

In [1]:

```
# se importa las libreria tkinter  
import tkinter as tk  
from tkinter.ttk import *  
import datetime  
import sqlite3  
from tkinter import messagebox  
from tkinter import ttk
```

In [2]:

```

conexion = sqlite3.connect('C:/Users/ADMINX/Documents/9ciclo/MATERIAS/SISTEMAS-IA/leccionpython.db')
cursordb = conexion.cursor()
global codigoclig
#ELIMINANDO TABLAS
#cursordb.execute("DROP TABLE pedidos")
#cursordb.execute("DROP TABLE detallescuentas")
#SE CREA LA TABLAS CON SQLITE
cursordb.execute("CREATE TABLE IF NOT EXISTS clientes (id INTEGER PRIMARY KEY AUTOINCREMENT,nombre TEXT,apellido TEXT,telefono TEXT,correo TEXT)")
print("SE CREA LA TABLA LOS CLIENTES")
cursordb.execute("CREATE TABLE IF NOT EXISTS cuentas (idc INTEGER PRIMARY KEY AUTOINCREMENT,tipo TEXT,fecha TEXT)")
print("SE CREA LA TABLA LOS CUENTAS")
cursordb.execute("CREATE TABLE IF NOT EXISTS detallescuentas (iddc INTEGER PRIMARY KEY AUTOINCREMENT,idcuenta INTEGER,fecha TEXT,detalle TEXT,debito TEXT,credito TEXT,saldo TEXT,FOREIGN KEY (idcuenta) REFERENCES cuentas (idc))")
print("SE CREA LA TABLA LOS DETALLES CUENTAS")
cursordb.execute("CREATE TABLE IF NOT EXISTS intermediacuecli (idi INTEGER PRIMARY KEY AUTOINCREMENT,idcliente INTEGER,idcuenta INTEGER,FOREIGN KEY (idcliente) REFERENCES clientes (id),FOREIGN KEY (idcuenta) REFERENCES cuentas (idc))")
print("SE CREA LA TABLA LOS INTERMEDIA CUENTAS Y CLIENTE")
cursordb.execute("CREATE TABLE IF NOT EXISTS productos (idpro INTEGER PRIMARY KEY AUTOINCREMENT,descripcion TEXT,cantidad INTEGER,stock INTEGER,precio TEXT)")
print("SE CREA LA TABLA LOS PRODUCTOS")
cursordb.execute("CREATE TABLE IF NOT EXISTS pedidos (idp INTEGER PRIMARY KEY AUTOINCREMENT,descripcion TEXT,costo TEXT,idinter INTEGER,idpro INTEGER,cantidad INTEGER,FOREIGN KEY (idinter) REFERENCES intermedioacuecli (idi),FOREIGN KEY (idpro) REFERENCES productos(idpro))")
print("SE CREA LA TABLA LOS PEDIDOS")

class Ventana:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        # SE CREA EL MENU DE LA APP
        menu = tk.Menu(master)

        submenu =tk.Menu(menu)

        submenu.add_command(label='Crear Cliente',font=("Arial Bold",25),command=self.crear_cliente)
        submenu.add_separator()
        submenu.add_command(label='Crear Pedido',font=("Arial Bold",25),command=self.crear_pedido)
        submenu.add_separator()
        submenu.add_command(label='Cuenta',font=("Arial Bold",25),command=self.cuenta_credito)
        #menucuenta = tk.Menu(submenu)
        #menucuenta.add_command(label='Crear Cuenta',font=("Arial Bold",25),command=self.cuenta_credito)
        menu.add_cascade(label='Sesion Clientes',menu=submenu)
        #submenu.add_cascade(label='Gestion Cuenta',submenu=menucuenta)
        master.config(menu=menu)

        submenu1 =tk.Menu(menu)
        submenu1.add_command(label='Gestion Pedido',font=("Arial Bold",25),command=self.gestion_pedido)

