

```

from tkinter import *
from tkinter import messagebox #msg emergentes
import sqlite3 # base de datos a crear
def conexionBD():
    miConexion=sqlite3.connect("Usuarios")
    miCursor=miConexion.cursor()

try:
    miCursor.execute('''
        CREATE TABLE DATOSUSUARIO (
        ID INTEGER PRIMARY KEY AUTOINCREMENT,
        NOMBRE_USUARIO VARCHAR(50),
        PASSWORD VARCHAR(50),
        APELLIDO VARCHAR(20),
        DIRECCION VARCHAR(50),
        COMENTARIOS VARCHAR(100))
    ''')

    messagebox.showinfo("BD", "BD creada con exito: ")
except:
    messagebox.showwarning("Atencion ","La BD ya existe")

def salirAplicacion():
    valor=messagebox.askquestion("Salir", "¿Desea salir de la aplicacion?: ")
    if valor=="yes":
        root.destroy()

def limpiarCampos():
    miNombre.set("")
    miId.set("")
    miApellido.set("")
    miDireccion.set("")
    miPass.set("")
    textoComentario.delete(1.0, END)

def crear():
    miConexion=sqlite3.connect("Usuarios")
    miCursor=miConexion.cursor()
    datos=
    miNombre.get(),miPass.get(),miApellido.get(),miDireccion.get(),textoComentario.g
    et("1.0", END)

    miCursor.execute("INSERT INTO DATOSUSUARIO VALUES(NULL,?, ?, ?, ?, ?)",(datos))

    """miCursor.execute("INSERT INTO DATOSUSUARIO VALUES(NULL,'" +miNombre.get() +
+
    "','" + miPass.get() +
    "','" + miApellido.get() +
    "','" + miDireccion.get() +
    "','" + textoComentario.get(1.0, END) + "')")"""

    miConexion.commit()
    messagebox.showinfo("BD", "Registro insertado con exito ")

def leer():
    miConexion=sqlite3.connect("Usuarios")
    miCursor=miConexion.cursor()

    miCursor.execute("SELECT * FROM DATOSUSUARIO WHERE ID=" +miId.get())
    elUsuario=miCursor.fetchall() # devuelve arreglo con los datos

```

```

for usuario in elUsuario:
    miId.set(usuario[0])
    miNombre.set(usuario[1])
    miPass.set(usuario[2])
    miApellido.set(usuario[3])
    miDireccion.set(usuario[4])
    textoComentario.insert(1.0, usuario[5])

miConexion.commit()

def actualizar():
    miConexion=sqlite3.connect("Usuarios")
    miCursor=miConexion.cursor()
    datos=
    miNombre.get(),miPass.get(),miApellido.get(),miDireccion.get(),textoComentario.g
et("1.0", END)
    """miCursor.execute("UPDATE DATOSUSUARIO SET NOMBRE_USUARIO='"
    miNombre.get() +
        "', PASSWORD='"+ miPass.get() +
        "', APELLIDO='"+miApellido.get() +
        "', DIRECCION='"+miDireccion.get() +
        "', COMENTARIOS='"+textoComentario.get("1.0", END) +
        "' WHERE ID=" + miId.get())"""
    miCursor.execute("UPDATE DATOSUSUARIO SET NOMBRE_USUARIO=?, PASSWORD=?,
APELLIDO=?, DIRECCION=?, COMENTARIOS=? " +
                    "WHERE ID=" + miId.get(),(datos))

    messagebox.showinfo("BD", "Registro actualizado con exito ")

def eliminar():
    miConexion=sqlite3.connect("Usuarios")
    miCursor=miConexion.cursor()

    miCursor.execute("DELETE FROM DATOSUSUARIO WHERE ID=" +miId.get())

    miConexion.commit()
    messagebox.showinfo("BD", "Registro eliminado con exito ")

root=Tk()

barraMenu=Menu(root)
root.config(menu=barraMenu, width=300, height=400)

bdMenu=Menu(barraMenu, tearoff=0)
bdMenu.add_command(label="Conectar", command=conexionBD)
bdMenu.add_command(label="Salir", command=salirAplicacion)

borraMenu=Menu(barraMenu, tearoff=0)
borraMenu.add_command(label="Borrar campos", command=limpiarCampos)

crudMenu=Menu(barraMenu, tearoff=0)
crudMenu.add_command(label="Crear", command=crear)
crudMenu.add_command(label="Leer", command=leer)
crudMenu.add_command(label="Actualizar", command=actualizar)
crudMenu.add_command(label="Borrar", command=eliminar)

ayudaMenu=Menu(barraMenu, tearoff=0)
ayudaMenu.add_command(label="Licencia")
ayudaMenu.add_command(label="Acerca de: ")

```

```

barraMenu.add_cascade(label="BD", menu=bdMenu)
barraMenu.add_cascade(label="Borrar", menu=borraMenu)
barraMenu.add_cascade(label="CRUD", menu=crudMenu)
barraMenu.add_cascade(label="Ayuda", menu=ayudaMenu)

#Comienzo de campos

miFrame=Frame(root)
miFrame.pack()

miId=StringVar()
miNombre=StringVar()
miApellido=StringVar()
miPass=StringVar()
miDireccion=StringVar()
# deben ser asignadas a a cada Entry Cadena de caracteres

cuadroID=Entry(miFrame, textvariable=miId)
cuadroID.grid(row=0, column=1,padx=10,pady=10)

cuadroNombre=Entry(miFrame, textvariable=miNombre)
cuadroNombre.grid(row=1, column=1,padx=10,pady=10)
cuadroNombre.config(fg="red", justify="right")

cuadroPass=Entry(miFrame, textvariable=miPass)
cuadroPass.grid(row=2, column=1,padx=10,pady=10)
cuadroPass.config(show="?")

cuadroApellido=Entry(miFrame, textvariable=miApellido)
cuadroApellido.grid(row=3, column=1,padx=10,pady=10)

cuadroDireccion=Entry(miFrame, textvariable=miDireccion)
cuadroDireccion.grid(row=4, column=1,padx=10,pady=10)

textoComentario=Text(miFrame, width=16, height=5)
textoComentario.grid(row=5, column=1, padx=10, pady=10)
scrollVert=Scrollbar(miFrame, command=textoComentario.yview)
scrollVert.grid(row=5,column=2, sticky="nsew")
textoComentario.config(yscrollcommand=scrollVert.set)

-----Labels o etiquetas de los cuadros Entry
idLabel=Label(miFrame, text="Id: ")
idLabel.grid(row=0 , column=0, sticky="e", padx=10,pady=10)

nombreLabel=Label(miFrame, text="Nombre: ")
nombreLabel.grid(row=1 , column=0, sticky="e", padx=10,pady=10)

passLabel=Label(miFrame, text="Password: ")
passLabel.grid(row=2 , column=0, sticky="e", padx=10,pady=10)

apellidoLabel=Label(miFrame, text="Nombre:Apellido: ")
apellidoLabel.grid(row=3 , column=0, sticky="e", padx=10,pady=10)

direccionLabel=Label(miFrame, text="Direccion: ")
direccionLabel.grid(row=4 , column=0, sticky="e", padx=10,pady=10)

comentariosLabel=Label(miFrame, text="Comentarios: ")
comentariosLabel.grid(row=5 , column=0, sticky="e", padx=10,pady=10)

```

```
# -----Botones inferiores
miFrame2=Frame(root)
miFrame2.pack()

botonCrear=Button(miFrame2, text="Create", command=crear)
botonCrear.grid(row=1,column=0, sticky="e", padx=10, pady=10)

botonLeer=Button(miFrame2, text="Read", command=leer)
botonLeer.grid(row=1,column=2, sticky="e", padx=10, pady=10)

botonActualizar=Button(miFrame2, text="Update", command=actualizar)
botonActualizar.grid(row=1,column=3, sticky="e", padx=10, pady=10)

botonBorrar=Button(miFrame2, text="Delete", command=eliminar)
botonBorrar.grid(row=1,column=4, sticky="e", padx=10, pady=10)

root.mainloop()
```