

Simulador de sistemas planetarios

SANTIAGO MESA VELÁSQUEZ.

ASESOR: EDWARD VILLEGAS



Una Introducción

► Python

Astronomía

Unión Python y astronomía



Python 3.6



Programming is an Art



¿Qué es Python?

- ▶ Un lenguaje de programación
- ▶ Permite el desarrollo de programas



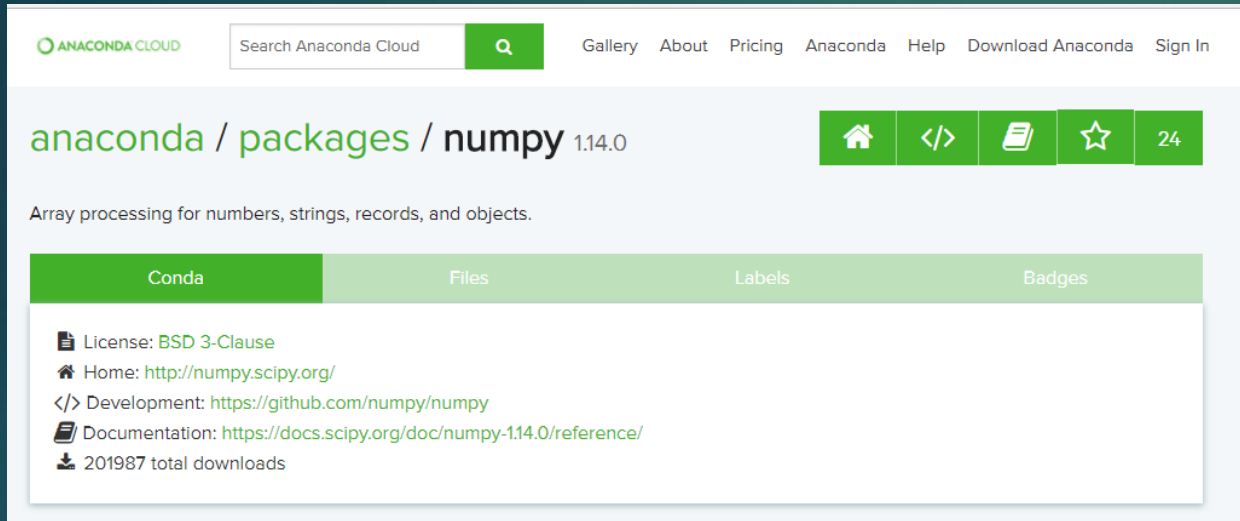


ANACONDA

Powered by Continuum Analytics[®]



¿Qué es?



ANACONDA CLOUD

Search Anaconda Cloud

Gallery About Pricing Anaconda Help Download Anaconda Sign In

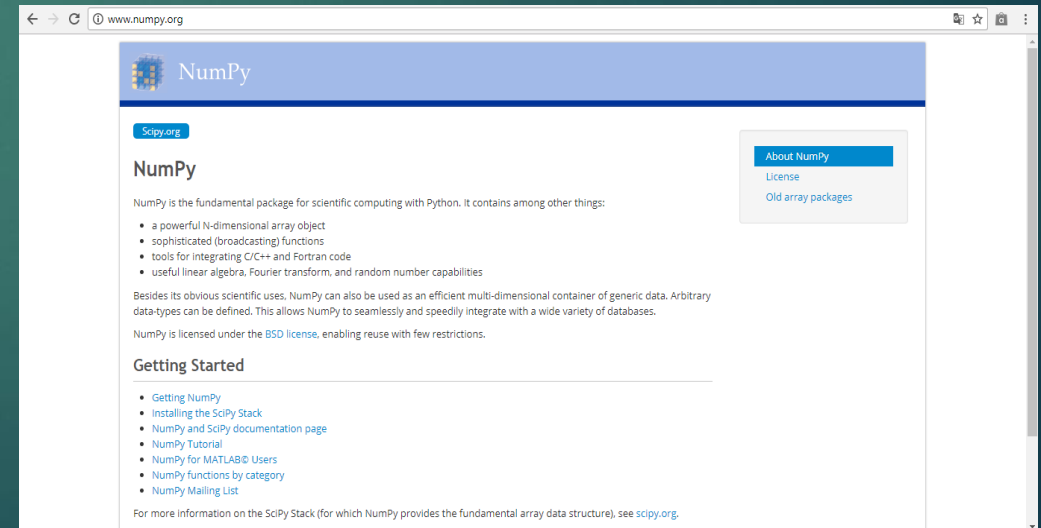
anaconda / packages / numpy 1.14.0

Array processing for numbers, strings, records, and objects.

Conda Files Labels Badges

License: BSD 3-Clause
Home: <http://numpy.scipy.org/>
Development: <https://github.com/numpy/numpy>
Documentation: <https://docs.scipy.org/doc/numpy-1.14.0/reference/>
201987 total downloads

► Donde una comunidad comparte programas



NumPy

SciPy.org

About NumPy
License
Old array packages

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

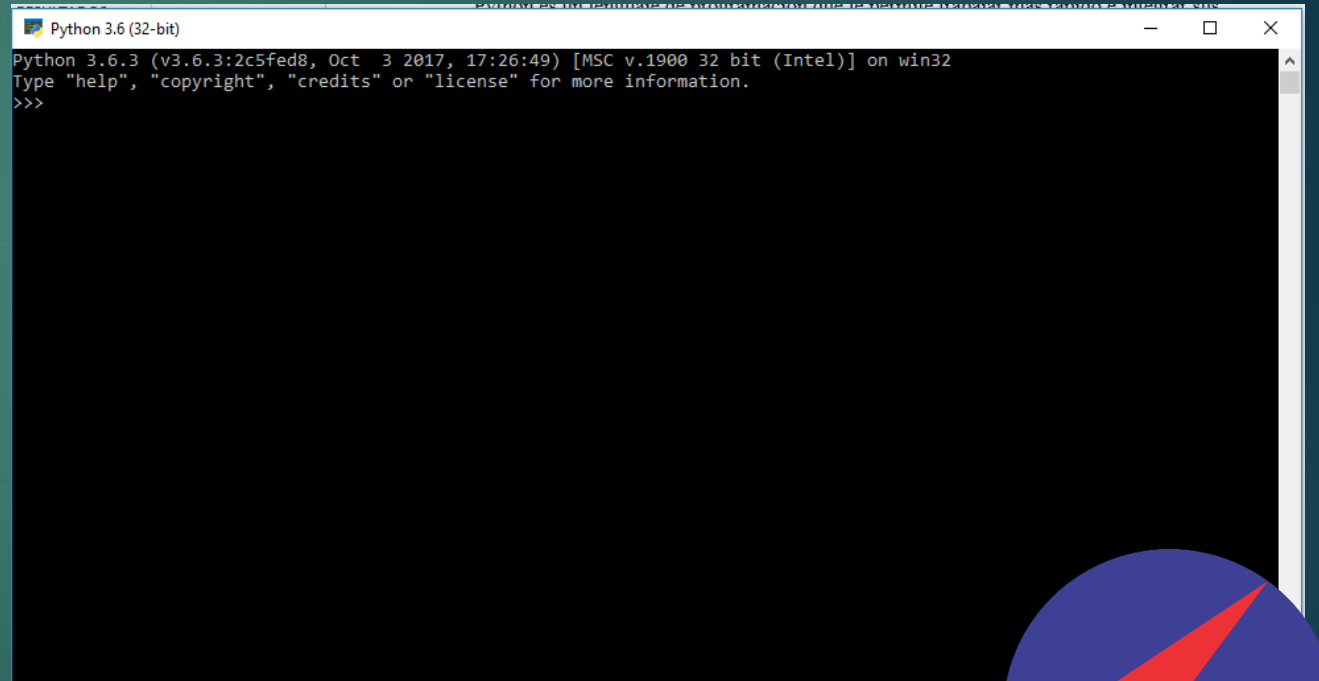
NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