```

```

        submenu1.add_separator()
        submenu1.add_command(label='Gestion Productos',font=("Arial Bold",25),command=self.gestion_productos)
        submenu1.add_separator()

        submenu1.add_command(label='Descuento Cuenta',font=("Arial Bold",25),command=self.descuento_cuenta)
        menu.add_cascade(label='Sesion Administrador',menu=submenu1)
        master.config(menu=menu)
        self.frame.pack()

    def crear_cliente(self):
        self.crear_cliente = tk.Toplevel(self.master)
        self.app = Crearcliente(self.crear_cliente)
    def crear_pedido(self):
        self.crear_pedido = tk.Toplevel(self.master)
        self.app = Crearpedido(self.crear_pedido)
    def cuenta_credito(self):
        self.cuenta_credito = tk.Toplevel(self.master)
        self.app = Cuentacredito(self.cuenta_credito)

    def gestion_pedido(self):
        self.gestion_pedido = tk.Toplevel(self.master)
        self.app = Gestionpedido(self.gestion_pedido)

    def descuento_cuenta(self):
        self.descuento_cuenta = tk.Toplevel(self.master)
        self.app = Descuentocuenta(self.descuento_cuenta)
    def gestion_productos(self):
        self.gestion_productos = tk.Toplevel(self.master)
        self.app = Crearproducto(self.gestion_productos)
# SE CREA CLASES DE CONECCION A LA BASE DE DATOS SQL LITE
#CLASES A GESTIONAR
class Crearcliente:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.etiqueta = tk.Label(master, text="Gestion usuarios",font=("Arial Bold",25))

        #self.etiqueta.grid(column=0,row=0)
        self.etiqueta.pack(ipadx=10)
        self.etiqueta2 = tk.Label(master, text="Nombre:",font=("Arial Bold",25))
        self.nombre = tk.Entry(master,width=40)
        self.etiqueta2.pack(ipadx=10)
        self.nombre.pack()

        self.etiqueta3 = tk.Label(master, text="Apellido",font=("Arial Bold",25))
        self.direccion = tk.Entry(master,width=40)
        self.etiqueta3.pack(ipadx=10)
        self.direccion.pack()

        self.etiqueta4 = tk.Label(master, text="Telefono",font=("Arial Bold",25))
        self.telefono = tk.Entry(master,width=40)
        self.etiqueta4.pack(ipadx=10)
        self.telefono.pack()

        self.etiqueta5 = tk.Label(master, text="Correo",font=("Arial Bold",25))
        self.correo = tk.Entry(master,width=40)
        self.etiqueta5.pack(ipadx=10)
        self.correo.pack()

```

```

        self.quitButton = tk.Button(master, text = 'CREAR PERSONAS', width = 25, command=self.crear_cliente)
        self.quitButton.pack()
        self.frame.pack()
    def crear_cliente(self):
        cursordb.execute("insert into clientes (id,nombre,apellido,telefono,correo) values (null,'" + self.nombre.get() + "','" + self.direccion.get() + "','" + self.telefono.get() + "','" + self.correo.get() + "')");
        print("SE EJECUTA CON EXITO")
    def recargar(self):
        print()

    def close_windows(self):
        self.master.destroy()
class Crearproducto:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.etiqueta = tk.Label(master, text="Crear Producto", font=("Arial Bold", 25))

        self.etiqueta.grid(row=0, column=2)
        self.etiqueta = tk.Label(master, text="Descripcion", font=("Arial Bold", 25))
        self.etiqueta.grid(row=1, column=1)
        self.descripcion = tk.Entry(master, width=40)
        self.descripcion.grid(row=1, column=2)

        self.etiqueta2 = tk.Label(master, text="Cantidad", font=("Arial Bold", 25))
        self.etiqueta2.grid(row=2, column=1)
        self.cantidad = tk.Entry(master, width=40)
        self.cantidad.grid(row=2, column=2)

        self.etiqueta3 = tk.Label(master, text="Stock", font=("Arial Bold", 25))
        self.etiqueta3.grid(row=3, column=1)
        self.stock = tk.Entry(master, width=40)
        self.stock.grid(row=3, column=2)

        self.etiqueta4 = tk.Label(master, text="Precio", font=("Arial Bold", 25))
        self.etiqueta4.grid(row=4, column=1)
        self.precio = tk.Entry(master, width=40)
        self.precio.grid(row=4, column=2)

        self.boton = tk.Button(master, text = 'CREAR PRODUCTO', width = 25, command=self.crear)
        self.boton.grid(row=5, column=3)
        self.boton2 = tk.Button(master, text = 'ACTUALIZAR PRODUCTO', width = 25)
        self.boton2.grid(row=5, column=4)
    def crear(self):
        pre = float(str(self.precio.get()))
        cursordb.execute("insert into productos (idpro,descripcion,cantidad,stock,precio) values(null,'" + str(self.descripcion.get()) + "','" + self.cantidad.get() + "','" + self.stock.get() + "','" + str(pre) + "')")
        cursordb.execute("SELECT * FROM productos")
        print(cursordb.fetchall())

class Cuentacredito:
    def __init__(self, master):
        self.master = master

