**Ejercicios Python - 17 - CRUD Básico 3 SUPERMERCADO**

## **Ejercicios de Python**

### **Ejercicio: CRUD Básico sobre la Tabla Clientes**

El objetivo de este ejercicio es ampliar nuestra aplicación en Python que permita realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la tabla clientes de la base de datos SUPERMERCADO, además de sobre la tabla categorías que ya tenemos hecho. La aplicación debe funcionar a través de dos **menús de opciones**, donde el usuario puede seleccionar primero la tabla sobre la que quiere operar y después la operación deseada de esa tabla.

[GIT](https://github.com/IgnacioAriasCampusfp/Programacion_Ignacio_Arias/tree/EjerciciosPython/Base_de_Datos/Ej17)

MenuCliente.py

from FuncionesCliente import DOW\_Cliente as dw

from colorama import Fore as fr

def Menu(cursor,conexion):

while True:

try:

print(fr.BLUE+"=== Gestión de Clientes ===\nSeleccione una opción:\n1. Crear nuevo cliente\n2. Leer cliente existente\n3. Actualizar un cliente\n4. Eliminar un cliente\n5. Salir"+fr.RESET)

opcion = int(input("> "))

match opcion:

case 1:

dw.Create(cursor,conexion)

case 2:

dw.List(cursor,conexion)

case 3:

dw.Actu(cursor,conexion)

case 4:

dw.Delete(cursor,conexion)

case 5:

print(fr.LIGHTRED\_EX+"Volviendo al Menu Prinicpal"+fr.RESET)

break

case \_:

print("Opción no valida")

except Exception as e:

print(f"Ocurrió un error inesperado: {e}")

except ValueError as e:

print(f"No has ingresado una opción valida: {e}")

DOW\_Cliente.py

from colorama import Fore as fr

def Create(cursor, conexion):

try:

idcliente = input(fr.LIGHTCYAN\_EX + "Ingrese el ID del cliente: " + fr.RESET)

cia = input(fr.LIGHTCYAN\_EX + "Ingrese el cia del cliente: " + fr.RESET).strip()

contacto = input(fr.LIGHTCYAN\_EX + "Ingrese el nombre del cliente: " + fr.RESET)

cargo = input(fr.LIGHTCYAN\_EX + "Ingrese el cargo del cliente: " + fr.RESET)

direccion = input(fr.LIGHTCYAN\_EX + "Ingrese la direccion del cliente: " + fr.RESET)

ciudad = input(fr.LIGHTCYAN\_EX + "Ingrese la ciudad del cliente: " + fr.RESET)

region = input(fr.LIGHTCYAN\_EX + "Insgrese la región del cliente: "+ fr.RESET)

cp = int(input(fr.LIGHTCYAN\_EX +"Ingrese el codigo postal del cliente: "+ fr.RESET))

pais = input(fr.LIGHTCYAN\_EX +"Ingrese el pais del cliente: "+ fr.RESET)

tlf = int(input(fr.LIGHTCYAN\_EX +"Ingrese el telefono del cliente: "+ fr.RESET))

fax = int(input(fr.LIGHTCYAN\_EX +"Ingrese el numero de FAX del cliente: " + fr.RESET))

consulta = """INSERT INTO cliente (idcliente, cia, contacto, cargo, direccion, ciudad, region, cp, pais, tlf, fax) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"""

cursor.execute(consulta, (idcliente, cia, contacto, cargo, direccion, ciudad, region, cp, pais, tlf, fax))

conexion.commit()

print(fr.GREEN + f"El cliente '{cia}' ha sido creado con éxito." + fr.RESET)

print(fr.CYAN + "Listado de Clientes:" + fr.RESET)

consulta = "SELECT \* FROM cliente ORDER BY idcliente;"

cursor.execute(consulta)

resultados = cursor.fetchall()

for idcliente, cia, contacto, cargo, direccion, ciudad, region, cp, pais, tlf, fax in resultados:

print(fr.YELLOW + f"--->ID: {idcliente} Cia: {cia} Contacto: {contacto}, Cargo: {cargo},\nDireccion: {direccion} Ciudad: {ciudad} Region: {region} Cp: {cp} Pais: {pais} Tlf: {tlf} Fax: {fax}" + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al crear un cliente: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)

def List(cursor, conexion):

try:

consulta = "SELECT \* FROM cliente;"

cursor.execute(consulta)

print(fr.CYAN + "Listado de Clientes:" + fr.RESET)

resultados = cursor.fetchall()

for idcliente, cia, contacto, cargo, direccion, ciudad, region, cp, pais, tlf, fax in resultados:

print(fr.YELLOW + f"--->ID: {idcliente} Cia: {cia} Contacto: {contacto}, Cargo: {cargo},\nDireccion: {direccion} Ciudad: {ciudad} Region: {region} Cp: {cp} Pais: {pais} Tlf: {tlf} Fax: {fax}" + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al listar clientes: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)

def Actu(cursor, conexion):

try:

idcliente = input(fr.LIGHTCYAN\_EX + "Ingrese el ID del cliente: " + fr.RESET)

cia = input(fr.LIGHTCYAN\_EX + "Ingrese el cia del cliente: " + fr.RESET).strip()

contacto = input(fr.LIGHTCYAN\_EX + "Ingrese el nombre del cliente: " + fr.RESET)

cargo = input(fr.LIGHTCYAN\_EX + "Ingrese el cargo del cliente: " + fr.RESET)

direccion = input(fr.LIGHTCYAN\_EX + "Ingrese la direccion del cliente: " + fr.RESET)

ciudad = input(fr.LIGHTCYAN\_EX + "Ingrese la ciudad del cliente: " + fr.RESET)

region = input(fr.LIGHTCYAN\_EX + "Insgrese la región del cliente: "+ fr.RESET)

cp = int(input(fr.LIGHTCYAN\_EX +"Ingrese el codigo postal del cliente: "+ fr.RESET))

pais = input(fr.LIGHTCYAN\_EX +"Ingrese el pais del cliente: "+ fr.RESET)

tlf = int(input(fr.LIGHTCYAN\_EX +"Ingrese el telefono del cliente: "+ fr.RESET))

fax = int(input(fr.LIGHTCYAN\_EX +"Ingrese el numero de FAX del cliente: " + fr.RESET))

consulta = """

UPDATE cliente SET cia = %s, contacto = %s, cargo = %s, direccion = %s, ciudad = %s, region = %s, cp = %s, pais = %s, tlf = %s, fax = %s WHERE idcliente = %s

"""

cursor.execute(consulta, (cia, contacto, cargo, direccion, ciudad, region, cp, pais, tlf, fax, idcliente))

conexion.commit()

print(fr.GREEN + f"El cliente '{cia}' ha sido actualizada con éxito." + fr.RESET)

print(fr.CYAN + "Listado de Clientes Actualizado:" + fr.RESET)

consulta = "SELECT \* FROM cliente ORDER BY idcliente;"

cursor.execute(consulta)

resultados = cursor.fetchall()

for idcliente, cia, contacto, cargo, direccion, ciudad, region, cp, pais, tlf, fax in resultados:

print(fr.YELLOW + f"--->ID: {idcliente} Cia: {cia} Contacto: {contacto}, Cargo: {cargo},\nDireccion: {direccion} Ciudad: {ciudad} Region: {region} Cp: {cp} Pais: {pais} Tlf: {tlf} Fax: {fax}" + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al actualizar un cliente: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)

def Delete(cursor, conexion):

try:

idcliente = input(fr.LIGHTCYAN\_EX + "Ingrese el ID del cliente que desea eliminar: " + fr.RESET)

consulta = "DELETE FROM cliente WHERE idcliente = %s"

cursor.execute(consulta, (idcliente,))

conexion.commit()

print(fr.GREEN + f"El cliente con ID '{idcliente}' ha sido eliminado con éxito." + fr.RESET)

except ValueError as e:

print(fr.RED + f"Error al eliminar un cliente: {e}" + fr.RESET)

except Exception as e:

print(fr.RED + f"Ocurrió un error inesperado: {e}" + fr.RESET)