Getting Started

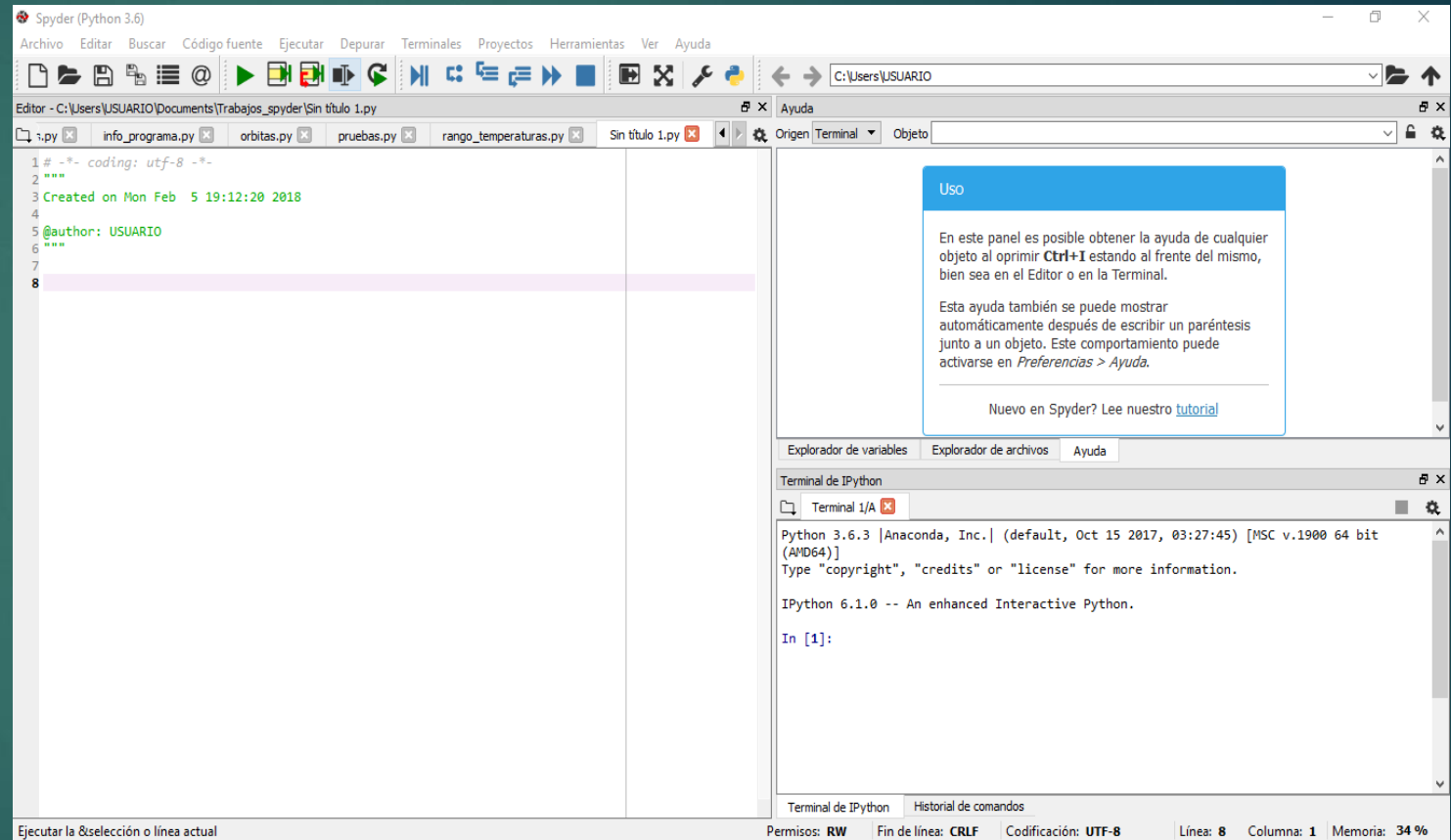
- Getting NumPy
- Installing the SciPy Stack
- NumPy and SciPy documentation page
- NumPy Tutorial
- NumPy for MATLAB® Users
- NumPy functions by category
- NumPy Mailing List

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see scipy.org.

¿Cómo se comparte lo que haces en python?



Spyder



Palabras claves

palabras



Variable

TENGWAR PER L'ITALIANO

CONSONANTI

I Tincotéma		II Parmatéma		III Calmatéma		IV Quessetéma	
P	t	P	p	q	c	q	
p	d	p	b	q	g	q	g
b		b	f	d	sc	d	qu
h		h	v	l		l	gl+i
m	n	m	m	z	z	z	gn
r	r	r		o	o	o	
l	r	l	rr	l	l	l	
s	s	s	s	ss	ss	ss	ss
h	h	h		h	e	h	u