```

```

self.frame = tk.Frame(self.master)
menu = tk.Menu(master)
submenu = tk.Menu(menu)
submenu.add_command(label='Ver Cuenta',font=("Arial Bold",25),command = self
.ver_cuenta)
submenu.add_separator()
menu.add_cascade(label='GESTION CUENTA',menu=submenu)

master.config(menu=menu)

self.etiqueta = tk.Label(master, text="Se crea la cuenta",font=("Arial Bold"
,25))
self.etiqueta.pack()

self.tipocuentat = tk.Label(master, text="Fecha Registro",font=("Arial Bold"
,20))
self.tipocuenta = tk.Entry(master,width=40)
self.tipocuenta.insert(0,str(datetime.datetime.now()))
self.tipocuentat.pack(ipadx=10)
self.tipocuenta.pack()

self.clientest = tk.Label(master,text="Seleccionar cliente")
self.clientes = ttk.Combobox(master,state="readonly")
    #Se lista los clientes
cursordb.execute("SELECT * FROM clientes")
datos = cursordb.fetchall()
#for d in datos:
self.clientes["values"] = datos
self.clientest.pack(ipadx=10)
self.clientes.pack()

self.numerot = tk.Label(master, text="Seleccione tipo Cuenta",font=("Arial Bo
ld",20))
self.numero = ttk.Combobox(master,state="readonly")

self.numerot.pack(ipadx=10)
tipos = ["CREDITO","AHORRO"]
self.numero["values"] = tipos
self.numero.pack()

self.quitButton = tk.Button(self.frame, text = 'CREAR CUENTA', width = 25, c
ommand = self.crear_cuenta)
self.quitButton.pack()
self.frame.pack()

def ver_cuenta(self):

    self.ver_cuenta = tk.Toplevel(self.master)
    self.app = Vercuenta(self.ver_cuenta)

def crear_cuenta(self):
    cursordb.execute("insert into cuentas (idc,tipofecha) values (null,'" +self.nu
mero.get()+ "','"+str(datetime.datetime.now())+"')");
    cursordb.execute("SELECT MAX(idc) FROM cuentas")
    codigocu = cursordb.fetchall()
    codigocuen = str(codigocu[0]).replace("'", "").replace("(", "").replace(",","").
replace(")","")
    cursordb.execute("insert into detallescuentas (iddc,idcuenta,fecha,detalle,deb
ito,credito,saldo) values (null,'" +codigocuen+ "','"+str(datetime.datetime.now())+'','Se
crea la cuenta','0.00','0.00','0.00')");
    cursordb.execute("insert into intermediacuecli (idi,idcliente,idcuenta) values

