In [ ]:

```
# Importar Bibliotecas
from tkinter import *
from tkinter import messagebox
from tkinter import filedialog as FileDialog
import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk
import sqlite3

# Desarrollo de la interfaz gráfica
root=Tk()
root.title("Aplicación para gestionar colecciones con Base de Datos")
root.geometry("700x500")

cid=StringVar()
coleccion=StringVar()
tipo=StringVar()
descripcion=StringVar()
casa=StringVar()
habitacion=StringVar()
lugar=StringVar()
tamanyo=StringVar()
procedencia=StringVar()
fabricante=StringVar()
comentario=StringVar()
imagen=StringVar()


def conexion_bbdd():
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()

    try:
        cursor.execute('''
            CREATE TABLE colecciones (
            ID INTEGER PRIMARY KEY AUTOINCREMENT,
            COLECCION VARCHAR(20) NOT NULL,
            TIPO VARCHAR(20) NOT NULL,
            DESCRIPCION VARCHAR(30),
            CASA VARCHAR(10),
            HABITACION VARCHAR(20),
            LUGAR VARCHAR(20),
            TAMANYO VARCHAR(20),
            PROCEDENCIA VARCHAR(20),
            FABRICANTE VARCHAR(20),
            COMENTARIO VARCHAR(50),
            IMAGEN VARCHAR(10))
            ''')
        messagebox.showinfo("CONEXION","Base de Datos creada correctamente")
#    except:
    except Exception as e:
        print(str(e))
        messagebox.showinfo("CONEXION","Conexión exitosa con la Base de Datos")

def eliminar_bbdd():
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()
    if messagebox.askyesno(message="¿Los datos se perderan definitivamente, desea
        cursor.execute("DROP TABLE colecciones")
    else:
        pass
```

```
def salir_aplicacion():
    valor=messagebox.askquestion("Salir", "¿Está seguro que quiere salir?")
    if valor=="yes":
        root.destroy()

def limpiar_campos():
    cid.set ("")
    coleccion.set ("")
    tipo.set ("")
    descripcion.set ("")
    casa.set ("")
    habitacion.set ("")
    lugar.set ("")
    tamanyo.set ("")
    procedencia.set ("")
    fabricante.set ("")
    comentario.set ("")
    imagen.set("")
    cls_image()                                 # <------------------ mere

def cls_image():       # <------------------ mere
    Label(root, text = "", width=15, height=6).place(x=500, y=100)    # <---------

def mensaje():
    acerca='''
    Aplicación Colecciones
    Versión 1.0
    Tecnología Python Tkinter
    '''
    messagebox.showinfo(title="INFORMACION", message=acerca)

######################################### Métodos CRUD #######

def crear():
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()
    try:
        datos=coleccion.get(),tipo.get(),descripcion.get(),casa.get(),habitacion.g
        cursor.execute("INSERT INTO colecciones VALUES(NULL,?,?,?,?,?,?,?,?,?,?,?)
        conexion.commit()
#    except:
    except Exception as e:
        print(str(e))
        messagebox.showwarning("ADVERTENCIA","Ocurrió un error al crear el registr
        pass
    limpiar_campos()
    mostrar()

def mostrar():
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()
    registros=tree.get_children()
    for elemento in registros:
        tree.delete(elemento)
    try:
        cursor.execute("SELECT * FROM colecciones")
        for row in cursor:
            tree.insert("",0,text=row[0],values=(row[1],row[2],row[3],row[4],row[5
#    except:
    except Exception as e:
        print(str(e))
        pass
```

```
###################### Tabla ######################

tree=ttk.Treeview(heigh=10, columns=('#0','#1','#2'))
tree.place(x=0, y=250)
tree.column('#0',width=100)
tree.heading('#0', text="ID", anchor=CENTER)
tree.heading('#1', text="Colección", anchor=CENTER)
tree.heading('#2', text="Tipo", anchor=CENTER)
tree.heading('#3', text="Descripción", anchor=CENTER)

def seleccionar_usando_click(event):
    item=tree.identify('item',event.x,event.y)
    cid.set(tree.item(item,"text"))
    coleccion.set(tree.item(item,"values")[0])
    tipo.set(tree.item(item,"values")[1])
    descripcion.set(tree.item(item,"values")[2])
    casa.set(tree.item(item,"values")[3])
    habitacion.set(tree.item(item,"values")[4])
    lugar.set(tree.item(item,"values")[5])
    tamanyo.set(tree.item(item,"values")[6])
    procedencia.set(tree.item(item,"values")[7])
    fabricante.set(tree.item(item,"values")[8])
    comentario.set(tree.item(item,"values")[9])
    imagen.set(tree.item(item,"values")[10])
    cls_image()        # <------------------ mere
    carga_imagen()     # <------------------ mere

tree.bind("<Double-1>", seleccionar_usando_click)

def actualizar():
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()
    try:
        datos=coleccion.get(),tipo.get(),descripcion.get(),casa.get(),habitacion.g
        cursor.execute("UPDATE colecciones SET COLECCION=?,TIPO=?,DESCRIPCION=?,CA
        conexion.commit()
#    except:
    except Exception as e:
        print(str(e))
        messagebox.showwarning("ADVERTENCIA","Ocurrió un error al actualizar el re
        pass
    limpiar_campos()
    mostrar()

def buscar_imagen():
    fichero = FileDialog.askopenfilename(initialdir = "./img/",title="Buscar image
    i=fichero[::-1].find("/")
    i=i*-1
    nom_imagen2=(fichero[i:])
    i=nom_imagen2[::-1].find(".")
    i=i+1
    cl=len(nom_imagen2)
    cl=cl-i
    nom_imagen=(nom_imagen2[:cl])
    print(nom_imagen)
    localizar_registro_imagen(nom_imagen)



def localizar_registro_imagen(nom_imagen):
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()
    registros=tree.get_children()
```

```
    for elemento in registros:
        tree.delete(elemento)
    try:
        cursor.execute(f"SELECT * FROM colecciones WHERE IMAGEN = '{nom_imagen}'")
        for row in cursor:
            tree.insert("",0,text=row[0],values=(row[1],row[2],row[3],row[4],row[5

    except Exception as e:
        print(str(e))
        messagebox.showwarning("ADVERTENCIA","Ocurrió un error al buscar el regist

def carga_imagen():
    fname = f"./img/{imagen.get()}.jpg"
    im1 = Image.open(fname)
    fname1 = fname.replace("jpg", "png")
    im1 = im1.resize((90, 90))
    im1.save(fname1)

    fname = f"./img/{imagen.get()}.png"
    stgImg = PhotoImage(file=fname)
    label=ttk.Label(root, image=stgImg)
    label.place(x=500, y=100)
    label.image = stgImg
    root.update_idletasks()
    return


def borrar():
    conexion=sqlite3.connect("./dat/base_colecciones")
    cursor=conexion.cursor()
    try:
        if messagebox.askyesno(message="¿Realmente desea eliminar el registro?", t
            cursor.execute("DELETE FROM colecciones WHERE ID="+cid.get())
            conexion.commit()
#    except:
    except Exception as e:
        print(str(e))
        messagebox.showwarning("ADVERTENCIA","Ocurrió un error al tratar de elimir
        pass
    limpiar_campos()
    mostrar()

###################### Colocar widgets en la VISTA ################

###################### Menús ######################
menubar=Menu(root)
menubasedat=Menu(menubar,tearoff=0)
menubasedat.add_command(label="Crear/Conectar Base de Datos", command=conexion_bbd
menubasedat.add_command(label="Eliminar Base de Datos", command=eliminar_bbdd)
menubasedat.add_command(label="Salir", command=salir_aplicacion)
menubar.add_cascade(label="Inicio", menu=menubasedat)

menuayuda=Menu(menubar,tearoff=0)
menuayuda.add_command(label="Resetear Campos", command=limpiar_campos)
menuayuda.add_command(label="Acerca", command=mensaje)
menubar.add_cascade(label="Ayuda", menu=menuayuda)

################## Etiquetas y cajas de texto ##############
e1=Entry(root, textvariable=cid)

l2=Label(root, text="Colección")
l2.place(x=50,y=10)
e2=Entry(root, textvariable=coleccion, width=20)
```

```
e2.place(x=150,y=10)

l3=Label(root, text="Tipo")
l3.place(x=340,y=10)
e3=Entry(root, textvariable=tipo, width=20)
e3.place(x=380,y=10)

l4=Label(root, text="Descripcion")
l4.place(x=50,y=40)
e4=Entry(root, textvariable=descripcion, width=30)
e4.place(x=150,y=40)

l5=Label(root, text="Casa")
l5.place(x=50,y=70)
e5=Entry(root, textvariable=casa, width=20)
e5.place(x=100,y=70)

l6=Label(root, text="Habitación")
l6.place(x=260,y=70)
e6=Entry(root, textvariable=habitacion, width=20)
e6.place(x=340,y=70)

l7=Label(root, text="Lugar")
l7.place(x=500,y=70)
e7=Entry(root, textvariable=lugar, width=20)
e7.place(x=550,y=70)

l8=Label(root, text="Tamaño")
l8.place(x=50,y=100)
e8=Entry(root, textvariable=tamanyo, width=20)
e8.place(x=100,y=100)

l9=Label(root, text="Procedencia")
l9.place(x=260,y=100)
e9=Entry(root, textvariable=procedencia, width=20)
e9.place(x=340,y=100)

l11=Label(root, text="Fabricante")
l11.place(x=50,y=130)
e11=Entry(root, textvariable=fabricante, width=20)
e11.place(x=115,y=130)

l11=Label(root, text="Imagen")
l11.place(x=270,y=130)
e11=Entry(root, textvariable=imagen, width=10)
e11.place(x=340,y=130)


l10=Label(root, text="Comentario")
l10.place(x=50,y=160)
e10=Entry(root, textvariable=comentario, width=50)
e10.place(x=150,y=160)


############################### Creando botones ##############

b1=Button(root, text="Crear registro", command=crear)
b1.place(x=30,y=200)
b2=Button(root, text="Modificar registro", command=actualizar)
b2.place(x=150,y=200)
b3=Button(root, text="Mostrar lista", command=mostrar)
b3.place(x=300,y=200)
b4=Button(root, text="Eliminar registro", bg="red",command=borrar)
```

```
b4.place(x=420,y=200)
b5=Button(root, text="Buscar imagen",command=buscar_imagen)
b5.place(x=560,y=200)
b6=Button(root, text="...",command=carga_imagen)        # <------------------ mere
b6.place(x=420,y=130)                                   # <------------------ mere

root.config(menu=menubar)

root.mainloop()
```

```
penjat-2
penjat-6
penjat-3
```

In [ ]:

In [ ]: