



# Desarrollo de Software en Astrofísica

Sala 763 Pabellón Forma, Profesor Sebastián Pérez  
14 de Marzo 2023

# Desarrollo de Software en Astrofísica

- La ciencia está experimentando una explosión de datos y la astronomía está a la cabeza. Los telescopios modernos producen terabytes de datos por cada observación y las simulaciones necesarias para modelar nuestro Universo observable llevan a las supercomputadoras al límite. Así, la astrofísica ha entrado en una era en la que el software bien diseñado es tan importante para el éxito de un proyecto como el hardware.

# Desarrollo de Software en Astrofísica

- Algunos ejemplos de software altamente complejos que se usan en astrofísica incluyen herramientas de reducción de datos para la instrumentación en telescopios o 'data reduction pipelines' (por ejemplo, CASA usado en radioastronomía), paquetes para análisis de datos, como astropy, por ejemplo, y códigos que simulan procesos físicos, como la hidrodinámica (por ejemplo, los códigos FARGO3D y Phantom), la evolución estelar (por ejemplo, el código MESA), la dinámica de N-cuerpos (el código GADGET) y la transferencia radiativa (por ejemplo, RADMC3D).

# Desarrollo de Software en Astrofísica

- Algunos ejemplos de software altamente complejos que se usan en astrofísica incluyen herramientas de reducción de datos para la instrumentación en telescopios o 'data reduction pipelines' (por ejemplo, CASA usado en radioastronomía), paquetes para análisis de datos, como astropy, por ejemplo, y códigos que simulan procesos físicos, como la hidrodinámica (por ejemplo, los códigos FARGO3D y Phantom), la evolución estelar (por ejemplo, el código MESA), la dinámica de N-cuerpos (el código GADGET) y la transferencia radiativa (por ejemplo, RADMC3D).