```

```

(null, '"+str(self.clientes.get()[0])+'', '"+codigocuen+'')");
    cursordb.execute("SELECT * FROM cuentas")
    print(cursordb.fetchall())
    cursordb.execute("SELECT * FROM detallescuentas")
    print(cursordb.fetchall())
    cursordb.execute("SELECT idcliente FROM intermediacuecli")
    codigocliente = cursordb.fetchall()
    codigocli = str(codigocliente).replace("(", "").replace(",", "").replace(")", "")
    print(codigocli)
    global codigoclig
    codigoclig = codigocli
    print(codigoclig)
    def close_windows(self):
        self.master.destroy()

class Vercuenta:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        print(codigoclig[1])
        cursordb.execute("SELECT idcuenta FROM intermediacuecli WHERE idcliente='"+s
tr(codigoclig[1])+"'")
        clienteid = cursordb.fetchall()
        self.codigoc = str(clienteid).replace("(", "").replace(",", "").replace(")", ""
)
        cursordb.execute("SELECT fecha,detalle,debito,credito,saldo FROM detallescue
ntas where idcuenta='"+str(self.codigoc[1])+"'")
        lista = cursordb.fetchall()

        # code for creating table
        for i in range(len(lista)+1):
            if i==0:
                self.etiqueta = tk.Label(master, text="Fecha Tramite", fg='red', font=(
"Arial Bold",25))
                self.etiqueta.grid(row=0, column=0)
                self.etiqueta1 = tk.Label(master, text="Detalle", fg='red', font=("Arial
Bold",25))
                self.etiqueta1.grid(row=0, column=1)
                self.etiqueta2 = tk.Label(master, text="Debito", fg='red', font=("Arial
Bold",25))
                self.etiqueta2.grid(row=0, column=2)
                self.etiqueta3 = tk.Label(master, text="Credito", fg='red', font=("Arial
Bold",25))
                self.etiqueta3.grid(row=0, column=3)
                self.etiqueta4 = tk.Label(master, text="Saldo", fg='red', font=("Arial B
old",25))
                self.etiqueta4.grid(row=0, column=4)

                self.etiqueta5 = tk.Label(master, text="Ingrese Cantidad", fg='black', f
ont=("Arial Bold",25))
                self.etiqueta5.grid(row=0, column=5)
                self.ingreso = tk.Entry(master,width=40)
                self.ingreso.grid(row=0, column=6)
                self.boton = tk.Button(master, text='Deposito', width = 25,command=sel
f.depositar)
                self.boton.grid(row=0, column=7)

                for j in range(len(lista[0])):
                    if i!=0:
                        self.e = tk.Entry(master, width=20, fg='blue', font=('Arial',10,'bo
ld'))

```

```

        self.e.insert(tk.END, lista[i-1][j])
        self.e.grid(row=i, column=j)

    def close_windows(self):
        self.master.destroy()
    def depositar(self):
        #where idcuenta='' + str(self.codigoc[1]) + '' and
        cursordb.execute("SELECT saldo FROM detallescuentas where idcuenta='' + str(self.
codigoc[1]) + '' and iddc=(SELECT MAX(iddc) FROM detallescuentas)")
        saldo = cursordb.fetchall()
        print(saldo)
        saldo1 = str(saldo).replace("(", "").replace(",", "").replace(")", "").replace("'",
, "").replace("[", "").replace("]", "")
        print("esto es " + saldo1)
        saldot = float(saldo1)
        igresod = float(str(self.ingreso.get()))
        totalsaldo = saldot + igresod
        print(totalsaldo)
        cursordb.execute("insert into detallescuentas (iddc,idcuenta,fecha,detalle,debi
to,credito,saldo) values (null,''' + str(self.codigoc[1]) + ''', ''' + str(datetime.datetime.now
()) + ''', 'DEPOSITO', '0.00', ''' + str(igresod) + ''', ''' + str(totalsaldo) + ''')");
        cursordb.execute("SELECT * FROM detallescuentas")
        saldo1 = ""
        print(cursordb.fetchall())
        cursordb.execute("SELECT fecha,detalle,debito,credito,saldo FROM detallescuenta
s where idcuenta='' + str(self.codigoc[1]) + ''")
        lista = cursordb.fetchall()

        # code for creating table
        for i in range(len(lista)+1):
            if i==0:
                self.etiqueta = tk.Label(self.master, text="Fecha Tramite", fg='red', font=
("Arial Bold", 25))
                self.etiqueta.grid(row=0, column=0)
                self.etiqueta1 = tk.Label(self.master, text="Detalle", fg='red', font=("Ari
al Bold", 25))
                self.etiqueta1.grid(row=0, column=1)
                self.etiqueta2 = tk.Label(self.master, text="Debito", fg='red', font=("Aria
l Bold", 25))
                self.etiqueta2.grid(row=0, column=2)
                self.etiqueta3 = tk.Label(self.master, text="Credito", fg='red', font=("Ari
al Bold", 25))
                self.etiqueta3.grid(row=0, column=3)
                self.etiqueta4 = tk.Label(self.master, text="Saldo", fg='red', font=("Arial
Bold", 25))
                self.etiqueta4.grid(row=0, column=4)

                self.etiqueta5 = tk.Label(self.master, text="Ingreso Cantidad", fg='black'
, font=("Arial Bold", 25))
                self.etiqueta5.grid(row=0, column=5)
                self.ingreso = tk.Entry(self.master, width=40)
                self.ingreso.grid(row=0, column=6)
                self.boton = tk.Button(self.master, text='Deposito', width = 25, command=s
elf.depositar)
                self.boton.grid(row=0, column=7)

            for j in range(len(lista[0])):
                if i!=0:
                    self.e = tk.Entry(self.master, width=20, fg='blue', font=('Arial', 10,
'bold'))
                    self.e.insert(tk.END, lista[i-1][j])

```



```

        self.e.grid(row=i, column=j)

class Crearpedido:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.etiqueta = tk.Label(self.master, text="SE CREA PEDIDOS", fg='red', font=(
"Arial Bold", 25))
        self.etiqueta.grid(row=0, column=2)
        self.etiqueta1 = tk.Label(self.master, text="Fecha Pedido", fg='black', font=(
"Arial Bold", 25))
        self.etiqueta1.grid(row=1, column=1)
        self.fecha = tk.Entry(master, width=40)
        self.fecha.insert(0, str(datetime.datetime.now()))
        self.fecha.grid(row=1, column=2)
        self.etiqueta2 = tk.Label(self.master, text="Seleccionar Producto", fg='black',
font=("Arial Bold", 25))
        self.etiqueta2.grid(row=2, column=1)
        self.productos = ttk.Combobox(self.master, state="readonly")
        cursordb.execute("SELECT idpro, descripcion, precio FROM productos")
        self.productos["values"] = cursordb.fetchall()
        self.productos.grid(row=2, column=2)
        self.etiqueta3 = tk.Label(self.master, text="Ingrese Cantidad", fg='black', fo
nt=("Arial Bold", 25))
        self.etiqueta3.grid(row=3, column=1)
        self.cantidad = tk.Entry(master, width=40)
        self.cantidad.grid(row=3, column=2)

        self.area = tk.Text(master, height=12, width=80, wrap=tk.WORD)
        self.area.grid(row=4, column=3)
        self.etiqueta4 = tk.Label(self.master, text="TOTAL:", fg='black', font=("Arial B
old", 25))
        self.etiqueta4.grid(row=3, column=3)

        self.costo = tk.Label(self.master, text="0.00", fg='black', font=("Arial Bold",
25))
        self.costo.grid(row=3, column=4)

        self.boton = tk.Button(master, text = 'CREAR PEDIDO', width = 25)
        self.boton.grid(row=1, column=3)
        self.boton1 = tk.Button(master, text = 'AGREGAR PEDIDO', width = 25, comman=s
elf.agregar_pedido)
        self.boton1.grid(row=2, column=3)
        self.totales = '0.00'

    def close_windows(self):
        self.master.destroy()

    def agregar_pedido(self):
        #, descripcion TEXT, c
osto TEXT, idinter INTEGER, idpro INTEGER
        costot = float(str(self.costo.cget("text")))
        print(costot)
        print(str(self.productos.get()[0]))
        productoid = str(self.productos.get()[0])
        cursordb.execute("SELECT precio from productos where idpro='"+productoid+"'")
        )
        preciop = cursordb.fetchall()
        productop = str(preciop).replace("(", "").replace(",", "").replace(")", "").rep
lace("'", "").replace("[", "").replace("]", "")

```



```

        cursordb.execute("SELECT idi FROM intermediacuecli WHERE idcliente='"+str(codigoclig[1])+"'")
        total = float(productop) * int(self.cantidad.get())
        self.totales +=total
        self.costo =tk.Label(self.master,text=str(total),fg='black',font=("Arial Bold",25))
        self.costo.grid(row=3,column=4)

        clienteid = cursordb.fetchall()[0][5]
        self.codigoc = str(clienteid).replace("(", "").replace(", ", "").replace(")", "")

    )

    cursordb.execute("insert into pedidos (idp,descripcion,costo,idinter,idpro,cantidad) values (null,'"+str(self.fecha.get())+"', '"+str(costot)+"', '"+str(self.codigoc[1])+"', '"+str(self.productos.get()[0])+"', '"+str(self.cantidad.get())+"')")
    cursordb.execute("SELECT * FROM pedidos")
    self.area.insert(tk.END, str(cursordb.fetchall()))
    print(cursordb.fetchall())

    self.etiqueta2 = tk.Label(self.master,text="Seleccionar Producto",fg='black',font=("Arial Bold",25))
    self.etiqueta2.grid(row=2,column=1)
    self.productos = ttk.Combobox(self.master,state="readonly")
    cursordb.execute("SELECT idpro,descripcion,precio FROM productos")
    self.productos["values"] = cursordb.fetchall()
    self.productos.grid(row=2,column=2)
    self.etiqueta3 = tk.Label(self.master,text="Ingrese Cantidad",fg='black',font=("Arial Bold",25))
    self.etiqueta3.grid(row=3,column=1)
    self.cantidad = tk.Entry(self.master,width=40)
    self.cantidad.grid(row=3,column=2)

class Gestionpedido:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.nombre = tk.Entry(master,width=40)
        self.nombre.pack()
        self.cantidad = tk.Entry(master,width=40)
        self.cantidad.pack()
        self.valor = tk.Entry(master,width=40)
        self.valor.pack()
        self.estado = tk.Entry(master,width=40)
        self.estado.pack()
        self.quitButton = tk.Button(self.frame, text = 'GESTION PEDIDO', width = 25,
command = self.close_windows)
        self.quitButton.pack()
        self.frame.pack()

    def close_windows(self):
        self.master.destroy()

class Descuentocuenta:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.quit.etiqueta = tk.Label(master, text="Mensaje para los usuarios",font=
("Arial Bold",25))

```

```

        self.quit.etiqueta.pack()
        self.numero = tk.Entry(master,width=40)
        self.numero.pack()
        self.tipocuenta = tk.Entry(master,width=40)
        self.tipocuenta.pack()
        self.saldo = tk.Entry(master,width=40)
        self.frame.pack()

    def close_windows(self):
        self.master.destroy()

#CLIENTE QUE TIENEN DIERO HACEN PEDIDOS

#REALIZAR UN PEDIDO UN CLIENTE PUEDE AGRUPARLOS EN PEDIDOS SIMPLEZ O COMPUESTOS

#PEDIDOS SIMPLES QUE COMPONEN A PEDIDOS COMPUESTOS
#SE HACEN EL PAGO CON CUENTAS DEL MISMO CLIENTE
#SOLO ES POSIBLE REALIZAR PETICIONES DEL PRODUCTOS STOCK.


#MAS VENTANAS
#Gestion pago
#Cobro ,orden de distribuicion y confirmacion de pedidos

#comprobar todos los pedidos pendientes de cobro..

def main():
    plantilla = tk.Tk()
    plantilla.title("Esta es mi APP")
    plantilla.geometry('650x600')
    app = Ventana(plantilla)
    plantilla.mainloop()

if __name__ == '__main__':
    main()

```

```

SE CREA LA TABLA LOS CLIENTES
SE CREA LA TABLA LOS CUENTAS
SE CREA LA TABLA LOS DETALLES CUENTAS
SE CREA LA TABLA LOS INTERMEDIA CUENTAS Y CLIENTE
SE CREA LA TABLA LOS PRODUCTOS
SE CREA LA TABLA LOS PEDIDOS

```

In [ ]:

In [ ]:

In [ ]:

In [ ]: