

```
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 # Diego Alberto Parra Garzón
13 # Dr Julian Andres Salamanca Bernal
14 # Colombia, Bogota D.C.
15
16 import numpy as np
17 import pylab as pl
18 import serial
19 import os
20 import subprocess
21 import math
22 import time
23 import Gnuplot
24 from Tkinter import *
25 import tkMessageBox
26 import Tkinter
27 import shutil
28
29 class App:
30     def Modulo(self):
31         bicho = Tk()
32         bicho.geometry("430x180+180+60")
33         bicho.config(bg="white")
34         bicho.title("Infrarossi")
35         bicho.resizable(width=0, height=0)
36
37         def Salir():
38             tkMessageBox.showinfo("Infrarossi", message= "! Cerrando el programa i")
39             os.system("rm MAC.txt MACD.txt puerto.txt conexion.txt direccion.txt
40             dispo.txt log.txt mac.txt macd.txt texput.log x.log")
41             os.system("rm MAC.txt MACD.txt puerto.txt")
42             os.system("rm bin/conexion.txt bin/direccion.txt")
43             os.system("sh bin/d_Blu.sh &")
44             exit()
45
46         def Verifica():
47             tkMessageBox.showinfo("Infrarossi", message= "! Conectando con el
48             dispositivo, por favor espere i")
49             os.system('xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e sh
50             c_Blu.sh ')
51             Conecta()
52
53         def Conecta():
54             os.system('rfcomm -a > conexion.txt | cut -d ":" -f 1,1
```

```

        conexion.txt > direccion.txt ')
53     puerto = open('direccion.txt', 'r')
54     self.puerto = puerto.read(7)
55     puerto.close()
56     self.puerta = "/dev/"+self.puerto
57     print self.puerta
58
59     Conectar()
60
61     def Conectar():
62         tkMessageBox.showinfo("Infrarossi", message= "! Verificando conexion
        i")
63     print self.puerta
64     try:
65         arduino = serial.Serial(self.puerta, 9600)
66         time.sleep(4)
67         arduino.write("aa")
68         Valido()
69         Valido()
70     except:
71         tkMessageBox.showinfo("Infrarossi", message= "! No hay conexion i")
72         # os.system('xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue
        -e sh c_Blu.sh &')
73
74
75     def Valido():
76     # ..... Botones menu de inicio .....
77         x1 = int(150)
78         y3 = int(300)
79         lblRapidez = Label(bicho, text="\nMENU\nDE INICIO", fg = ("black"), bg
        = ("white"), font = ("Century Schoolbook L",10)).place(x=15+x1, y=280
        -y3)
80         btnComenzar = Button(bicho, text= "Difracción", width=6, height=1,
        command= Difraccion).place(x=20+x1, y=350-y3)
81         btnDetener= Button(bicho, text= "Atenuación", width=6, height=1,
        command= Atenuacion).place(x=20+x1, y=380-y3)
82         btnLimpiar = Button(bicho, text= "Absorción", width=6, height=1, command=
        Absorcion).place(x=20+x1, y=410-y3)
83
84
85
86     #----- Funcion limpiar pantalla
        -----
87     def Reset():
88         tkMessageBox.showinfo("Infrarossi", message= "! Limpiando, por favor
        espere i")
89         os.system("rm MAC.txt MACD.txt puerto.txt conexion.txt direccion.txt
        datos/dats1.dat")
90         os.system("sh bin/d_Blu.sh &")
91         os.system("python infrarossi.py &")
92         os.system("rm MAC.txt MACD.txt puerto.txt")
93         exit()
94
95     #----- Funciones menu
        -----

```

```

96     def Difraccion():
97         tkMessageBox.showinfo("Infrarossi", message= "Espere por favor,
           Preparando todo para empezar con el experimento.")
98         os.system("python bin/Difraccion.py &")
99     # os.system("python bin/estadis2.py &")
100
101
102     def Atenuacion():
103         tkMessageBox.showinfo("Infrarossi", message= "Espere por favor,
           Preparando todo para empezar con el experimento.")
104         os.system("python bin/Atenuacion.py &")
105     # os.system("xterm -T Infrarossi -geom 50x8+185+100 +cm -bg blue -e
python bin/Atenuacion.py & ")
106
107     def Absorcion():
108         tkMessageBox.showinfo("Infrarossi", message= "Espere por favor,
           Preparando todo para empezar con el experimento.")
109         os.system("python bin/Absorcion.py &")
110
111
112     #----- Bluetooth desconectado
-----
113     def BL_off():
114         tkMessageBox.showinfo("Infrarossi", message= "!Bluetooth desconectado!")
115         os.system("rm MAC.txt MACD.txt puerto.txt conexion.txt direccion.txt")
116         os.system("sh bin/d_Blu.sh &")
117         os.system("python infrarossi.py &")
118         os.system("rm MAC.txt MACD.txt puerto.txt")
119         exit()
120
121
122     # ----- Definiendo Funcion firmware
-----
123
124     def Firmware():
125         tkMessageBox.showinfo("Infrarossi", message= "Conecte la tarjeta
           microcontroladora arduino uno, con un microcontrolador listo para su
           uso.\n\nProcediendo con el instalador del firmware")
126         os.system("python bin/firmware/G_firmware.py &")
127
128     #-----Definiendo función
Documentación-----
129     def Documentacion():
130         tkMessageBox.showinfo("Infrarossi", message= "! Abriendo
           documentación, tenga paciencia i")
131         os.system("xdg-open 'Montaje/Articulo_montaje_infrarossi.pdf' &")
132
133
134
135
136     #----- Botones Bluetooth
-----
137     y1 = int(300)
138     lblBlue = Label(bicho, text="BLUETOOTH ", fg = ("black"), bg = ("white"),
           font = ("Century Schoolbook L",10)).place(x=10, y=320-y1)

```

```
139         btnConectar= Button(bicho, text= " ON ", width=5, height=1, command=
        Verifica).place(x=20, y=350-y1)
140     #         btnConectar= Button(bicho, text= " ON ", width=5, height=1, command=
Valido).place(x=20, y=350-y1)
141         btnDesconectar= Button(bicho, text= " OFF ", width=5, height=1, command=
        Bl_off).place(x=20, y=380-y1)
142         btnSalir=Button(bicho, text = "Salir", command=Salir, height=1, width =5).
        place(x=20, y=410-y1)
143
144
145     #----- Botones firmware -----
146         y2 = int(270)
147         btnfirmware=Button(bicho, text = "Firmware", command=Firmware, height=1, width
        =5).place(x=340, y=380-y2)
148         btnDocumentacion=Button(bicho, text = "Ayuda", command=Documentacion, height=1
        , width =5).place(x=340, y=350-y2)
149         btnLimpiar = Button(bicho, text= "limpiar", width=5, height=1, command=
        Reset).place(x=340, y=320-y2)
150         bicho.mainloop()
151
152
153     def __init__(self):
154         self.Modulo()
155         self.__del__()
156
157     def __del__(self):
158         print ("PROGRAMA TERMINADO")
159
160 if __name__ == "__main__":
161     App()
162
163
```