

# HITO 2 DEL 1º TRIMESTRE DE PROGRAMACIÓN

Alejandro Cortés Díaz

## Índice

<b>Enlace a GitHub .....</b>	<b>2</b>
<b>Fase 1. ....</b>	<b>3</b>
<b>Fase 2. ....</b>	<b>6</b>
<b>Archivo donde ejecuto; .....</b>	<b>6</b>
<b>Archivo conexión padre: .....</b>	<b>7</b>
<b>Archivo conexión hijo; .....</b>	<b>7</b>
<b>Menú Main;.....</b>	<b>8</b>
<b>Menú cliente;.....</b>	<b>10</b>
<b>Menú producto; .....</b>	<b>11</b>
<b>DAO CLIENTE; .....</b>	<b>12</b>
<b>DAO producto; .....</b>	<b>19</b>
<b>Bibliografía .....</b>	<b>29</b>

## Enlace a GitHub

<https://github.com/Cortes-cmd/Programacion.git>

## Fase 1.

En este primer paso, se van a definir los pasos necesarios para resolver el registro de un nuevo cliente y cómo realizar una compra, así como diseñar para cada uno de ellos el diagrama de flujo correspondiente. En este primer paso, no es necesario desarrollar la aplicación en ningún lenguaje de programación. Se trata en todo caso, de analizar la organización y métodos de resolución de la aplicación.

-Para resolver el problema planteado se propuso el siguiente modelo de código;

-Primero, establecí una conexión a una base de datos de tipo SQL para resolver el problema, me resultó un acercamiento más realista al tratamiento de los clientes a una base de datos, así como más interactiva.

Poder visualizar los cambios realizados en la misma base de datos, me cuidó de errores que, de otra forma, me hubiera costado más tiempo encontrar, además, prefiero tener un estímulo visual más claro de mis cambios.

Para la realización de este paso, es necesario ser conocedor de las librerías pertinentes a las que debemos acceder para realizar esta comunicación, construir ese puente, esto nos lo permite la funcionalidad `mysql.connector`, que será empleada en el código para rellenar los campos necesarios, pudiendo acceder al sistema con un usuario en particular.

También requeriremos de otra funcionalidad para manipular información de la base de datos, este es al caso de `cursor()`, que nos permitirá ejecutar las sentencias SQL necesarias, para poder implementar los cambios que deseemos en la misma BD. Será implementada esta herramienta, en el archivo “Conexión”.

-Después, decidí implementar un menú principal que me permitiera acceder a cada tabla y, posteriormente, en su propio menú, poder seleccionar los cambios que deseo llevar a cabo.

Para este objetivo, fue necesario llamar, en el archivo del ejercicio a la función, llamada “Menu”, usando la importación del archivo donde se encuentra el menú principal como “Mm”, resultando la llamada en “Mm.Menu”, e introduciendo, a continuación conexión y cursor, pues, como se ha comentado en el párrafo previo, ambos serán necesarios, para conectarse a la base de datos, y para operar con los datos encontrados en la misma.

Dentro de este menú principal, o “Main Menu”, proporciono acceso a las tablas “cliente”, y “producto”, llamando a las funciones de aquellos menús correspondientes, así como hice en su momento con el menú main, mas, esta vez, desde el propio “Main Menu”, lo que me brinda acceso a las tablas

Cabe destacar, que dentro de la propia función Menu, tanto en el menú Main, como en los menús de cada tabla, hago uso de “conexión” y de “cursor”, pues, como se ha explicado, son necesarias para poder conectarme a la base de datos, y hacer uso de las sentencias SQL.

-Una vez introducido a cualquiera de los menús, la estructura sigue el mismo proceso de llamado a las funciones de los distintos archivos, donde se albergan las funcionalidades disponibles para cada tabla.

En el caso del Menú cliente;

\*Se permite seleccionar una de las opciones disponibles para manipular datos dentro de la tabla clientes (insertar cliente, seleccionar todos los clientes, seleccionar cliente, y hacer seguimiento del cliente).

