```
#Proyecto Clima comportamiento
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def rungekutta4(f,g,j,t0,x0,y0,z0,h,muestras):
    # Runge Kutta de 4do orden

    estimado = np.zeros(shape=(muestras,4),dtype=float)

    # incluye el punto [x0,y0]
    estimado[0] = [t0,x0,y0,z0]
    ti = t0
    xi = x0
    yi = y0
    zi = z0
    i= 1
    while not(i>=muestras):
        K1x = h * f(t0,xi,yi,z0)
        K1y = h * g(t0,xi,yi,z0)
        K1z = h * j(t0,xi,yi,z0)

        K2x = h * f(ti+h/2, xi + K1x/2,yi + K1y/2,zi +
K1z/2)
        K2y = h * g(ti+h/2, xi + K1x/2,yi + K1y/2,zi +
K1z/2)
        K2z = h * j(ti+h/2, xi + K1x/2,yi + K1y/2,zi +
K1z/2)

        K3x = h * f(ti+h/2, xi + K2x/2,yi + K2y/2,zi +
K2z/2)
        K3y = h * g(ti+h/2, xi + K2x/2,yi + K2y/2,zi +
K2z/2)
        K3z = h * j(ti+h/2, xi + K2x/2,yi + K2y/2,zi +
```

```
K2z/2)

        K4x = h * f(ti+h, xi + K3x, yi + K3y, zi + K3z)
        K4y = h * g(ti+h, xi + K3x, yi + K3y, zi + K3z)
        K4z = h * j(ti+h, xi + K3x, yi + K3y, zi + K3z)


        xi=  xi + (1/6)*(K1x+2*K2x+2*K3x +K4x)
        yi = yi + (1/6)*(K1y+2*K2y+2*K3y +K4y)
        zi = zi + (1/6)*(K1z+2*K2z+2*K3z +K4z)

        ti = ti + h


        estimado[i] = [ti,xi,yi,zi]
        i=i+1

    return(estimado)

alfa=10
beta=-8/3
ro=28
f= lambda t,x,y,z: alfa*(y-x)
g= lambda t,x,y,z: ro*x-y-x*z
j= lambda t,x,y,z: beta*z +x*y
#condiciones iniciales

muestras= 10049
t0= 0
x0= 10
y0= 7
z0= 7
h= 0.025
```

```
tabla= rungekutta4(f,g,j,t0,x0,y0,z0,h,muestras)

print(tabla)

# Plot
xs=tabla[:,1]
ys=tabla[:,2]
zs=tabla[:,3]

fig = plt.figure()
ax = fig.gca(projection='3d')

ax.plot(xs, ys, zs)
ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("Lorenz Attractor-Efecto Mariposa")

plt.show()
```