reducción/limpieza	24
2.2.6. Generación del MASTER BIAS y MASTER FLAT	24
2.2.7. Corrección por BIAS (zerocor) y corrección por FLATS (flatcor)	28
2.2.8. Saturación y obtención de centros de brillo	29
2.2.9. Análisis de apertura	35
3. Agradecimiento especiales	43

Prefacio

Como bien dice el título del presente documento, éste pretende ser una guía base para el análisis fotométrico de apertura sintética (que tiene todo su potencial centrado al estudio fotométrico estelar). Aquí, se muestran los pasos a seguir para obtener los datos instrumentales necesarios para realizar nuestro análisis fotométrico, partiendo de la premisa de que el lector ya ha instalado y verificado el correcto comportamiento de los software DS9 y IRAF y, además, habiendo entendido teóricamente que son cada una de las imágenes que utilizaremos. Si aún el lector no tiene instalado alguno de los dos software mencionados puede dirigirse al link, [link](#), en el cual se encuentra un archivo descargable (o legible) que, en su contenido, expone una posible forma de instalación de ambos software. Si el lector tiene dudas sobre algún punto de este documento, se pide, encarecidamente, enviar un mensaje al correo que aparece arriba. Se aceptan retribuciones.

1. Introducción

Es astronomía, la fotometría de apertura sintética mediante CCD es una técnica que nos permite obtener las magnitudes aparentes de forma instrumental de objetos astronómicos y es potencialmente efectivo para estrellas que se encuentren relativamente aisladas.^{en} el cielo nocturno. Existen muchas técnicas en concreto que nos permite realizar esta tarea pero aquí se muestra una en particular mediante el uso de software anteriormente indicado.

2. Método

2.1. Ordenamiento de los archivos

2.1.1. Carpeta Maestra

Cuando hayamos obtenido nuestras imágenes capturadas por nuestra CCD, en el telescopio, lo ideal es colocar todo en una carpeta representativa al proyecto que estemos llevando a cabo. Se recomienda fuertemente que el nombre de la carpeta en la cual tendremos nuestras imágenes sea corto y no tenga espacios (esto para motivos de comodidad), por ejemplo, si nuestra carpeta esta nombrada como

- **Carpeta 1**, puede traer problemas a la hora ejecutar iraf, pero,
- **Carpeta1 o Carpeta_1**, son nombramientos más conveniente, ya que nos evitan ciertos problemas que pueden existir.

En resumen, es conveniente tener algo así:

2.1.2. Creación de listas de imágenes

Las tareas que realizaremos para la reducción/limpieza de nuestras imágenes, en general, requieren de listas de estas mismas imágenes para ser ejecutadas. El proceso de la generación de dichas listas se muestra acontinuación:

- Volvemos (con nuestro cursor) a nuestra carpeta donde están nuestras imágenes y abriremos una nueva terminal (sin cerrar nada de lo anterior)
- En esta nueva terminal, para crear listas con archivos que están en la carpeta tecleamos, consecutivamente,

ls inicialesdelarchivo.externsióndelarchivo > nombredelalista.list*

- Lo anterior debe realizarse para cada tipo de imagen sean FLATS, BIAS, imágenes de ciencia, estándares y otra lista que contenga todas nuestras imágenes y, además, en cada filtro (excepto los BIAS), es decir, debemos crear listas para los objetos de ciencia y estándares en cada filtro y listas para los FLATS en cada filtro.

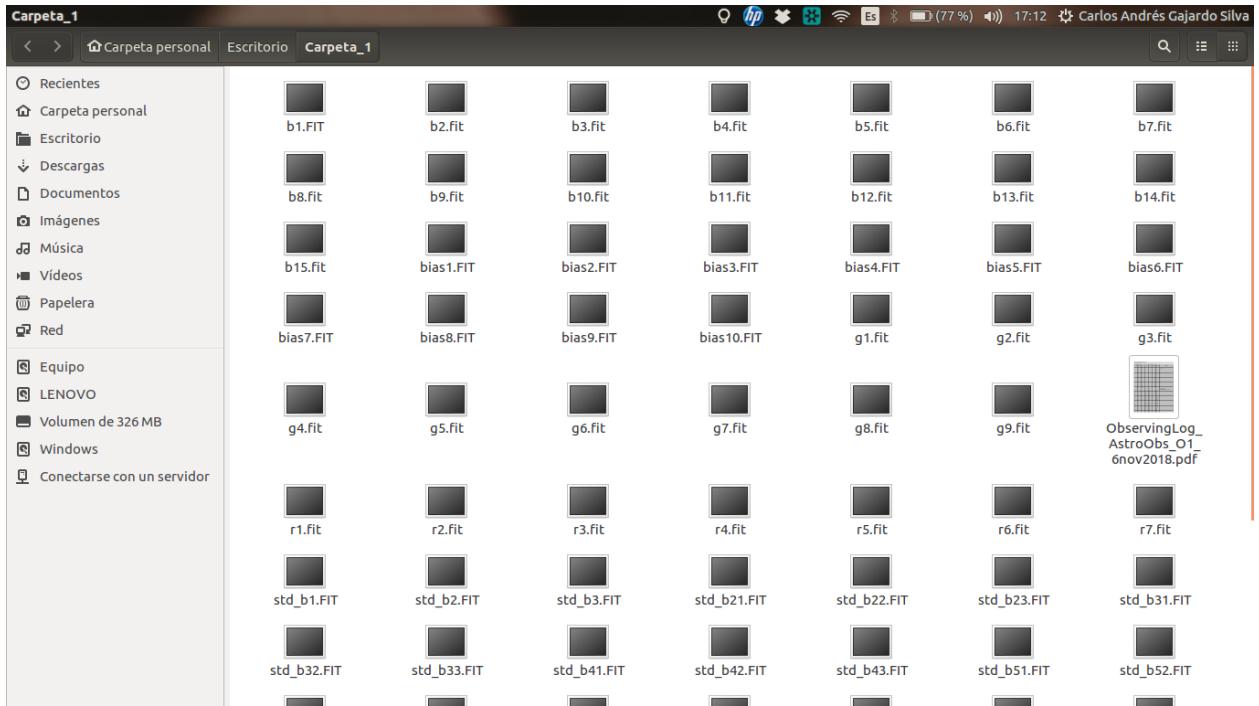


Figura 1: Carpeta para guardar nuestros archivos

- Luego de acabar el proceso anterior, en nuestra carpeta, se abrán creado N listas (por ejemplo, si el proceso se llevó a cabo N veces) que son, en realidad, archivos de texto comunes y corrientes, pero que necesariamente necesitan llevar la extensión `.list`.
- Dichas listas, tambien pueden crearse de forma manual, creando un archivo de texto común (en la carpeta que estamos trabajando) que contiene los nombres de los archivos respectivos (los que debe llevar y con sus extensiones correspondientes) ordenados en columna (hacia abajo, es decir, se escribe el nombre de su archivo con su extensión correspondiente, luego se presiona la tecla INTRO/ENTER y se escribe el siguiente archivo de la misma manera y así sucesivamente).
- Observación: Para los archivos de extensión `.fit`, `.fits`, `.FIT` o `.FITS`, dicha extensión puede de ser cambiada a una de las anteriormente mencionadas simplemente renombrándolas y cambiando su extensión, es decir, por ejemplo, si tenemos una archivo llamado `x.fit`, este puede ser renombrado a `x.fits`.

Lo anterior hecho se muestra acontinuación:

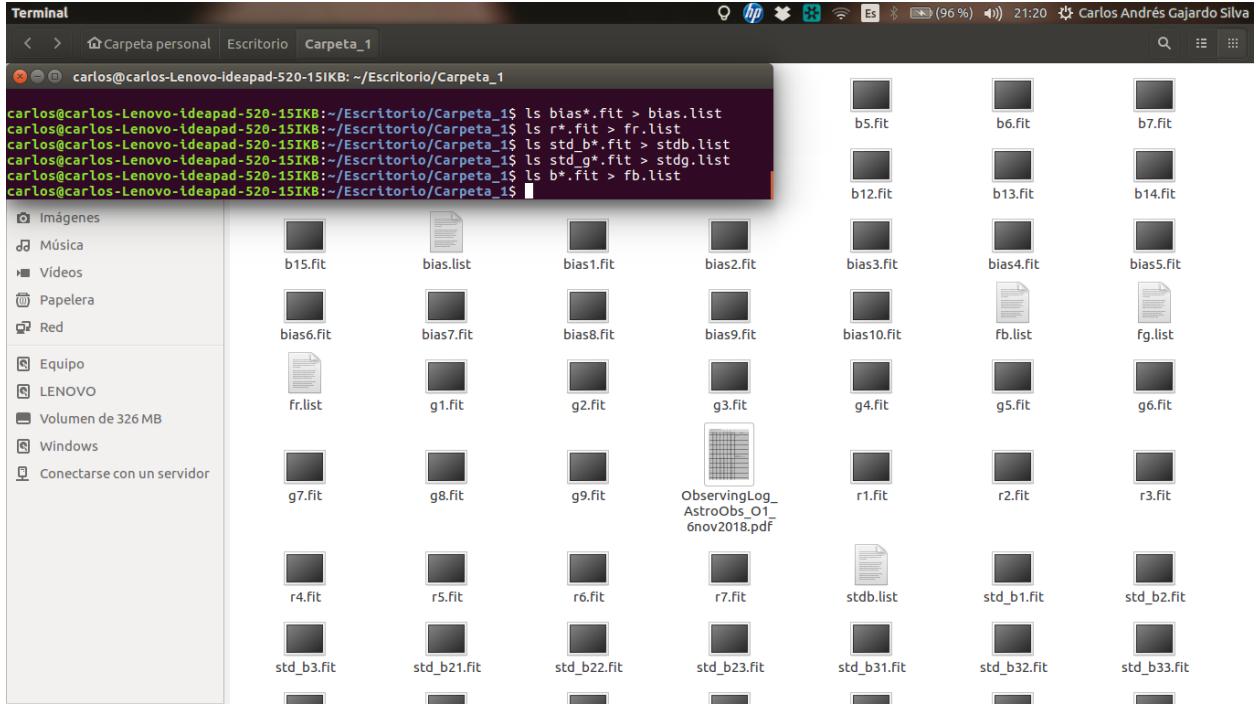


Figura 2: Creación de listas.

2.2. Reducción/Limpieza de imágenes

Es ahora cuando comienza la parte entretenida. Como sabemos, cuando hacemos una captura de imagen con nuestra CCD esta presenta, aun que no necesariamente, dos secciones importantes, el BIASSEC y el DATASEC. Nuestro interés de carácter científico se encuentra en DATASEC, entonces queremos quitar la sección BIASSEC de nuestra imagen, pero, sin embargo, antes de cortar o quitar la sección BIASSEC tenemos que tener en cuenta que esta sección es tambien la sección del OVERSCAN, lo que nos permite modelar el paso de la corriente a traves de los pixeles de nuestra CCD al momento de capturar una imagen. Acontinuación se muestra el proceso completo.

2.2.1. Ejecución de IRAF

Como sabemos, el software que ocuparemos para realizar nuestro objetivo es IRAF, por tanto, debemos ejecutarlo. El proceso de ejecución de IRAF se muestra acontinuación:

- Primero nos aseguramos de ejecutar IRAF en la carpeta en la que están nuestras imágenes. Para esto abrimos una terminal en la carpeta que están nuestras imágenes y luego, en la terminal, introducimos el comando de activación de iraf:

```
source activate iraf27
```

- Luego, desde nuestra terminal (en la misma que se ha escrito el comando de activación de iraf) ejecutamos otra terminal, una terminal interactiva, a conveniencia **xgterm**.

Para esto tecleamos:

xgterm

- En la terminal interactiva, entramos a iraf introduciendo el código

cl

- Con esto ya tendremos IRAF ejecutándose en nuestro ordenador.

2.2.2. Corrección por OVERSCAN de todas nuestras imágenes y obtención del DATASEC/TRIMSEC

En general, todas nuestras imágenes son, a conveniencia, generadas, electrónicamente, con una zona de OVERSCAN para modelar el paso de la corriente a través de los píxeles de nuestra CCD, que de cierta forma perturba el análogo eléctrico (imagen) con 'ruido' que no queremos a la hora de hacer ciencia. Por otro lado, esta zona no nos entrega nada más de información acerca de nuestro objetivo aparte de lo ya mencionado. Es por esto que luego de modelar el paso de la corriente, es conveniente quitar esta zona de todas nuestras imágenes. Este proceso se muestra a continuación volviendo al último punto de la del apartado 2.2.1., es decir, a la terminal **xgterm** donde el último comando introducido fue

cl.

Ahora se muestra una serie de comandos para lograr la corrección por OVERSCAN de nuestras imágenes:

- Luego de haber introducido **cl**, introducimos los siguientes comandos, uno a la vez:

**noao
imred
ccdred**

- El panorama al cual debiéramos estar enfrentados se muestra en la siguiente imagen:

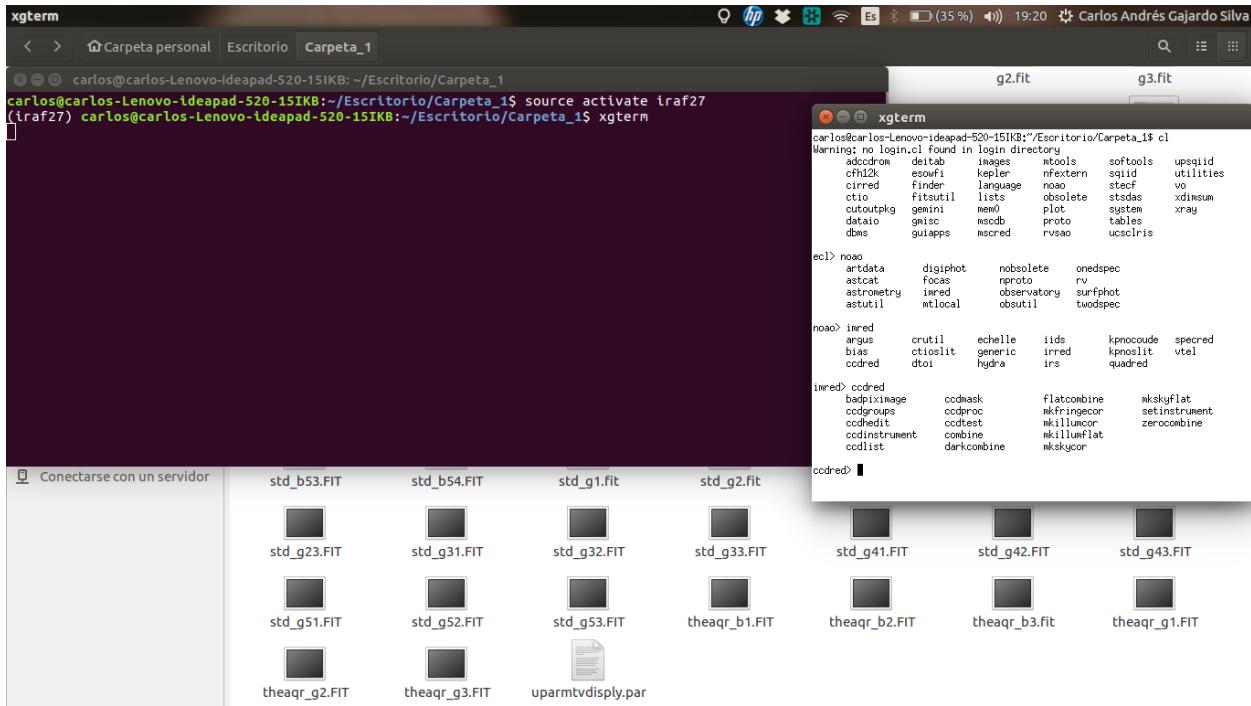


Figura 3: Panorama hasta ccdred.

- Una vez ahí, nos enfocamos en ***setinstrument***, tarea que nos permite realizar nuestro objetivo. Por defecto, esta tarea tiene parametros ya establecidos y no necesariamente los que nosotros necesitamos. Para ver/editar los parámetros por defecto que tiene esta tarea tecleamos

epar setinstrument,

que nos permite editar los parámetros de la sección ***setinstrument*** para acomodarlos a lo que nosotros queremos hacer. Tenga cuidado y sea cauteloso desde ahora. Al teclear esto, debe conseguir algo como esto:

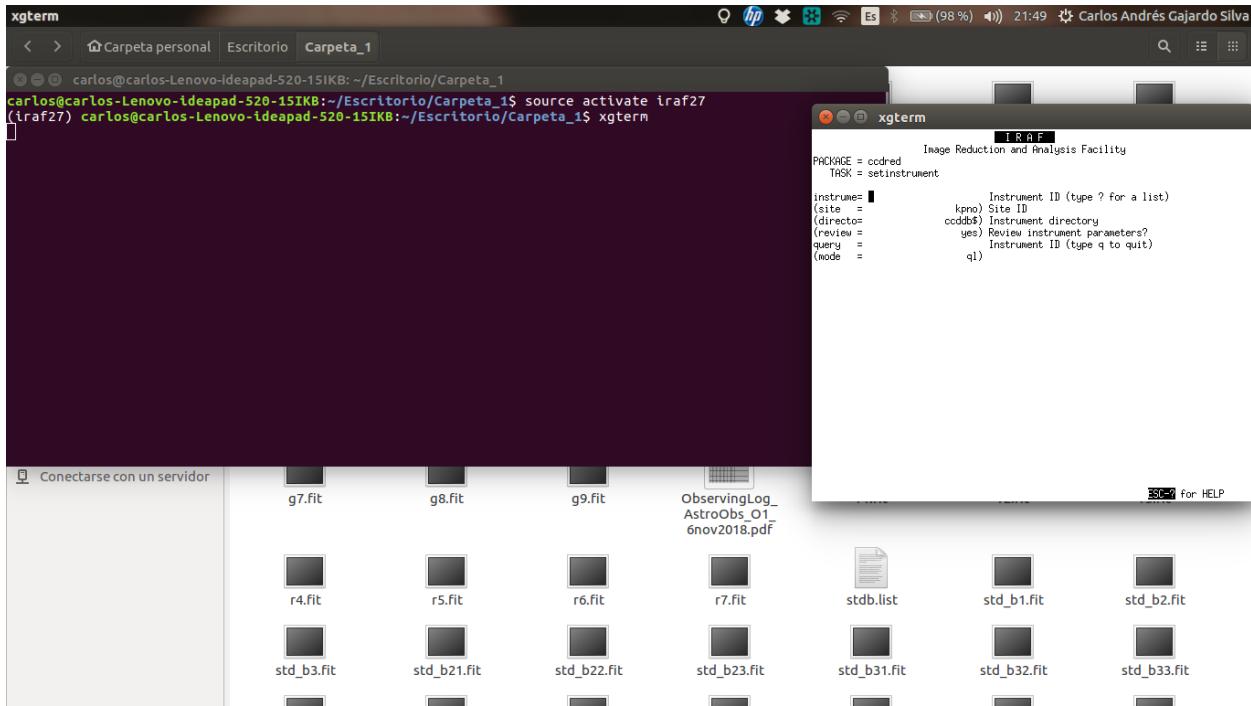


Figura 4: Interfaz de epar setinstrument.

- Luego, en la interfaz de *epar setinstrument* tecleamos :go, sin cambiar nada de los que esta en su interior.
- Si accidentalmente cambió algo, antes de escribir :go, muévase utilizando las flechas del teclado hasta donde algo ha sido escrito, y presione la tecla más común para borrar texto que tenga su computador o bien presione la tecla *suprimir/Supr*.
- Si accidentalmente ha cambiado algo y, además, ha presionado INTRO/ENTER, antes de escribir :go, muévase con la flechas del teclado y hasta donde a escrito algo y luego presione ESPACIO y luego INTRO/ENTER. Luego continue con :go.
- Si accidentalmente cambió algo y tecleó :go, usted puede ser llevado a otra interfaz que, en general, le pedirá cierta información, pero antes de entregarle alguna información lea bien, si en alguna parte dice algo como: "type q to quit", usted sabrá que hacer (si no tiene experiencia con el inglés busque un diccionario). A continuación si logra salir, generalmente, será llevado hasta *ccdred* en donde podra repetir el proceso de *epar setinstrument* desde cero, o bien, enmendar su error.
- En general, no es conveniente editar, sin conocimiento previo, algo de la interfaz de *epar setinstrument*. Infórmese.
- Si usted no se equivocó , es decir, hizo lo que muestra el primer punto despues de la figura 4, le aparecerá un linea en la que usted puede escribir. Dicha linea esta relacionada con la ID del instrumento que estamos usando, sin embargo, acostumbramos a dejar en blanco esta linea y simplemente presionamos INTRO/ENTER. Luego, usted debe llegar a una interfaz similar a la que se muestra acontinuación:

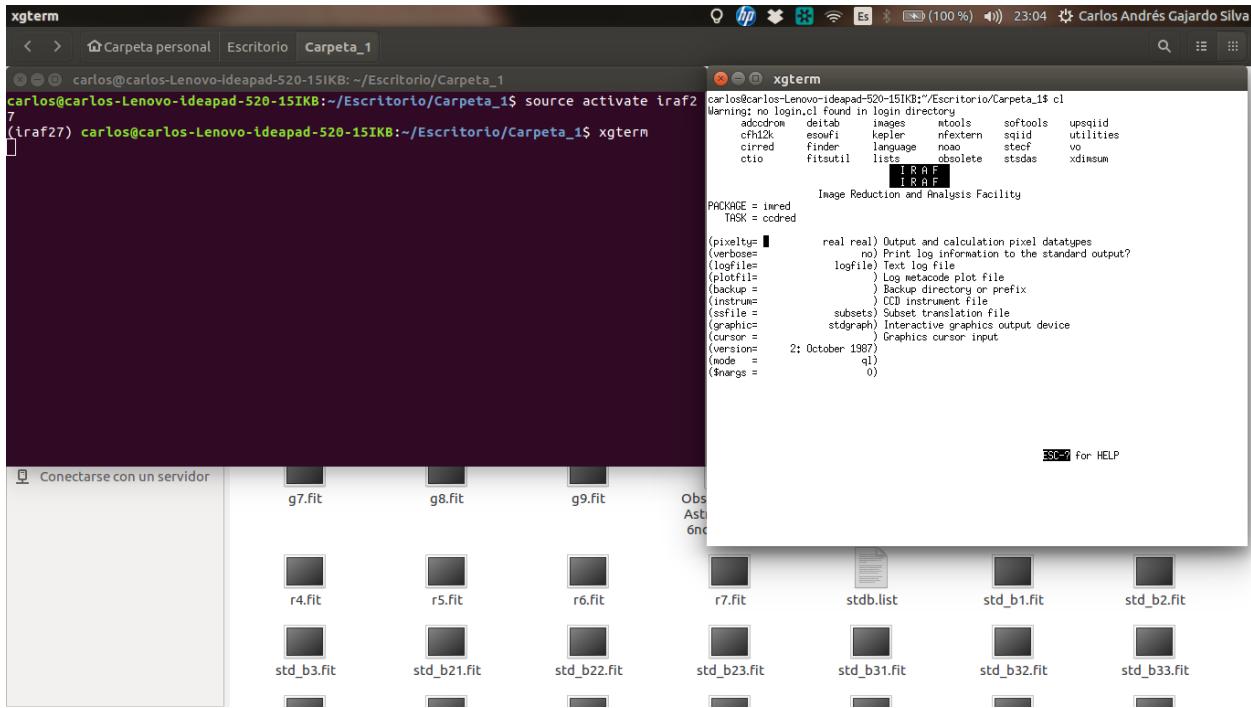


Figura 5: Interfaz luego de escribir :go y INTRO/ENTER, en la sección de *epar setinstrument*.

- En esta interfaz, a conveniencia y por la seguridad de nuestros datos, es conveniente editar la opción **backup** (recordemos que esto se hace utilizando las flechas del teclado para llegar a donde queremos, escribiendo y pulsando INTRO/ENTER), por ejemplo podemos escribir, en **backup**, r"(sin las comillas). Lo que hará esto es crear archivos de respaldo que contienen la misma información que los archivos originales. Dichos archivos se crearán en la misma carpeta en la que se encuentran los archivos originales y estarán escritos con una letra r.^al inicio para poder diferenciarlo. Intuitivamente, ya sabemos para qué podríamos usar dichos archivos de respaldo.
- Luego de editar (o no editar) la opción **backup**, introducimos nuevamente :go. Aquí hay que poner atención. Deberíamos llegar a algo como así:

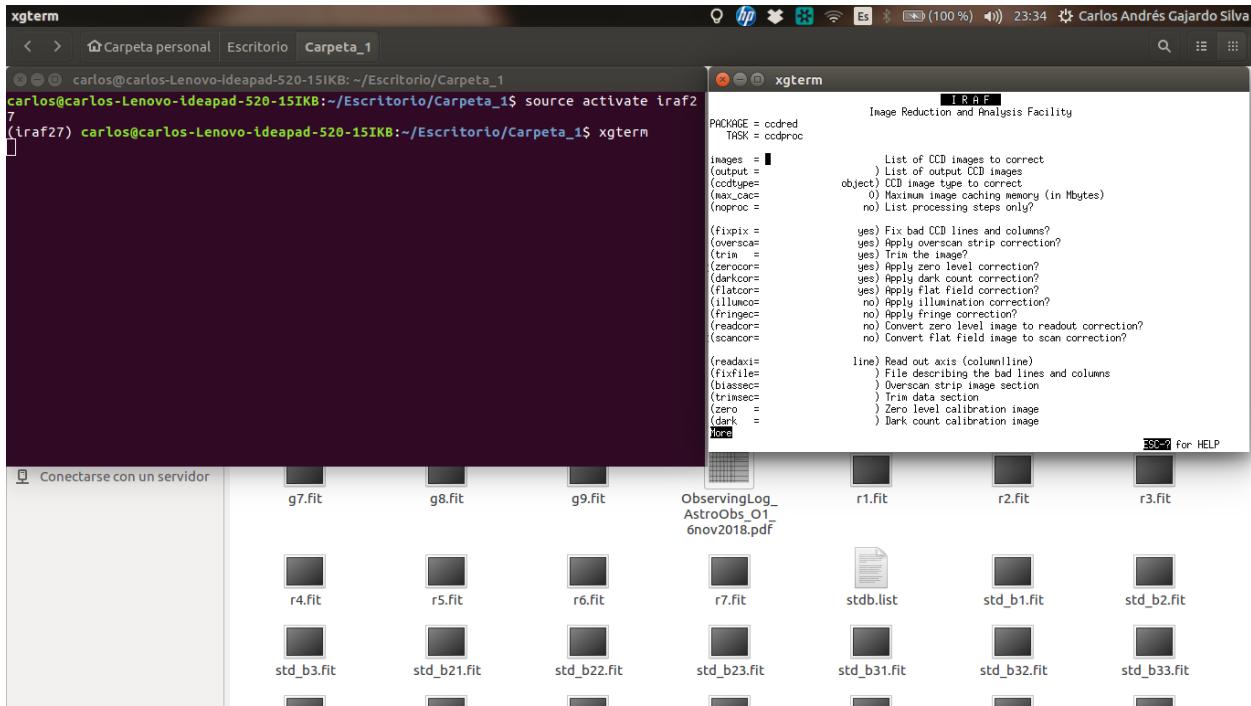


Figura 6: Interfaz 6.

- En la interfaz de la Figura 6, es donde introduciremos las listas que habíamos creado para procesarlas, en este caso comenzaremos con **bias.list**.
- Editamos **images** con **@bias.list**, para que se pueda identificar esta lista (es necesario escribir **@**)
- Editamos **ccdtype** con que no halla nada en esa opción.
- Entre las opciones a editar, **fixpix** y **scancor**, incluyéndolas, todas deben ser editadas con **no**, excepto **oversca** y **trim**, las cuales deben estar editadas con **yes**.
- En la siguiente sección, entre **readaxi** y **nscan**, primero nos preocupamos por las opciones **biassec** y **trimsec**, esto debemos indicarle a iraf para que haga los “cortes” de las imágenes de forma correcta.
- Lo que debemos introducir en **biassec** y **trimsec**, para editar, son básicamente las coordenadas en pixeles”, para que iraf haga los cortes correctos. Estos debemos encontrarlos estudiando las imágenes.

2.2.3. Determinación de la zona de OVERSCAN (BIASSEC) y de la zona de datos útiles (DATASEC/TRIMSEC)

Para, valientemente, salir a encontrar dichas coordenadas del último punto del apartado anterior, debemos dejar de lado por un momento las tareas que estabamos haciendo. Para salir (esperando que nuestras ediciones queden guardadas) de la interfaz de la Figura 6 tecleamos

:q

y posteriormente introducimos, aproximadamente, cinco veces el comando

bye

para llegar al inicio, tal como cuando ejecutamos IRAF, es decir, a **ecl>**

- Una vez ahí, introducimos

plot,

seguido de

implot

- Cuando introducimos **implot**, IRAF nos indicará que debemos introducir el nombre de una imagen. Por conveniencia, abriremos una imagen que corresponda a un FLAT. La razón de por qué usamos un FLAT es sencilla, sin embargo quedará como una interrogante al lector.
- En nuestro caso introducimos la imagen **b4.fit**. Esto abrirá otra interfaz de **xgterm**, esta vez, una interfaz que muestra un gráfico. Dicho gráfico indica el número de cuentas de un píxel en una linea de píxeles.
- **Advertencia: no intente mover ni agrandar la interfaz que se acaba de abrir. Tampoco la cierre.**
- Esto se ve algo así:

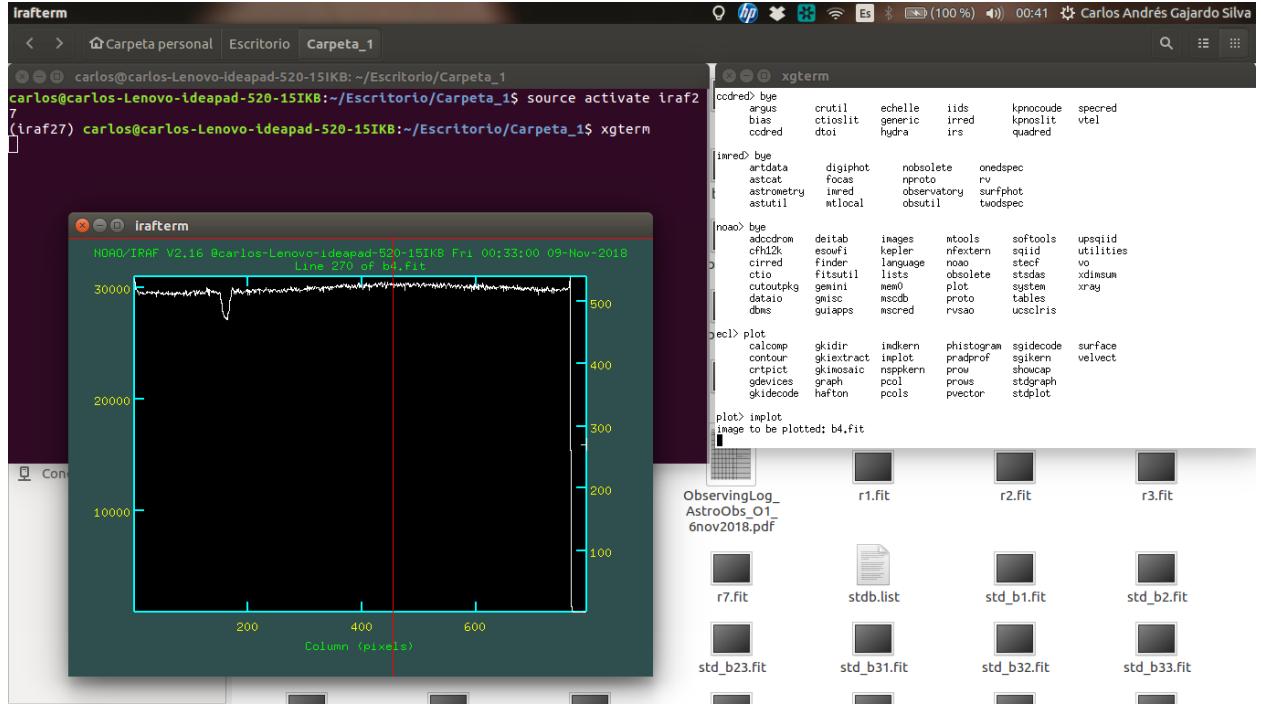


Figura 7: Interfaz gráfica: Número de cuentas v/s pixel en fila 270 de pixeles.