Un dato relevante a destacar en este menú, es que únicamente utilizo “conexion”, en “Insertar cliente”, esto se debe a que para las consultas de tipo “select”, no es necesario utilizar “conexion”, puesto que no se van a realizar cambios en la base de datos, sino que vas a consultar información.

En el caso del Menú producto;

\*Se permite seleccionar una de las opciones disponibles para manipular datos dentro de la tabla producto (comprar producto, seleccionar todos los productos, y seleccionar producto)

Las funciones reunidas en cada archivo DAO, con respecto a cada tabla necesitada realizan siguientes operaciones;

Con respecto a cliente;

- Registro de cliente via función `ingresar_cliente(conexion,cursor)`
- Visualizar todos los clientes registrados via función `select_clientes(cursor)`
- Seleccionar búsqueda de un solo cliente via función `select_cliente(cursor)`
- Seguimiento de compras de un cliente a través de la id del mismo via función `seguimiento_cliente(cursor)`

Con respecto al producto;

- Realizar compra de producto via función `compra_producto(conexion,cursor)`
- Seleccionar todos los productos disponibles via función `selet_productos(cursor)`
- Sacar el precio de un solo producto del interés del usuario por medio de la id via función `select_producto(cursor)`

## Fase 2.

A continuación, nos dedicamos a la implementación de la aplicación. En este punto, sí es necesario utilizar Python para su desarrollo, usando las estructuras de datos que consideres más adecuadas. Debemos tener presente que no se nos pide realizar el aplicativo visual, únicamente la funcionalidad de cada caso. La selección de cada una de las funcionalidades se realizará mediante el uso de un menú.

### Archivo donde ejecuto;

Ejecuto el archivo con los archivos “Conexión”, “Menu\_Main (el menú principal)”, y llamo a conexión y cursor de estos mismos archivos, así como, de Menu\_Main, la función menú, cuya utilidad se explica en su correspondiente captura.

```
# IMPORTO CONEXION
import Conexion as Cx

# IMPORTO MAIN MENU
import Menu_Main as Mm

# ASIGNO CONEXION A CONEXION PARA UTILIZAR SUS FUNCIONES CON LA BASE DE
DATOS
conexion = Cx.conexion

# ASIGNO CURSOR A CURSOR PARA HACER EXECUTES
cursor = Cx.cursor

# LLAMO A MENU MAIN PARA PODER USAR CURSOR Y CONEXION EN EL
Mm.Menu(conexion, cursor)
```

## Archivo conexión padre:

En este archivo inicio sesión en mysql.

```
import mysql.connector
def conectar (BaseDatos):

    # Establecer conexión con la base de datos
    conexion = mysql.connector.connect(
        host="localhost",      # Dirección del servidor (localhost
                                para base de datos local)
        user="root",           # Usuario de la base de datos
        password="curso",      # Contraseña del usuario
        database=BaseDatos     # Nombre de la base de datos
    )
    return conexion
```

## Archivo conexión hijo;

Vía el archivo conexión, importando el anterior con la sesión iniciada, especifico la conexión a la base de datos del hito, y llamo a cursor para ejecutar sentencias.

```
#IMPORTO DESDE CONEXIÓN PADRE LAS FUNCIONES DE CONEXION
import Conexion_Padre as bdd
#USO CONEXIÓN PARA CONECTARME A BD
conexion=bdd.conectar("SUPERMERCADOPython")
#EMPLEO CURSOR PARA EJECUTAR
cursor= conexion.cursor()
```