VOCALI

Tehtar		Tengwar
i	a	c
i	e	h
i	i	i
i	o	o
i	u	o



Lista

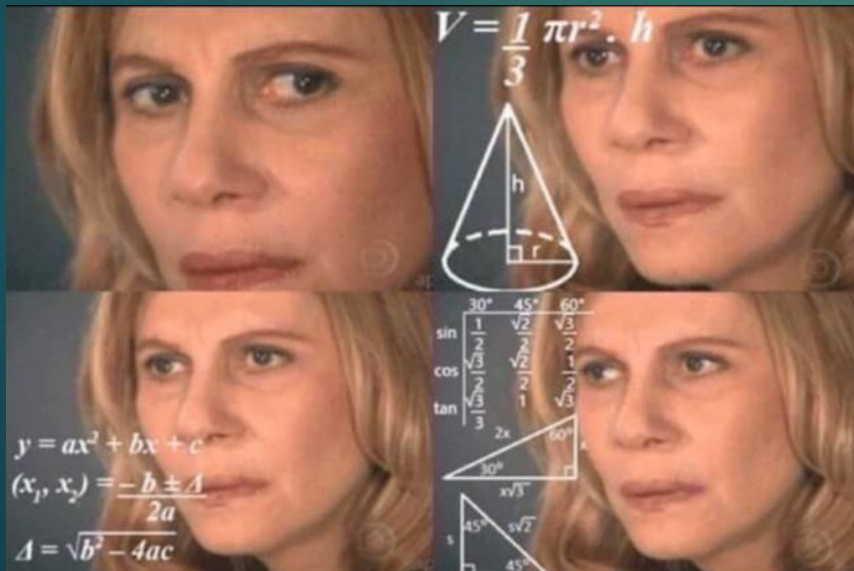
```
>>> jugadores[3]='Pity'
>>> jugadores1=jugadores[1:4]
>>> jugadores1
['Driussi', 'Alario', 'Pity']
>>> jugadores2=jugadores[3:7]
>>> jugadores2
['Pity', 'Auzqui', 'Rojas', 'Ponzio']
>>> jugadores3=jugadores[:4] # Esta nueva lista incluye los ítems
>>> jugadores3
['Batalla', 'Driussi', 'Alario', 'Pity']
>>> jugadores4=jugadores[2:] # Esta lista incluirá los ítems ubic
>>> jugadores4
['Alario', 'Pity', 'Auzqui', 'Rojas', 'Ponzio']
>>>
```

```
Me=[0.055,0.382,0.39] #
Ve=[0.8149,0.949,0.72] #
Ti=[1,1,1.00] # Informa
Ma=[0.169,0.532,1.52] #
Ju=[317.81,10.9,5.20] #
Sa=[95.16,9.14,9.54] #
Ur=[15.3,3.98,19.19] #
Ne=[17.146,3.864,30.06]
```



Posiciones en listas

```
Me=[0.055,0.382,0.39]
print(Me[0])
print(Me[-1])
print(Me[1])
```



```
In [3]: runfile('C:/Users/USUARIO/Documents/Trabajos_spyder/Sin título 0.py', wdir='C:/Users/USUARIO/Documents/Trabajos_spyder')
```

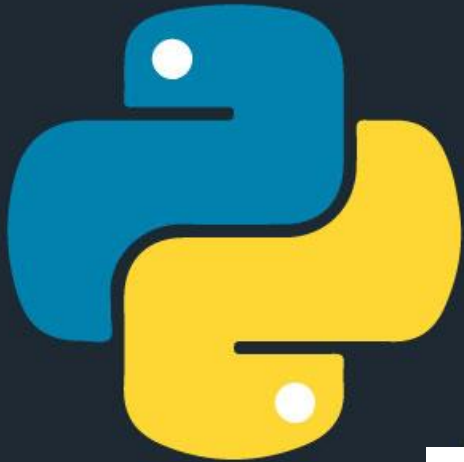
0.055

0.39

0.382



Función



```
def my_function(param1, param2, ...):  
    pass
```

```
def Rt_a_m(Rt):  
    """  
    se ingresa el valor en radios terrestres y se convierte en metros  
    """  
    m=6371000*Rt  
    return m
```

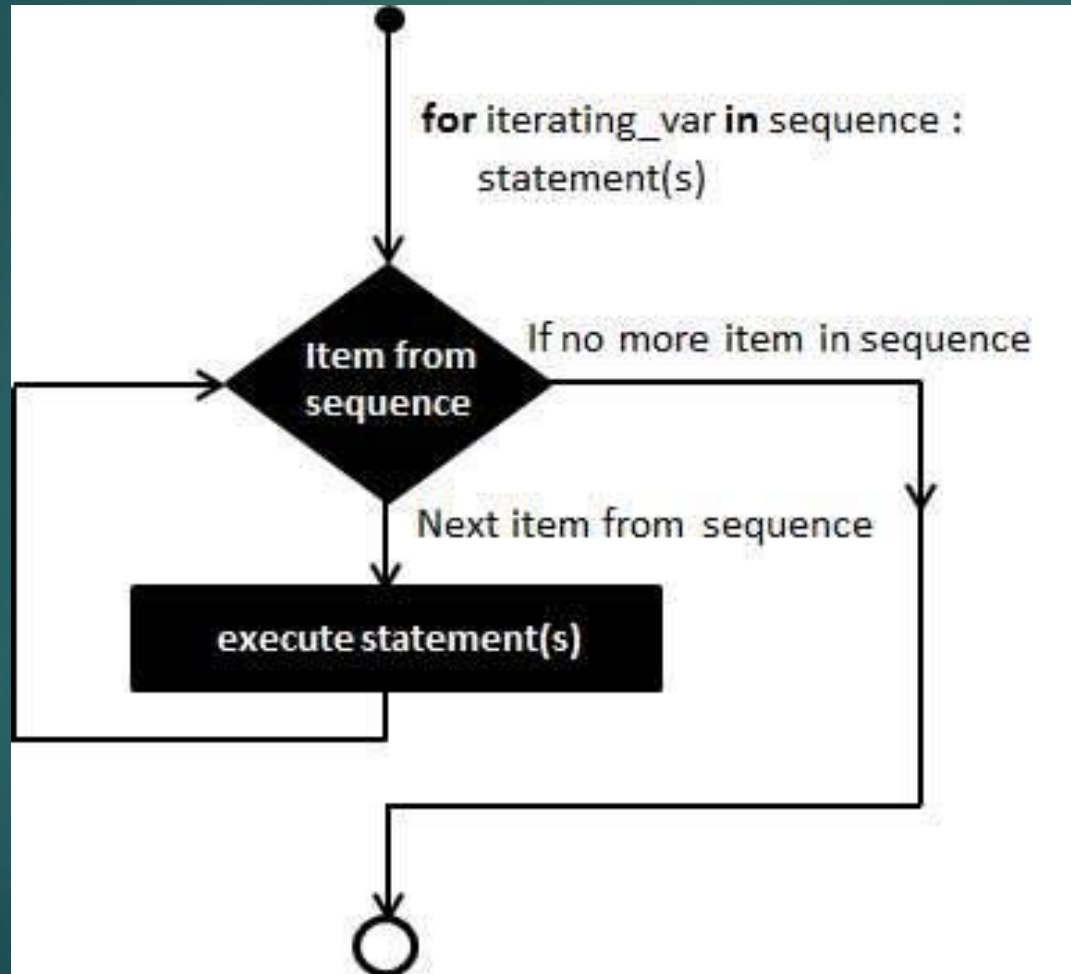
¿Cómo se llama a una función?

```
def excentricidad(c,a):  
    e=c/a  
    return e
```

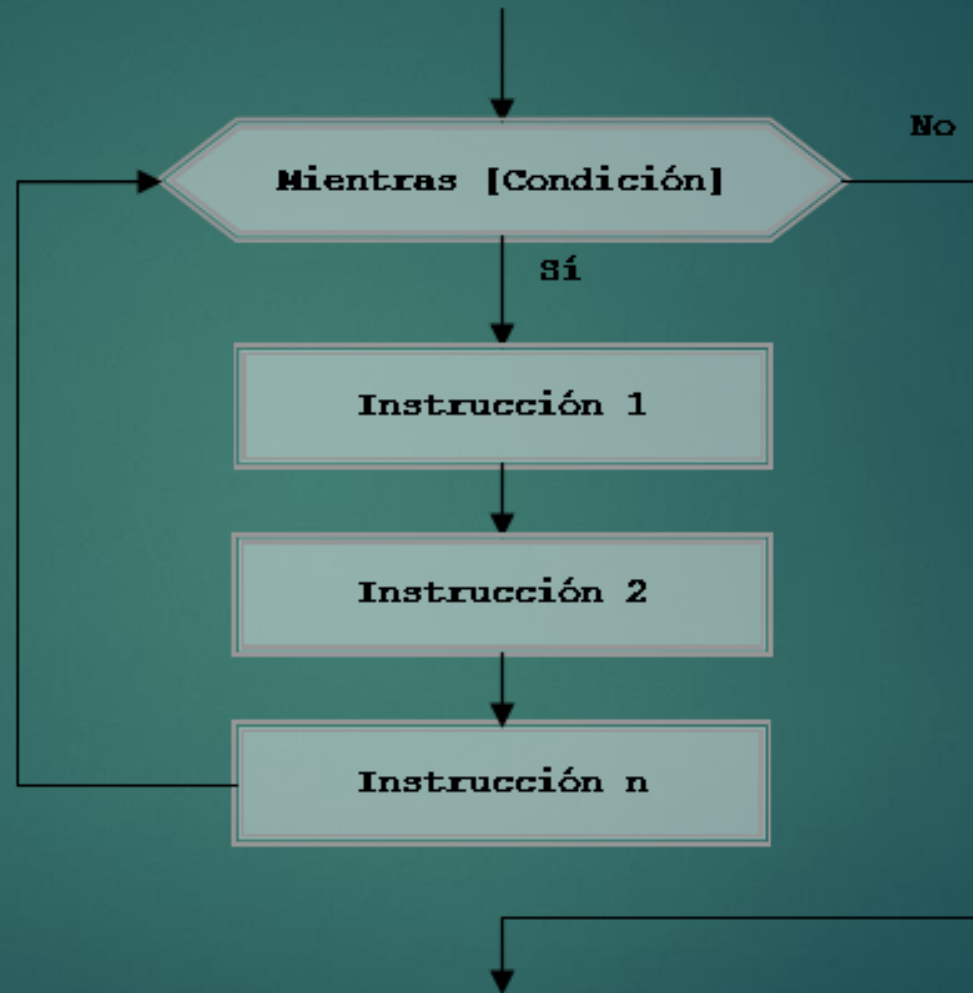
```
orbitas.excentricidad(c,a)
```



Ciclos “for”



Ciclos while



Astronomía *_*



Planos de coordenadas

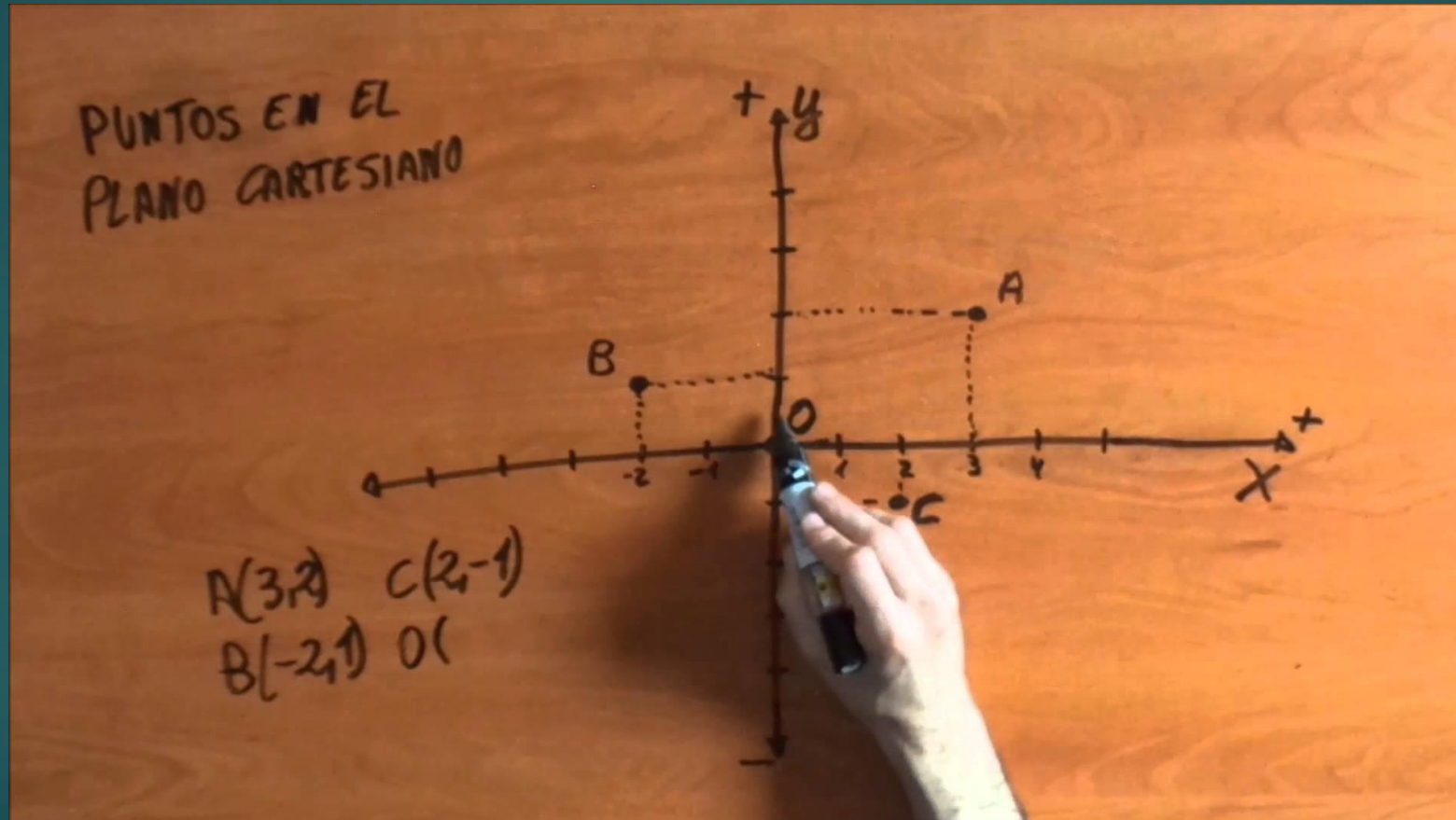
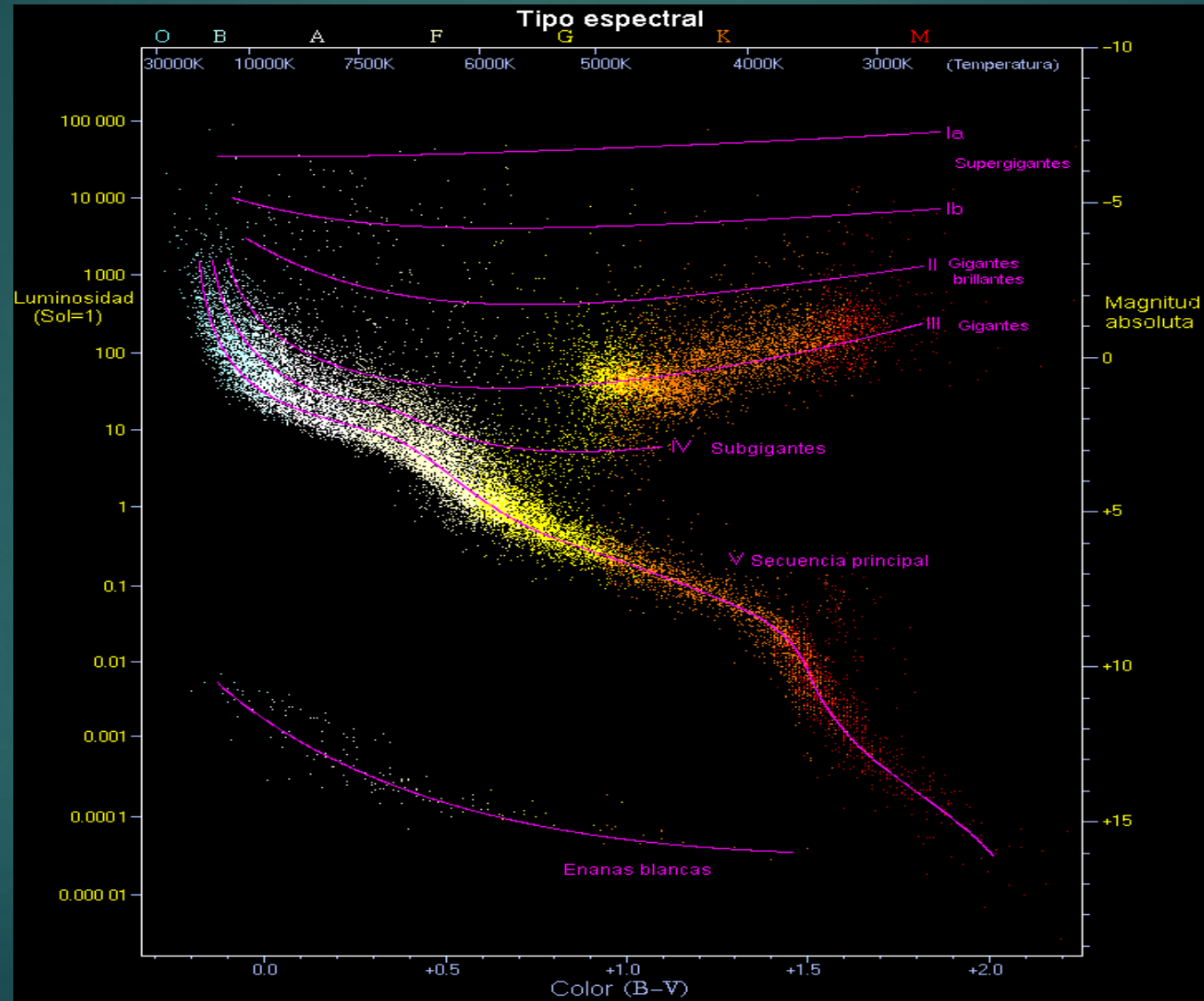


Diagrama HR



Unidades de medida



Distancia:

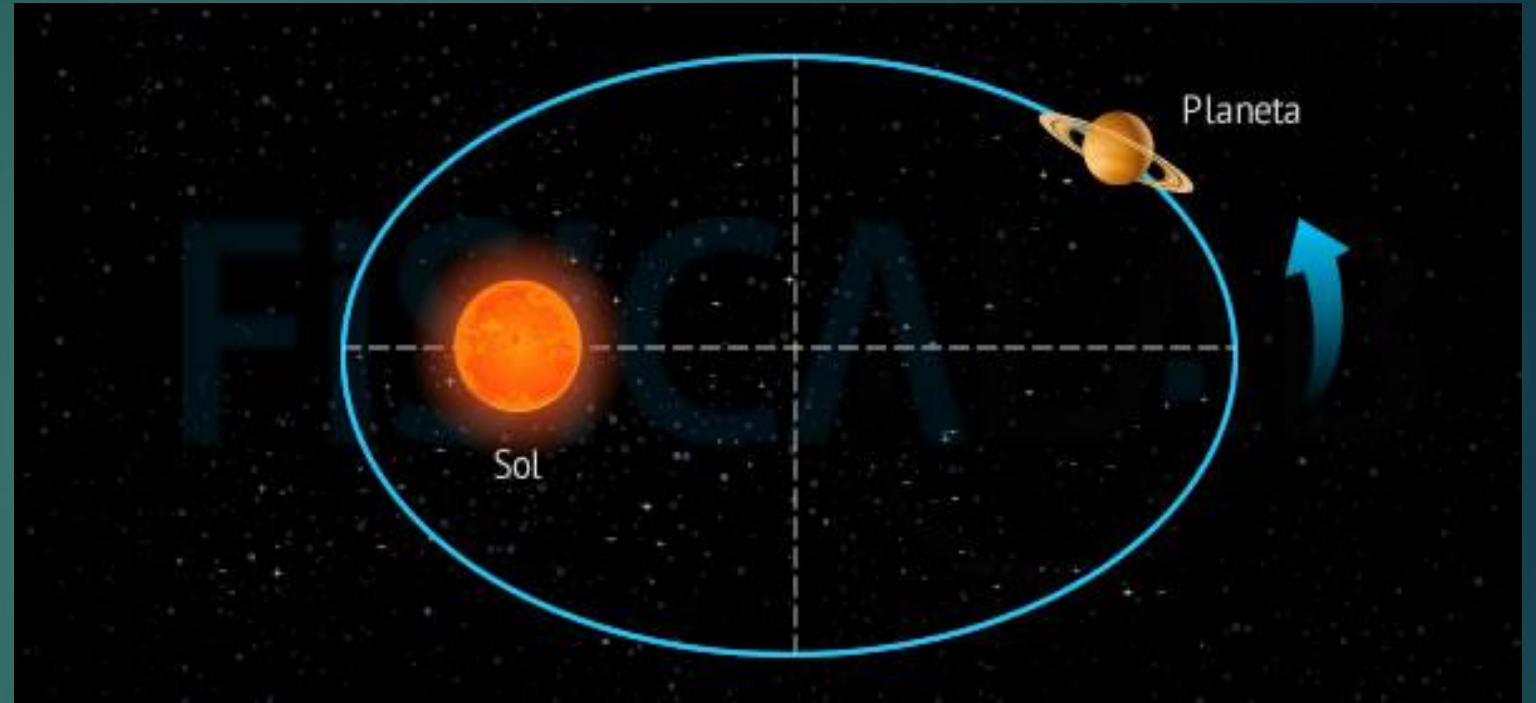
Unidad Astronómica
149597870700 m

Masa:

Temperatura



Órbitas



A full-page background image featuring Leonardo DiCaprio in a tuxedo, smiling and holding a glass of champagne. The background is a blurred party scene with confetti and lights. A solid red rectangle is located in the top right corner.

Unión Python y astronomía

Proceso para llegar al programa



Factores de conversión empleados

```
6 """
7 import numpy#se importa para las funciones de seno, coseno y tangente
8
9 def luminosidad_solar_a_vatios(Lsol):
10     """
11     se ingresa el valor en luminosidades solares, y se convierte en vatios
12     """
13     vatios=Lsol*3.827e26
14     return vatios
15
16 def Mt_a_Kg(Mt):
17     """
18     se ingresa el valor en masas terrestres y se convierte en kg
19     """
20     Kg=5.972e24*Mt
21     return Kg
22
23 def Rt_a_m(Rt):
24     """
25     se ingresa el valor en radios terrestres y se convierte en metros
26     """
27     m=6371000*Rt
28     return m
29
30 def UA_a_m(UA):
31     """
32     se ingresa el valor en unidades astronómicas y se convierte a metros
33     """
34     m=149597870700*UA
35     return m
36
```

```
def coordenadas_polares_a_cartesianas(r,A):
    """
    Se ingresan coordenadas polares, con los grados en radianes, y se convierte a coordenadas
    """
    x=r*numpy.cos(A)
    y=r*numpy.sin(A)
    return [x,y]

def segundos_a_años(s):
    d=s/31536000
    return d
```



El ingreso de las leyes

```
1 |# -*- coding: utf-8 -*-
2 |"""
3 |Editor de Spyder
4 |Acá se encuentra las leyes físicas con las que funcionará el programa (lista_planetas.py)
5 |"""
6 |from numpy import pi
7 |G=6.67408e-11
8 |sigma= 5.67e-8
9 |pi4=4*pi
10 |pi4e2=pi**2*4
11 |
12 |
13 |def ley_stefan_boltzmann(T,L):
14 |    """
15 |    Se usa la ley de boltzmann para calcular del diagrama HR el radio de la estrella, el radio
16 |    """
17 |    R=(L/(T**4*pi4*sigma))**0.5
18 |    return R
19 |
20 |def ecuacion_masa (T,L):
21 |    """
22 |    Se usa esta ecuación, para relacionar la masa, luminosidad en vatios y temperatura de una
23 |    """
24 |    M=L/(T/30029898)
25 |    return M
26 |
27 |def tercera_ley(M,a):
28 |    """
29 |    Con la tercera ley de Kepler se calculará el periodo de rotacion de los plaentas que el us
30 |    """
31 |    P=(a**3*pi4e2/(G*M))**0.5
32 |    return P
33 |
```

```
def tercera_ley_a(M,P):
    """
    En este caso, se usa la misma ley, pero despejando el valor del eje mayor y con las misma:
    """
    a=(P**2*G*M/pi4e2)**(1/3)
    return a
```



La creación de las orbitas

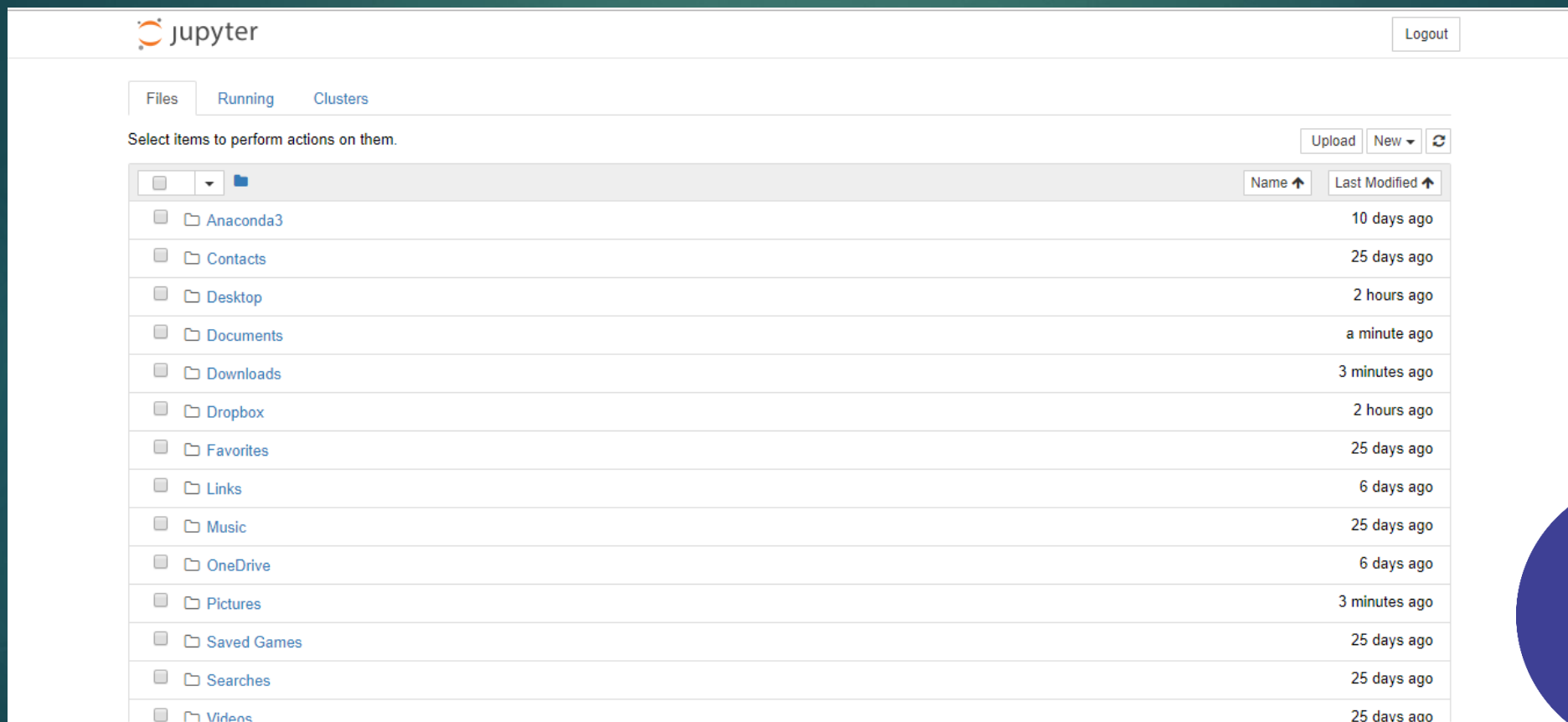
```
4
5 @author: USUARIO
6 """
7 import numpy #se usa numpy para las funciones de seno y coseno
8 import conversiones
9 import calculos_planetas
10 import info_programa
11 def excentricidad(c,a):
12     """
13     con la posición de los focos y el eje mayor, se haya la excentricidad de la elipse
14     """
15     e=c/a
16     return e
17
18 def radio_orbital_tiempo(a,e,w,t):
19     """
20     funcion que con la excentricidad y el angulo orbital, en radianes, regresa el radio
21     """
22     r=a*(1-e**2)/(1+e*numpy.cos(w*t))
23     return r
24
25 def angulo_orbital_tiempo(w,t):
26     """
27     con el angulo orbital o la frecuencia angular, se multiplica con el periodo para obtener
28     """
29     return w*t
30
31 def posicion_foco(a,e):
32     """
33     Función que relaciona la excentricidad con el eje mayor para sacar la posición de los focos
34     """
35     c=e*a
36     return c
```



El entorno gráfico



Usando Jupyter Notebook



The screenshot displays the Jupyter Notebook web interface. At the top left is the Jupyter logo, and at the top right is a 'Logout' button. Below the header, there are three tabs: 'Files' (selected), 'Running', and 'Clusters'. A message 'Select items to perform actions on them.' is shown above a table of files. The table has columns for 'Name' and 'Last Modified'. The files listed are folders: Anaconda3, Contacts, Desktop, Documents, Downloads, Dropbox, Favorites, Links, Music, OneDrive, Pictures, Saved Games, Searches, and Videos. Each folder has a checkbox to its left and a timestamp to its right. At the top right of the table, there are buttons for 'Upload', 'New', and a refresh icon.

	Name	Last Modified
<input type="checkbox"/>	Anaconda3	10 days ago
<input type="checkbox"/>	Contacts	25 days ago
<input type="checkbox"/>	Desktop	2 hours ago
<input type="checkbox"/>	Documents	a minute ago
<input type="checkbox"/>	Downloads	3 minutes ago
<input type="checkbox"/>	Dropbox	2 hours ago
<input type="checkbox"/>	Favorites	25 days ago
<input type="checkbox"/>	Links	6 days ago
<input type="checkbox"/>	Music	25 days ago
<input type="checkbox"/>	OneDrive	6 days ago
<input type="checkbox"/>	Pictures	3 minutes ago
<input type="checkbox"/>	Saved Games	25 days ago
<input type="checkbox"/>	Searches	25 days ago
<input type="checkbox"/>	Videos	25 days ago



Todas las partes de la interfaz

```
In [1]: # -- Aquí Los módulos requeridos -- #
# Requerido para crear la interfaz
from ipywidgets import VBox, HBox, Button, Label, BoundedFloatText, BoundedIntText
from IPython.display import clear_output, display
# Requerido para los gráficos
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
# from matplotlib import animation
import time
# -- Módulos propios -- #
from rango_temperaturas import color_estrellas
```

```
# Función para crear la animación
def graficar_sistema(datos):
    fig = plt.figure()
    instantaneas = len(datos[0])
    plt.show(block=False)
    max_x = max(max(abs(planeta[:, 0]))) for planeta in datos[1:]
    max_y = max(max(abs(planeta[:, 1]))) for planeta in datos[1:]

    for inst in range(instantaneas):
        ax = plt.axes(xlim=(-max_x, max_x), ylim=(-max_y, max_y))
        plt.axis('equal')
        ax.plot([0], [0], 'oy')
        for planeta in datos[1:]:
            ax.plot([planeta[inst, 0]], [planeta[inst, 1]], 'ob')
        fig.canvas.draw()
        time.sleep(0.01)
```




```
# Variables para guardar info del sistema
estrella = [] # temperatura, luminosidad
planetas = [] # lista con excentricidad y semieje mayor por planeta
# Estilo
estilo_texto = {'description_width': 'initial'}
# Solicitud de información de la estrella
titulo_estrella = Label(value="Propiedades estrella")
temp = BoundedIntText(description='Temperatura (K)', value=5778, min=3000, max=35000, style=estilo_texto)
lum = BoundedFloatText(description='Luminosidad (Sol=1)', value=1, min=1e-5, max=1e5, style=estilo_texto)
gui_estrella = VBox(children=[titulo_estrella, temp, lum])
# Solicitud de información de los planetas
titulo_planetas = Label(value="Propiedades planetas")
excentricidad = BoundedFloatText(description="Excentricidad", value=0.2, min=0.0, max=1.0)
semi_mayor = BoundedFloatText(description="Semieje mayor (UA)", value=1.0, min=0.01, style=estilo_texto)
gui_planetas = VBox(children=[titulo_planetas, excentricidad, semi_mayor])
# Unión de solicitud de info
gui_info = HBox(children=[gui_estrella, gui_planetas])
# Botones
bt_reiniciar = Button(description="Borrar planetas")
bt_agregar = Button(description="Agregar otro planeta")
bt_guardar = Button(description="Guardar sistema")
bt_simular = Button(description="Simular")
gui_botones = HBox(children=[bt_reiniciar, bt_agregar, bt_guardar, bt_simular])
# Unión GUI
gui_sistema = VBox(children=[gui_info, gui_botones])
# Acciones de los botones
```



```
def clic_reiniciar(boton):
    #estrella[:] = []
    planetas[:] = []
    clear_output()
    display(gui_sistema)

def clic_agregar(boton):
    planetas.append([excentricidad.value, semi_mayor.value])

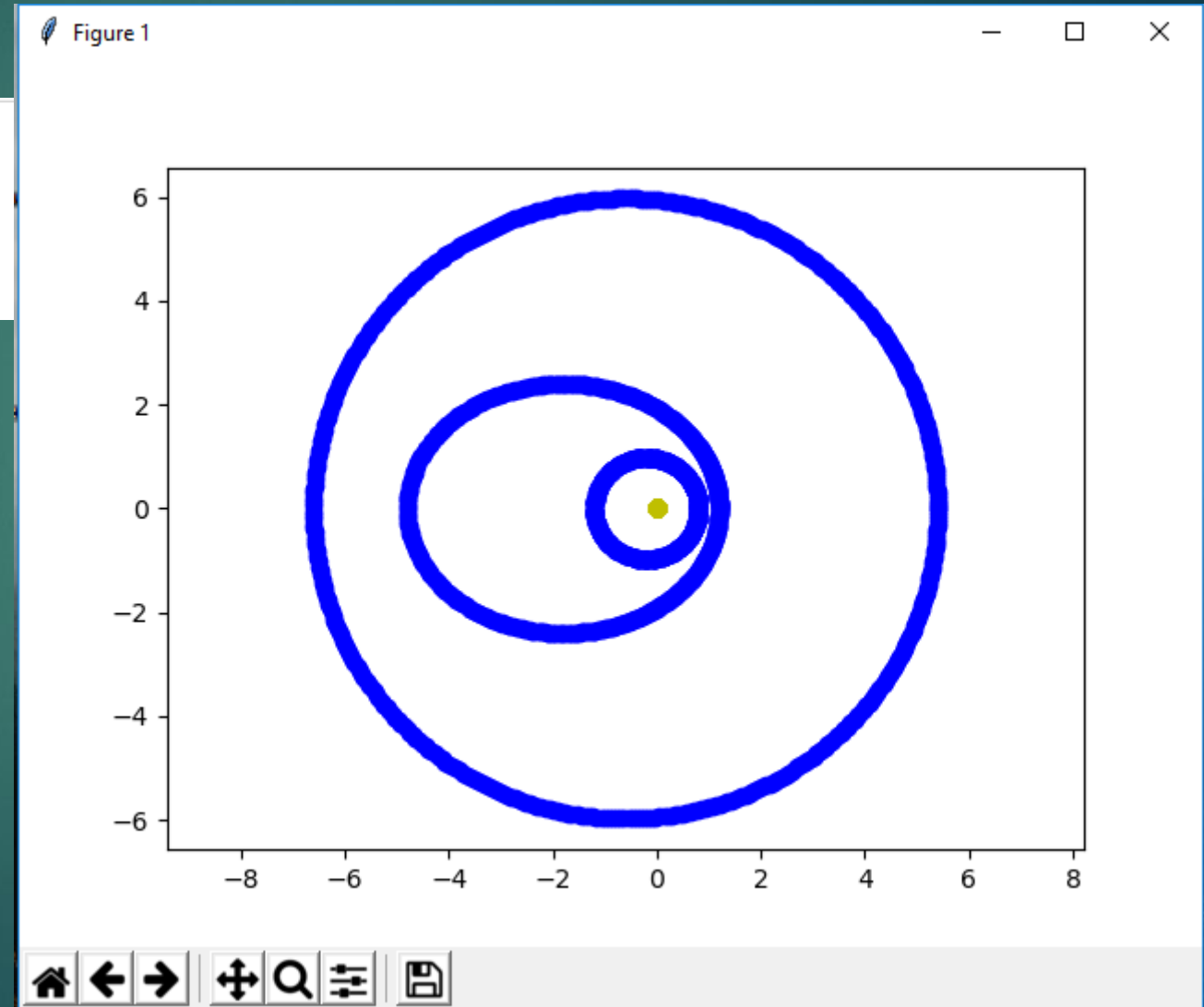
def clic_guardar(boton):
    planetas.append([excentricidad.value, semi_mayor.value])
    estrella.extend([temp.value, lum.value])
    print("Ha elegido una estrella {}".format(color_estrellas(estrella[0])))
    print("Número de planetas: {}".format(len(planetas)))
    info = open('info_programa.py', 'w')
    info_planetas = "planetas = {}".format(planetas)
    info_estrella = "estrella = {}".format(estrella)
    info.write(info_planetas)
    info.write(info_estrella)
    info.close()

def clic_simular(boton):
    if len(estrella) == 0:
        clic_guardar(boton)
    from orbitas import función_orbitas_entera
    sistema_datos = función_orbitas_entera()
    graficar_sistema(sistema_datos)
```



La recopilación de todo

Propiedades estrella		Propiedades planetas	
Temperatura (K)	<input type="text" value="5778"/>	Excentricidad	<input type="text" value="0,2"/>
Luminosidad (Sol=1)	<input type="text" value="1"/>	Semieje mayor (UA)	<input type="text" value="1"/>
<input type="button" value="Borrar planetas"/>		<input type="button" value="Agregar otro planeta"/>	<input type="button" value="Guardar sistema"/>
<input type="button" value="Simular"/>			



Bibliografía

- ▶ <https://github.com/saint-germain/Python3Espanol>
- ▶ <http://www.universoformulas.com/matematicas/geometria/elipse/>
- ▶ <https://pwg.gsfc.nasa.gov/stargaze/Mmotion.htm>
- ▶ <http://www2.elo.utfsm.cl/~elo352/2010/Exp4/DocumentacionRS2008/Ecuaciones%20de%20orbita.pdf>
- ▶ <https://www.pycon.co/speakers/santiago-mesa-velasquez/>
- ▶ <https://www.python.org/about/>
- ▶ <https://anaconda.org/about>
- ▶ https://live.osgeo.org/es/quickstart/jupyter_quickstart.html

Github

► <https://github.com/samez21/leyes-kepler>



¡Muchas gracias!