- **Como indican las Figuras 8 y 9**, moviendo el cursor sobre el gráfico hacia la esquina inferior derecha en donde pulsaremos la tecla 'e' (una vez) y luego moviendo el cursor hacia la esquina superior derecha del grafico, en donde nuevamente pulsaremos 'e' (una vez). Esto nos dará una visión más amplia para poder diferenciar la zona de datos útiles y lo que debemos cortar de la imagen. Gráficamente,

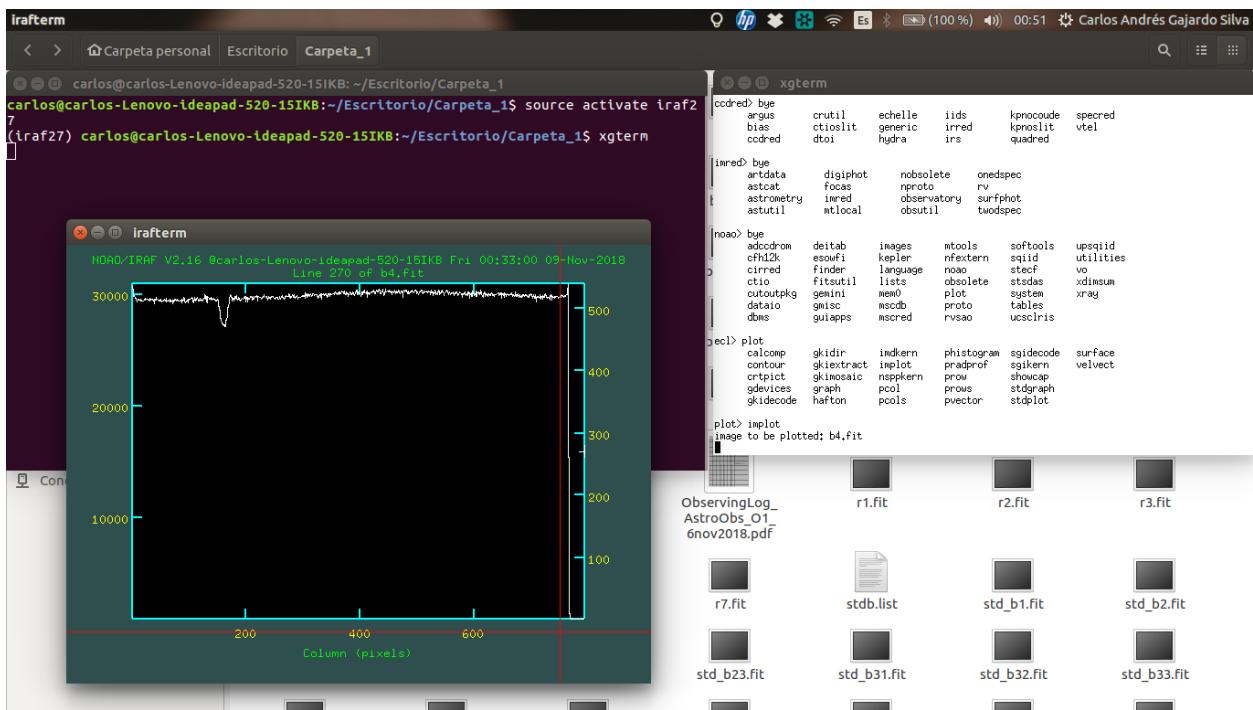


Figura 8: Pulse 'e', una vez, ahí.

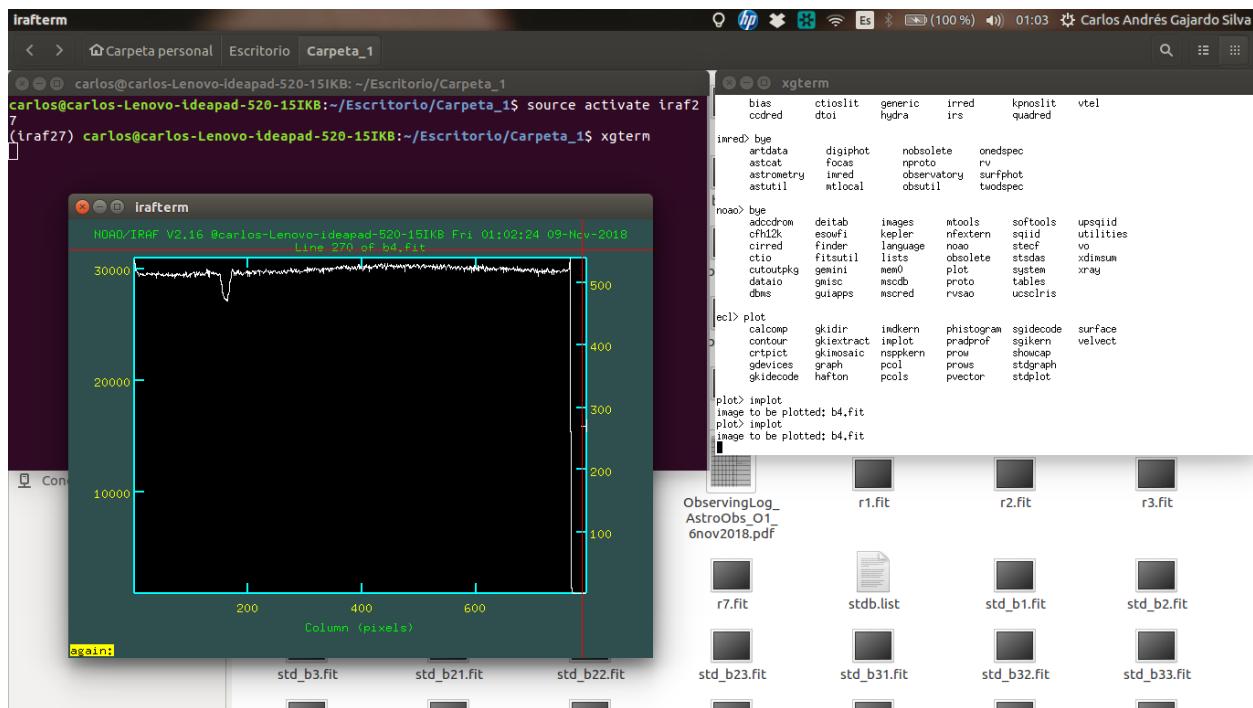


Figura 9: Pulse nuevamente 'e', una vez, ahí.

- Deberíamos conseguir algo como esto:

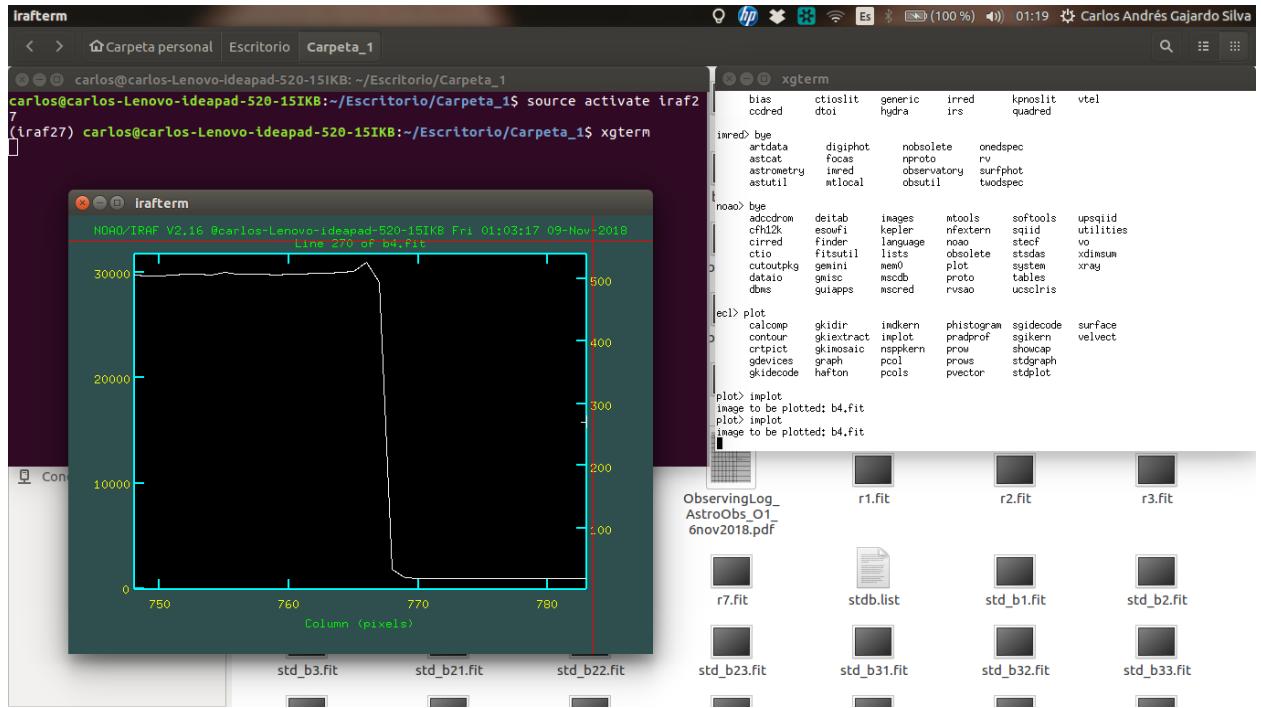


Figura 10: Menudo zoom.

- Luego de esto, procedemos a ubicar el cursor en el centro de la curva casi vertical que se puede notar facilmente. Cuando tengamos el cursor ahí, inmediatamente presionamos (solo una vez) la tecla ESPACIO. Esto nos debería indicar, aproximadamente, el número del pixel de la fila 270 en la cual el número de cuentas por pixel se reduce drásticamente (la curva casi vertical). En nuestro caso, los resultados son:

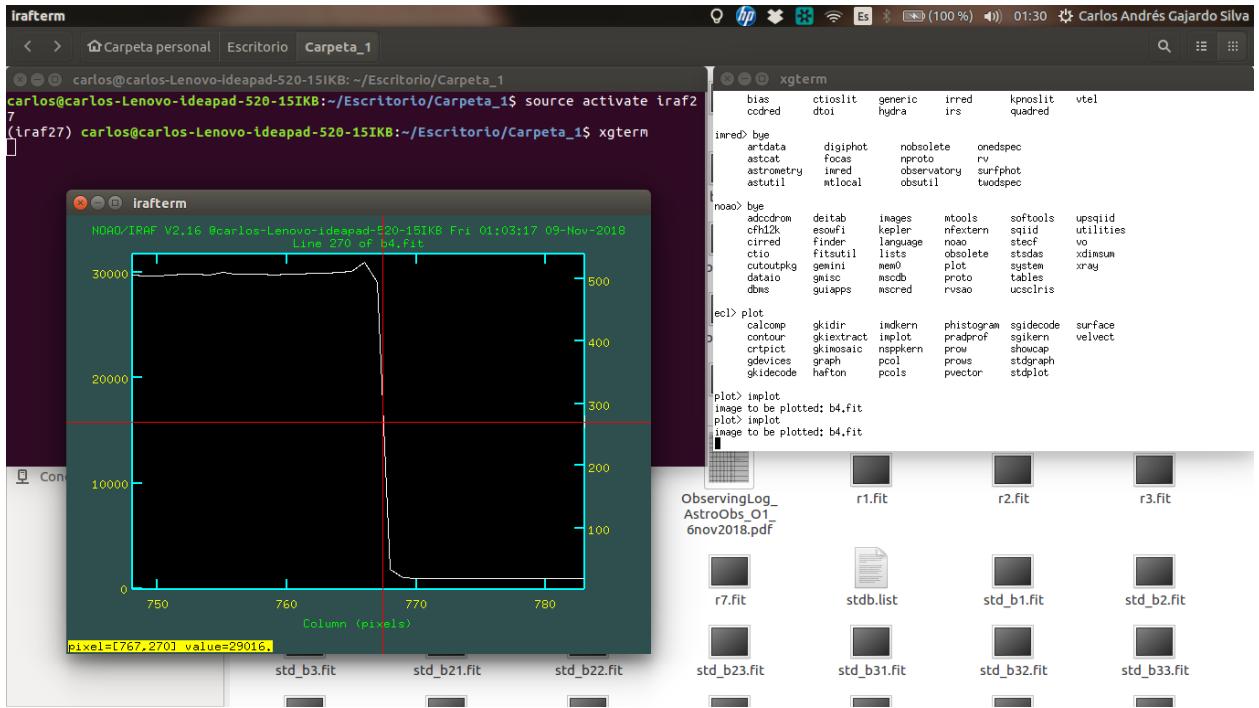


Figura 11: Indica que el del pixel buscado en la fila 270 es el número 767.

- Este número debemos guardar o anotarlo en alguna parte, será de gran importancia ya que nos dice, por ejemplo, hasta que pixel se encuentra nuestra zona de datos útiles.
- Ahora, para analizar una columna de pixeles debemos pulsar 'c' una vez. Con esto obtendríamos lo siguiente:

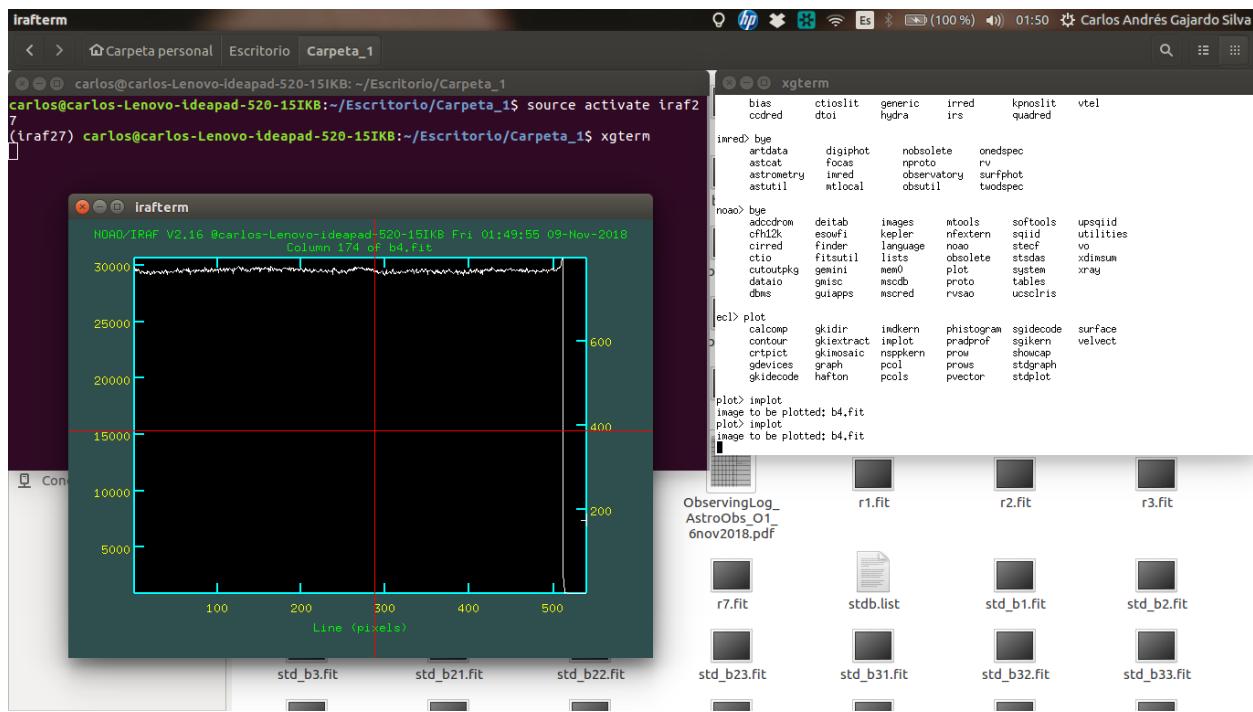


Figura 12: Menudo zoom 2.

- Procediendo de manera análoga, a lo hecho anteriormente, obtenemos que :

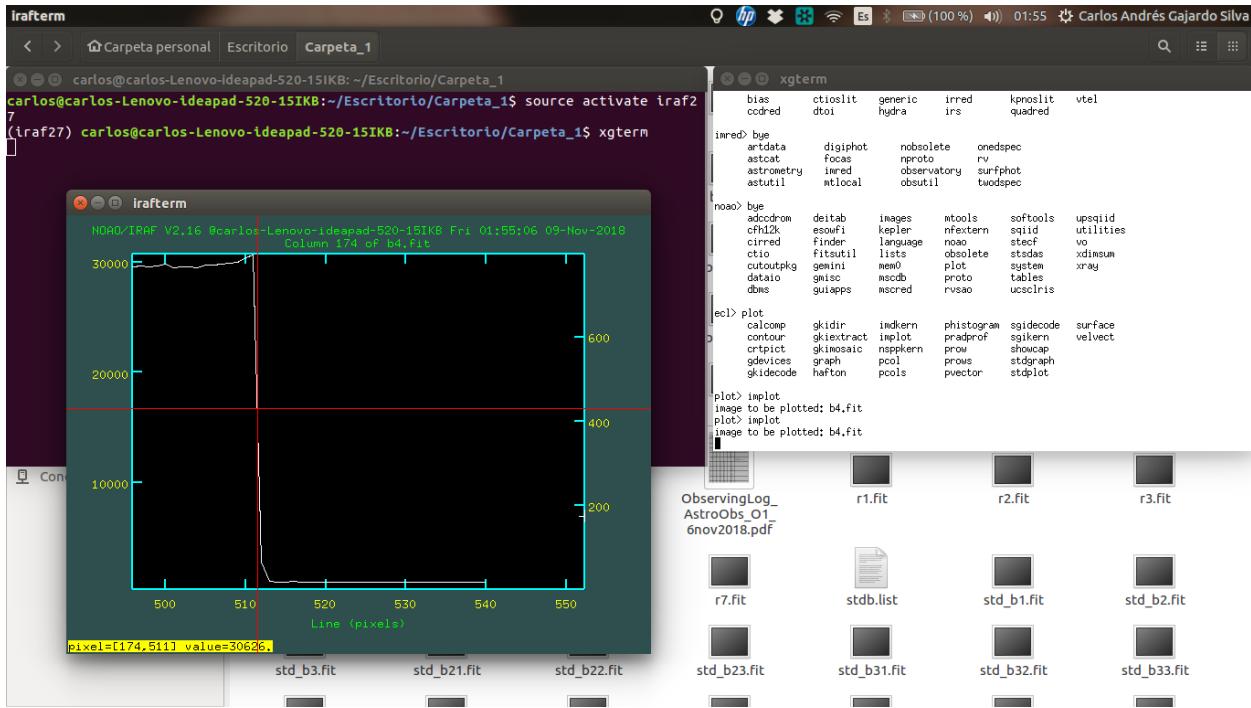


Figura 13: Indica que el del pixel buscado en la columna 174 es el número 511.

- Podemos extrapolar que estos píxeles representativos de una fila y una columna en cada una de nuestras imágenes es el mismo. Las razones del por qué quedan a interpretación del lector.
- Procedemos a guardar, entonces, ambos resultados.
- Ahora para cerrar seguramente nuestra interfaz gráfica primero debemos hacer un click derecho dentro de la gráfica y luego pulsamos la letra **q**. Ahora puede cerrar dicha interfaz como usualmente usted cierra una ventana cualquiera.

2.2.4. Modelación de OVERSCAN y separación TRIMSEC-BIASSEC

Procedemos ahora a ejecutar la tarea pendiente del último punto del apartado 2.2.2.. El procedimiento de indíca acontinuación:

- Considerando el último punto del apartado 2.2.3., una vez cerrada la pestaña de la interfaz gráfica, en nuestro terminal **xgterm** volveremos a la sección *plot*
- Ahí introducimos **bye**, la veces que sea necesario, para volver a *ecl >*.
- Desde ahí, debemos volver a la interfaz de la Figura 6. Una vez en esa interfaz, asegúandonos de que los parámetros están editados correctamente, procedemos a editar **biassec** y **trimsec**. Éstos, los editamos de la siguiente manera:

- **biassec= [768:795,512:540]**

- **trimsec= [1:767,1:511]**
- Luego de haber editado correctamente, más abajo, entre **interac** y **mode**, solo debemos fijarnos en la opciones **interac** y **functio**.
- **interact**, nos ayuda a ajustar manualmente una curva relacionada a **functio**. Si queremos hacer ajustes manuales introducimos *yes*. Si no queremos acerlo de forma manual y que el programa lo haga como está predeterminado por defecto, entonces introducimos *no*
- **functio**, nos ayudará a modelar, con una función la zona de OVERSCAN. En general, la función *legendre* está por defecto, pero esta función no es muy buena para este tipo de datos. En este caso utilizaremos las función *spline3*. Entonces editamos **functio** introduciendo *spline3*.
- Una vez editado correctamente esta sección procedemos con **:go**. Luego, iraf debería dar un mensaje con opción para introducir texto al cual responderemos *yes*. Luego de esto pulsaremos INTRO/ENTER. Nos aparecerá algo como lo que sigue:

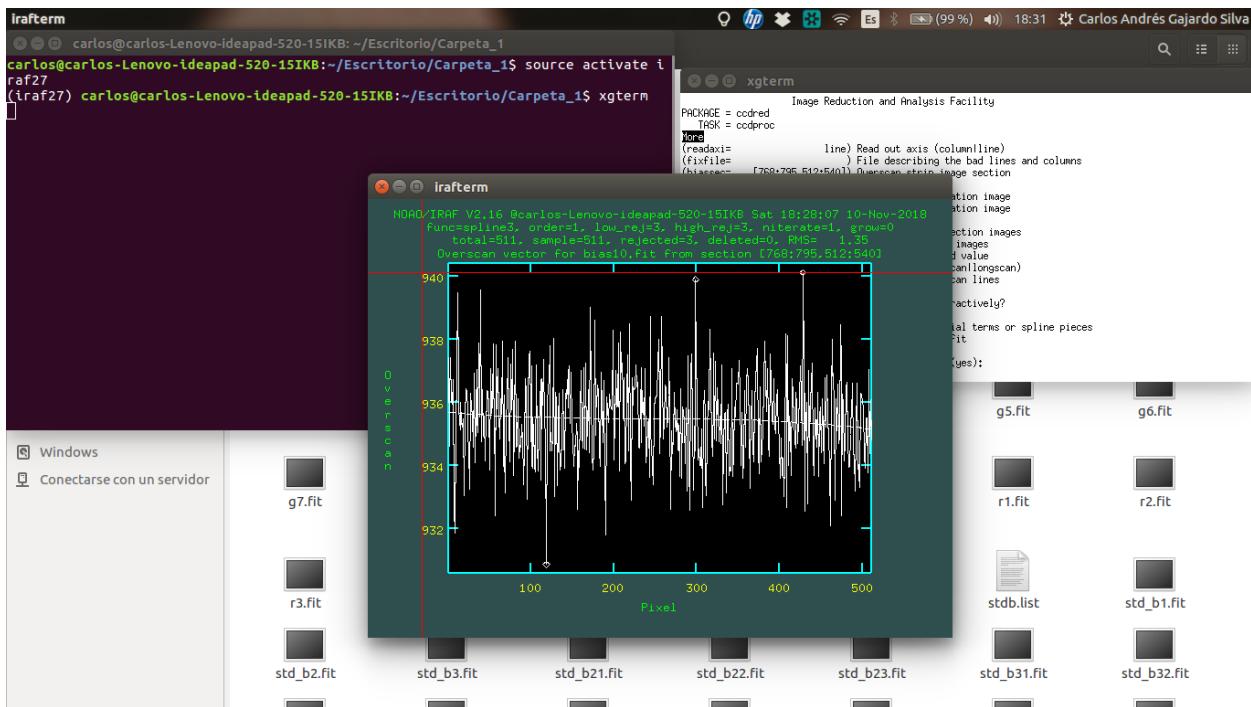


Figura 14: Modelación de OVERSCAN .

- NO SE ASUSTE, este gráfico es su amigo. Este gráfico corresponde al 'ruido' de la corriente que fluye a través de los píxeles de nuestra CCD y en la sección de datos útiles, cosa que no queremos a la hora de hacer ciencia.
- Como podemos ver, hay una especie de curva que recorre horizontalmente en los píxeles, este es el ajuste que modela el ruido BIAS. La idea es elegir una curva que

modele optima esto, una curva no muy plana pero que sea suave. la curva que aparece en la Figura 14 es del orden uno.

- Para cambiar el orden de ajuste, en la interfaz, hacemos click una vez y luego introducimos :o (número del orden deseado) (sin paréntesis), luego para refrescar el gráfico pulsamos una vez la letra f. Por ejemplo:

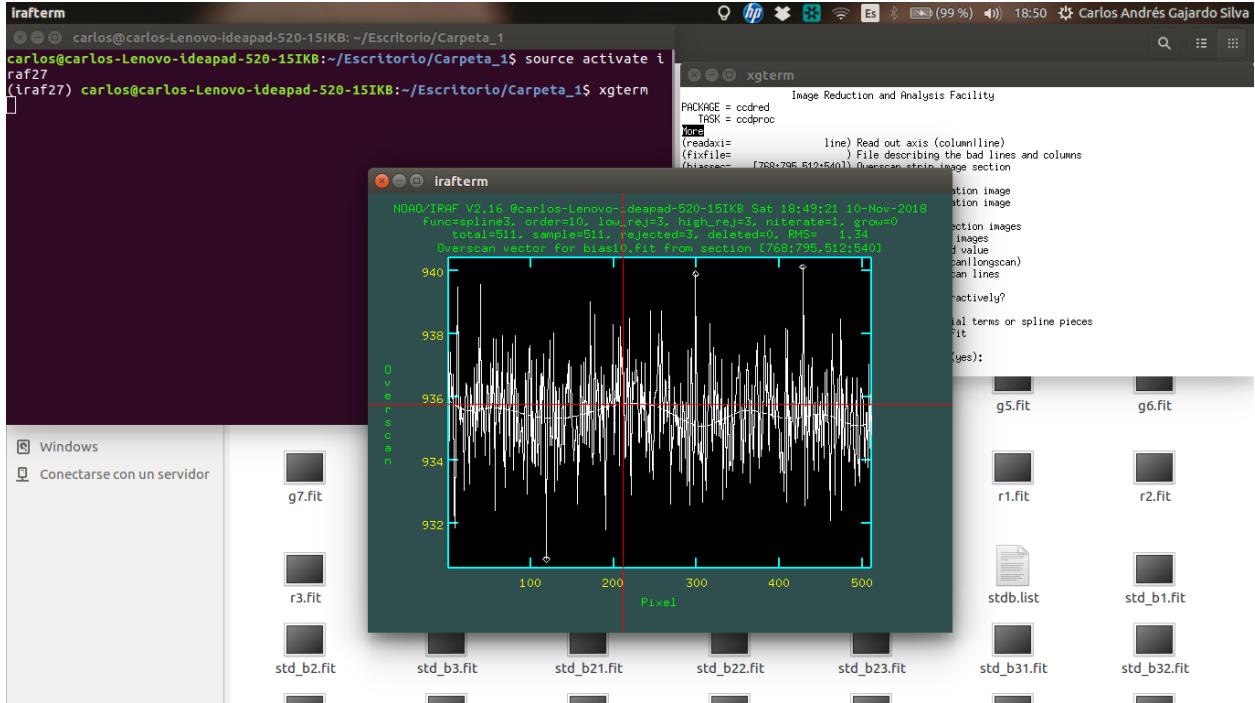


Figura 15: Curva de ajuste de orden 10.

- Si estamos conforme con nuestro ajuste, ahora procedemos a realizar el mismo proceso para cada una de las imágenes, para esto pulsamos, una vez, la tecla q. Luego de esto, en nuestra terminal de **xgterm** nos preguntará nuevamente si queremos pasar a la siguiente imagen. Intuitivamente haremos el proceso de hacer los ajustes para cada imagen.
- Una vez hayamos terminado con todas la imágenes, **xgterm** nos enviará a *ccdred* y ya tendremos nuestros BIAS listos. Esto es:

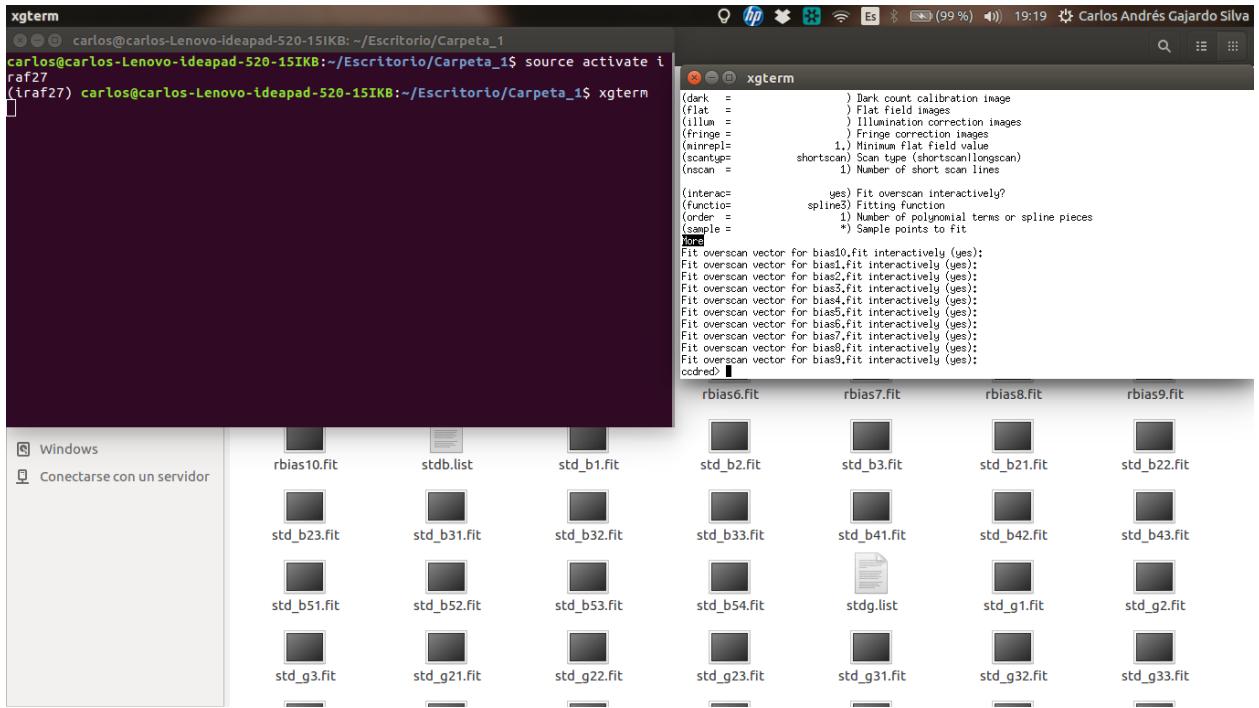


Figura 16: Nuevamente en *ccdred*.

- Un paso no necesario pero suficiente para darnos cuenta de los cambios que hemos hecho es ver nuestras imágenes que han sido reescritas y compararlas con las que han sido creadas para el **backup** (Aquí no se muestra el proceso de abrir las imágenes, sin embargo más adelante se hará):

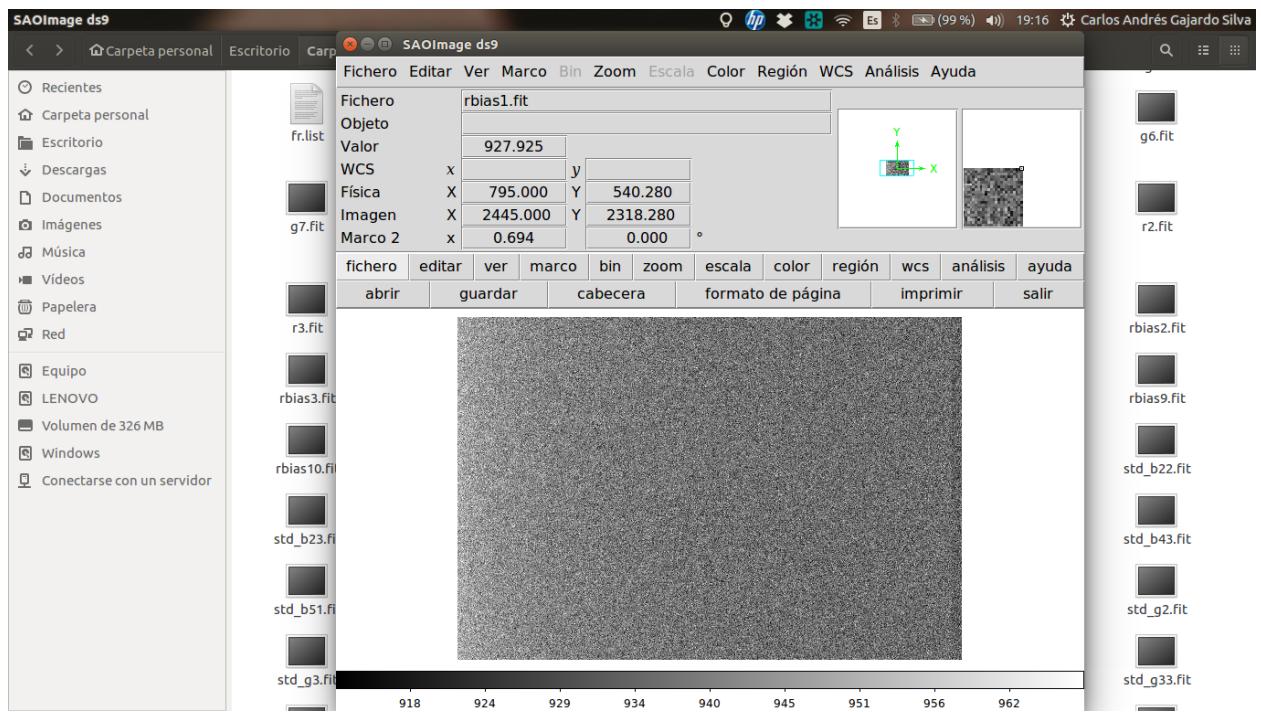


Figura 17: Imagen del backup de bias1.fit.

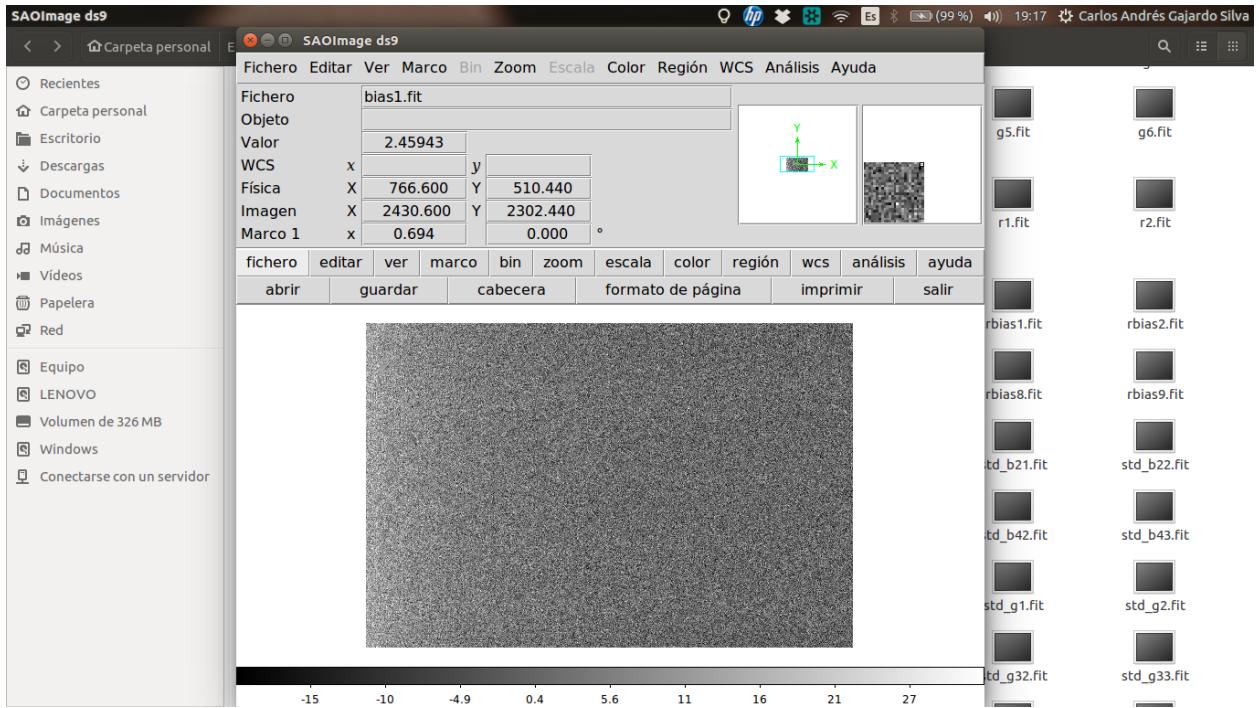


Figura 18: Imagen de la imagen procesada bias1.fit.

- Quizá aquí no se puede apreciar, a simple vista, si se ha hecho un cambio, pero si el lector es detallista se dará cuenta que la imagen efectivamente a ha sido cortada y procesada.
- Debemos repetir el proceso para cada una de nuestras listas. Se asumira que el lector ya esta familiarizado con el proceso que se acaba de realizar, por lo tanto, en este informe se omitirá el proceso para las demás listas.

2.2.5. Combinación de imágenes de reducción/limpieza

Estamos ahora interesados en crear una caracterización estadística de nuestras imágenes, algo que nos ayude, por ejemplo, a caracterizar el ruido de nuestra CCD, o bien, la reacción de nuestros píxeles a la propia luz que estos reciben. Estamos hablando de la creación de un Master FLAT y un Master BIAS, que, por ejemplo, puede corresponder, respectivamente, a un promedio de la información en cada imagen respectiva. Procedemos entonces a la creación de estas estadísticas.

2.2.6. Generación del MASTER BIAS y MASTER FLAT

- Para comenzar, luego de haber terminado con lo anterior, debemos volver a **xterm** (podemos cerrar la terminal anterior y volver a entrar mediante otra integral o, simplemente, volver desde donde quedamos con **bye**).
- Procedemos a volver a la sección *ccdred*. Y nos enfocamos en las opciones *flatcombine* y *zerocombine*.

- *flatcombine* nos permite combinar los FLATS de la forma que queramos y obtener nuestro Master FLAT.
- *zerocombine* nos permite combinar los BIAS de la forma que queramos y obtener nuestro Master BIAS.
- Comenzaremos por crear nuestro Master BIAS. Para esto tipeamos *epar zerocombine*. Debemos llegar a algo como esto:

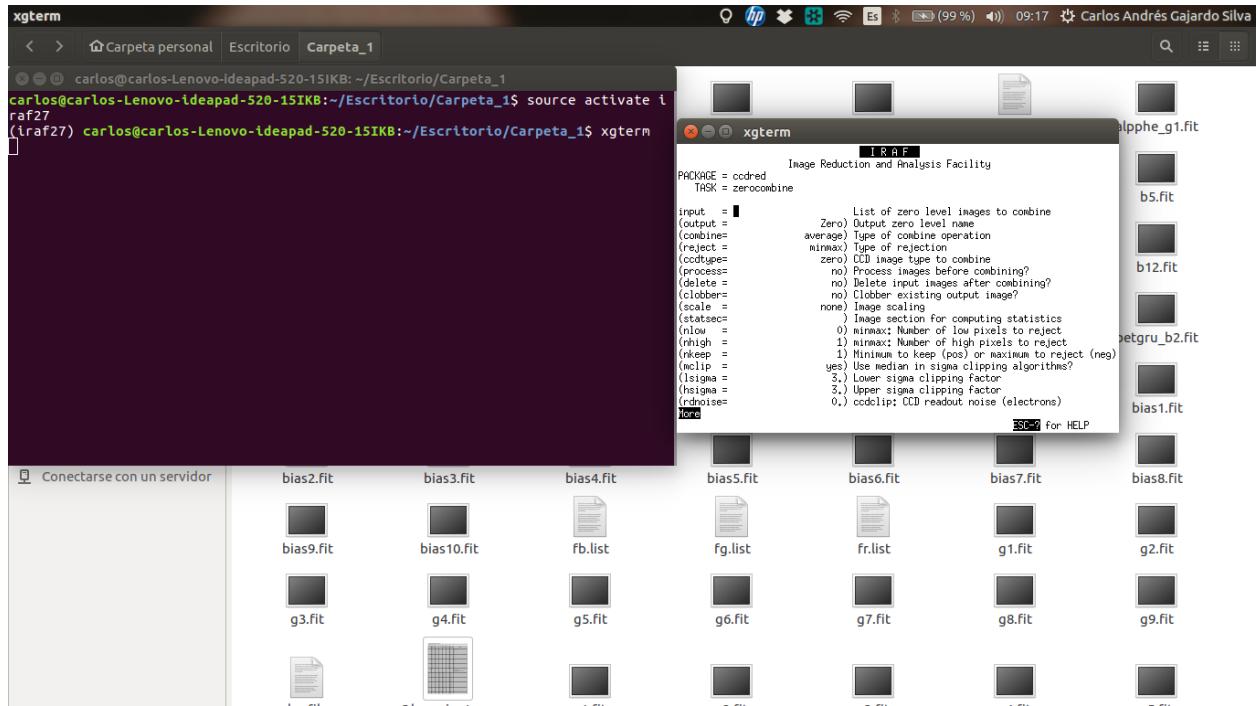


Figura 19: Interfaz de *epar zerocombine*

- El lector ya debe estar familiarizado con este tipo de interfaz.
- Procedemos ahora a editar los parámetros de esta sección. Ponga atención.
- Primero debemos ingresar en **input**, `@nombredelalistadebias.list`
- En **output** debemos ingresar un nombre que será el nombre de nuestro Master BIAS. Por defecto, en general, IRAF lo llama **Zero**. Podemos elegir quedarnos con este nombre, o bien, cambiarlo. El resultado, específicamente, será, si se deja por defecto, un archivo llamado **Zero.fit**, o **Zero.fits**, o **Zero.FIT**, o bien, **Zero.FITS**
- Más abajo, en **combine**, debemos elejir la forma de combinación de nuestros BIAS. Existen muchas formas de combinación, pero en este caso, haremos un promedio, para lo cual editamos con **average**.
- En **reject** editamos como **minmax**. A groso modo esto elimina (o reyecta) los valores que esten muy alejados del promedio de valores.

- En **ccdtype** debemos asegurarnos que este quede en blanco.
- Generalmente, los demás parametros están, por defecto, editados de forma correcta, pero, sin embargo, si tiene dudas, debe asegurarse de consultar a su ayudante.
- Dicho esto, procedemos a realizar esta tarea con :**go**. Una vez introducido :**go**, al cabo de unos segundos (depende de el poder de su computadora), será enviado nuevamente a *ccdred*.
- Antes de proceder, debe asegurarse de que su archivo esperado haya sido creado. Véalo en su carpeta en donde tiene todas sus imágenes.
- Si quiere ver la imagen creada usted tiene, al menos, dos opciones de hacerlo y ambas es utilizando DS9.
- Puede abrir DS9 directamente desde **xgterm**, primero volviendo a *ecl >* y luego tipeando **&ds9 !**.
- Si lo anterior parece no funcionar, vaya a su carpeta, habra una nueva terminal y escriba **ds9**.
- Si no puede ejecutar DS9 de ninguna forma debe consultar.
- En el caso de que por cualquier método usted haya ejecutado con éxito DS9, ahora debe dirigirse a su terminal **xgterm** (que debe estar en *ecl >*)y tipear los siguientes comandos:
 - **1-tv**
 - **2-set stdimage=imt4096**
 - **3-display** y luego introducir el nombre de su imagen y con su respectiva extensión.
 - Así podremos ver nuestra imagen y nuestro Master BIAS estará listo. A continuación se muestra la imagen de nuestro Master BIAS:

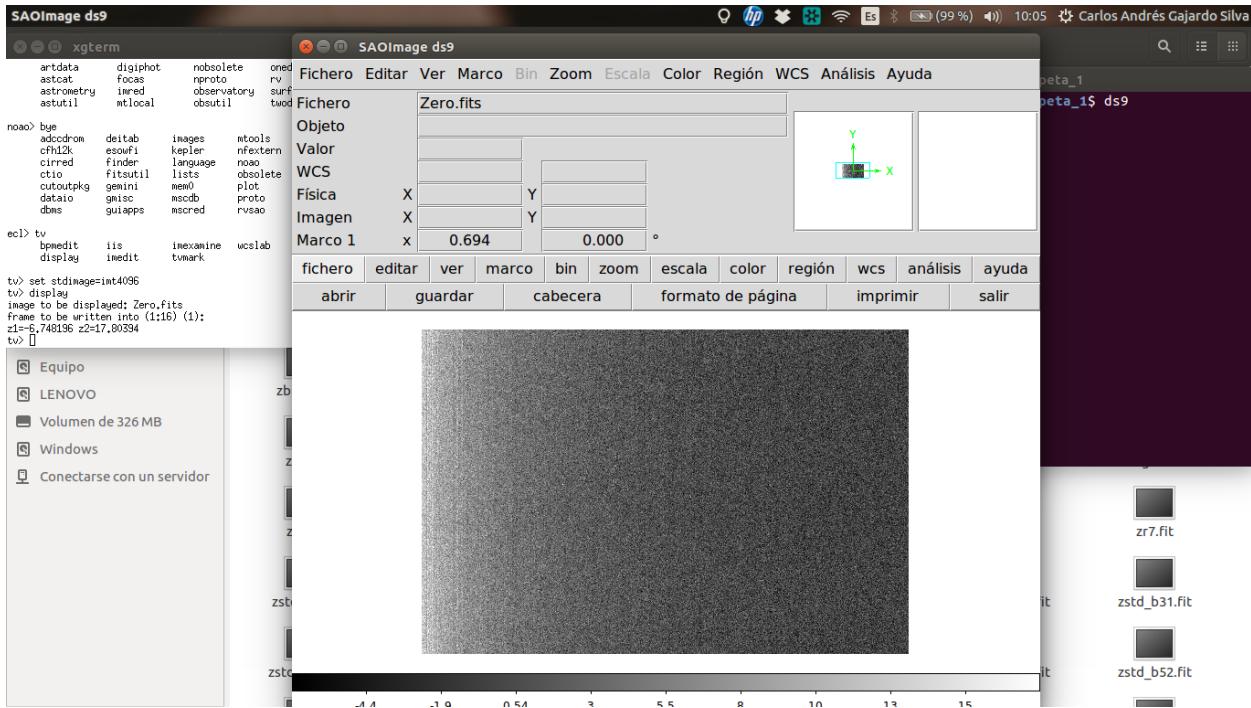


Figura 20: Master BIAS.

- Una vez obtenido nuestro Master BIAS, procedemos de igual forma para obtener nuestro Master FLAT para cada filtro. Para esto nos dirigimos ahora a *ccdred*. Aquí nos dirigimos a *epar flatcombine*
- Y, nuevamente, entraremos a una interfaz con la que ya debemos estar familiarizado. Aquí editamos de la siguiente forma:
 - En **input** escribimos el nombre de nuestra lista de FLATS, en un filtro.
 - En **output** ya es trivial lo que tenemos que hacer.
 - En **combine** escribimos *average*.
 - **ccdtype** debe estar en blanco.
 - **process**, **subsets**, **delete** y **clobber** deben ser editadas como *no*.
 - Luego, procedemos con **:go**
 - Luego de unos segundos, iraf nos enviará nuevamente a *ccdred* y nuestro Master FLAT, en un filtro específico, estará listo.
 - Debemos repetir el mismo proceso para los demás FLATS en los filtros restantes. En realidad, el lector lo hará, aquí omitiremos ese paso.

2.2.7. Corrección por BIAS (`zerocor`) y corrección por FLATS (`flatcor`)

Una vez terminado el proceso anterior, debemos comenzar ahora con el proceso de 'limpieza' de las imágenes. Primero queremos quitar el ruido BIAS de nuestro Master FLAT. También queremos quitar el ruido BIAS de nuestras imágenes de ciencia e imágenes estándares. Y, finalmente, queremos corregir por FLAT sin ruido BIAS nuestras imágenes de ciencia a las que ya se les ha quitado el ruido BIAS. Esto puede parecer escabroso y sin sentido, pero si el lector piensa lógicamente el esquema que estamos armando aquí, se dará cuenta que, en efecto, el esquema tiene mucho sentido y, además, es la forma más corta para obtener una imagen de ciencia " limpia ", sin ruidos y lista para analizar. Comenzamos primero por quitar el ruido BIAS de nuestros Master Flats. El resultado de esto será una imagen llamada Super Master Flat, esto se obtiene de la siguiente forma:

- Pimero nos dirigimos, en nuestra terminal *xterm*, a la sección *ccdred*.
- Luego vamos a *epar setinstrument* y teclamos dos veces **:go**.
- Editamos **images** con el nombre de nuestra imagen Master Flat.
- Entre **fixpix** y **scancor** sólo debe estar activado el parámetro **zerocor**, es decir, este debe ser editado como *yes*.
- Más abajo, entre **readaxi** y **nscan**, debemos editar el parámetro **zero**, que debe ser editado con el nombre de la imagen de nuestro Master BIAS.
- Finalmente, más abajo, debemos editar, para evitar complicaciones, el parámetro **interac**, al cual escribimos *no*.
- Luego, procedemos con **:go**. El hacer esto generará una imagen FLAT sin ruido BIAS que llamaremos SUPER MASTER FLAT.
- Ahora, se procede, de igual forma a quitar el ruido BIAS de nuestras imágenes de ciencia y imágenes standard.
- En resumen, acabamos de hacer nuestra corrección por BIAS a todas nuestras imágenes. Ahora haremos la corrección por FLAT a nuestras imágenes de ciencia y standards con su filtro respectivo.
- Para esto volvemos a *ccdred*. Luego nos dirigimos a *epar setinstrument* en donde escribimos dos veces seguidas **:go**
- Así volveremos a la interfaz con la cual ya estamos familiarizados.
- Aquí, editamos de forma similar, a cuando hicimos la corrección por BIAS, es decir, introducimos nuestra lista para trabajar. Lo que tenemos que editar ahora es el parámetro **flatcor**, al cual le pondremos *yes*, (el parámetro **zerocor** y **zero** deben quedar vacíos, en el caso de que haya quedado guardado de la tarea anterior) y más abajo, debemos editar **flat**, en donde pondremos nuestra imagen Super Master Flat, en el filtro correspondiente.
- Acabado esto tendremos nuestras imágenes listas para hacer ciencia.

2.2.8. Saturación y obtención de centros de brillo

Hemos acabado una etapa tediosa pero muy importante para nuestro análisis final que nos ayudará a obtener toda la riqueza e información de nuestra observación para hacer nuestra ciencia. Este análisis es el análisis fotométrico del que hablábamos en un principio, el cual es el motivo principal de este 'Manual'. Los procedimientos que haremos a continuación son, como los anteriores, extensos y tediosos pero, sin embargo, son mucho más interactivos y entretenidos, lo que debe ser la motivación para el lector a realizar todo este proceso y no morir en el intento. Sin más que agragar en esta pequeña introducción procedemos avanzar.

- Lo primero que hacemos para empezar de forma cómoda es cerrar todo vestigio de lo que hayamos hecho anteriormente, con esto nos referimos a las ventanas, terminales, etc.
- Luego, abrimos nuevamente nuestra carpeta donde estuvimos trabajando todo el tiempo anterior.
- Un paso cómodo, pero no necesario, es crear una nueva carpeta que contenga solo las imágenes de ciencia y standars, las cuales ya han sido limpiadas y corregidas.
- Ahora sí, para comenzar, debemos analizar si nuestras imágenes que acabamos de procesar son realmente útiles para poder ser usadas para hacer alguna ciencia.
- Un primer paso es ver si nuestra imagen y el brillo de nuestros objetos saturaron o no, de cierta forma, nuestra CCD. Esto dependerá directamente de la sensibilidad de nuestra CCD y su respuesta a las fuentes de luz.
- En concreto para notar si nuestras imágenes están saturadas o no, lo primero que hacemos es, como ya hemos aprendido, ejecutar una terminal **xgterm** en la carpeta donde tenemos nuestras imágenes procesadas, y llegar al entorno *ecl >*, luego escribimos *tv*. También tenemos que abrir DS9. Luego, en la terminal **xgterm** introducimos el comando para que la imagen se vea completa *set stdimage=imt4096* , luego para analizar una imagen introducimos *imexamine nombrede la imagen.extension* . En consecuencia, se abrirá la imagen señalada en DS9. Esto es:

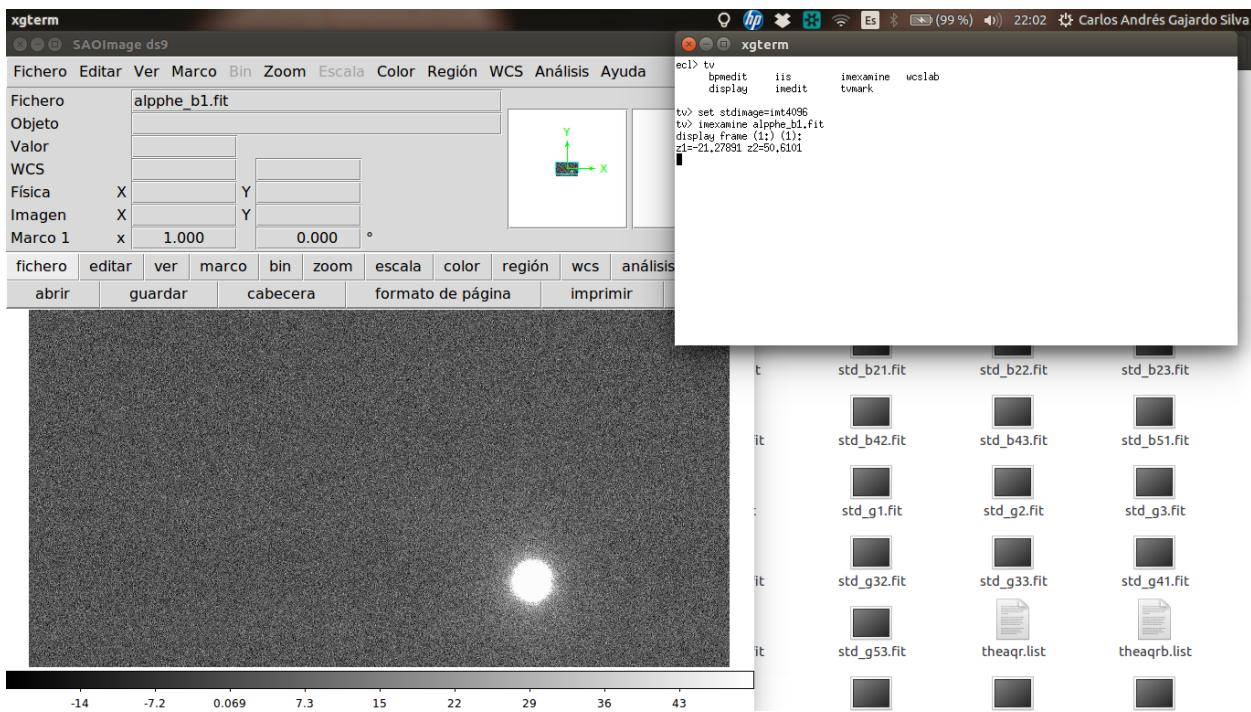


Figura 21: Imexamine.

- Si nos vamos a la pestaña de DS9 donde se ha abierto nuestra imagen notaremos que el curso ahora es extraño, pareciera variar. NO SE ASUSTE, eso quiere decir que todo está en orden.
- Ahora es cuando analizamos nuestra imagen. Para eso posamos el cursor en el centro de nuestro objeto y luego pulsamos, una vez, la letra **r**, o bien, la letra **s**. Los resultados se muestran graficamente:

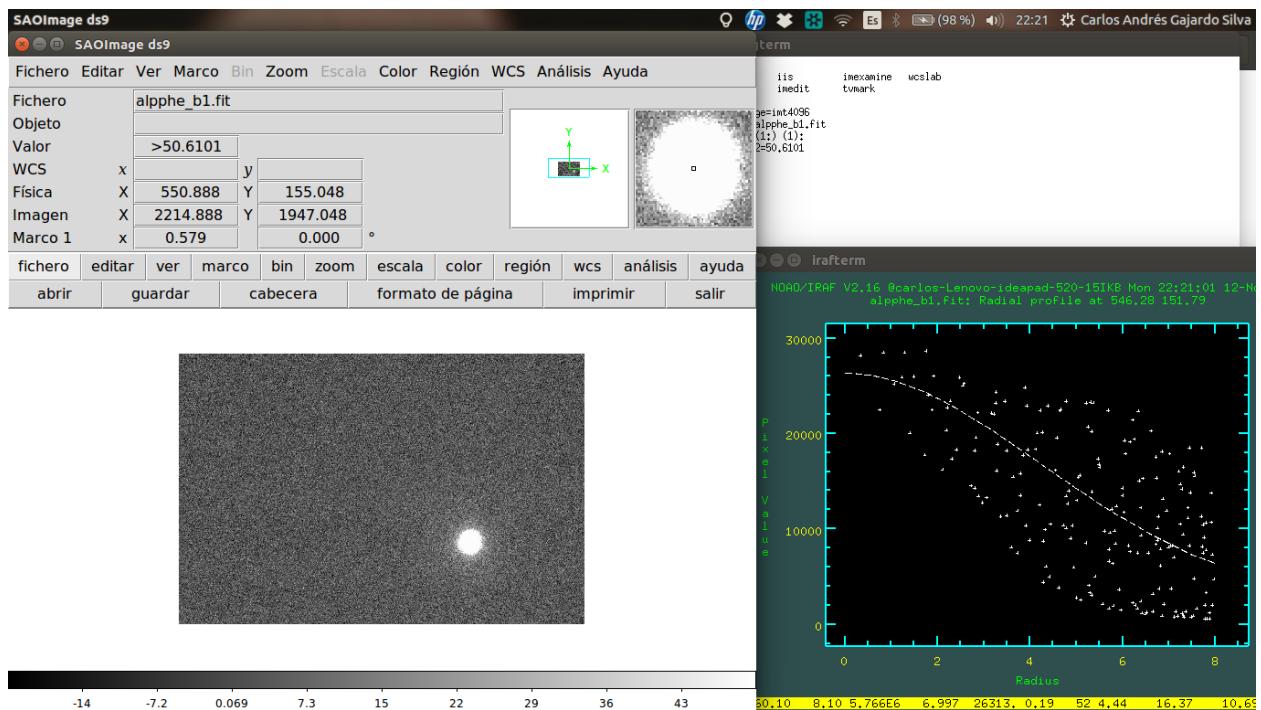


Figura 22: Al pulsar e.

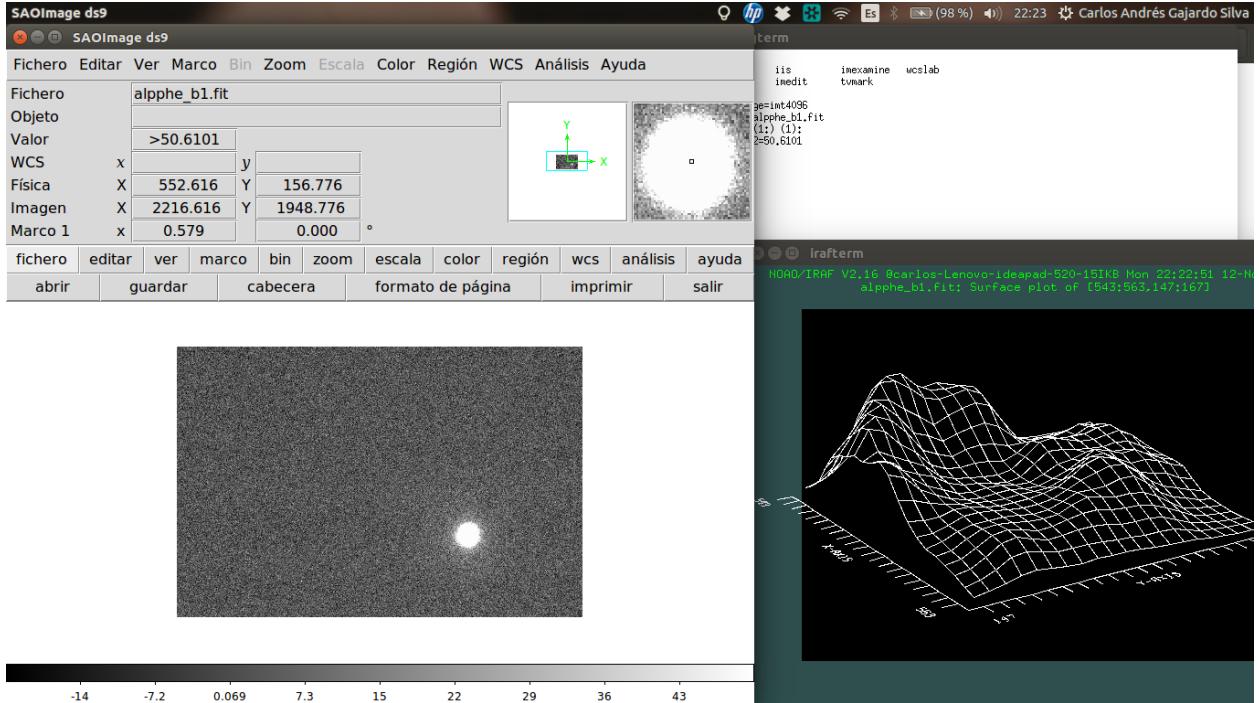


Figura 23: Al presionar s.

- Para el caso de la Figura 22, si queremos saber si nuestra estrella u objeto está saturada o no, nos fijamos en primera instancia en el número de cuentas (eje vertical) de nuestro gráfico. Si el numero de cuentas esta muy cerca del número de cuentas que satura la CCD, o bien, si sospechamos que el número de cuentas sobrepasa el valor máximo de cuentas que captura la CCD, entonces puede que la información que obtengamos de la estrella no sea fidedigna. Otra forma de observar lo mismo es fijarse en la linea por tramos (perfil Gaussiano) que se traza en la gráfica, esta linea es una tipo de ajuste que estima el comportamiento del número de cuentas obtenido. Si dicho perfil Gaussiano no se encuentra completamente en el gráfico (es decir, que no podamos distinguir su tope), entonces la información puede, con mucha probabilidad, no ser certera.
- Para el caso de la Figura 23, la interpretación es análoga a la interpretación de la Figura 22 en el sentido de evidenciar si nuestra imagen (el objeto), se encuentra saturada o no.
- Otro aspecto importante, que aprovechamos de cubrir ahora es descubrir en que 'coordenadas' de los pixeles se encuentran nuestros objetos (en específico, su centro) para eso, también con la acción *imexamine*, pulsamos una vez **j** para ver la coordenada en el eje horizontal y **k** para ver la coordenada en el eje vertical. Al presionar **j** o **k** sobre aproximadamente el centro de nuestro objeto se abrirá un gráfico parecido a los que hemos tratado, pero a diferencia de los demás, este trae un marco colorido y dentro de ese marco colorido hay una sección que dice **center** el cual nos da un número de un pixel. Dicho pixel no necesariamente es el pixel o coordenada de donde nosotros pulsamos **j** o **k**, sino que corresponde al pick del perfil gaussiano (línea grillada) pro-

ducido por nuestra estrella en la linea o columna de pixeles. Estas coordenadas deben ser anotadas en un archivo de texto con la extensión .coo, único para cada objeto. Gráficamente:

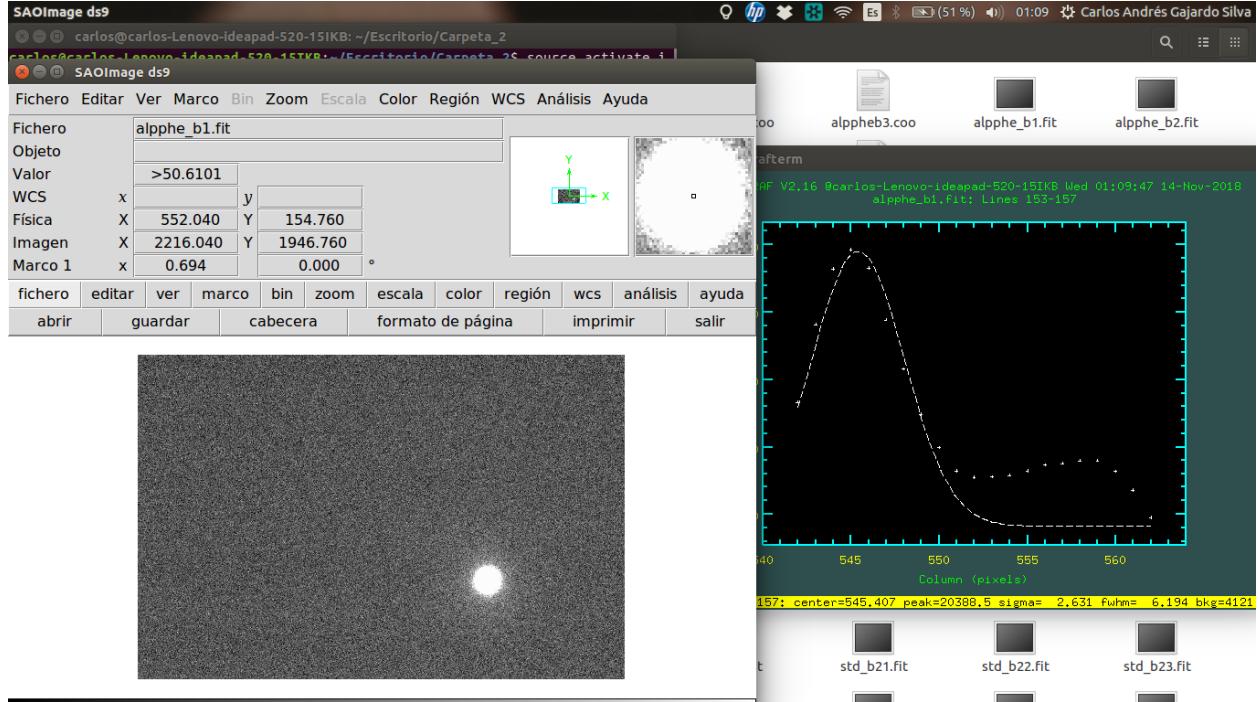


Figura 24: Al presionar j.

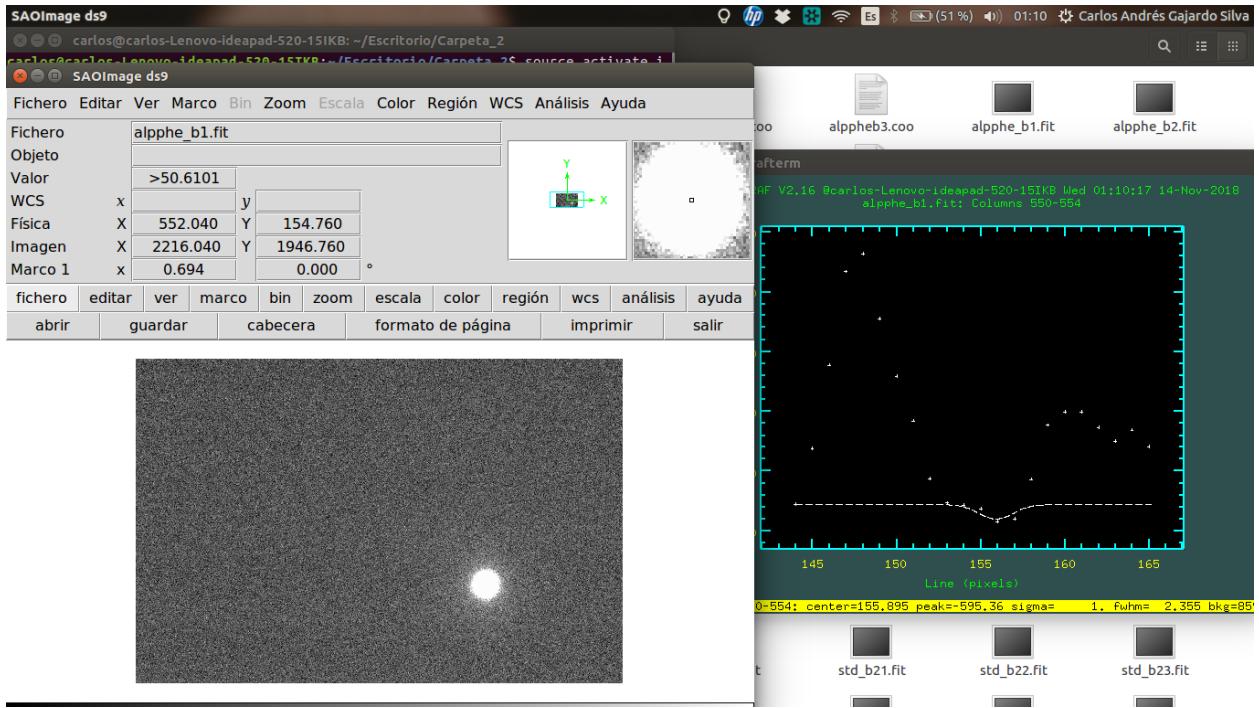


Figura 25: Al presionar k.

- A pesar de que existen técnicas para modelar y obtener información válida de imágenes saturadas, esto requiere otro proceso aún más tedioso de realizar, por lo que, en este caso, trabajaremos con el consenso de que no es válido obtener cualquier tipo de información a partir de imágenes saturadas.
- Usted puede ajustar el gráfico para, por ejemplo, tener una visión un poco más amplia de los datos. Para eso debemos ajustar editar algunos parámetros de la sección *rimexam* (en *tv*), esto es, tecleamos *epar rimexam* y dentro de esa interfaz editamos solamente el parámetro *rplot* asignándole un valor que tenga sentido y luego damos :go y luego para salir damos :q.
- Debemos repetir este proceso de 'Checkeo' para cada estrella/imagen que tengamos, algo que omitiremos en este documento. Sin embargo, se advierte al lector que no debe obviar este proceso.
- **Observación:** En ocasiones, y como ocurre en este caso, por ciertos problemas técnicos, digamos, por ejemplo, de calibración o otros factores, nuestra estrella puede presentar una perfil extraño, incluso puede llegar a presentar dos o, incluso, más perfiles gaussianos. Por ejemplo, para las Figuras 24 y 25, si seguimos los puntos graficados (no la linea grillada) veremos que tenemos dos peaks, o bien, dos perfiles gaussianos tanto en el eje horizontal como en el vertical. Más adelante plantearemos otros problemas que pueden estar asociados a esto y plantearemos distintas soluciones. Por ahora decidimos que cuando tengamos este problema trabajaremos con el pixel medio entre los píxeles de cada peak, lo cual no es tan alocado, ya que, al tratarse de estrellas individuales o

'aisladas' (lo que no aplica para el caso específico de la observación de estrellas binarias o bien que tengamos dos estrellas en un campo visual muy reducido), la información intrínseca del brillo de la estrella se mantiene, solo que esta "dispersa".

2.2.9. Análisis de apertura

Una vez acabado el proceso anterior, estamos a pasos de alcanzar nuestro objetivo. Se pide, de ahora en adelante, la mayor de la motivación, a pesar de lo largo y tedioso que ha significado realizar cada uno de los pasos en los procesos anteriores. Para comenzar con nuestro último proceso, comenzamos por trabajar en un entorno cómodo, cerrar aquellas pestañas que están demás, ojalá volver a tener solo nuestra carpeta en la cual tenemos nuestras imágenes que ya han sido procesadas y, además, que ya han pasado por una especie de filtro hecho en el proceso anterior.

- Comenzamos, como ya es de costumbre, activando iraf en una terminal que haya sido abierta en la carpeta en la cual estén nuestros datos y ejecutando una **xgterm**.
- Aquí estaremos en *ecl >*. Ahora nos dirigimos a *noao*. Ahí vamos a *digiphot* y luego *apphot*.
- *apphot* contiene una gran cantidad de tareas que podemos realizar con imágenes del mismo carácter que las que hemos estado trabajando en el transcurso de este documento.
- Una tarea, bastante especial es *phot*, la que nos permite hacer fotometría. Sin embargo, nada puede ser tan sencillo como simplemente introducir una imagen y obtener un valor de, por ejemplo, una magnitud instrumental de un objeto que hayamos observado, por el contrario, la tarea *phot* tiene, como prácticamente todas las tareas, parámetros e incluso, muchos de esos parámetros son, además, editables.
- Procedemos entonces a editar estos parámetros para concretar nuestro objetivo. Introducimos *epar phot*.
- Dentro de la interfaz, los parámetros editables que nos interezan principalmente son **datapar**, **centerpar**, **fitskyp** y **photpar**. Para editar cualquiera de estos parámetros, nos posamos sobre ellos y luego introducimos :e.
- Comenzamos con **datapar**. Aquí estamos con información técnica de nuestros instrumentos. Por ejemplo, **scale** indica la escala en arcosegundos por pixel.
- Luego nos fijamos en **ccdread**, que corresponde al ruido intrínseco de nuestra cámara. Para conocer este valor, una opción es acceder al header de la imagen a la que queremos aplicar nuestra tarea.
- Para acceder al header de una imagen rápidamente, debemos volver un poco atrás introduciendo :q dos veces y de ahí llegar nuevamente a *ecl >*. Ahí introducimos *imheader nombredelaimagen.extensión l+*.

- Haciendo esto la terminal arrojará un texto que es el header de nuestra imagen. Intuitivamente, dentro de ese texto se encuentra información de nuestra imagen y de como fue tomada dicha imagen. Por ejemplo, el tiempo de exposición de la toma, la temperatura de la CCD, etc. Sólo basta con leer para enterarse de la información existente ahí. Mucha de esta información posiblemente será requerida al momento de editar ciertas tareas. En general, para ver toda la información que contiene el header es conveniente expandir nuestra terminal **xterm**
- Un ejemplo de header es el siguiente:

```

xterm
carlos@carlos-Lenovo-ideapad-520-15IKB:~/Escritorio/Carpeta_2$ source activate i
raf27
(caraf27) carlos@carlos-Lenovo-ideapad-520-15IKB:~/Escritorio/Carpeta_2$ xterm
[...]

```

```

No bad pixels, ninc=0., max=0., (old)
Line storage mode, physdim [767,811], length of user area 2673 s.u.
Created Mon 14:39:09 12-Nov-2018, Last modified Mon 14:39:09 12-Nov-2018
Pixel File "std_b1.fit" [ok]

EXTEND = F / File may contain extensions
ORIGIN = 'NOAO-IRAF FITS Image Kernel July 2003' / FITS file originator
DATE = '2018-11-12T16:39:03' / Date FITS file was generated
IRAF-TIME = '2018-11-12T16:39:03' / Time of last modification
OBJECT =
TELESCOP='Unknown Telescope'
INSTRUME='SBIG ST-7'
DISPERSER='None Name'
DATE-OB = '2018-11-12T02:34:32.000' / GHT START OF EXPOSURE
EXPTIME = '+5.0000000000E+000' / EXPOSURE IN SECONDS
CCD-TEMP = '-2.493675007139E+001' / CCD TEMP IN DEGREES C
XPIXSZ = '+9.0000000000E+000' / PIXEL WIDTH IN MICRONS
YPIXSZ = '+9.0000000000E+000' / PIXEL HEIGHT IN MICRONS
XBINNING = 1 / HORIZONTAL BINNING FACTOR
YBINNING = 1 / VERTICAL BINNING FACTOR
XORIGIN = 0 / SUB-FRAME ORIGIN X POS
YORIGIN = 0 / SUB-FRAME ORIGIN Y POS
EGAIN = '+2.6600000000E+000' / ELECTRONS PER ADU
FOCALLEN = '+2.0320000000E+003' / FOCAL LENGTH IN MM
APERTDIA = '+2.0320000000E+002' / APERTURE DIAMETER MM
REFTHRESH = '+4.8387000000E+004' / REFLECTION THRESHOLD -1.00-1.00
CLKLOCK = '1000' / BLACK ADU FOR DISPLAY
CMWHITE = '1000' / WHITE ADU FOR DISPLAY
PEDESTAL = '-100' / ADD TO ADU FOR 0-BASE
SISTIMVER = 'SBIFITSEXT Version 1.0.' / SBIG FITS EXTENSIONS VER
SWACQUIR = 'WinDOPS Ver 5.63 Build 28-NT' / DATA ACQ SOFTWARE
SWACQUIR = 'SBIG Win CCDOPS Version 5.63 Build 28-NT'
FILTRNAME = 'Blue' / OPTICAL FILTER NAME
SNAPSHOT = 1 / NUMBER IMAGES COPIED
RESMODE = 6 / RESOLUTION MODE
EXPSTATE = '125' / EXPOSURE STATE (HEX)
RESPONSE = '+3.0000000000E+003' / CCD RESPONSE FACTOR
NOTE = 'Local time:11/6/2018 at 23:34:32'
WCSTHM = 2
LTM1_1 = 1
LTM1_2 = 1
LTM2_1 = 1
LTM2_2 = 1
WAT0_001='systemphysical'
WAT1_001='wttype=linear'
WAT2_001='wttype=linear'
TRIMSEC = 'Nov 11 20:53 Trim data section is [1:767,1:511]'
OVERSCAN = 'Nov 11 20:53 overscan section is [768:795,512:540] with mean=933.779'
CCDSEC = '[1:767,1:511]'
CCDMEAN = '31.09678'
CCDMEANT = '1226500749'
CCDPROC = 'Nov 12 14:39 CCD processing done'
ZEROCOR = 'Nov 12 12:54 Zero level correction image is Zero.Fits'
FLATCOR = 'Nov 12 14:39 Flat field image is SFFlat.fits with scale=28954.5'
ed> [...]

