

# Desarrollo de Software en Astrofísica

Sala 763 Pabellón Forma,  
Profesor Sebastián Pérez  
23 de mayo 2023



Un poco de astronomía primero...

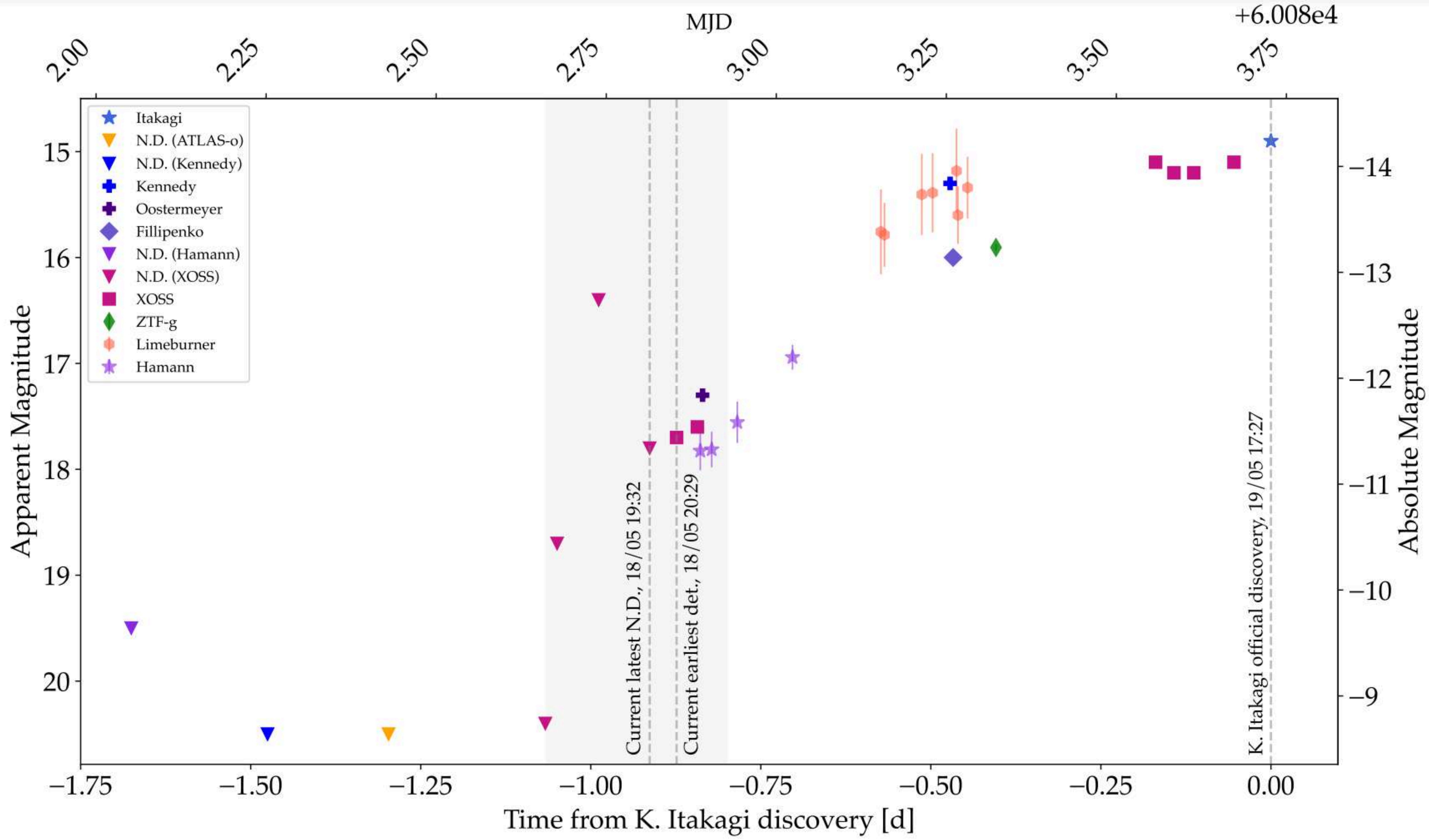
Aaron Wasserman  
SN 2023ixf in M101

2023/04/28

#SN2023ixf Koichi Itakagi  
closest Type 2 supernova in a decade.









Day 000





# Veamos el DATA CHALLENGE de la clase pasada

Escribe un código en Python de **programación orientada a objetos** que lea datos de una **tabla ASCII (bajar del link más abajo)** con información sobre exoplanetas y crea un gráfico de masa vs radio orbital. También se pueden proponer otros gráficos para visualizar otras propiedades de los exoplanetas.

Los datos en la tabla son 5 columnas para: Nombre del planeta, Masa relativa a la Tierra, Radio relativo a la Tierra, Semieje mayor de la orbita (unidad astronómica), Periodo Orbital (días)

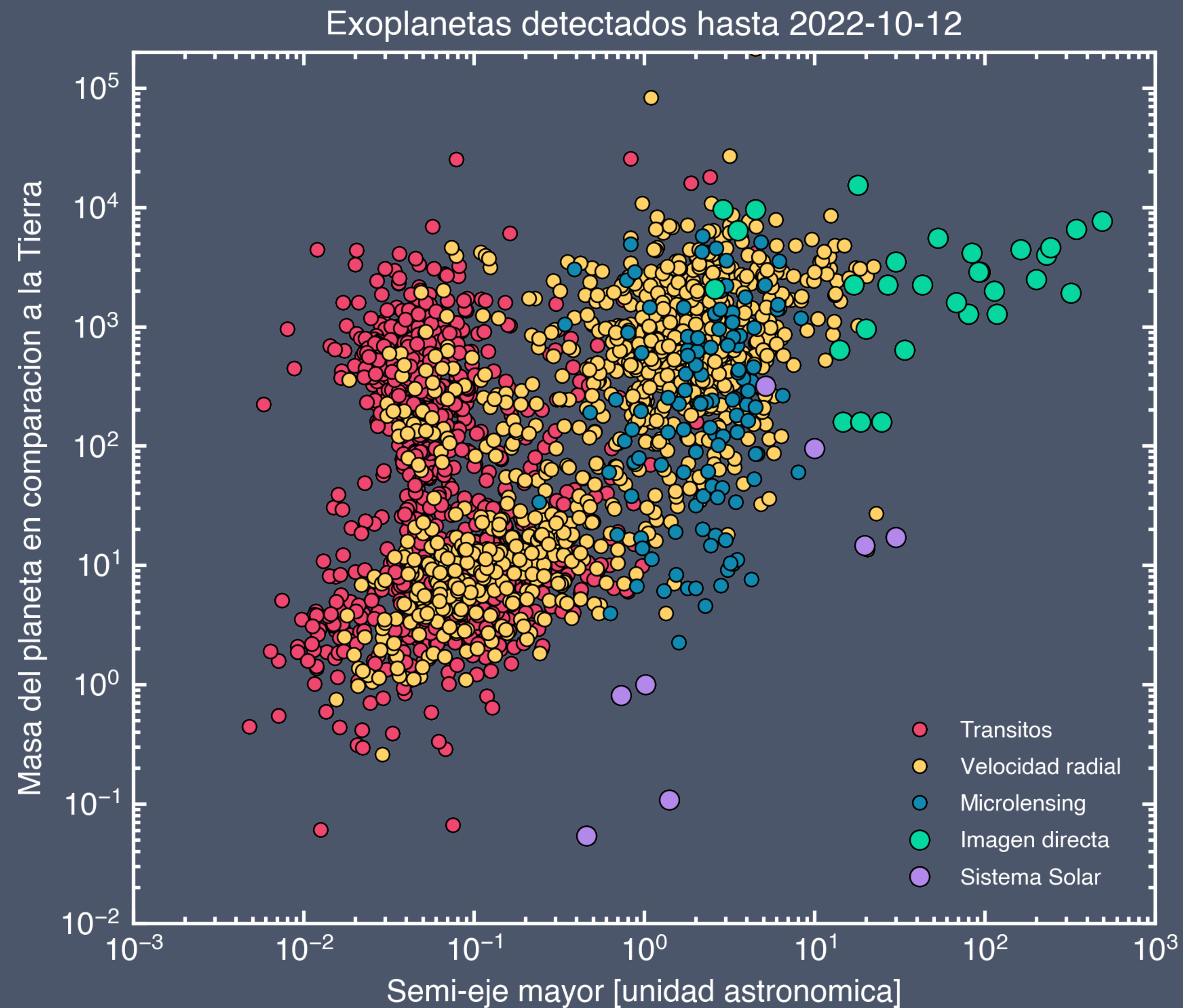
**Sugerencia:** define dos clases, una para representar un objeto “exoplaneta” que tenga los atributos “nombre, masa, radio, orbita, periodo” y una clase para la tabla de datos de exoplanetas.

Usa matplotlib y trata de usar escala log-log (el rango de tamaños de órbita y masas es muy grande!).

[https://www.dropbox.com/s/36h1j5nwzp947a9/exoplanets\\_all.txt?dl=0](https://www.dropbox.com/s/36h1j5nwzp947a9/exoplanets_all.txt?dl=0)

```
2M 0746+20 b,9534.85,10.87,2.897,4640.00
2M 2140+16 b,6356.57,10.31,3.530,7340.00
2M 2206-20 b,9534.85,14.57,4.480,8686.00
51 Eri b,635.66,11.21,14.000,15000.00
55 Cancri e,7.98,1.88,0.015,0.74
AU Mic b,57.21,4.20,0.066,8.46
BD+20 594 b,16.30,2.23,0.241,41.69
BD+45 564 b,432.25,13.56,0.830,307.88
BD+55 362 b,228.84,14.01,0.780,265.59
BD+63 1405 b,1258.60,13.00,2.060,1198.48
CFHTWIR-Oph 98 A b,2479.06,20.85,200.000,8040000.00
CoRoT-10 b,874.03,10.87,0.105,13.24
CoRoT-11 b,791.39,15.58,0.044,2.99
CoRoT-12 b,291.45,16.14,0.040,2.83
CoRoT-13 b,415.72,9.92,0.051,4.04
CoRoT-14 b,2415.50,12.22,0.027,1.51
CoRoT-16 b,168.45,13.11,0.062,5.35
CoRoT-17 b,778.68,11.43,0.046,3.77
```