## Menú Main;

Aquí muestro las opciones posibles de las respectivas tablas de las que se permite acceso en el archivo, “Producto” y “Cliente”. Por ello importo los menús de las tablas correspondientes.

```
#IMPORTO LOS DOS MENUS PARA ENLAZARLOS AL MAIN
import Menu_Producto as Mp

import Menu_Cliente as Mc
#DEF MENU
def Menu (conexion,cursor):
    try:
        while True:
            #MUESTRO OPCIONES Y ASIGNO CADA UNA A CADA CASO POSIBLE
            opcion=int(input("Menú de opciones\n1/Tabla Producto\n2/Tabla
Cliente\n3/Salir\n"))

            match opcion:
                case 1:
                    print("Marchando al menu de Producto")
                    Mp.Menu(conexion,cursor)
                case 2:
                    print("Marchando al menu de Cliente")
                    Mc.Menu(conexion,cursor)
                case 3:
                    print("Saliendo del programa")
                    cursor.close()
                    conexion.close()
                    break

    except ValueError as e:
        #ERROR SI TIPO DE DATO ERRONEO
        print("No aplica ese valor, error{e}")
```

```
except Exception:  
    #ERROR SI FALLA ALGUNA COSA DISTINTA  
    print("Error al ejecutar la funcion")
```

## Menú cliente;

En este menú la finalidad es la misma, entregar las distintas opciones. En este caso; Ingresar cliente, seleccionar todos los clientes, seleccionar clientes de manera individual, y hacer un seguimiento de los pedidos del cliente

```
# IMPORTO CLIENTE PARA USARLO EN EL MENU
import DAW_Cliente as dwC

# DEF MENU
def Menu(conexion, cursor):
    try:
        while True:
            # MUESTRO OPCIONES
            opcion = int(input("Menú de opciones \n1/Insertar nuevo
cliente\n2/Seleccionar todos los clientes\n3/Seleccionar un cliente en
específico\n4/Seguimiento de cliente \n5/Salir al menu principal"))

            # ASIGNO UNA OPCION A CADA FUNCION CON MATCH CASE
CORRESPONDIENTE
            match opcion:
                case 1:
                    print("Insertar cliente")
                    dwC.ingresar_cliente(conexion, cursor)
                case 2:
                    print("Seleccionar a todos los clientes")
                    dwC.select_clientes(cursor)
                case 3:
                    print("Seleccionar cliente")
                    dwC.select_cliente(cursor)
                case 4:
                    print("Seguimiento de cliente")
                    dwC.seguimiento_cliente(cursor)
                case 5:
                    print("Marchando al menú principal")
                    break

            #ERROR SI TIPO DE DATO NO ES CORRECTO
        except ValueError as e:
            print(f"No aplica ese valor, error {e}")
        except Exception:
            #ERROR SI SE DA OTRO TIPO DE FALLO
            print("Error al ejecutar la función")
```

## Menú producto;

Lo mismo aplica a este menú, las funcionalidades a elegir son; Comprar producto, seleccionar todos los productos, y seleccionar un único producto.

```
# IMPORTO DOW PRODUCTO PARA USARLO EN LOS DISTINTOS CASES
import DAW_Producto as dwp

# DEFINO MENU
def Menu(conexion, cursor):
    try:

        while True:
            # OPCIONES
            opcion = int(input("Menú de opciones \n1/Comprar
producto\n2/Seleccionar todos los productos\n3/Seleccionar un producto en
particular\n4/Salir"))

            # SEGUN LA OPCION UNA FUNCION PARA LA NECESIDAD
CORRESPONDIENTE
            match opcion:
                case 1:
                    print("Comprar producto")
                    dwp.compra_producto(conexion, cursor)
                case 2:
                    print("Seleccionar todos los productos")
                    dwp.select_productos(cursor)
                case 3:
                    print("Seleccionar producto")
                    dwp.select_producto(cursor)
                case 4:
                    print("Marchando al menu principal")
                    break

    except ValueError as e:
        # ERROR SI EL TIPO DE DATO NO ES VALIDO
        print(f"No aplica ese valor, error {e}")
    except Exception:
        # ERROR PARA CUALQUIER OTRO TIPO DE FALLO
        print("Error al ejecutar la funcion")
```

## DAO CLIENTE;

La función `ingresar_cliente`; Introduce los datos solicitados del cliente en la tabla cliente (`idcliente` es `auto_increment`), puesto que `idcliente` es `auto_increment`, lo recojo en base a los datos solicitados para construir su perfil, y le comunico el id que se le ha asignado desde la base de datos. Después, introduzco todos esos datos del perfil del cliente en un archivo txt, previamente manejados por un array.