astropy  
A Community Python Library for Astronomy



```
=====
R_u IRAF64 pre-alpha based on N060/IRAFNET PC-IRAF Revision 2.14.1
('v') This is NOT the released version of IRAF64.
Visit http://www.ir.isas.jaxa.jp/~cyanauch/iraf64/ to report problems.

This product includes results achieved by the IRAF64 project conducted
by Chisato Yamauchi (ISAS/JAXA).

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.

The following commands or packages are currently defined:

dataio. language. obsolete. softools. utilities.
dms. lists. plot. system.
images. noao. proto. tables.

c> lds%
c> displ dev$pix
frame to be written into (116) (1):
z1=35, z2=546,0218
c> instat dev$pix
*          IMAGE      NPIX      MEAN      STDEV      MIN      MAX
dev$pix: 282144      108.3      131.3      -1.      19936.
c> inexam dev$pix
display frame (1) (1):
[]

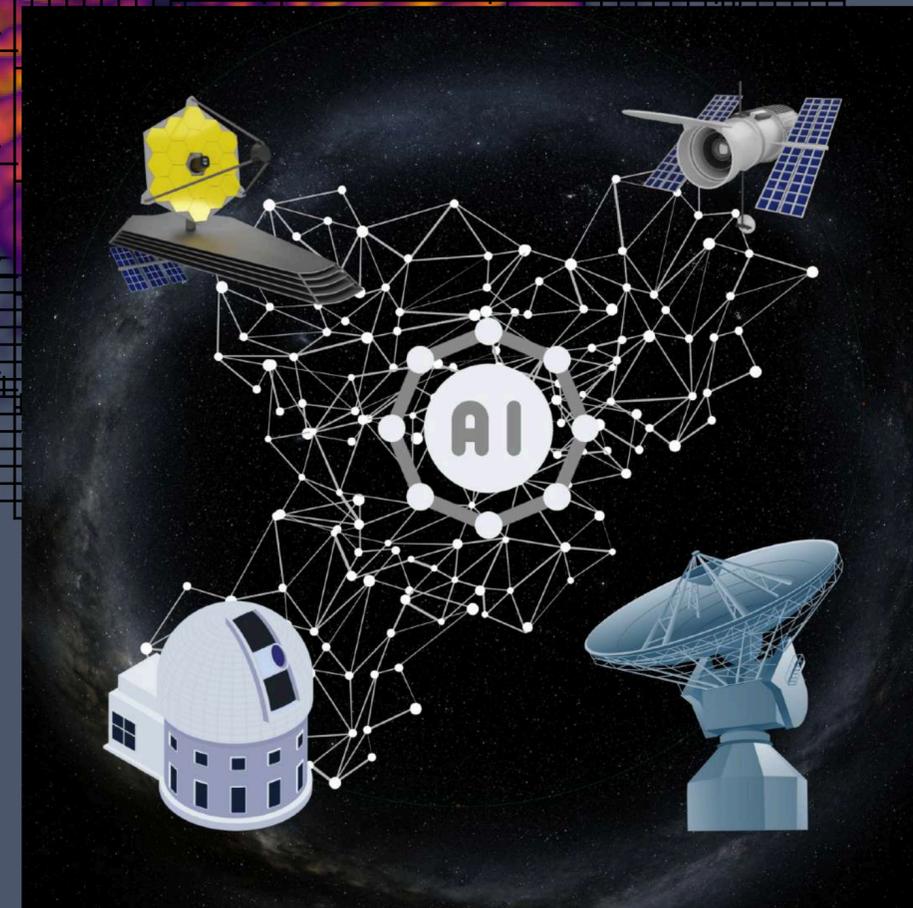
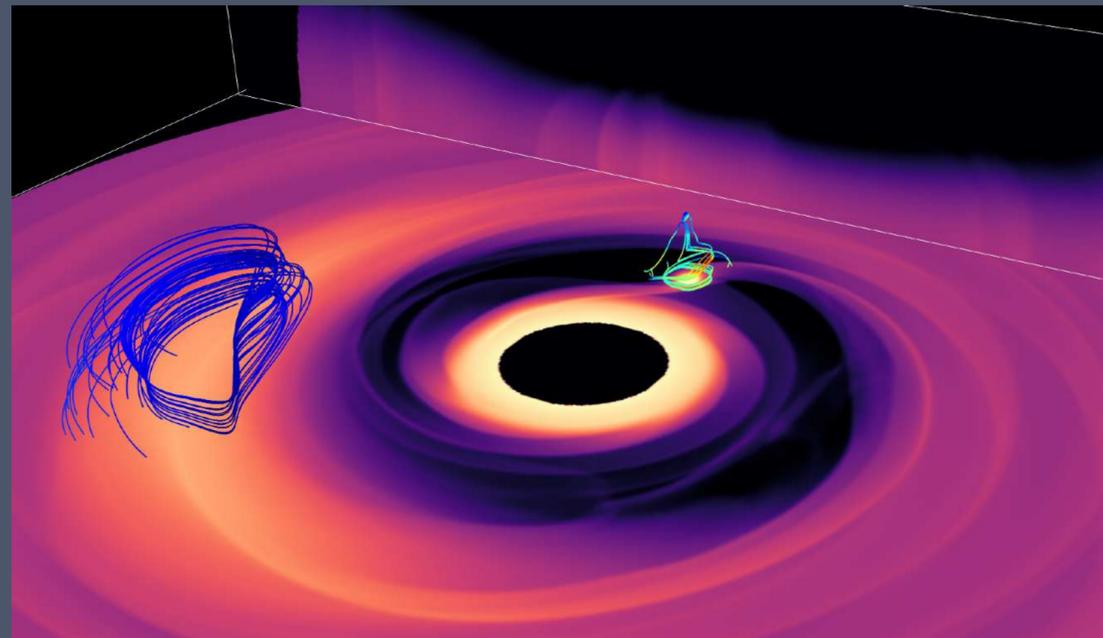
=====
N/A07/IRAF V2.14.1 cyanauch@zugo.plain.isas.jaxa.jp Sat 22:50:29 20-Jun-2009
pix: Surface plot of [248:266,250:270]
w51 b 600s

=====
LOCALHOST: localhos 31DEC21 TST: Cpu= 0.1 Real= 0 IO= 4
LOCALHOST: RLDIF: Task RLDIF (release of 31DEC21) begins
LOCALHOST: RLDIF: UVGET: Using flag table version 1 to edit data
LOCALHOST: RLDIF: Calibrating gains polarization
LOCALHOST: RLDIF: Not calibrating bandpass
LOCALHOST: RLDIF: Using known POLANGLE for 1328-307
LOCALHOST: RLDIF: IF 1 avg RL phase 40.179 -- 0.068
LOCALHOST: RLDIF: IF 2 avg RL phase 171.848 -- 0.101
LOCALHOST: RLDIF: returns adverbs to AIPS
LOCALHOST: LISTR: appears to have ended successfully
LOCALHOST: LISTR: localhos 31DEC21 TST: Cpu= 0.0 Real= 0 IO= 5
LOCALHOST: LISTR: Task LISTR (release of 31DEC21) begins
LOCALHOST: LISTR: localhos 31DEC21 TST: Cpu= 0.0 Real= 1
LOCALHOST: UVPLT: Task UVPLT (release of 31DEC21) begins
LOCALHOST: UVPLT: UVGET: Using flag table version 1 to edit data
LOCALHOST: UVPLT: Calibrating gains polarization
LOCALHOST: UVPLT: Not calibrating bandpass
LOCALHOST: UVPLT: SCRL at visibility 1
LOCALHOST: UVPLT: PPLAN at visibility 1
LOCALHOST: UVPLT: PPLAN: 2532 Points plotted
LOCALHOST: UVPLT: appears to have ended successfully
LOCALHOST: UVPLT: localhos 31DEC21 TST: Cpu= 0.1 Real= 0 IO= 4
```