# La única manera de aprender P00 es practicando DATA CHALLENGE

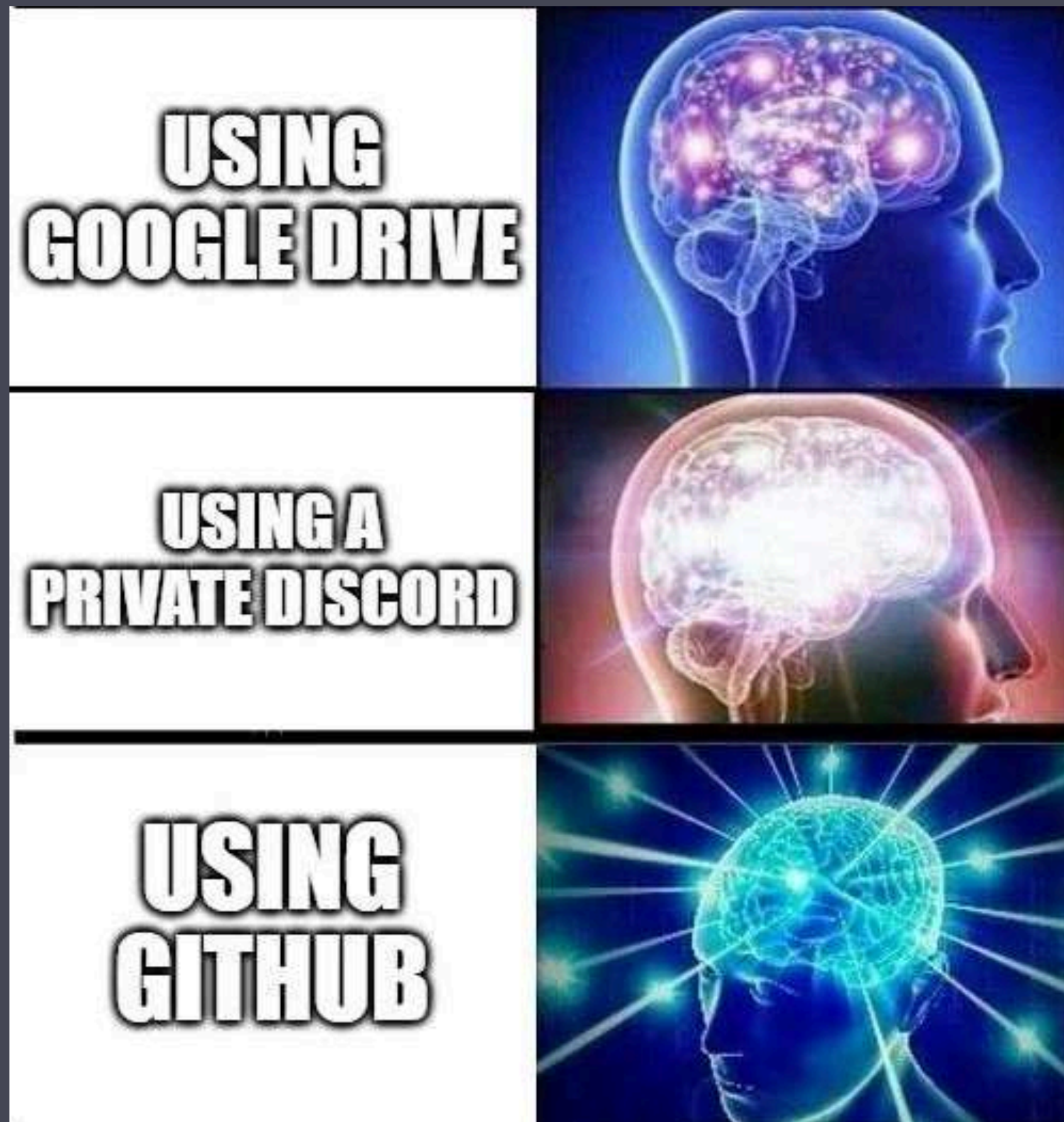


<https://gist.github.com/seba-perez>



# Introducción a Git (para astrónomos)

o cómo usar repositorios para sus proyectos de desarrollo



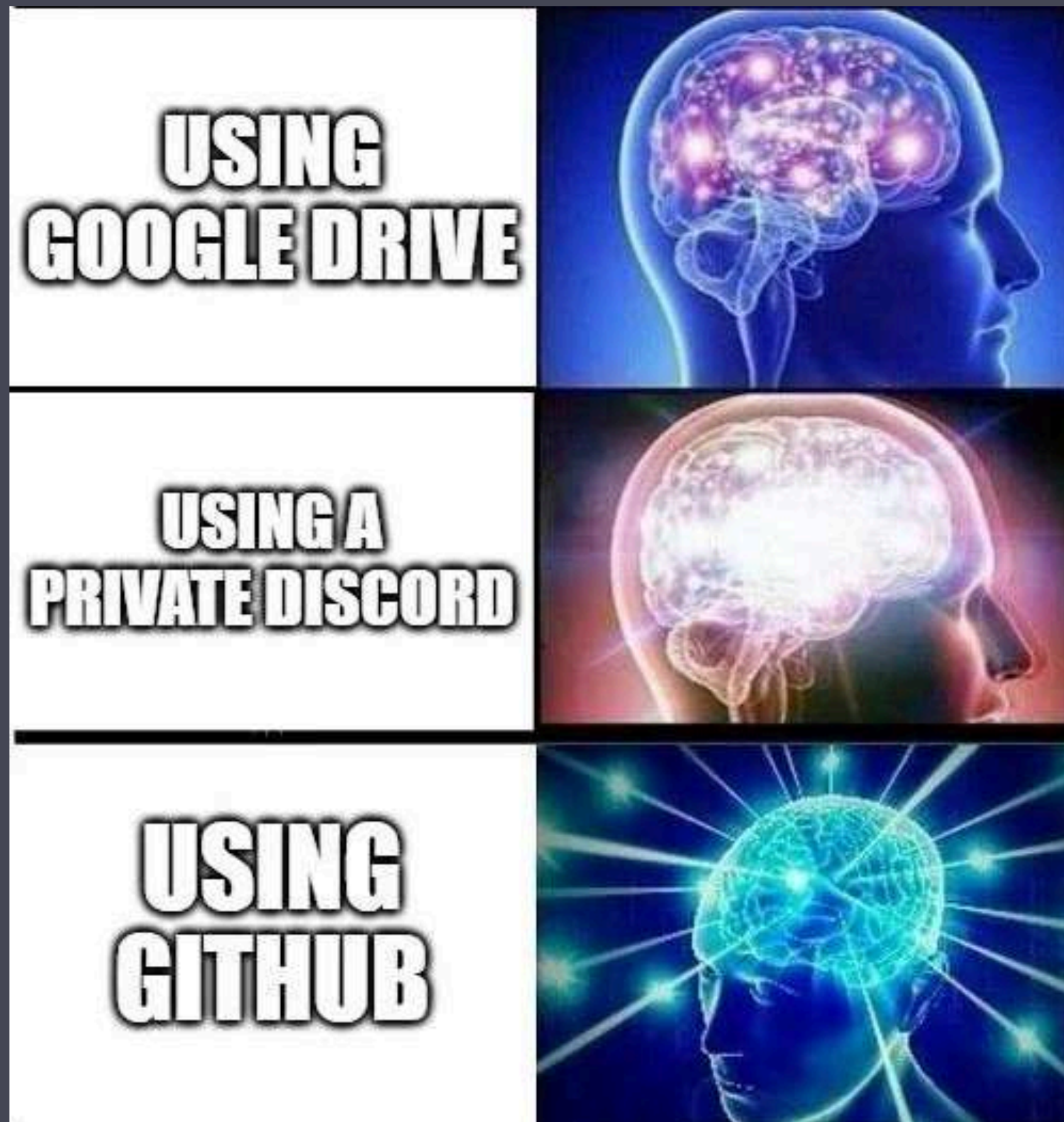
Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Permite a los equipos de desarrollo gestionar de manera eficiente los cambios en el código fuente y colaborar en proyectos de forma efectiva.

Además, GitHub es una plataforma de alojamiento de repositorios Git que facilita el trabajo colaborativo y proporciona herramientas adicionales para el desarrollo de software.



# Introducción a Git (para astrónomos)

o cómo usar repositorios para sus proyectos de desarrollo



Git fue creado por Linus Torvalds en 2005, el mismo desarrollador del kernel de Linux. Surgió como una solución a los problemas que presentaban los sistemas de control de versiones centralizados en ese momento. Git se diseñó para ser rápido, escalable y resistente a fallas, lo que lo convirtió en una herramienta popular tanto para proyectos pequeños como para proyectos de gran envergadura.



# Introducción a Git (para astrónomes)

Conceptos básicos de Git:

- **Repositorio**: Un repositorio de Git es una colección de archivos y directorios, junto con el historial de cambios de cada archivo. Puedes tener un repositorio local en tu máquina y también alojarlo en servicios como GitHub.
- **Commit**: Un commit es un registro de cambios en el repositorio. Representa un punto en el historial donde se guarda una versión del código. Los commits contienen metadatos como el autor, la fecha y un mensaje descriptivo.
- **Ramas (Branches)**: Las ramas permiten crear líneas de desarrollo independientes en un repositorio. Puedes crear nuevas ramas para trabajar en nuevas funcionalidades o solucionar problemas sin afectar la rama principal (normalmente llamada "master" o "main"). Las ramas se pueden fusionar más tarde para incorporar los cambios.
- **Clonar**: Clonar un repositorio implica crear una copia local de un repositorio remoto en tu máquina. Esto te permite trabajar en el código sin afectar el repositorio principal.
- **Push y Pull**: Push es la acción de subir los cambios locales a un repositorio remoto, mientras que pull es la acción de obtener los cambios más recientes del repositorio remoto y fusionarlos con tu copia local.