

```
1  #!/usr/bin/python
2  #!*_ coding:utf-8 -*-
3  # Este script es software libre. Puede redistribuirlo y/o modificarlo bajo
4  # los terminos de la licencia pública general de GNU, según es publicada
5  # por la free software fundation bien la versión 3 de la misma licencia
6  # o de cualquier versión posterior. (según su elección ).
7  # Si usted hace alguna modificación en esta aplicación, deberá siempre
8  # mencionar el autor original de la misma.
9  # Autor:
10 # Universidad Distrital Francisco Jose
11 # Grupo de fisica e informatica
12 # Dr Julian Andres Salamanca Bernal
13 # Diego Alberto Parra Garzón
14 # Colombia, Bogota D.C.
15
16 import numpy as np
17 import os
18 import serial
19 import subprocess
20 import math
21 import time
22 import Gnuplot
23 import shutil
24 import matplotlib.pyplot as plt
25 class App:
26     def Analisis(self):
27         # os.system("octave bin/estadistica1.m")
28         os.system("octave estadistica2.m")
29         self.BX = np.loadtxt("datos/BX.dat")
30         self.BY = np.loadtxt("datos/BY.dat")
31         self.error_CM1 = np.loadtxt("datos/ECM1.dat")
32         self.q1 = np.loadtxt("datos/q1.dat")
33         self.Amplitud = np.loadtxt("datos/Amplitud.dat")
34         self.Ampli = round(self.Amplitud, 6)
35         self.T = np.loadtxt("datos/Transmitancia.dat")
36         self.A = np.loadtxt("datos/Vteo.dat")
37         self.V_A = np.loadtxt("datos/V_aproximado.dat")
38         f = np.loadtxt("datos/F.dat")
39         a = np.loadtxt("datos/a.dat")
40         b = np.loadtxt("datos/b.dat")
41         self.Yest = np.loadtxt("datos/Yest.dat")
42         self.Yteo = np.loadtxt("datos/Yteo.dat")
43         self.ECM = np.loadtxt("datos/ECM.dat")
44         self.error_CM11 = round(self.error_CM1, 11)
45         self.X = f[:,0]
46         self.V = f[:,1]
47         self.a = round(a, 6)
48         self.b = round(b, 3)
49         print self.X, self.V, "\n\n", self.a, " X^(", self.b, ") "
50
51     def Grafica(self):
52         plt.subplot(221)
53         plt.title('Distancia vs Intensida \n')
54         plt.xlabel('Distancia [m]')
55         plt.ylabel('Intensidad [W]')
```

```

56     pl.plot(self.X, self.V, 'g:o')
57     pl.axis([0, 0.65, -0.001, 0.022])
58     pl.text(0.2, 0.019, r'Datos recolectados')
59
60     def Grafica1(self):
61         pl.subplots_adjust(left=0.13)
62         pl.subplots_adjust(bottom=0.13)
63         pl.subplots_adjust(right=0.97)
64         pl.subplots_adjust(top=0.87)
65         pl.subplots_adjust(wspace=0.37)
66         pl.subplots_adjust(hspace=0.68)
67
68     def Grafica2(self):
69         pl.subplot(222)
70         pl.title('APROXIMACION')
71         pl.xlabel('Distancia [m]')
72         pl.ylabel('Intensidad [W]')
73         pl.plot(self.X, self.A, 'B', self.q1, self.BY, 'y', self.q1, self.BX, 'b')
74         pl.plot(self.X, self.V_A, 'Y', self.X, self.V, 'g:o')
75         pl.axis([0, 0.65, -0.001, 0.022])
76         pl.text(0.15, 0.015, r' ' + str(0.000277) + ' + X^(' + str(-2) + ')')
77         pl.text(0.3, 0.017, r'Azul')
78         pl.text(0.2, 0.010, r'      Amarillo')
79         pl.text(0.19, 0.008, r' ' + str(self.Ampli) + ' X^(-2)')
80
81
82     def Grafica3(self):
83         pl.subplot(212)
84         pl.plot(self.X, self.V, 'g:o', self.X, self.V_A, 'Y', self.X, self.A, 'B',
85               self.q1, self.BY, 'y', self.q1, self.BX, 'b')
86         pl.xlabel('Distancia [m]')
87         pl.ylabel('Intensidad [W]')
88         pl.title('Distancia vs Intensidad irradiada \n')
89         pl.legend(loc='upper left')
90         pl.axis([0, 0.65, -0.001, 0.022])
91         pl.text(0.3, 0.019, r' ideal = ' + str(0.000277) + ' X^(' + str(-2) + ')')
92         pl.text(0.3, 0.016, r' Exp = ' + str(self.Ampli) + ' X^(-2)')
93         pl.text(0.12, 3.5, r'V1')
94         pl.text(0.01, 2, r'V2*')
95         pl.text(0.3, 0.013, r'Fp = ' + str(round(self.T,4)))
96         pl.text(0.3, 0.010, r'ECM = ' + str(round(self.error_CM1,11)))
97         pl.savefig('datos/Absorcion.png')
98         pl.show()
99
100    def Ordenar(self):
101        os.system("python bin/o_Carpetas2.py")
102
103    def __init__(self):
104        self.Analisis()
105        self.Grafica()
106        self.Grafica1()
107        self.Grafica2()
108        self.Grafica3()
109    # self.Ordenar()

```

```
110         self.__del__()
111
112     def __del__(self):
113         print ("PROGRAMA TERMINADO")
114
115
116 Iniciar = App()
117
```