La función `select_clientes`; Emplea una consulta sql para seleccionar todos los clientes ordenándolos por su id específico, los datos asociados a esa id.

La función `select_cliente`; Pide el `idcliente` para mostrar los datos asociados al mismo (nombre, ciudad, y teléfono).

La función `seguimiento_cliente`; Realiza una consulta sql, y en base a un `idcliente` ya solicitado. Une las tablas correspondientes para mostrar todos los datos del cliente, el `idpedido` del pedido o pedidos que haya realizado, y de cada pedido, muestra; El nombre, el precio del producto, y la cantidad comprada de este.

```
#DEF FUNCION PARA INTRODUCIR CLIENTE A BASE DE DATOS
```

```
def ingresar_cliente(conexion, cursor):
```

```
    try:
```

```
        # SOLICITO ATRIBUTOS DE LA TABLA CLIENTE
```

```
        Nombre = input("Ingrese su nombre: ")
```

```
        ciudad = input("Dime la ciudad del nuevo cliente: ")
```

```
        tlf = input("Ingrese su número de teléfono: ")
```

```
        # LOS INSERTO EN SENTENCIA SQL Y EJECUTO
```

```

        consulta = "INSERT INTO cliente(Nombre, ciudad, tlf) VALUES ( %s,
%s, %s)"

        cursor.execute(consulta, ( Nombre, ciudad, tlf))

        conexion.commit()

        #EXTRAIGO EL ID EN BASE A LOS DATOS INTRODUCIDOS PARA SACAR EL ID
Y MOSTRÁRSELO POSTERIORMENTE

        consulta="select idcliente from cliente where Nombre=%s and
ciudad=%s and tlf=%s"

        cursor.execute(consulta,(Nombre,ciudad,tlf))
        resultado=cursor.fetchall()

        for linea in resultado:
            idcliente=linea
            print(f"Posees id:{idcliente}")

        # GUARDO LA INSERCIÓN EN UN ARRAY, Y ESTE LO PRINTEO EN UN
ARCHIVO

        array = [ idcliente,Nombre, ciudad, tlf]
        with open("Clientes.txt", "a") as archivo:
            archivo.write("Datos insertados:\n\n")
            for dato in array:
                archivo.write(f"{dato}\n")

        print(f"Se insertó en la base de datos el cliente: {Nombre}\n con
el idcliente:{idcliente}\n de la ciudad: {ciudad}\n con el número de
teléfono:{tlf}")

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS
POSIBLES)

        except ValueError as e:
            print(f"Tipo de dato incorrecto, error {e}")

        except Exception as unknown:

```

```
print(f"No pudo ejecutarse la función, error {unknown}")
```

```
Marchando al menu de Cliente
Menú de opciones
1/Insertar nuevo cliente
2/Seleccionar todos los clientes
3/Seleccionar un cliente en específico
4/Seguimiento de cliente
5/Salir al menu principal
Insertar cliente
Ingrese su nombre: Paco
Dime la ciudad del nuevo cliente: PacoLandia
Ingrese su número de teléfono: 888
Posees id:(13,)
Se insertó en la base de datos el cliente: Paco
con el idcliente:(13,)
de la ciudad: PacoLandia
con el número de teléfono:888
```

```
67
68 (12,)
69 Alejandro
70 Madrid
71 723
72 Datos insertados:
73
74 (13,)
75 Paco
76 PacoLandia
77 888
78
```

