



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE CIENCIAS ASTRONÓMICAS Y GEOFÍSICAS

# BREVE INTRODUCCIÓN AL USO DE IRAF



ELABORACIÓN: LIC. MARINA SOSA  
COORDINACIÓN: DR. ILEANA ANDRUCHOW

OBSERVATORIO PEDAGÓGICO  
LA PLATA, 2015

# Nota

Este instructivo tiene como objetivo brindar al estudiante una breve y sencilla introducción a las herramientas básicas del programa IRAF.

Si lo desea, en caso que encuentre algún error o tenga alguna sugerencia para mejorar este instructivo, agradeceríamos que se contacte con:

Observatorio Pedagógico: <pedagogica@fcaglp.fcaglp.unlp.edu.ar>

Lic. Nérida M. González: <ngonzalez@fcaglp.fcaglp.unlp.edu.ar>

# 1. Introducción

IRAF corresponde a las siglas *Image Reduction and Analysis Facility*. Este sistema proporciona un amplio conjunto de herramientas para el análisis y el procesado de imágenes astronómicas. Fue desarrollado a mediados de los años 80 por la institución *National Optical Astronomy Observatories (NOAO)* en Tucson, Arizona. Se ha hecho popular entre la comunidad astronómica entre otras cosas por haber sido seleccionado como el sistema sobre el cual se basan los programas de reducción y análisis de datos del Telescopio Espacial (HST). Su arquitectura permite que se añadan paquetes externos con facilidad y si se conoce el lenguaje en el que se basa este sistema es posible agregar tareas propias y ejecutarlas dentro del mismo. En lo que sigue veremos como utilizar algunas de las tantas que posee IRAF, aquellas que sean útiles para materias como Astronomía Estelar, Sistemas Estelares o Astronomía Extragaláctica sin morir en el intento.

Es importante tener en cuenta que este instructivo se basará en la versión de IRAF instalada en la sala de máquinas para Junio de 2014.

## 1.1. Para arrancar:

Para abrir IRAF es necesario seguir un protocolo: de una terminal cualquiera, y controlando que estemos en nuestro *home* (el comando `pwd` me dice en que directorio estoy), abrimos una *xgterm* (equivalente a una terminal regular, pero que habilita la posibilidad de trabajar también con una interacción gráfica) de esta manera: `>xgterm &`, luego desde la misma accedemos al sistema IRAF con `>ec1` (ver Fig. 1).

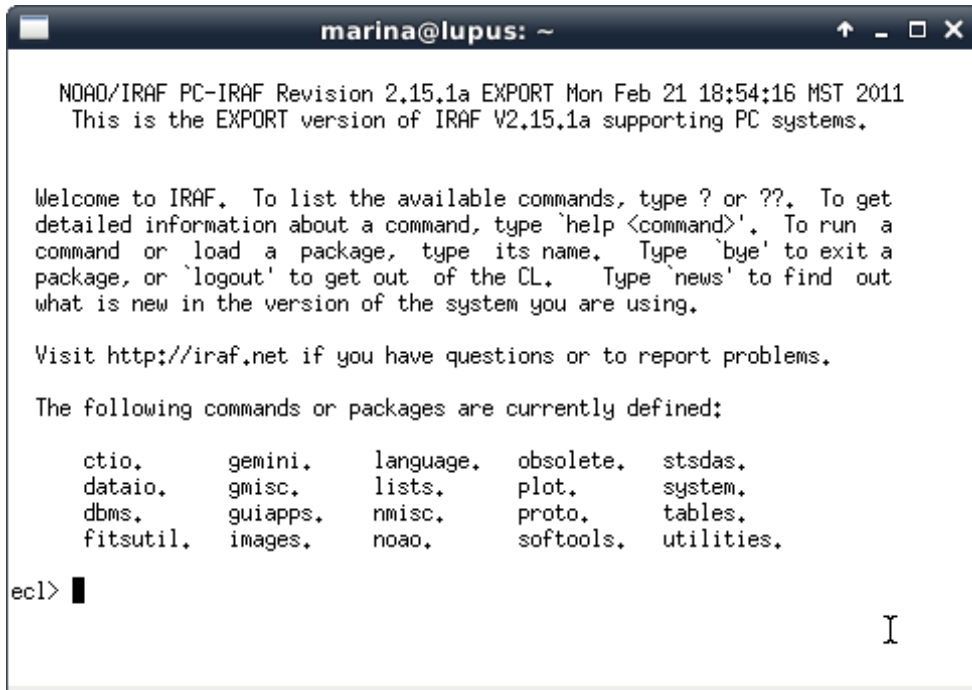
Para salir de IRAF: `>logout`.

Al trabajar con imágenes astronómicas vamos a necesitar una herramienta de visualización, que IRAF no posee, pero permite utilizar alguna externa, como es SAOImage DS9. Este programa debemos abrirlo antes de entrar a IRAF (o sea, antes de tipear “ec1”) poniendo en la *xgterm* abierta: `>ds9 -fifo dev/imt1 &`. La primera vez que ejecute el IRAF como hemos explicado arriba, automáticamente se generará un archivo de inicio “login.cl” que será propio de cada usuario y contiene definiciones para iniciar el sistema y un directorio llamado “uparm” en el cual se guardarán los valores de los parámetros que se usaran en cada tarea. Ambos son muy importantes y cada usuario deberá procurar de NO borrarlos por error o modificar, a no ser que sea necesario.

## 2. Paquetes y tareas

Una vez dentro de IRAF, necesitamos acceder al directorio donde vamos a trabajar, donde estén las imágenes. Esto se hace con el comando `cd`. Veamos un ejemplo, supongamos que las imágenes que necesito para realizar la Práctica 3 están en un directorio que se llama “MisGalaxias” y este último está dentro de otro directorio que se llama “Materias”, por lo tanto para llegar al directorio de las imágenes debo hacer:

```
>cd Materias/MisGalaxias
```



```
marina@lupus: ~
NOAO/IRAF PC-IRAF Revision 2.15.1a EXPORT Mon Feb 21 18:54:16 MST 2011
This is the EXPORT version of IRAF V2.15.1a supporting PC systems.

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type `help <command>`. To run a
command or load a package, type its name. Type `bye` to exit a
package, or `logout` to get out of the CL. Type `news` to find out
what is new in the version of the system you are using.

Visit http://iraf.net if you have questions or to report problems.

The following commands or packages are currently defined:

    ctio.      gemini.    language.  obsolete.  stsdas.
    dataio.    gmisc.     lists.     plot.      system.
    dbms.      guiapps.   nmisc.     proto.     tables.
    fitsutil.  images.    noao.      softools.  utilities.

ecl> |
```

Figura 1: Captura de pantalla de la *xterm*.

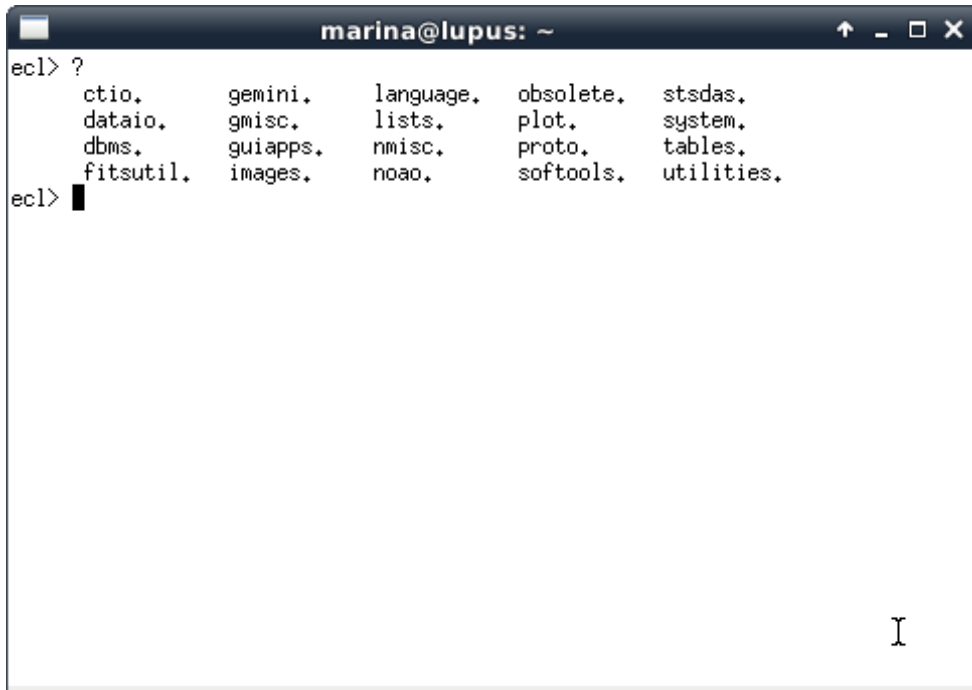
Además, existen otros comandos similares a los usados en UNIX como son **mkdir**, **ls**, **mv**, **dir**, **delete** que podremos usar en IRAF. Cualquier otro comando de UNIX se puede utilizar dentro de IRAF anteponiendo al mismo “!” para que IRAF lo reconozca, por ejemplo, `>!rm`.

Los comandos que sirven para realizar determinadas acciones dentro de IRAF se denominan “tareas” y están agrupadas en paquetes de acuerdo a su funcionalidad, a su vez los paquetes se suelen organizar en otros paquetes, esto se especifica con un punto en medio de los nombres de los paquetes (por ejemplo, *noao.imred*). Para saber la lista de paquetes disponibles tipeamos “?” (ver Fig. 2).

Para ver los paquetes dentro de otros paquetes, tipeamos “??” y en este caso se desplegará una lista más larga.

Al arrancar el IRAF, hay paquetes que cargan automáticamente. Con el comando **package** podemos ver los paquetes que ya están cargados (ver Fig. 3). Si quiero cargar un paquete que necesito ejecutar, sólo basta con tipear el nombre, de esta manera se mostrará en pantalla una lista con todas las tareas y sub-paquetes que contiene. También notaremos que el *prompt* cambia, ya que aparecerá el nombre del paquete cargado. Si quiero abandonar un paquete cargado por que ya no lo voy a utilizar se usa el comando **bye**.

**Para ejecutar cualquier tarea es necesario tener cargado el paquete que la contiene. Si no sé en que paquete está, entonces tipeo “help nombredelatarea” y aparece en el encabezado del help donde tengo que ir para ejecutarla.**



```
marina@lupus: ~
ecl> ?
ctio.      gemini.   language. obsolete.  stsdas.
dataio.    gmisc.   lists.    plot.     system.
dbms.      guiapps. nmisc.    proto.    tables.
fitsutil.  images.  noao.     softools. utilities.
ecl> █
```

Figura 2: Lista de paquetes disponibles.

