

Funciones 3D, tiro parabólico y órbitas en Python



Roberto López
Eric Martín

ÍNDICE

CÓDIGO	2
EXPLICACION	6
IMÁGENES	8

CÓDIGO

```
import math as m
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

print("Selecciona una de las acciones introduciendo el número correspondiente:")
print("1- Dibuja la función  $6x^2y^5$ ")
print("2- Dibuja la función  $3x + y$ ")
print("3- Dibuja la función  $x^2y - 2xy^2 + 3xy + 4$ ")
print("4- Dibuja la función  $5x^9 - 3y^8$ ")
print("5- Dibuja la función  $(1/2)x^2$ ")
print("6- Calcula el tiro parabólico con tus propios parámetros")
print("7- Calcula la velocidad de un cometa en el afelio")
print("8- Calcula la velocidad de un cometa en el perihelio")
print("9- Calcula el radio de un cometa en el afelio")
print("10- Calcula el radio de un cometa en el perihelio")
nfuncion=input()
nfuncion=int(nfuncion)

if nfuncion==1:
    def fun(x, y):
        return (6*x**2*y**5)

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    x = y = np.arange(-3.0, 3.0, 0.05)
    X, Y = np.meshgrid(x, y)
    zs = np.array([fun(x,y) for x,y in zip(np.ravel(X), np.ravel(Y))])
    Z = zs.reshape(X.shape)

    ax.plot_surface(X, Y, Z)

    ax.set_xlabel('X Label')
    ax.set_ylabel('Y Label')
    ax.set_zlabel('Z Label')

    plt.show()

if nfuncion==2:
    def fun2(x, y):
        return (3*x+y)

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    x = y = np.arange(-3.0, 3.0, 0.05)
    X, Y = np.meshgrid(x, y)
    zs = np.array([fun2(x,y) for x,y in zip(np.ravel(X), np.ravel(Y))])
    Z = zs.reshape(X.shape)
```

```

ax.plot_surface(X, Y, Z)

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

plt.show()

if nfuncion==3:
    def fun3(x, y):
        return (x**2*y-2*x*y**2+3*x*y+4)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = y = np.arange(-3.0, 3.0, 0.05)
X, Y = np.meshgrid(x, y)
zs = np.array([fun3(x,y) for x,y in zip(np.ravel(X), np.ravel(Y))])
Z = zs.reshape(X.shape)

ax.plot_surface(X, Y, Z)

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

plt.show()

if nfuncion==4:
    def fun4(x, y):
        return (5*x**9-3*y**8)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = y = np.arange(-3.0, 3.0, 0.05)
X, Y = np.meshgrid(x, y)
zs = np.array([fun4(x,y) for x,y in zip(np.ravel(X), np.ravel(Y))])
Z = zs.reshape(X.shape)

ax.plot_surface(X, Y, Z)

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

plt.show()

if nfuncion==5:
    def fun5(x, y):
        return ((1/2)*x**2)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

```



```

x = y = np.arange(-3.0, 3.0, 0.05)
X, Y = np.meshgrid(x, y)
zs = np.array([fun5(x,y) for x,y in zip(np.ravel(X), np.ravel(Y))])
Z = zs.reshape(X.shape)

```

```

ax.plot_surface(X, Y, Z)

```

```

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

```

```

plt.show()

```

```

if nfuncion==6:

```

```

    g=9.81
    j=int(input('Dame el valor de la altura'+ " "))
    lista=np.array([10, 20, 30, 40, 50, 60, 70, 80, 90,200])
    r=np.deg2rad(lista)
    Vi=int(input('Dame el valor de la velocidad inicial'+""))
    t=int(input('Dame el valor del tiempo'+""))
    x=np.arange(0,t,0.1)
    Y_0=j
    X_0=0
    Vix1=Vi * np.cos(r[0])
    Viy1=Vi * np.sin(r[1])
    a1= Y_0 + Viy1 * ((x - X_0) / Vix1) - 0.5 *g *((x - X_0) /
Vix1)**2

```

```

    #a1=(Viy1 * (x/Vix1)) + ((( g/2) * x/Vix1)**2)
    #tv=(Viy1 + m.sqrt((Viy1 **2)+( 2 * g * Viy1)/2))*x
    #ts= tv/2
    #xmax=Vix1 * tv
    #ymax= Viy1 + ts - (g*(ts**2)/2)

```

```

    vox=((1/(Viy1 * x) )- (1/a1))* ((g**2)*(x**2))
    #print tv
    #print ts

```

```

    plt.plot(a1)

```

```

    plt.show()

```

```

if nfuncion==7:

```

```

    print("Calcula la velocidad a la que se mueve un cometa en el
afelio.")

```

```

    Vperihelio=int(input('Dime el valor de la velocidad en el
perihelio '))
    Rafelio=int(input("Dime el valor del radio en el afelio "))
    Rperihelio=int(input("Dime el valor del radio en el perihelio "))

```

```

    Vafelio= (Vperihelio*Rperihelio)/Rafelio

```

```

Vafelio=str(Vafelio)
print("La velocidad del cometa en el afelio es de " + Vafelio + "
m/s")

```

```

if nfuncion==8:
    print("Calcula la velocidad a la que se mueve un cometa en el
perihelio.")

    Vafelio=int(input('Dime el valor de la velocidad en el afelio '))
    Rafelio=int(input("Dime el valor del radio en el afelio "))
    Rperihelio=int(input("Dime el valor del radio en el perihelio "))

    Vperihelio= (Vperihelio*Rafelio)/Rperihelio
    Vperihelio=str(perihelio)
    print("La velocidad del cometa en el afelio es de " + Vperihelio +
" m/s")

```

```

if nfuncion==9:
    print("Calcula el radio de la órbita de un cometa en el afelio.")

    Vperihelio=int(input('Dime el valor de la velocidad en el
perihelio '))
    Rperihelio=int(input("Dime el valor del radio en el perihelio "))
    Vafelio=int(input('Dime el valor de la velocidad en el afelio '))

    Rafelio= (Vperihelio*Rperihelio)/Vafelio
    Rafelio=str(Rafelio)
    print("El radio de la órbita del cometa cuando se encuentra en el
afelio es de " + Rafelio + " m")

```

```

if nfuncion==10:
    print("Calcula el radio de la orbita de un cometa en el
perihelio.")

    Vperihelio=int(input('Dime el valor de la velocidad en el
perihelio '))
    Rafelio=int(input("Dime el valor del radio en el afelio "))
    Vafelio=int(input("Dime el valor del radio en el perihelio "))

    Rperihelio= (Vafelio*Rafelio)/Vperihelio
    Rperihelio=str(Rperihelio)
    print("El radio de la órbita del cometa cuando se encuentra en el
perihelio es de " + Rperihelio + "

```

EXPLICACION

Lo primero que hacemos es importar todas las librerías que necesitaremos en los modos a los que corresponden a las primeras 4 líneas. Importamos math, numpy, matplotlib.pyplot y Axes 3D. Posteriormente damos las indicaciones de lo que vamos a realizar con el código, de tal forma que explicamos que con cada número del teclado se realizara una función y operación diferente.

Posteriormente definimos una variable nfunción como el número del teclado al que corresponderán las indicaciones anteriormente señaladas.

Una vez acabada la introducción comenzamos con la parte principal que se trata de la realización de un total de 10 funciones diferentes.

Con un condicional en el que observamos cual es la variable nfunción, es decir, vemos cual es la opción deseada por el usuario.

Lo primero que nos sale es un mensaje que nos indica el número que debemos presionar para cada función del programa, las 5 primeras (correspondientes al 1,2,3,4 y 5) representan las funciones $6x^2y^5$, $3x + y$, $x^2y - 2xy^2 + 3xy + 4$, $5x^9 - 3y^8$, $(1/2)x^2$ correspondientemente. Las cinco siguientes corresponden a un tiro parabólico, y el cálculo del radio y la velocidad en el afelio y perihelio de un cometa con respecto al cuerpo que los atrae.

