from tkinter import \*

from tkinter.ttk import \*

from time import strftime

import tkinter as tk

import random

#Funcion para la hora del reloj digital

def reloj\_digital():

etiqueta\_hm.config(text= strftime ("%H:%M"))

etiqueta\_s.config(text= strftime ("%S"))

etiqueta\_fecha.config(text= strftime ("%A, %d/%m/%Y"))

if cb[0] == 100:

boton\_cargador.config(fg="yellow")

funcion\_cargar\_texto()

etiqueta\_s.after(1000, reloj\_digital)

#FUNCION DE BLUETOOTH

def bluetooth():

contador\_blue = [0]

# Creamos la lista de dispositivos móviles cercanos

devices = ["iPhone de Esteban", "iPhone de Carolina", "iPhone de Alberto", "iPhone de Luismi","Redmi Note 11S"]

# Creamos la lista de dispositivos cercanos en la interfaz

listbox = tk.Listbox(app, width=50, height=10)

for device in devices:

listbox.insert(tk.END, device)

listbox.place(x= 200, y = 200)

# Función que se ejecuta al pulsar el botón "Enlazar"

def enlazar():

if contador\_blue[0] == 0:

# Obtenemos el elemento seleccionado de la lista

seleccionado = listbox.get(tk.ACTIVE)

mensaje\_conectado = Label(app, background="white", text="Conectado a {} correctamente !!".format(seleccionado))

mensaje\_conectado.place(x= 250, y = 400)

contador\_blue[0] = 1

#nuevo boton

button.destroy()

def borrarblue():

listbox.destroy()

mensaje\_conectado.destroy()

button\_continuar.destroy()

luz()

button\_continuar = tk.Button(app, text="Continuar", command=borrarblue)

button\_continuar.place(x= 350, y = 450)

# Configuramos el botón de enlazar

button = tk.Button(app, text="Enlazar", command=enlazar)

button.place(x= 350, y = 400)

#Funcion luz

def luz():

if cb[0] <= 80:

#ENCENDER

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

etiqueta\_bateria.place(x = 600,y = 750)

elif primera\_vez[0] == True:

#grado de encendido

primera\_vez[0] = False

# crear los labels y los inputs

nombre\_label = tk.Label(app, text="Nombre:")

nombre\_input = tk.Entry(app)

apellido\_label = tk.Label(app, text="Apellidos:")

apellido\_input = tk.Entry(app)

nacimiento\_label = tk.Label(app, text="Año de nacimiento:")

nacimiento\_input = tk.Entry(app)

altura\_label = tk.Label(app, text="Altura:")

altura\_input = tk.Entry(app)

peso\_label = tk.Label(app, text="Peso:")

peso\_input = tk.Entry(app)

localizacion\_label = tk.Label(app, text="Localización:")

localizacion\_input = tk.Entry(app)

actividades\_label = tk.Label(app, text="Nº de actividades semanales:")

actividades\_input = tk.Entry(app)

deportes\_label = tk.Label(app, text="Deportes favoritos:")

deportes\_input = tk.Entry(app)

# agregar los widgets a la ventana principal

espacio1 = Label(app, text=" ", background="black")

espacio1.pack()

nombre\_label.pack()

nombre\_input.pack()

espacio2 = Label(app, text=" ", background="black")

espacio2.pack()

apellido\_label.pack()

apellido\_input.pack()

espacio3 = Label(app, text=" ", background="black")

espacio3.pack()

nacimiento\_label.pack()

nacimiento\_input.pack()

espacio4 = Label(app, text=" ", background="black")

espacio4.pack()

altura\_label.pack()

altura\_input.pack()

espacio5 = Label(app, text=" ", background="black")

espacio5.pack()

peso\_label.pack()

peso\_input.pack()

espacio6 = Label(app, text=" ", background="black")

espacio6.pack()

localizacion\_label.pack()

localizacion\_input.pack()

espacio7 = Label(app, text=" ", background="black")

espacio7.pack()

actividades\_label.pack()

actividades\_input.pack()

espacio8 = Label(app, text=" ", background="black")

espacio8.pack()

deportes\_label.pack()

deportes\_input.pack()

espacio9 = Label(app, text=" ", background="black")

espacio9.pack()