# Desarrollo de Software en Astrofísica

- Por otro lado, las técnicas estadísticas avanzadas y el aprendizaje automático juegan un papel cada vez más importante en la reducción de grandes volúmenes de datos en astronomía, y también requieren de códigos complejos. Muchos paquetes de software son desarrollados por grandes equipos y deben hacer uso de tipos heterogéneos de plataformas de hardware, desde CPUs de propósito general que se ejecutan en computadoras portátiles, hasta computación multinúcleo (clusters) que hacen uso de paralelización masiva y unidades de procesamiento gráfico (GPU).

$$v \rightarrow \vec{v}[\text{km/s}]$$



# Desarrollo de Software en Astrofísica

- El desarrollo de software se ha convertido en una parte esencial de todos los subcampos de la astronomía y el ecosistema científico está implementando cada vez más maneras de acreditar, citar, y medir el impacto de la investigación y desarrollo de software. Hasta ahora el mundo académico y científico solo se fijaba en la contribución a nivel de "publicaciones" o "papers" para evaluar cuán exitosa es la carrera de una o un científico. Hoy entramos a la era de la astronomía de los "petabytes" donde hacer contribuciones significativas a software será tan valioso como el impacto en papers. **En la Licenciatura en Astrofísica con mención en Ciencia de Datos reconocemos que el desarrollo de software es parte importante de la formación de futuras astrofísicas y astrofísicos.**

# Desarrollo de Software en Astrofísica

- Las habilidades clave necesarias para la investigación científica intensiva en datos también son muy valoradas en sectores de la industria, el gobierno y los medios/comunicaciones. Estas habilidades requieren conocimiento del oficio de desarrollar software.

# Objetivo del Curso

- A lo largo de este curso, trabajarás con un pequeño grupo de colaboradores (entre 1 y 3 personas máximo) en un proyecto final, un paquete Python real de código abierto para astronomía. Si ya tienes compañeros de grupo en mente, escriban al correo electrónico del profesor de cátedra. De lo contrario, les asignaremos grupos de forma aleatoria. Esta es una oportunidad para poner en práctica el conocimiento que han aprendido en los cursos anteriores de la carrera. Al final del curso tendrán la oportunidad de presentar su paquete a los demás estudiantes del curso. Antes de comenzar, con su grupo logren:
  - \* Crear una cuenta GitHub

# Estructura del Curso

- El curso se separa en dos partes: (1) la parte técnica, de programación, y (2) la parte más humana, en cuanto al trabajo en equipo y el contacto con el usuario.

# Evaluación del Curso

Desarrollar un paquete Python de código abierto para astronomía desde cero. Su proyecto debe cumplir con los siguientes requerimientos:

- \* [Versionamiento] => Mantenerse en GitHub, generar releases.
- \* [Documentación] => Comentar cada función del código, y en las funciones que son más "complejas", indicar la lógica del algoritmo. Cada función debe llevar nombres representativos (que indiquen acciones) y variables.
- \* [Modulación] => Generar módulos que en su conjunto cumplan el objetivo del software. Cada módulo se compone de funciones, métodos y/o atributos que resuelven una parte del software.
- \* [Pruebas] => Contener tests de funcionamiento, que son las que indican si el software cumple con los objetivos planteados.
- \* [Compatibilidad] => Ser pip instalable y ejecutable en otras computadoras.
- \* [Distribución] => Ser una herramienta con una API (es decir, no un solo script o notebook). Deberíamos poder importar el paquete y llamarlo.

# Evaluación del Curso

Desarrollar un paquete Python de código abierto para astronomía desde cero. Su proyecto debe cumplir con los siguientes requerimientos:

- \* [Versionamiento] => Mantenerse en GitHub, generar releases.
- \* [Documentación] => Comentar cada función del código, y en las funciones que son más "complejas", indicar la lógica del algoritmo. Cada función debe llevar nombres representativos (que indiquen acciones) y variables.
- \* [Modulación] => Generar módulos que en su conjunto cumplan el objetivo del software. Cada módulo se compone de funciones, métodos y/o atributos que resuelven una parte del software.
- \* [Pruebas] => Contener tests de funcionamiento, que son las que indican si el software cumple con los objetivos planteados.
- \* [Compatibilidad] => Ser pip instalable y ejecutable en otras computadoras.
- \* [Distribución] => Ser una herramienta con una API (es decir, no un solo script o notebook). Deberíamos poder importar el paquete y llamarlo.

# Data Challenge

Leer y graficar la imagen de su objeto astronómico favorito.

1. Decidir cuál es el objeto a visualizar y en qué longitud de onda (elegir un survey de todo el cielo como 2mass para el infrarrojo o sdss para el óptico).
2. Obtener la imagen FITS desde una base de datos (ejemplo: skyview virtual observatory).
3. Escribir un pequeño código PYTHON que lea la imagen FITS.
4. Plotear la imagen con alguna librería ad-hoc.
5. Usar coordenadas apropiadas para graficar usando el HEADER.