- **Las funciones representadas son:**
 1. Dibuja la función $6x^2y^5$
 2. Dibuja la función $3x + y$
 3. Dibuja la función $x^2y - 2xy^2 + 3xy + 4$
 4. Dibuja la función $5x^9 - 3y^8$
 5. Dibuja la función $(1/2)x^2$
 6. Calcula el tiro parabólico con tus propios parámetros
 7. Calcula la velocidad de un cometa en el afelio
 8. Calcula la velocidad de un cometa en el perihelio
 9. Calcula el radio de un cometa en el afelio
 10. Calcula el radio de un cometa en el perihelio

Una vez definido esto, para los números 1,2,3,4,5 definimos una función con los parámetros de x e y. luego introducimos la función determinada y representamos la figura de dicha función y los niveles de la x, z e por lo tanto tenemos diversas funciones en tres dimensiones.

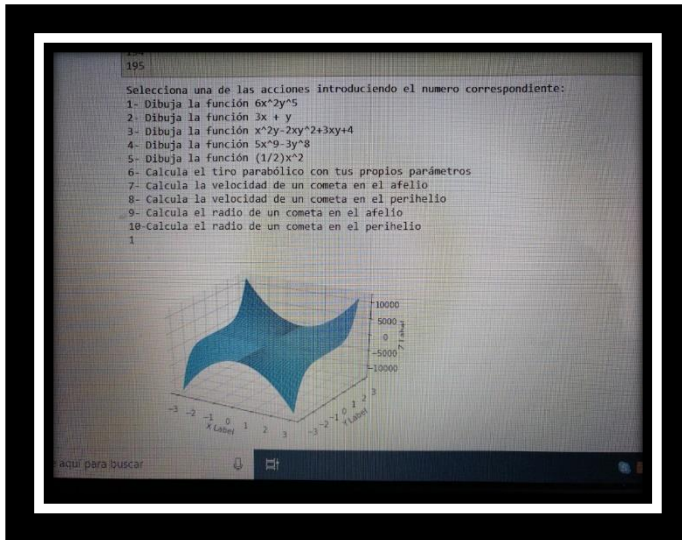
El funcionamiento de las cinco primeras funciones se basa en que el programa al pulsar el número asignado a esa función nos devuelva las funciones con x e y, representado en tres dimensiones.

En cuanto al tiro parabólico, si pulsamos el numero 6 saldrá un mensaje que nos pedirá introducir la velocidad inicial, el tiempo y la altura inicial. Una vez hecho esto calculará la trayectoria con las fórmulas introducidas y las representará de manera que veamos cómo se mueve el objeto a lo largo de su recorrido.

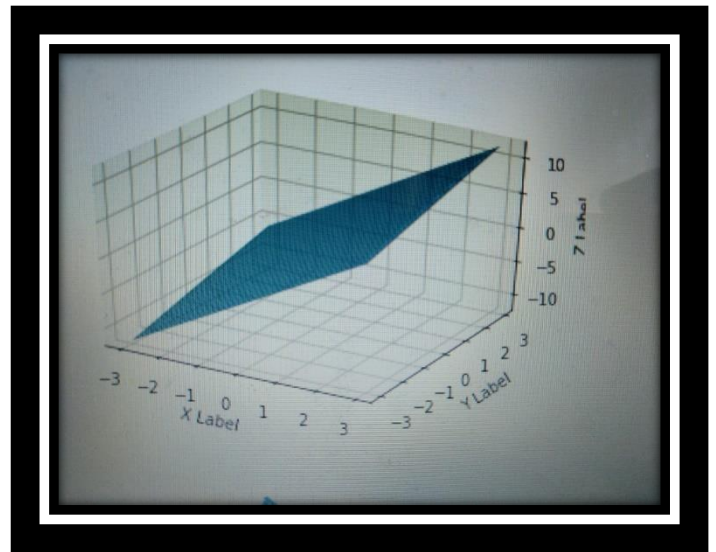
Por último encontramos 4 funciones para las órbitas, cada una de ellas calcula la velocidad y el radio en dos puntos diferentes, de manera que están asignados el 7,8,9 y 10 para ello. La función trabaja con una fórmula que hemos calculado para cada uno de los apartados despejando la variable que queremos obtener en función de las otras tres.

IMÁGENES

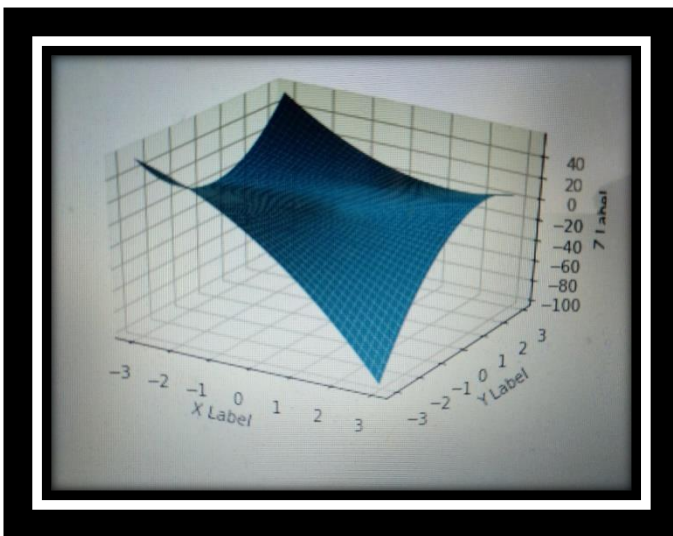
1.



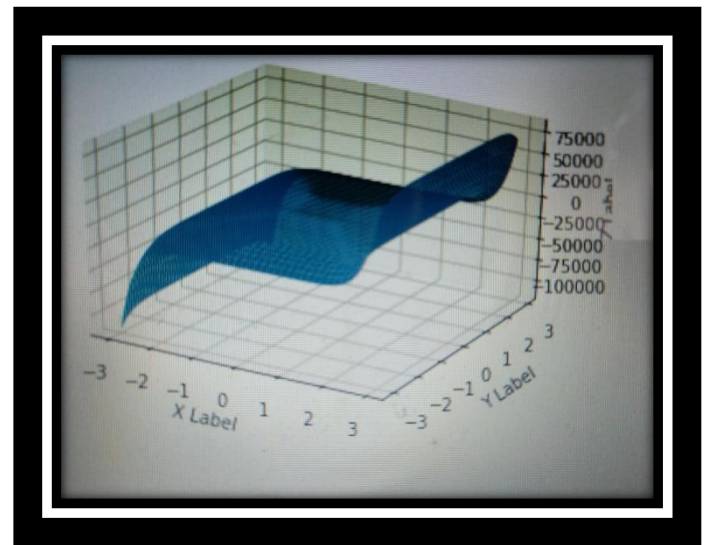
2.



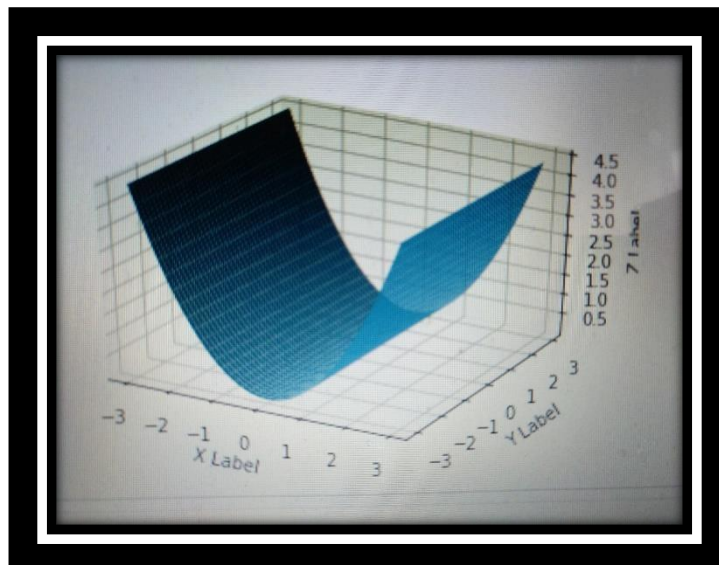
3.



4.



5.



6.

```

12 X_0=0
13 Vix1=Vi * np.cos(r[0])
14 Viy1=Vi * np.sin(r[1])
15 a1= Y_0 + Viy1 * ((x - X_0) / Vix1) - 0.5 * g * ((x - X_0) / Vix1)**2
16
17 #a1=(Viy1 * (x/Vix1)) + ((( g/2) * x/Vix1)**2)
18 #tv=(Viy1 + m.sqrt((Viy1 **2)+( 2 * g * Viy1)/2))*x
19 #ts= tv/2
20 #xmax=Vix1 * tv
21 #ymax= Viy1 + ts - (g*(ts**2)/2)
22
23 vox=((1/(Viy1 * x) )- (1/a1))* ((g**2)*(x**2))
24 #print tv
25 #print ts
26
27 plt.plot(a1)
28
29 plt.show()

```

Dame el valor de la altura

[]: 1

[]: 1

```

Viy1=Vi * np.sin(r[1])
a1= Y_0 + Viy1 * ((x - X_0) / Vix1) - 0.5 * g * ((x - X_0) / Vix1)**2

#a1=(Viy1 * (x/Vix1)) + ((( g/2) * x/Vix1)**2)
#tv=(Viy1 + m.sqrt((Viy1 **2)+( 2 * g * Viy1)/2))*x
#ts= tv/2
#xmax=Vix1 * tv
#ymax= Viy1 + ts - (g*(ts**2)/2)

vox=((1/(Viy1 * x) )- (1/a1))* ((g**2)*(x**2))
#print tv
#print ts

plt.plot(a1)

plt.show()

```

Dame el valor de la altura

Dame el valor de la velocidad inicial

```

15 a1= Y_0 + Viy1 * ((x - X_0) / Vix1) - 0.5 * g * ((x - X_0) / Vix1)**2
16
17 #a1=(Viy1 * (x/Vix1)) + ((( g/2) * x/Vix1)**2)
18 #tv=(Viy1 + m.sqrt((Viy1 **2)+( 2 * g * Viy1/2)))*x
19 #ts= tv/2
20 #xmax=Vix1 * tv
21 #ymax= Viy1 + ts - (g*(ts**2)/2)
22
23 vox=((1/(Viy1 * x) )- (1/a1))* ((g**2)*(x**2))
24 #print tv
25 #print ts
26
27 plt.plot(a1)
28
29 plt.show()

```

Dame el valor de la altura 2

Dame el valor de la velocidad inicial 9

Dame el valor del tiempo

]: 1

]: 1

```

27 plt.plot(a1)
28
29 plt.show()

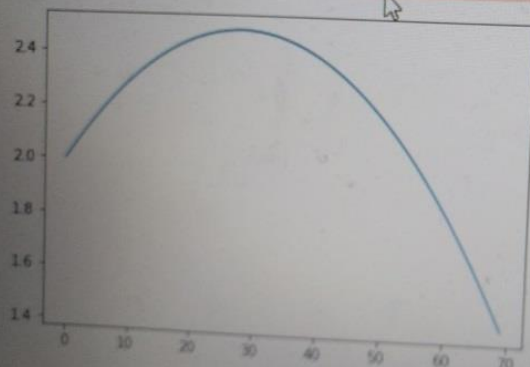
```

Dame el valor de la altura 2

Dame el valor de la velocidad inicial 9

Dame el valor del tiempo 7

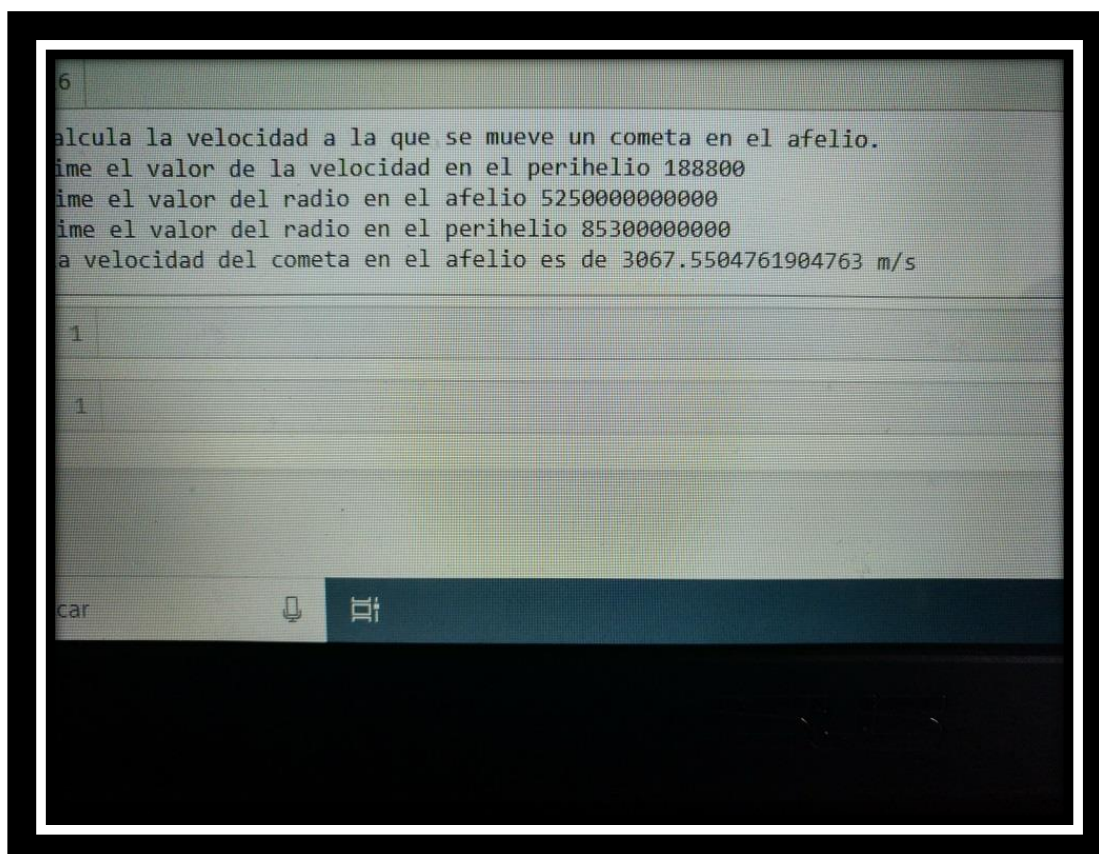
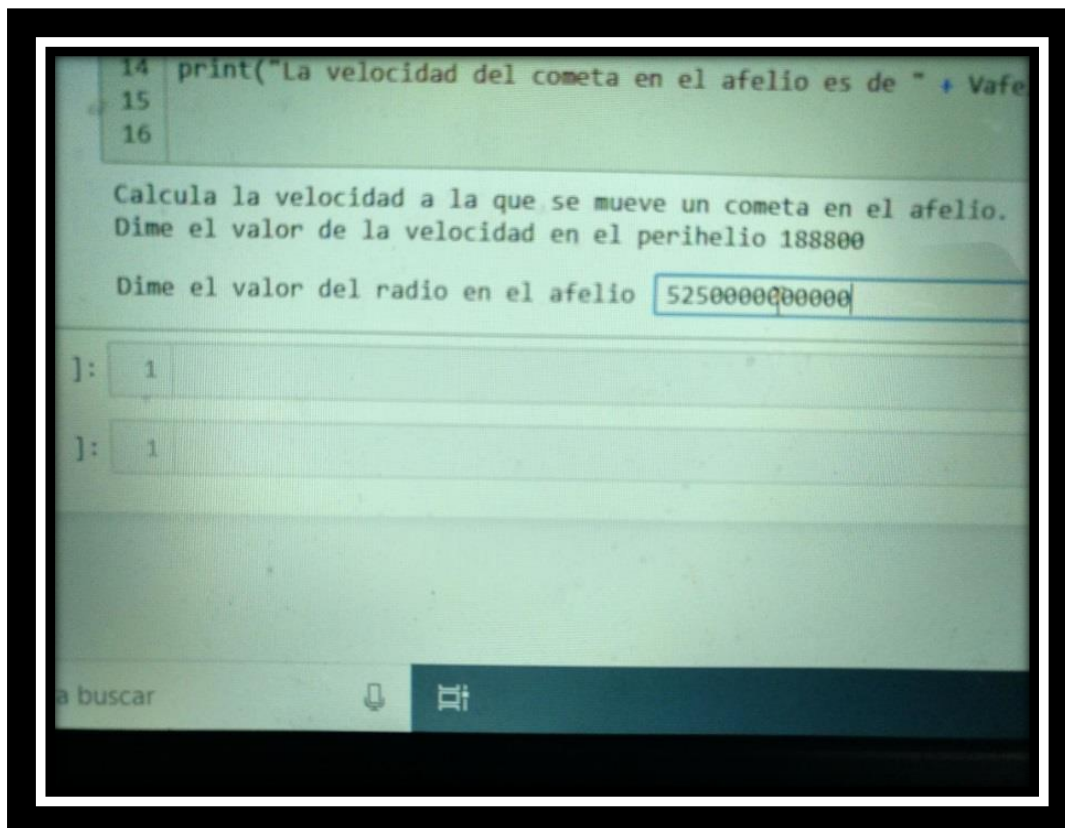
C:\Users\Fran\Anaconda3\lib\site-packages\ipykernel_launcher.py:23: RuntimeWarning:
C:\Users\Fran\Anaconda3\lib\site-packages\ipykernel_launcher.py:23: RuntimeWarning:



]: 1

]: 1

7.



9.

```

8 Rperihelio=int(input('Dime el valor del radio en el perihelio '))
9 Vafelio=int(input('Dime el valor de la velocidad en el afelio '))
10
11 Rafelio= (Vperihelio*Rperihelio)/Vafelio
12 Rafelio=str(Rafelio)
13 print("El radio de la órbita del cometa cuando se encuentra en el afelio es de " + Rafelio)
14
15
16
17

```

Calcula el radio de la órbita de un cometa en el afelio.
Dime el valor de la velocidad en el perihelio 70163398
Dime el valor del radio en el perihelio 85300000000
Dime el valor de la velocidad en el afelio 3067
El radio de la órbita del cometa cuando se encuentra en el afelio es de 1951398059797848.0 m

1

1

uscar

10.

```

12 Rperihelio=str(Rperihelio)
13 print("El radio de la órbita del cometa cuando se encuentra en el
14
15
16
17

```

Calcula el radio de la orbita de un cometa en el perihelio.
Dime el valor de la velocidad en el perihelio 188800
Dime el valor del radio en el afelio

]: 1

]: 1

a buscar


```

9 vafelio=int(input( Dime el valor de la velocidad en el afelio ))
10
11 Rperihelio= (Vafelio*Rafelio)/Vperihelio
12 Rperihelio=str(Rperihelio)
13 print("El radio de la órbita del cometa cuando se encuentra en el perihelio es de " + Rperihelio)
14
15
16
17

```

Calcula el radio de la órbita de un cometa en el perihelio.
 Dime el valor de la velocidad en el perihelio 188800
 Dime el valor del radio en el afelio 525000000000
 Dime el valor de la velocidad en el afelio 3067
 El radio de la órbita del cometa cuando se encuentra en el perihelio es de 85284692796.61017 m

1
 1

Buscar

CONCLUSIÓN

El trabajo que hemos realizado no ha supuesto grandes problemas más allá del tiempo y las ideas, ya que no teníamos muy claro que hacer y cuando nos pusimos a trabajar decidimos hacer un tiro parabólico, cuando lo habíamos hecho, las órbitas del cometa y posteriormente cuando esas dos cosas estaban hechas, la representación de funciones usando las gráficas en 3D, lo que supuso un trabajo duro de investigación que requería mucho tiempo para recabar toda la información necesaria y que como considerábamos necesario hacer este proyecto con todas las partes, hizo que el tiempo estuviera muy justo.

Por otro lado, realizar los códigos fue difícil en cuanto a que no sabíamos con qué librerías tratar ni cómo hacerlo, pero una vez hecho eso nos resultó bastante sencillo.