#Funcion que borra el imput y empieza el reloj

def destruir\_widgets():

nombre\_label.destroy()

nombre\_input.destroy()

apellido\_label.destroy()

apellido\_input.destroy()

nacimiento\_label.destroy()

nacimiento\_input.destroy()

altura\_label.destroy()

altura\_input.destroy()

peso\_label.destroy()

peso\_input.destroy()

localizacion\_label.destroy()

localizacion\_input.destroy()

actividades\_label.destroy()

actividades\_input.destroy()

deportes\_label.destroy()

deportes\_input.destroy()

limpiar\_btn.destroy()

espacio1.destroy()

espacio2.destroy()

espacio3.destroy()

espacio4.destroy()

espacio5.destroy()

espacio6.destroy()

espacio7.destroy()

espacio8.destroy()

espacio9.destroy()

bluetooth()

limpiar\_btn = tk.Button(app, text="EMPEZAR", command=destruir\_widgets)

limpiar\_btn.pack()

else:

label\_imagen\_negro.destroy()

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

etiqueta\_bateria.place(x = 600,y = 750)

etiqueta\_bateria.destroy()

#HORA

frame\_hora.pack()

frame\_hora.place(relx=0.5, rely=0.5, anchor=CENTER)

#FECHA

etiqueta\_fecha.pack(anchor="center")

#Funcion para que cambie a negro

def cambiar\_color\_a\_negro():

app.config(bg='black')

Label\_imagen\_usb.config(background= "black")

#Cambiamos a azul claro

app.config(background= "light blue")

Label\_imagen\_usb.config(background= "light blue")

#Indicamos tiempo de espera

app.after(3000, cambiar\_color\_a\_negro)

#Funcion que cambia la imagen del cargador

def cambiar\_imagen\_cargador():

if cargando[0] == False:

Label\_imagen\_usb.config(image=imagen\_usb\_conectado)

boton\_cargador.config(fg="green")

cargando[0] = True

app.after(3000, funcion\_cargar)

elif cargando[0] == True:

# Crear un widget Label y asignarle la imagen

label\_imagen\_negro = Label(app, image=imagen\_negro, background="black")

label\_imagen\_negro.place(x= 180, y= 730)

Label\_imagen\_usb.config(image=imagen\_usb)

boton\_cargador.config(fg="yellow")

cargando[0] = False

#Funcion cargar

def actualizar\_bateria():

if cargando[0] == True:

if cb[0] == 100:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

elif cb[0] < 10:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

else:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

etiqueta\_bateria.place(x = 600,y = 750)

if cb[0] == 51:

cb[0] = 60

elif cb[0] < 100:

cb[0] += 5

app.after(2000, actualizar\_bateria)

def funcion\_cargar():

if cb[0] >= 100:

cb[0] = 100

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

elif cb[0] < 10:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

else:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

etiqueta\_bateria.place(x = 600,y = 750)

app.after(500, actualizar\_bateria)

def funcion\_cargar\_texto():

if cb[0] >= 100:

cb[0] = 100

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

elif cb[0] < 10:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

else:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

etiqueta\_bateria.place(x = 600,y = 750)

#Funcion descargar

def descargar():

cb[0] = cb[0] - 5

if cb[0] < 0:

cb[0] = 0

if cb[0] == 100:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

elif cb[0] < 10:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

else:

etiqueta\_bateria = Label(app,font=("digitalk", 15),text ="BATERIA: {} %".format(cb[0]))

etiqueta\_bateria.place(x = 600,y = 750)

etiqueta\_s.after(30000, descargar)

app = Tk ()

#VARIABLES

cargando = [False]

primera\_vez=[True]

# Cargar la imagen

imagen\_sol = PhotoImage(file="sol.png")

imagen\_nube = PhotoImage(file="nube.png")

imagen\_negro = PhotoImage(file= "negro.png")

label\_imagen\_negro = Label(app, image=imagen\_negro, background="black")

label\_imagen\_negro.place(x= 180, y= 730)

def llamando\_carga\_solar():

carga\_solar()

app.after(5000, llamando\_carga\_solar)

def carga\_solar():

label\_imagen\_nube = Label()

label\_imagen\_sol = Label()

label\_imagen\_negro = Label()

if cargando[0] == False and primera\_vez[0] == False and cb[0] < 100:

label\_imagen\_negro.destroy()

label\_imagen\_nube.destroy()

label\_imagen\_sol.destroy()

#Esta amaneciendo

if strftime("%H") == "06" or strftime("%H") == "07" or strftime("%H") == "08":

luz = "amaneciendo"

# Crear un widget Label y asignarle la imagen

label\_imagen\_sol = Label(app, image=imagen\_sol, background="black")

label\_imagen\_sol.place(x= 195, y= 728)

cb[0] = cb[0] + 1

#Es de dia

elif strftime("%H") == "09" or strftime("%H") == "10" or strftime("%H") == "11" or strftime("%H") == "12" or strftime("%H") == "13" or strftime("%H") == "14" or strftime("%H") == "15":

numero\_aleatorio = random.randint(0, 1)

#soleado

if numero\_aleatorio == 0:

luz = "dia soleado"

# Crear un widget Label y asignarle la imagen

label\_imagen\_sol = Label(app, image=imagen\_sol, background="black")

label\_imagen\_sol.place(x= 195, y= 728)

cb[0] = cb[0] + 3

#nublado

elif numero\_aleatorio == 1:

luz = "dia nublado"

cb[0] = cb[0] + 2

# Crear un widget Label y asignarle la imagen

label\_imagen\_nube = Label(app, image=imagen\_nube, background="black")

label\_imagen\_nube.place(x= 195, y= 730)

elif strftime("%H") == "16" or strftime("%H") == "17" or strftime("%H") == "18" or strftime("%H") == "19" or strftime("%H") == "20":

numero\_aleatorio = random.randint(0, 1)

#soleado

if numero\_aleatorio == 0:

luz = "tarde soleada"

cb[0] = cb[0] + 2

# Crear un widget Label y asignarle la imagen

label\_imagen\_sol = Label(app, image=imagen\_sol, background="black")

label\_imagen\_sol.place(x= 195, y= 728)

#nublado

elif numero\_aleatorio == 1:

luz = "tarde nublada"

cb[0] = cb[0] + 1

# Crear un widget Label y asignarle la imagen

label\_imagen\_nube = Label(app, image=imagen\_nube, background="black")

label\_imagen\_nube.place(x= 195, y= 730)

else:

luz = "noche"

if luz != "noche":

boton\_cargador.config(fg="green")

else:

# Crear un widget Label y asignarle la imagen

label\_imagen\_negro = Label(app, image=imagen\_negro, background="black")

label\_imagen\_negro.place(x= 180, y= 728)

#Titulo

app.title('Reloj digital')

#Modificar si es ecalable

app.resizable(False,False)

#Fondo

app.config(bg= 'black')

app.geometry('800x800')

#Frame de la hora y fecha

frame\_hora = Frame(app, width="800", height="800")

#Etiqueta horas y minutos

etiqueta\_hm = Label(frame\_hora, font=("digitalk",70),text="H:M")

etiqueta\_hm.grid(row=0, column = 0)

#Etiqueta segundos

etiqueta\_s = Label(frame\_hora, font=("digitalk", 30),text = "s")

etiqueta\_s.grid(row=0, column=1, sticky="n")

#Etiqueta fecha

etiqueta\_fecha = Label(font=("digitalk", 30),text ="dia d/m/aaaa")

#Bonton light

boton\_luz = tk.Button(app, text="Light", font=("digitalk", 12), borderwidth=8,command=luz)

boton\_luz.place(x = 10,y = 100)

#Imagen usb

imagen\_usb = PhotoImage(file="usb\_hembra.png")

imagen\_usb\_conectado = PhotoImage(file="usb\_hembra\_conectado.png")

Label\_imagen\_usb = Label(app,image=imagen\_usb, background="black")

Label\_imagen\_usb.place(x= 280, y= 730)

#Boton

boton\_cargador = tk.Button(app, text="C", font=("digitalk", 15), command=cambiar\_imagen\_cargador,fg="yellow" ,bg="black", borderwidth=0)

boton\_cargador.place(x= 490, y= 750)

#BATERIA

#carga bateria

cb = [51]

#tiempo de carga

t = 2

#Descargar

app.after(10000, descargar)

reloj\_digital()

carga\_solar()

llamando\_carga\_solar()

app.mainloop()