10	Carlos	yuca	1252
11	Luca	Palermo	987...
12	Alejandro	Madrid	723
13	Paco	PacoLa...	888
NULL	NULL	NULL	NULL

```
# DEF FUNCION PARA SELECCIONAR TODOS LOS CLIENTES RECOGIDOS (SOLO HACE
FALTA CURSOR PORQUE ES SENTENCIA DE TIPO SELECT)
```

```
def select_clientes(cursor):
```

```
    try:
```

```
        # CONSULTA A TODOS LOS CLIENTES DE LA TABLA ORDENANDOLOS POR EL
ID A TRAVES DE SENTENCIA SQL
```

```
        consulta = "SELECT * FROM cliente ORDER BY idcliente"
```

```
        cursor.execute(consulta)
```

```

# PRINTEO A LOS CLIENTES RECOGIDOS

resultado = cursor.fetchall()

for fila in resultado:

    print(f"IDCliente: {fila[0]}, Nombre: {fila[1]}, Ciudad: {fila[2]}, Teléfono: {fila[3]}")

#LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)

except ValueError as e:

    print(f"Tipo de dato incorrecto, error {e}")

except Exception as unknown:

    print(f"No pudo ejecutarse la función, error {unknown}")

# FUNCION PARA SELECCIONAR CLIENTE INDIVIDUAL (SENTENCIA SELECT POR LO QUE NO HACE FALTA CONEXION)

# Regresar al menu principal
5/Salir al menu principal2
Seleccionar a todos los clientes
IDCliente: 1, Nombre: Juan, Ciudad: asd, Teléfono: 5346
IDCliente: 2, Nombre: jasd, Ciudad: alksd, Teléfono: 1234
IDCliente: 3, Nombre: asljkd, Ciudad: asd, Teléfono: 1234
IDCliente: 4, Nombre: wawa, Ciudad: wawawa, Teléfono: 99
IDCliente: 5, Nombre: Carlos, Ciudad: oaksnf, Teléfono: 123
IDCliente: 6, Nombre: carlos, Ciudad: sd, Teléfono: 313
IDCliente: 7, Nombre: carlos, Ciudad: dhno, Teléfono: 123
IDCliente: 8, Nombre: Iguana, Ciudad: Lagarto, Teléfono: 99
IDCliente: 9, Nombre: Carlos, Ciudad: guanaba, Teléfono: 5342
IDCliente: 10, Nombre: Carlos, Ciudad: yuca, Teléfono: 1252
IDCliente: 11, Nombre: Luca, Ciudad: Palermo, Teléfono: 987969
IDCliente: 12, Nombre: Alejandro, Ciudad: Madrid, Teléfono: 723
IDCliente: 13, Nombre: Paco, Ciudad: PacoLandia, Teléfono: 888
Menú de opciones

def select_cliente(cursor):

    try:

        # PIDO ID PARA SACAR SUS DATOS

        idcliente = int(input("Dime el ID del cliente que quieres seleccionar: "))

        # EXTRAIGO CADA DATO DEL CLIENTE CON EL ID DADO

```



```
        consulta = "SELECT * FROM cliente WHERE idcliente = %s ORDER BY  
idcliente"
```

```
        cursor.execute(consulta, (idcliente,))
```

```
        # LOS IMPRIMO
```

```
        resultado = cursor.fetchall()
```

```
        for fila in resultado:
```

```
            print(f"IDCliente: {fila[0]}, Nombre: {fila[1]}, Ciudad:  
{fila[2]}, Teléfono: {fila[3]}")
```

```
        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS  
POSIBLES)
```

```
        except ValueError as e:
```

```
            print(f"Tipo de dato incorrecto, error {e}")
```

```
        except Exception as unknown:
```

```
            print(f"No pudo ejecutarse la función, error {unknown}")
```

```
Dime el ID del cliente que quieres seleccionar: 12  
IDCliente: 12, Nombre: Alejandro, Ciudad: Madrid, Teléfono: 723  
Menú de opciones  
1/Insertar nuevo cliente  
2/Seleccionar todos los clientes  
3/Seleccionar un cliente en específico  
4/Seguimiento de cliente  
5/Salir al menu principal
```

```
# DEF FUNCION PARA MOSTRAR LOS DATOS DE LAS COMPRAS O PEDIDOS DE UN SOLO  
CLIENTE
```

```
def seguimiento_cliente(cursor):
```

```
    try:
```

```
        # PIDO SU ID
```

```
        idcliente = int(input("Introduce el ID del cliente para realizar  
un seguimiento: "))
```

```
        # INNER JOINS DE LAS TABLAS DE LAS QUE SELECCIONO TODOS LOS DATOS  
SOBRE EL CLIENTE Y SUS PEDIDOS
```

```

        consulta = """
            SELECT c.idcliente, c.Nombre, c.ciudad, c.tlf, p.idpedido,
            prod.nombre, d.precio, d.unidades
            FROM cliente c
            JOIN pedido p ON c.idcliente = p.idcliente
            JOIN detalle d ON d.idpedido = p.idpedido
            JOIN producto prod ON prod.idproducto = d.idproducto
            WHERE c.idcliente = %s
            ORDER BY p.idpedido
        """

        #EJECUTO
        cursor.execute(consulta, (idcliente,))

        # LOS PRINTEA
        resultado = cursor.fetchall()
        for fila in resultado:
            print(f"ID Cliente: {fila[0]}")
            print(f"Nombre: {fila[1]}")
            print(f"Ciudad: {fila[2]}")
            print(f"Teléfono: {fila[3]}")
            print(f"Pedido ID: {fila[4]}")
            print(f"Producto: {fila[5]}, Precio: {fila[6]}, Cantidad:
            {fila[7]}\n")

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS
        POSIBLES)
        except ValueError as e:
            print(f"Error en la entrada de datos: {e}")
        except Exception as unknown:
            print(f"No pudo ejecutarse la función, error: {unknown}")

```

```
1/Insertar nuevo cliente
2/Seleccionar todos los clientes
3/Seleccionar un cliente en específico
4/Seguimiento de cliente
5/Salir al menu principal4
Seguimiento de cliente
Introduce el ID del cliente para realizar un seguimiento: 10
ID Cliente: 10
Nombre: Carlos
Ciudad: yuca
Teléfono: 1252
Pedido ID: 3
Producto: Salsa verde original Frankfurter, Precio: 13.0, Cantidad: 1

Menú de opciones
1/Insertar nuevo cliente
2/Seleccionar todos los clientes
3/Seleccionar un cliente en específico
4/Seguimiento de cliente
5/Salir al menu principal■
```

## DAO producto;

- La función compra\_producto; Primero comprueba si el cliente ya está registrado en la base de datos, seleccionando los datos del cliente usando el id solicitado para su identificación, lo introduzco en resultado, y en el caso de que sea TRUE, entonces muestra un mensaje de bienvenida, si fuera FALSE, por un error al buscar un cliente inexistente, mostraría un mensaje de error, haciendo referencia a la no existencia de este cliente.

En el caso de que el cliente no se encuentre en la base de datos, le posibilita registrarse directamente introduciendo sus datos y procediendo con ellos, tal y como en la tabla cliente.

Posteriormente, pido el nombre y el id del cliente, para asignarle a esos datos los de compra posteriores, le imprimo toda la lista de productos, con su id, nombre, y precio.

Automáticamente, ingresa su id en la tabla pedido. Luego, pido los datos de unidades y el idproducto que desea comprar, todo esto para, por un lado, servirme de sus respuestas en las variables para rellenar los campos pertinentes de la tabla detalle, puesto que está realizando una compra, y , por otro lado, para introducir esos datos en un txt llamado factura, donde se podrían observar, los datos del movimiento.

Después, le pregunto si desea continuar comprando, de lo contrario, termina la función de compra.

- Mi función select\_producto; Sirve para saber el precio de un producto en particular, primero le imprimo la lista de los productos, sin el precio, luego le pregunto, de qué producto desea saber el precio, y a continuación procede la impresión con la lista, y el dato deseado.
- La función select\_productos; Simplemente via sentencia SQL, solicita todos los datos de la tabla productos, y los ejecuta, luego, por medio de un bucle for, imprime las líneas de todos los productos de la propia tabla.

```

#DEF FUNNCION PARA LA COMPRA DE PRODUCTO
def compra_producto (conexion,cursor):
    try:
        while True:
            try:
                #PRIMERO ME ASEGURO DE QUE EXISTA EL CLIENTE QUE VA A
                COMPRAR
                cliente=input("Se encuentra en la base de datos? Si/No")
                if cliente.lower() == "si":
                    print("Inicio de sesión")
                    idcliente=int(input("Introduzca su id de cliente"))

                    #SELECCIONO HIPOTETICO CLIENTE CON LA ID QUE ME DAN
                    consulta = "SELECT * FROM cliente WHERE idcliente =
%s"

                    cursor.execute(consulta, (idcliente,))
                    resultado = cursor.fetchone()

                    #SI SE PUEDE REALIZAR LA CONSULTA, LUEGO EXISTE
                    CLIENTE

                    if resultado:
                        print(f"Bienvenido!")
                        break
                    #SINO,
                    else:
                        print("El ID de cliente no existe en la base de
datos.")

                        #PERMITO REGISTRO SI NO ESTA REGISTRADO EN LA DB
                        else:
                            Nombre=input("Ingrese su nombre: ")
                            ciudad=(input("Ingrese su ciudad: "))
                            tlf=input("Ingrese su telefono")

                            #EJECUTO CONSULTA CON LOS DATOS DADOS
                            consulta="insert into cliente(Nombre,ciudad,tlf)
values(%s,%s,%s)"

                            cursor.execute(consulta,(Nombre,ciudad,tlf))

                            conexion.commit()

                            #GUARDO AL CLIENTE EN EL TXT COMO ARRAY UNA VEZ SE
                            REGISTRE

                            consulta="select idcliente from cliente where
Nombre=%s and ciudad=%s and tlf=%s"

```

```

        cursor.execute(consulta,(Nombre,ciudad,tlf))
        resultado=cursor.fetchall()

        #INFORMO AL CLIENTE DE SU ID PARA PODER HACER COMPRA
A POSTERIORI

        for linea in resultado:
            idcliente=linea
            print(f"Se te ha asignado el id:{idcliente}")

        #INTRODUCZCO LOS DATOS EN UN ARRAY Y POSTERIORMENTE
UN ARCHIVO

        array = [idcliente, Nombre, ciudad, tlf]
        with open("Clientes.txt", "a") as archivo:
            archivo.write("=== Lista de Clientes ===\n")
            archivo.write("Datos insertados:\n\n")
            for dato in array:
                archivo.write(f"{dato}\n")
            print(f"Se insertó el cliente: {Nombre}\n con el
idcliente: {idcliente}\n la ciudad: {ciudad}\n y el telefono {tlf}")
            break

        #LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES
VARIOS POSIBLES)
        except ValueError as e:
            print(f"Tipo de dato incorrecto, error {e}")
        except Exception as unknown:
            print(f"No pudo ejecutarse la función, error {unknown}")

    while True:
        try:
            #PIDO DATOS PARA COMPRA
            print("Iniciando compra...")
            idcliente=int(input("Cual es su id de cliente?\n>"))
            Nombre=input("A nombre de quién es la compra?\n>")

            #MUESTRO LISTA DE PRODUCTOS
            lista= "select idproducto,Nombre,precio from producto"

            cursor.execute(lista)

            #PRINTEO PRODUCTOS POR MEDIO DE LISTADO
            resultado=cursor.fetchall()
            for idproducto, nombre,precio in resultado:
                print(f"IDProducto: {idproducto}, Nombre: {nombre},
Precio: {precio}")