Por ejemplo, quiero usar la tarea *ellipse*. En el encabezado del *help* dice: *stsdas.analysis.isophote*, por lo tanto necesito cargar el paquete *isophote* para poder ejecutar la tarea.

## 2.1. Parámetros

Cada tarea tiene un set de parámetros que controlan el comportamiento de la misma. Los resultados que arroje al ejecutarla y su confiabilidad dependerán de los valores que le asigne a los parámetros. Siempre que estén por trabajar con una tarea vayan al *help*<sup>1</sup> de la misma.

Con **lpar** se listan los parámetros de una tarea, por ejemplo, si tipeo `lpar imexamine` me lista los parámetros de la tarea **imexamine** como se muestra en la Fig. 4 (si pongo el nombre de la tarea abreviada, `lpar imexa`, también funciona, siempre que la abreviación no sea igual al nombre y/o abreviación de otra tarea). Existen dos tipos de parámetros, aquellos que se requieren para ejecutar la tarea y los que tienen valores predeterminados que no necesitan ser especificados (parámetros ocultos). Los parámetros necesarios para que la tarea se ejecute son los que no están encerrados por paréntesis, por ejemplo, en el

---

<sup>1</sup>Puedo guardar el *help* de una tarea en un archivo tipo ASCII, por ejemplo para imprimirlo, de la siguiente manera: `> help nombredelatarea > nombredelatarea.hlp`.

```

marina@lupus: ~
dataio.  gmisc.  lists.  plot.  system.
dbms.   guiapps. nmisc.  proto. tables.
fitsutil. images.  noao.  softtools. utilities.
ecl> package
clpackage
language
tv
dataio
plot
user
system
lists
noao
nttools
utilities
proto
imutil
ismatch
imgeom
imfit
imfilter
imcoords
images
ecl>

```

Figura 3: Captura de pantalla de IRAF al ejecutar el comando `package`.

caso de `imexamine` estos son: `input`, `frame`, `image`, si se omite alguno de estos parámetros IRAF les pedirá que den un valor.

**Siempre que tengas duda sobre como funciona una tarea, para que sirve o que parámetros tiene puedes recurrir a la ayuda, por ejemplo, `>help imexamine`.**

Para editar los parámetros de `imexamine`, por ejemplo, ejecutamos: `>epar imexamine`, y se mostrará una lista (ver Fig. 5). Ahora se nos permite interactuar con la lista de parámetros. Utiliza las flechas para moverte hacia arriba o hacia abajo, al llegar al parámetro que deseas editar, tipeá el valor que quieras darle (si te equivocaste al escribir se borra con “`supr`”, únicamente) y luego “`enter`”. Una vez que hayas terminado de editar todos los parámetros (`input`, `frame`, `image`) podemos salir de tres maneras:

- “CTRL-c” saldrá y cancelará cualquier cambio.
- “:q” saldrá sin ejecutar la tarea, pero quedarán los cambios guardados.
- “CTRL-d” o “: go” saldrá guardando los cambios y ejecutando.

**El editor de IRAF es VI<sup>2</sup>, por eso se utilizan estos comandos.**

---

<sup>2</sup>VI es el editor estándar de UNIX.

```

marina@lupus: ~
ec1> lpar imexa
      input =          images to be examined
      frame = 1        display frame
      image =          image name
      (output = "")    output root image name
      (ncoutput = 101) Number of columns in image output
      (nloutput = 101) Number of lines in image output
      (logfile = "")   logfile
      (keeplog = no)   log output results
      (defkey = "a")   default key for cursor list input
      (autoredraw = yes) automatically redraw graph
      (allframes = yes) use all frames for displaying new images
      (nframes = 0)    number of display frames (0 to autosense)
      (ncstat = 5)     number of columns for statistics
      (nlstat = 5)     number of lines for statistics
      (graphcur = "")  graphics cursor input
      (imagecur = "")  image display cursor input
      (wcs = "logical") Coordinate system
      (xformat = "")   X axis coordinate format
      (yformat = "")   Y axis coordinate format
      (graphics = "stdgraph") graphics device
      (display = "display(image='$1',frame=$2)") display command template
      (use_display = yes) enable direct display interaction
      (mode = "ql")
ec1> █

```

Figura 4: Lista de parámetros de la tarea **imexamine** que se muestran al ejecutar **lpar imexamine**.

Para volver los parámetros de las tareas a los valores que tiene IRAF por *default* utilizamos el comando **unlearn**:

```
>unlearn tarea
```

### 3. ¡A trabajar con imágenes!

El formato de las imágenes más usado es FITS (*Flexible Image Transport System*), consiste de un *header* con la información sobre el instrumento, tiempo de observación, filtro, etc., y los datos propiamente dichos, es decir imagen + texto, en una sola cosa.

Como se mencionó anteriormente, toda la información sobre la imagen se almacena en el *header*. Es posible verlo y editarlo.

Para ver la información que contiene, usamos la tarea **imheader**, eligiendo la opción “l+” que esta indicando el *header* completo. Por ejemplo, supongamos que quiero leer el *header* completo de la imagen AEG1026.fits, entonces utilizo el comando:

```
>imheader AEG1026.fits l+
```

Si no agrego “l+” me mostrará sólo el header “corto” de la imagen.

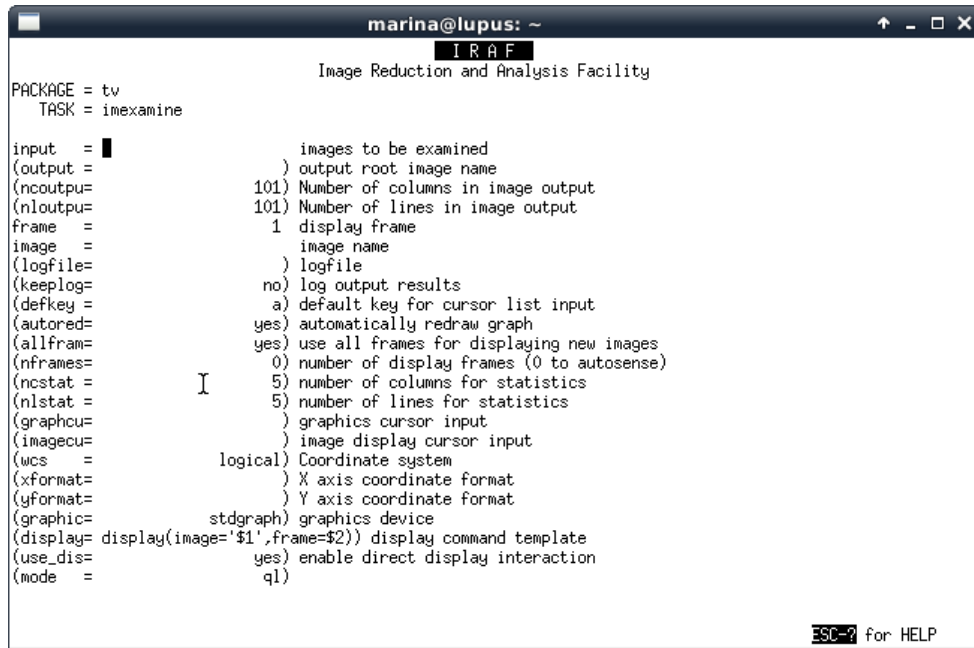


Figura 5: Captura de pantalla al ejecutar `epar imexamine`.

En caso de que necesite ver sólo algunos campos, por ejemplo, nombre de la imagen [I], filtro [filter], tiempo de exposición [exptime], podemos usar la tarea `hselect`:

```
>hselect *.fits $I,filter,exptime yes
```

En este ejemplo “.fits” está indicando que me muestre el nombre de la imagen, filtro y tiempo de exposición de todas las imágenes que tengan extensión “.fits” que se encuentran dentro del directorio donde estoy trabajando.

Puedo también editar un parámetro o agregar otro que necesite, por ejemplo, si quiero darle un valor específico a un parámetro que esté en el *header* de la imagen, como ser, EXP-TIME y darle el valor 80, puedo hacerlo mediante la tarea `hedit` y por línea de comando sin necesidad de utilizar `epar`, de la siguiente manera:

```
>hedit *.fits EXPTIME '80' ver-
```

## ATENCIÓN

**Manejen con cuidado la tarea HEDIT por que se pueden perder datos del *header* ya que modifica los archivos, como cuando borro algo con *rm*. Por esto es muy útil tener las imágenes crudas siempre en un directorio de *backup*.**

Veamos un ejemplo más completo. En la Fig. 6 se muestra el *header* de la imagen MSS\_0153.fits sin editar ni procesar. El parámetro EQUINOX vale 0,000, y quiero darle el valor 2000. Esto lo podemos hacer como mostramos más arriba, usando la tarea `hedit`. En la Fig. 7 se muestran todos los parámetros que posee la misma y en particular, los que son



```

MSS_0153.fits[1024,1024][ushort]: HB89 Pol.67.5
No bad pixels, min=0., max=5658.
Line storage mode, physdim [1024,1024], length of user area 8303 s.u.
Created Fri 22:50:31 29-Jul-2011, Last modified Mon 14:39:39 08-Aug-2011
INSTRUME= 'CAFOS 2.2' / instrument ID
TELESCOP= 'CA-2.2 ' / telescope-id
OBJECT = 'HB89 Pol.67.5' / Target description
OBSERVER= 'XX' / Name of observer
ORIGIN = 'DSAZ ' / Max Planck Institut
FILENAME= 'MSS0153.fits' / Name of this File
EXPTIME = 120.00 / Total integration time (sec)
DARKTIME= 121.45 / Total elapsed time (sec)
RA = 331.230854 / 22:04:55.4 RA [deg]
DEC = 4.73137 / +04:43:52.9 DEC [deg]
HA = 1.000000 / DD:MM:JJ DEC (J2000) pointing (deg)
AIRMASS = 1.18592 / airmass
DATE = '2011-07-30T01:50:31' / file creation
LST = 79786.105000 / 22:09:46.1 sid. time [sec]
UTC = 6630.684000 / 01:50:30.7 U.T. at start [sec]
UT_START= 6630.684000 / 01:50:30.7 U.T. at start [sec]
UT_END = 6811.184000 / 01:53:31.2 U.T. [sec]
EQUINOX = 0.000 / Equinox in years
RADECSYS= 'NN ' / FK5?
MJD-OBS = 55772.07670000 / modified jul. date
DATE-OBS= '2011-07-30T01:50:31' / file creation
OBSERVAT= 'DSAZ ' / observatory
IMAGETYP= 'science ' / Type of observation
CAMID = 'CCD15 ' / Camera used
CCDNAME = 'SITE#1d_15' / Detector used
CCDPSIZ = 24.00 / Pixelsize in microns
CCDORI = 'XX ' / Orientation of CCD
CCDSPEED= 'normal ' / CCD readout speed
CCDBINX = 1 / Binning factor along X
CCDBINY = 1 / Binning factor along Y
CCDGAIN = 20.000 / total analog Gain
CCDSENS = 2.300 / Sensitivity at selected Gain (e-/DN)
CCDMEAN = 64.100 / Bias of CCD at selected Gain (DN)
CCDSAT = 190000.000 / Saturation of CCD (e-)
CCDRON = 5.060 / Readout-nois of CCD at selected Gain (e-)
CCDTEMP = -113.7 / dewar temperature in deg. C
BIASSEC = '[0,513;0,1536]' / overscan portion of frame
CCDSEC = '[513,513;1536,1536]' / orientation to full frame [x1:y1,x2:y2]

```

Figura 6: *Header* de una imagen sin editar ni procesar.

indispensables (*images, fields, value*). Vamos a realizar el cambio de valores de la siguiente manera, por línea de comandos:

```
>hedit MSS\_0153.fits equinox '2000' ver-
```

Observen que seguido de la imagen a la que quiero ejecutarle la tarea **hedit**, se especificó el parámetro y el valor que quiero asignarle, “ver-” esta indicando que haga el cambio sin preguntar. Luego, podemos comprobarlo en la Fig. 8 que el parámetro tomó el valor que le indique. Si ahora quisiera darle otro valor al parámetro BIASSEC (en la Fig. 6 vale [0:513,0:1536]), tengo que proceder de igual manera:

```
>hedit MSS\_0153.fits BIASSEC '[1:1024,1:1024]' ver-
```

```

PACKAGE = imutil
TASK = hedit

images =          IM1002 images to be edited
fields =          gain  fields to be edited
value =          16.2  value expression
(add =           no) add rather than edit fields
(addonly=        no) add only if field does not exist
(delete =        no) delete rather than edit fields
(verify =        yes) verify each edit operation
(show =          yes) print record of each edit operation
(update =        yes) enable updating of the image header
(mode =          ql)

```

Figura 7: Archivo de parámetros de la tarea *hedit*.

Y nuevamente, en la Fig. 8 vemos que tomó el nuevo valor. El mismo valor que le asigné al parámetro BIASSEC quiero asignárselo a CCDSEC, entonces:

```
>hedit MSS\_0153.fits CCDSEC '[1:124,1:124]' ver-
```

En este caso, me equivoqué al tipear los números y ejecuté la tarea. Esta acción modificó el archivo de datos de la imagen como se muestra en la Fig. 8. En caso de que me pase esto y necesite recuperar los valores originales lo que conviene es volver a cargar los *headers* originales de la imagen.

**IMPORTANTE:** Siempre hacer una copia de seguridad de las imágenes sin procesar.

## 4. Tareas

En esta sección vamos a explicar el uso y funcionamiento de algunas tareas que les van a ser útiles.

### 4.1. Display - DS9

Es una tarea que nos permite visualizar la imagen en el DS9. Supongamos que quiero desplegar la imagen *sttpv2.fits* en el *frame* 1, entonces tengo que ejecutar la siguiente línea de comando:

```
>display sttpv2.fits 1
```

Y automáticamente, en la ventana del DS9 se mostrará la imagen (ver Fig. 9). El “1” esta indicando en que *frame* despliego la imagen. Si no indico el número del *frame* me va a preguntar en pantalla en que *frame* quiero desplegar la imagen. El DS9 me permite desplegar varias imágenes si indico distintos números de *frames*, y para cambiarme de *frame*, en la ventana del DS9 (ver Fig. 9), seleccionando **frame** puedo moverme libremente entre un *frame* y otro o incluso borrar alguna imagen cargada.

Veamos otro ejemplo, despliego dos imágenes:

```

MSS_0153.fits[1024,1024][real]: HB89 Pol.67.5
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1024,1024], length of user area 9963 s.u.
Created Tue 12:35:26 30-Jul-2013, Last modified Sat 10:55:23 23-Jun-2012
INSTRUME= 'CAFOS 2.2'           / instrument ID
TELESCOP= 'CA-2.2 '           / telescope-id
OBJECT   = 'HB89 Pol.67.5'     / Target description
OBSERVER= 'XX'                 / Name of observer
FILENAME= 'MSS0153.fits'      / Name of this File
EXPTIME =          80.00 / Total integration time (sec)
DARKTIME=          121.45 / Total elapsed time (sec)
RA      =          331.230854 / 22:04:55.4 RA [deg]
DEC     =           4.73137 / +04:43:52.9 DEC [deg]
HA      =           1.000000 / DD:MM:JJ DEC (J2000) pointing (deg)
AIRMASS =           1.18592 / airmass
LST     =          79786.105000 / 22:09:46.1 sid. time [sec]
UTC     =          6630.684000 / 01:50:30.7 U.T. at start [sec]
UT_START= 6630.684000 / 01:50:30.7 U.T. at start [sec]
UT_END  = 6811.184000 / 01:53:31.2 U.T. [sec]
EQUINOX =           2000. / Equinox in years
DARKTIME=          121.45 / Total elapsed time (sec)
RA      =          331.230854 / 22:04:55.4 RA [deg]
DEC     =           4.73137 / +04:43:52.9 DEC [deg]
HA      =           1.000000 / DD:MM:JJ DEC (J2000) pointing (deg)
AIRMASS =           1.18592 / airmass
LST     =          79786.105000 / 22:09:46.1 sid. time [sec]
UTC     =          6630.684000 / 01:50:30.7 U.T. at start [sec]
UT_START= 6630.684000 / 01:50:30.7 U.T. at start [sec]
UT_END  = 6811.184000 / 01:53:31.2 U.T. [sec]
EQUINOX =           2000. / Equinox in years
RADECSYS= 'NN'                / FK5?
MJD-OBS  =          55772.07670000 / modified jul. date
DATE-OBS= '2011-07-30T01:50:31' / file creation
OBSERVAT= 'DSAZ'              / observatory
IMAGETYP= 'science'           / Type of observation
CAMID    = 'CCD15'            / Camera used
CCDNAME  = 'SITE#1d_15'       / Detector used
CCDPSIZ  =          24.00 / Pixelsize in microns
CCDORI   = 'XX'               / Orientation of CCD
CCDSPEED= 'normal'           / CCD readout speed
CCDBINX  =          1 / Binning factor along X
CCDBINY  =          1 / Binning factor along Y
CCDGAIN  =          20.000 / total analog Gain
CCDSENS  =          2.300 / Sensitivity at selected Gain (e-/DN)
CCDMEAN  =          64.100 / Bias of CCD at selected Gain (DN)
CCDSAT   =          190000.000 / Saturation of CCD (e-)
CCDRON   =          5.060 / Readout-nois of CCD at selected Gain (e-)
CCDTEMP  =          -113.7 / dewar temperature in deg. C
BIASSEC  = '[1:1024,1:1024]' / overscan portion of frame
CCDSEC   = '[1:124,1:124]' / orientation to full frame [x1:y1,x2:y2]

```

Figura 8: *Header* de una imagen editada y procesada.

```

>display sttpv2.fits 1
>display sttpr1.fits 2

```

y quiero verlas juntas, entonces en la ventana del DS9, voy a *frame* y luego seteo la opción *tile*, de esta manera en la ventana del DS9 se verá la imagen sttpv2.fits en el *frame 1* y la imagen sttpr1.fits en el *frame 2* (ver Fig. 10). Si quiero borrar del DS9 la imagen sttpv2.fits, me aseguro que el correspondiente *frame* esté resaltado en azul y seteo la opción *delete*. Otra

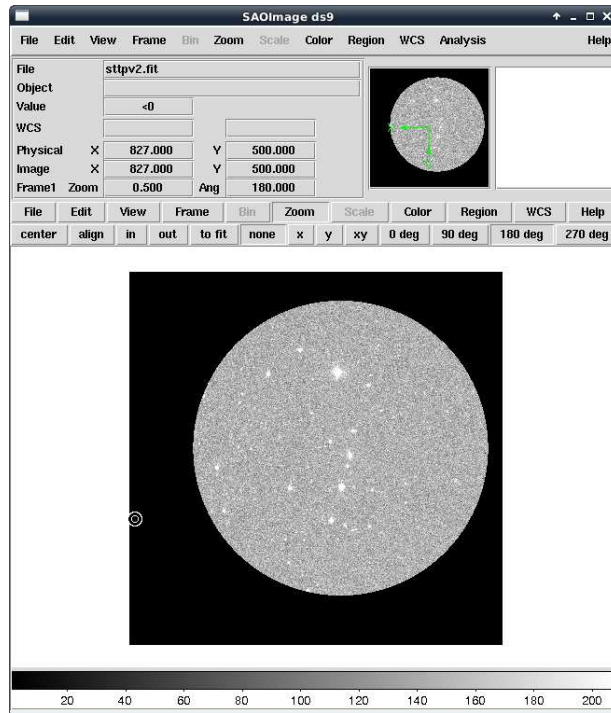


Figura 9: Ventana del DS9.

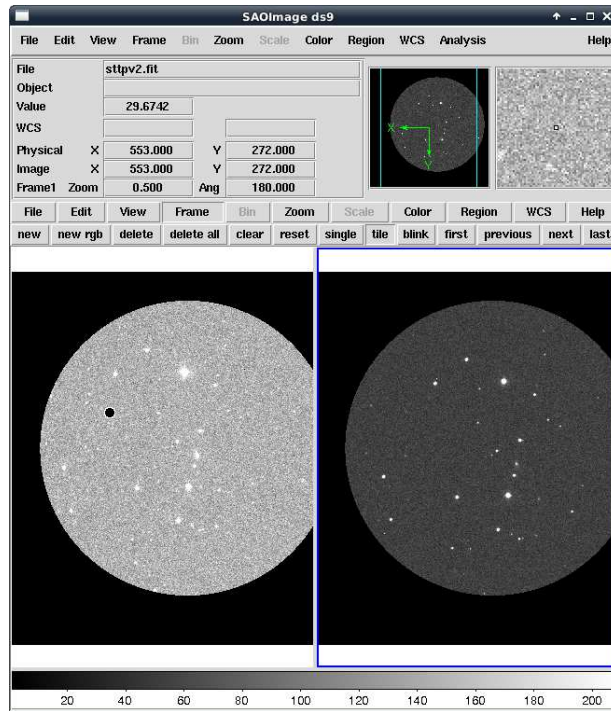


Figura 10: Captura de pantalla de la ventana del DS9 utilizando la opción *tile*.

opción útil que posee este “visualizador” es la de *zoom*, como pueden ver en la Fig. 10, la opción **zoom** me permite realizar un acercamiento o alejamiento en el campo de la imagen de forma simple. Además puedo modificar el contraste de la imagen desplazándome sobre la misma con el mouse a la vez que presiono “click” derecho o cambiar los colores si elijo la opción **color** en la barra de menú del DS9.

**Puede pasar que aparezca un círculo verde en el campo de la imagen al hacer “click” sobre el DS9. Estos círculos se pueden borrar si te parás en el centro, presionas “click” y luego “delete”**

## 4.2. Imexamine

Esta tarea permite hacer un análisis muy detallado interactivamente de una imagen previamente visualizada con el DS9. Es muy completa y posee muchas opciones útiles, acá veremos las más usadas, pero puedes recurrir a la ayuda y conocer las restantes (>help imexamine).

Una vez desplegada la imagen en el DS9, ejecutamos la tarea tipeando:

```
>imexa MSS120.fits
```

el cursor cambiará a un cursor circular parpeadeante y se moverá a la ventana del DS9. Ahora estamos habilitados para estudiar regiones de interés con alguna de las tantas opciones que veremos a continuación:

- “r” Perfil radial del objeto en el cual se centro el cursor, además nos muestra otros parámetros interesantes como el *FWHM*, que es el último valor de la fila de abajo en la ventana que emerge al ejecutar **imexamine** (ver Fig. 11).
- “c” Dibuja la columna marcada por el cursor.
- “l” Dibuja la fila marcada por el cursor.
- “m” Calcula la estadística de una zona alrededor del cursor.
- “h” Histograma de una zona alrededor del cursor.
- “s” Diagrama de superficie en torno a la posición del cursor.
- “e” Diagrama de contorno en torno a la posición del cursor.
- “q” sale del **imexamine**.

Además, es posible pasar de la ventana del DS9 a la ventana gráfica tipeando “g” sobre la ventana del DS9, de esta manera el foco estará ahora sobre la ventana gráfica. También podemos volver a la ventana del DS9 presionando “i” sobre la ventana gráfica y si todo está bien el cursor volverá a la ventana del DS9. Esto es útil si, por ejemplo, quiero guardar algún gráfico que generé utilizando la opción “r” de **imexamine**. En este caso, luego de

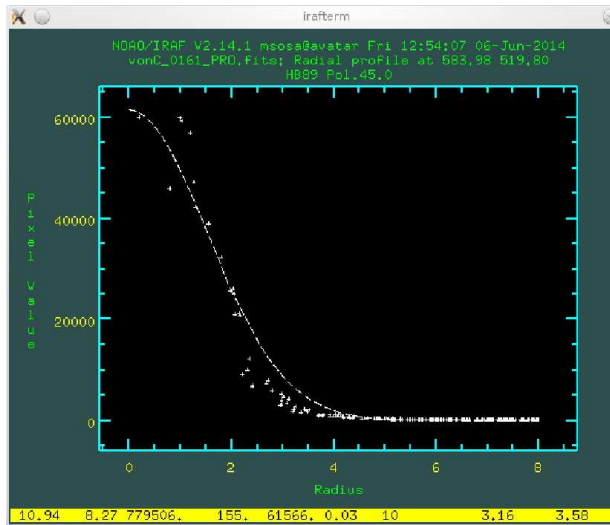


Figura 11: Ventana gráfica que se despliega al utilizar la opción “r” de la tarea **imexamine** sobre el centro de una estrella del campo estudiado.

presionar “r” sobre el centro del objeto que quiero estudiar, presiono “g”, el foco se mueve a la ventana gráfica, y allí tipeo: **.snap eps** y presiono *enter*. De esta manera, dentro del directorio donde estoy aparecerá un gráfico con extensión *eps* del estilo “sgi15127.eps”, al cual después podremos cambiarle el nombre si lo deseamos. Una vez hecho esto, presiono “i”, para volver a la ventana del DS9 y si termine con el **imexamine**, presiono “q” y el foco pasa a la ventana de IRAF.

**Es importante ser cuidadosos a la hora de ejecutar comandos como “g”, “i” o “q” de manera tal de hacerlo dentro de las ventanas correspondientes por que es posible que si no se hace correctamente no se pueda continuar con la sesión de IRAF.**

### 4.3. Phot

Esta tarea puede utilizarse para hacer fotometría. Podemos ir a ver el *help* para ver los parámetros que contiene:

```
> help phot
```

Como se muestra en la Fig. 12, vemos que esta tarea es parte del paquete *apphot*, por lo tanto cuando la ejecutemos tenemos que asegurarnos de que éste paquete esté cargado y por ende los paquetes *noao* y *digiphot*. Con **>epar phot**, podemos editar los parámetros necesarios. En la Fig. 13 vemos que estos son: *image* y *skyfile*. Pero la tarea **phot** requiere que se editen los parámetros de otras tareas que utiliza al ejecutarla, estas son: *datapars*, *centerpars*, *fitskypars* y *photpars* (en la Fig. 13 se muestran dentro de la lista de parámetros). Esto podemos hacerlo normalmente con **epar datapars**, o bien, desde la

```

marina@lupus: ~
PHOT (May00)      noao.digiphot.apphot      PHOT (May00)

NAME
  phot -- do aperture photometry on a list of stars

USAGE
  phot image

PARAMETERS

  image
  The list of images containing the objects to be measured.

  skyfile = ""
  The list of text files containing the sky values, of the
  measured objects, one object per line with x, y, the sky value,
  sky sigma, sky skew, number of sky pixels and number of
  rejected sky pixels in columns one to seven respectively. The
  number of sky files must be zero, one, or equal to the number
  of input images. A skyfile value is only requested if
  fitstups, salgorithm = "file" and if PHOT is run
  non-interactively.

  coords = ""
  The list of text files containing initial coordinates for the
  objects to be centered. Objects are listed in coords one object
  per line with the initial coordinate values in columns one and
  two. The number of coordinate files must be zero, one, or equal
  to the number of images. If coords is "default",
  "dir$default", or a directory specification then a coords file
  name of the form dir$root.extension.version is constructed and
  searched for, where dir is the directory, root is the root
  image name, extension is "coo" and version is the next
  available version number for the file.

  output = "default"
  The name of the results file or results directory. If output is
  "default", "dir$default", or a directory specification then an
  output file name of the form dir$root.extension.version is
  constructed, where dir is the directory, root is the root image
  name, extension is "mag" and version is the next available
  version number for the file. The number of output files must be
  zero, one, or equal to the number of image files. In both
  interactive and batch mode full output is written to output. In
  interactive mode an output summary is also written to the
  standard output.

  plotfile = ""
  The name of the file containing radial profile plots of the
  [c=quit,d=downhalf,flsp=downfull,iler=downline,N=next]

```

Figura 12: Captura de pantalla del *Help* de la tarea **phot**

lista de parámetros de la tarea **phot** que se despliega al hacer `>epar phot`, nos paramos sobre **datapars** y tipeamos “e”, y se listaran los parámetros, ahora, de la tarea **datapars**, para que podamos editarlos, como se muestra en la Fig. 14. Lo mismo para las demás (ver Fig. 15, 16 y 17). De cada una de estas tareas que se encuentran dentro de **phot** salimos con “q” y se graban los parámetros que editamos. Una vez que seteamos los parámetros que nos interesan, ejecutamos la tarea **phot** de la manera usual.

## 5. Para seguir leyendo

- Introduction to IRAF - <http://www.twilightlandscapes.com/IRAFtutorial/index.html>.
- Reducción de datos astronómicos usando IRAF, Instituto de Astrofísica de Canarias - [http://www.iac.es/galeria/jap/IRAF\\_notes/notas.html#\\_Toc467774861](http://www.iac.es/galeria/jap/IRAF_notes/notas.html#_Toc467774861).

```

marina@lupus: ~
I R A F
Image Reduction and Analysis Facility

PACKAGE = apphot
TASK = phot

image =
skyfile =
(coords =
(output = default)
(plotfil=
(datapar=
(centerp=
(fitskyp=
(photpar=
(interac= yes)
(radplot= no)
(icomman=
(gcomman=
)_wcsin =
)_wcsout =
(cache =
)_verify =
)_update =
(verbose=
)_graphics =
)_display =
(mode = ql)
I

```

Figura 13: Lista de parámetros de la tarea **phot**.

```

marina@lupus: ~
I R A F
Image Reduction and Analysis Facility

PACKAGE = apphot
TASK = datapars

(scale = 1.)
(fhwhp= 2.5)
(emissio= yes)
(sigma = INDEF)
(datamin= INDEF)
(datamax= INDEF)
(noise = poisson)
(ccdread=
)_gain =
(readnoi= 0.)
(epadu = 1.)
(exposur=
)_airmass=
)_filter =
)_obstime=
)_itime =
)_vairmass=
)_ifilter=
)_otime =
(mode = ql)
I
ESC-? for HELP

```

Figura 14: Lista de parámetros de la tarea **datapars**.



```

marina@lupus: ~
I R A F
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = centerpars

(salgori=  centroid) Centering algorithm
(cbox = 5.) Centering box width in scale units
(cthresh= 0.) Centering threshold in sigma above background
(minsnr= 1.) Minimum signal-to-noise ratio for centering algorithm
(cmaxite= 10) Maximum number of iterations for centering algorithm
(maxshif= 1.) Maximum center shift in scale units
(clean = no) Symmetry clean before centering ?
(rclean = 1.) Cleaning radius in scale units
(rclip = 2.) Clipping radius in scale units
(kclean = 3.) Rejection limit in sigma
(mkcente= no) Mark the computed center on display ?
(mode = q1)

I

ESC=? for HELP

```

Figura 15: Lista de parámetros de la tarea **centerpars**.

```

marina@lupus: ~
I R A F
Image Reduction and Analysis Facility
PACKAGE = apphot
TASK = fitskypars

(salgori=  mean) Sky fitting algorithm
(annulus= 18.) Inner radius of sky annulus in scale units
(dannulu= 6.) Width of sky annulus in scale units
(skyvalu= 0.) User sky value
(smaxite= 10) Maximum number of sky fitting iterations
(sloclip= 0.) Lower clipping factor in percent
(shiclip= 0.) Upper clipping factor in percent
(snrreje= 50) Maximum number of sky fitting rejection iterations
(sloreje= 3.) Lower K-sigma rejection limit in sky sigma
(shireje= 3.) Upper K-sigma rejection limit in sky sigma
(khist = 3.) Half width of histogram in sky sigma
(binsize= 0.1) Binsize of histogram in sky sigma
(smooth = no) Boxcar smooth the histogram
(rgrow = 0.) Region growing radius in scale units
(mksky = no) Mark sky annuli on the display
(mode = q1)

I

ESC=? for HELP

```

Figura 16: Lista de parámetros de la tarea **fitskypars**.

```
marina@lupus: ~  
I R A F  
Image Reduction and Analysis Facility  
PACKAGE = apphot  
TASK = photpars  
(weighti=  constant) Photometric weighting scheme for wphot  
(apertur=  3.) List of aperture radii in scale units  
(zmag =  25.) Zero point of magnitude scale  
(nk-apert= no) Draw apertures on the display  
(mode =  q)  
  
I  
  
ESC for HELP
```

Figura 17: Lista de parámetros de la tarea **photpars**.