```

The file browser window shows the following files:

- Windows
- Conectarse con un servidor
- std_g42.fit
- std_g43.fit
- std_g51.fit
- theaqrb1.coo
- theaqrb2.coo
- theaqrb3.coo
- theaqrg1.coo
- theaqrg2.coo
- theaqrg3.coo
- uparmccdsetint.par
- uparmimcdred.par
- uparmimlimhear.par
- uparmimlimpar.par
- uparmimlimpypar.par
- uparmimlimxpar.par
- uparmtvrimexm.par

Figura 26: Header de std_b1.fit.

- Una vez editados los parámetros de la tarea **datapar** en relación al Header introducimos :q para guardar esta configuración, volver a la interfaz de *epar phot* para luego editar la tarea **centerpar**. En **centerpar**, en este caso, conviene editar solo **cbox** que es una 'caja de pixeles' centrada en las coordenadas que fueron ingresadas a través del archivo de extensión **coo** cuya misión es, medianete un algoritmo matemático encontrar, dentro de la caja de pixeles", el centro real de, en este caso, el brillo de nuestra estrella.
- **Observación:** En general, no necesariamente conviene tener un **cbox** demasiado grande ya que se podría hacer una mal cálculo del centro de brillo de nuestra estrella, sin embargo, la opción **cmaxite** controla el número de datos para hacer una cierta estadística y determinar el centro de brillo. Así, por ejemplo, cuando tenemos un **cbox** relativamente grande conviene aumentar el número de iteraciones y reducir el error en el cálculo de centro de brillo.

- Cuando ya estemos conforme con nuestra edición de **centerpar** introducimos :**q** para pasar a la edición de **photpar**
- En **photpar** nos concentraremos en la opción **apertur**. Esta opción nos permitirá reconstruir en brillo de las estrellas a ciertos radios del centro que fue calculado en la tarea anterior, en otras palabras, nos permitirá obtener el radio de brillo de nuestra estrella. Es por eso que esta opción debe ser editada con muchos datos de radios. Por ejemplo, puede ser editada como 2.,3.,4.,5.,6. y así como uno estime conveniente. El patrón debe ser creciente pero no necesariamente regular. Una vez hecho esto, introducimos :**q** para luego editar **interac**, al cual de daremos el valor *no*.
- Ahora miramos la tarea **fitskyp**. Dentro de **fitskyp** nos fijamos en **salgorim**, **annulus**, **dannulus** y también **smaxite**, aun que este último no es tan importante.
- Como se dijo en el punto anterior, sólo estamos haciendo una mirada a **fitskyp**, puesto que será el paso más fundamental de lo que viene.
- Entonces, luego de haber editado correctamente **datapar**, **centerpar** y **photpar** podemos introducir :**go**.
- Luego de introducir :**go**, IRAF nos preguntará algunas cosas. E en general, es para preguntar si estamos seguros de la edición que hicimos en las tareas anteriores. Si estamos completamente seguros, damos INTRO/ENTER a todo.
- Finalmente usted será llevado nuevamente a *apphot*.
- Si todo fue hecho correctamente, en la carpeta en la que usted trabaja se habrá creado un archivo de texto con extensión **.mag**. Este archivo es fundamental. Para tener una idea de como debe ser este archivo veamos las siguientes figuras:

```

Editor de textos
carlos@carlos-Lenovo-ideapad-520-15IKB:~/Escritorio/Carpeta_2$ source activate iraf27
(caraf27) carlos@carlos-Lenovo-ideapad-520-15IKB:~/Escritorio/Carpeta_2$ xterm
[...]
std_b1.fit.mag.1 (~/Escritorio/Carpeta_2) - gedit
Guardar
#K IRAF      = NOAO/IRAFV2.16      version   %-23s
#K USER      = name               %-23s
#K HOST      = carlos-Lenovo-ideapad-5 computer  %-23s
#K DATE      = 2018-11-15        yyyy-mm-dd %-23s
#K TIME      = 00:19:28          hh:mm:ss  %-23s
#K PACKAGE   = apphot            name      %-23s
#K TASK      = phot               name      %-23s
#
#K SCALE      = 0.6              units     %-23.7g
#K FWHMPSF   = 2.5              scaleunit %-23.7g
#K EMISSION   = yes               switch    %-23b
#K DATAMIN   = INDEF             counts    %-23.7g
#K DATAMAX   = INDEF             counts    %-23.7g
#K EXPOSURE   = EXPTIME          keyword   %-23s
#K AIRMASS   = ""                keyword   %-23s
#K FILTER     = FILTER             keyword   %-23s
#K OBSTIME   = ""                keyword   %-23s
#
#K NOISE      = poisson           model    %-23s
#K SIGMA      = INDEF             counts    %-23.7g
#K GAIN       = EGAIN             keyword   %-23s
#K EPADU      = 2.66              e-/adu    %-23.7g
#K CCDREAD   = ""                keyword   %-23s
#K READNOISE  = 0.                e-        %-23.7g
#
#K CALGORITHM = centroid          algorithm %-23s
#K CBOXWIDTH  = 10.              scaleunit %-23.7g
#K CTHRESHOLD = 0.                sigma    %-23.7g
#K MINSNRATIO = 1.                number   %-23.7g
#K CMAXITER   = 50.              number   %-23d
)
#) (CR or value);
#) centroid
#) units (10.) (CR or value);
#) #; 10, scale units 16.66667 pixels
#) (CR or value);
#) scale units (10.) (CR or value);
#) #; annulus; 10, scale units 16.66667 pixels
#) scale units (10.) (CR or value);
#) annulus; 10, scale units 16.66667 pixels
#) in scale units (3.,6.,9.,12.,15.,18.,21.,24.,27.,30.,
#) scale units 5, pixels
#) scale units 10, pixels
#) scale units 15, pixels
#) scale units 20, pixels
#) scale units 25, pixels
#) scale units 30, pixels
#) scale units 35, pixels
#) scale units 40, pixels
#) scale units 45, pixels
#) scale units 50, pixels
#) round in counts (INDEF) (CR or value);
#) on of background; INDEF counts
#) (CR or value);
#) #; value; INDEF counts
#) (CR or value);
#) #; value; INDEF counts
stdb2.coo
stdb33.coo
stdb54.coo
std_b23.fit

```

«std_b1.fit.mag.1» seleccionado (5,4 kB)

Figura 27: std_b1.fit.mag (parte 1).

```

Editor de textos
carlos@carlos-Lenovo-ideapad-520-15IKB:~/Escritorio/Carpeta_2$ source activate iraf27
(caraf27) carlos@carlos-Lenovo-ideapad-520-15IKB:~/Escritorio/Carpeta_2$ xterm
std_b1.fit.mag.1 (~/Escritorio/Carpeta_2) - gedit
Guardar
Abrir ▾
#K SALGORITHM = centroid          algorithm %-23s
#K ANNULUS   = 10.                 scaleunit %-23.7g
#K DANNULUS   = 10.                 scaleunit %-23.7g
#K SKYVALUE   = 0.                  counts  %-23.7g
#K KHIST      = 3.                  sigma   %-23.7g
#K BINSIZE    = 0.1                sigma   %-23.7g
#K SMOOTH     = no                 switch  %-23b
#K SMAXITER   = 10.                number  %-23d
#K SLOCLIP    = 0.                 percent  %-23.7g
#K SHICLIP    = 0.                 percent  %-23.7g
#K SNREJECT   = 50.                number  %-23d
#K SLOREJECT  = 3.                sigma   %-23.7g
#K SHIREJECT  = 3.                sigma   %-23.7g
#K RGROW      = 0.                 scaleunit %-23.7g
#
#K WEIGHTING  = constant          model   %-23s
#K APERTURES  = 3.,6.,9.,12.,15.,18.,21 scaleunit %-23s
#K ZMAG       = 25.                zeropoint %-23.7g
#
#N IMAGE      XINIT    YINIT    ID     COORDS    LID   \
#U imagename  pixels   pixels   ##    filename   ##   \
#F %-23s      %-10.3f  %-10.3f  %-6d  %-23s    %-6d  \
#
#N XCENTER    YCENTER   XSHIFT  YSHIFT  XERR   YERR   CIER CERROR \
#U pixels     pixels   pixels   pixels  pixels  pixels  ##  errors \
#F %-14.3f    %-11.3f  %-8.3f  %-8.3f  %-8.3f  %-15.3f  %-5d %-9s  \
#
#N MSKY       STDEV     SSKEW    NSKY    NSREJ   SIER SERROR \
#U counts     Counts   counts   npix   npix   ##  serrors \
#F %-18.7a    %-15.7a  %-7d    %-9d    %-5d   %-9s

```

Texto plano ▾ Anchura de la pestaña: 8 ▾ Ln 1, Col 1 ▾ INS

«std_b1.fit.mag.1» seleccionado (5,4 kB)

Figura 28: std_b1.fit.mag (parte 2).

Figura 29: std_b1.fit.mag (parte 3).

- Debemos generar este archivo, de extensión **.mag** para cada imagen. Este archivo se lee de una manera complicada que no explicaremos aquí, sin embargo, para este caso, daremos la información útil que nos servirá en nuestro propósito.
- Una vez que tengamos dichos archivos listos para cada estrella abriremos uno de estos archivos y dirigiremos a la parte inferior, donde tenemos valores numéricos en plan de arreglo numérico, como en la Figura 29.
- La información que se necesita se muestra a continuación:

Figura 30: Información útil.

- Los datos marcados nos ayudarán al objetivo de encontrar el radio del brillo de nuestros objetos. Podemos hacer una gráfica mediante algún software que lo permita y, valga la redundancia, graficar los datos asociados a **RAPERT** y **FLUX**. Por ejemplo, [aquí](#) se encuentra un código escrito en Python que genera este gráfico. Para los datos de la Figura 30 el gráfico es el siguiente:
- El radio de brillo de nuestra estrella queda determinado cuando el número de cuentas permanece constante". En la Figura 31, el número de cuentas permanece constante cuando el radio de apertura es aproximadamente 17. La idea es obtener este radio para cada una de nuestras estrellas estándar (que no necesariamente es el mismo).
- Una vez obtenido este valor para cada estrella debemos re-ejecutar la misma tarea, para cada objeto, pero esta vez debemos editar, además de los parámetros que fueron editados, los parámetros de **fitskyp** que son **anuulus** y **dannulus**. En general, **anuulus** debe ser un poco más grande que nuestro radio de apertura en donde el número de cuentas permanece constante". Por ejemplo, si en la Figura 31 dicho radio es 17, entonces nuestro **annulus** puede ser editado, digamos, como **19..** El **dannulus** es una especie de **annulus** pero que se encuentra aun más lejos del centro del **annulus**, esto es, si nuestro **dannulus** es, digamos 5, quiere decir que se encuentra 5 unidades más lejano que el centro del **annulus**. **Annulus** y **dannulus** forman una especie de anillo que rodea nuestro objeto de interés y lo que hace es establecer una estadística del brillo del cielo para caracterizar aun mejor a nuestro objeto. Luego de editar **dannulus**, con un valor razonable damos :q

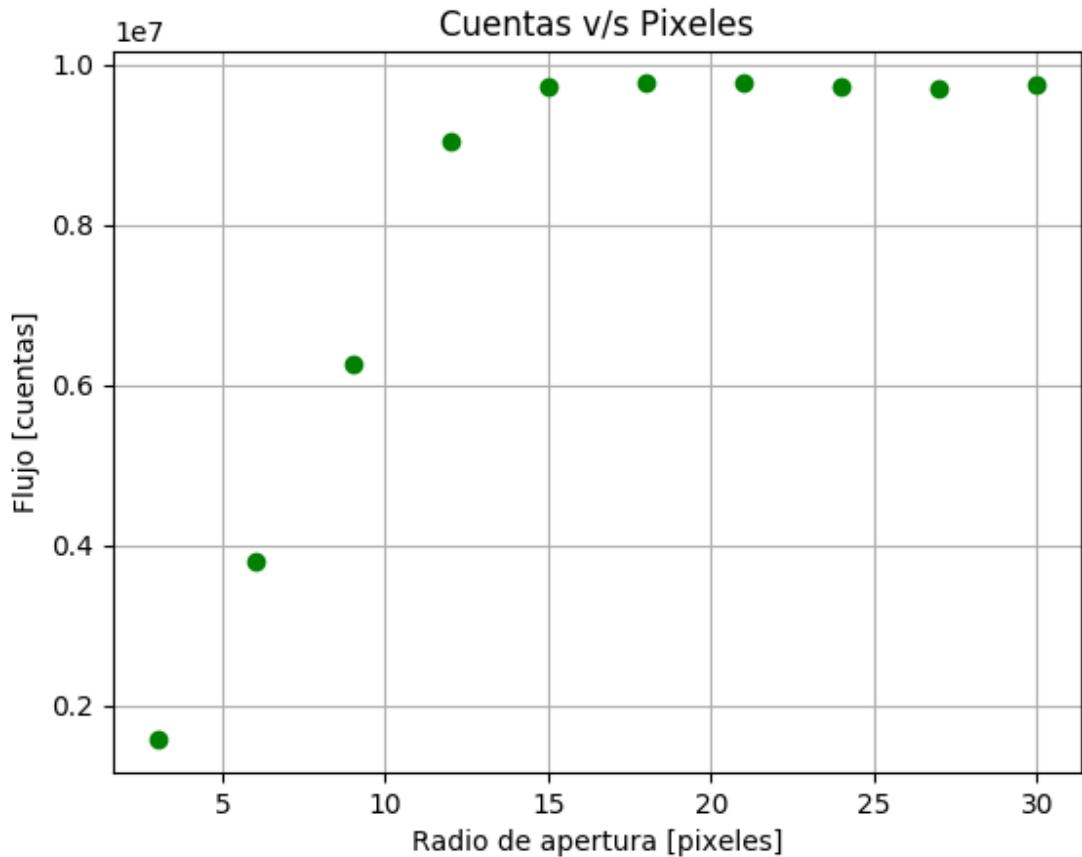


Figura 31: FLUX V/S RAPERT.

- Por último, vamos a editar nuevamente **photpar**, específicamente, editamos **aperture**, en donde solo consideraremos el radio que hemos obtenido (en el cual el número de cuentas permanece constante). Luego de esto damos :**q** y luego :**go**.
- El resultado de esto es otro archivo como el de la Figura 30 pero esta vez tenemos solo un valor de RAPERT Y FLUX. Pero eso no es todo porque además hemos obtenido nuestra magnitud instrumental que tanto hemos buscado, y que se encuentra justo al lado derecho de nuestro valor de FLUX.
- Este procedimiento se debe realizar para cada objeto al cual queremos obtener una magnitud instrumental.

Con esto hemos terminado lo que es el objetivo de este manual y con lo cual hemos hecho gran parte de lo que posiblemente pueden ser resultados científicos dispuestos a nuestra investigación. Posiblemente, la versión de este documento no sea la última y se pide, apenas este material llegue a las manos del lector, o bien, a su computadora, tablet, smartphone o cualquier otro dispositivo de lectura, comunicarse al correo electrónico carlosanga@udec.cl. También, posiblemente, se agregarán anexos para complementar lo que aquí se trabaja.

3. Agradecimiento especiales

Yo, Carlos Gajardo Silva, quiero agredecer encarecidamente a quienes me han aportado con mucho en la redacción, estructura y contenido de este documento, ellos son Juan Pablo Benavente, Carolina Ortíz, Luis Gonzalez, Hugo Parischewsky y Raúl Baier, todos ellos estudiantes de la carrera Astronomía de la Universidad de Concepción.