```

```

#INSERT INTO EN PEDIDO ANTES DE COMPRA
consulta1=("insert into pedido(idcliente) values(%s)")

cursor.execute(consulta1,(idcliente,))

conexion.commit()

#INTRODUCCIÓN DE DATOS NECESARIOS PARA ASIGNARLOS A LOS
REGISTROS DE LA TABLA DETALLE
unidades = int(input("Cuántas unidades deseas comprar del
producto al que echo un ojo?"))
idproducto = int(input("De nuestros productos
disponibles, introduce el id de aquel que quiera comprar:\n"))

# OBTENCIÓN DEL PRECIO DEL PRODUCTO SELECCIONADO
consulta2 = "select precio from producto where idproducto
= %s"

cursor.execute(consulta2, (idproducto,))

resultado = cursor.fetchall()

precio_compra = resultado[0][0]

#INSERT INTO EN DETALLE PARA COMPRA
consulta3=("insert into
detalle(idproducto,precio,unidades) values(%s,%s,%s)")

cursor.execute(consulta3,(idproducto,precio_compra,unidades))

conexion.commit()

print(f"Se realizó la compra del producto con el id:
{idproducto} con el coste: {precio_compra}, y la cantidad: {unidades}")

array=[f"El cliente con el id:{idcliente}Y el
nombre:{Nombre},Realizo la compra del producto con el id:{idproducto} ,A
un precio de:{precio_compra} En la cantidad de :{unidades}"]

#ARCHIVO TXT PARA INTRODUCIR LOS DATOS DE COMPRA
with open("Factura.txt", "a") as archivo:
    archivo.write("=== Factura de Compra ===\n")

    archivo.write("Detalles de la compra:\n\n")

```

```
        for dato in array:
            archivo.write(f"{dato}\n")

#PREGUNTO SI DESEA CONTINUAR CON LA COMPRA
continuar=input("Deseas seguir comprando? Si/No\n")

if continuar.lower() == "si":
    print("Siguiente compra")
elif continuar.lower() == "no":
    break

#LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS
POSIBLES)
except ValueError as e:
    print(f"Tipo de dato incorrecto, error {e}")
except Exception as unknown:
    print(f"No pudo ejecutarse la función, error {unknown}")
```



```

Menú de opciones
1/Tabla Producto
2/Tabla Cliente
3/Salir
1
Marchando al menu de Producto
Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir
1
Comprar producto
Se encuentra en la base de datos? Si/Nosi
Inicio de sesión
Introduzca su id de cliente
Bienvenido!
Iniciando compra...
Cual es su id de cliente?
>7
A nombre de quién es la compra?
>Carlos
IDProducto: 1, Nombre: Té Dharamsala, Precio: 18
IDProducto: 2, Nombre: Cerveza tibetana Barley, Precio: 19
IDProducto: 3, Nombre: Sirope de regaliz, Precio: 10
IDProducto: 4, Nombre: Especias Cajun del chef Anton, Precio: 22
IDProducto: 5, Nombre: Mezcla Gumbo del chef Anton, Precio: 21
IDProducto: 6, Nombre: Mermelada de grosellas de la abuela, Precio: 25
IDProducto: 7, Nombre: Peras secas orgánicas del tío Bob, Precio: 30
IDProducto: 8, Nombre: Salsa de arándanos Northwoods, Precio: 40
IDProducto: 9, Nombre: Buey Mishi Kobe, Precio: 97
IDProducto: 10, Nombre: Pez espada, Precio: 31
IDProducto: 11, Nombre: Queso Cabrales, Precio: 21
IDProducto: 12, Nombre: Queso Manchego La Pastora, Precio: 38
IDProducto: 13, Nombre: Algas Konbu, Precio: 6
IDProducto: 14, Nombre: Cuajada de judías, Precio: 23
IDProducto: 15, Nombre: Salsa de soja baja en sodio, Precio: 16
IDProducto: 16, Nombre: Postre de merengue Pavlova, Precio: 17
IDProducto: 17, Nombre: Cordero Alice Springs, Precio: 39
IDProducto: 18, Nombre: Langostinos tigre Carnarvon, Precio: 63
IDProducto: 19, Nombre: Pastas de té de chocolate, Precio: 9
IDProducto: 20, Nombre: Mermelada de Sir Rodney's, Precio: 81
IDProducto: 21, Nombre: Bollos de Sir Rodney's, Precio: 10
IDProducto: 22, Nombre: Pan de centeno crujiente estilo Gustaf's, Precio: 21
IDProducto: 23, Nombre: Pan fino, Precio: 9
IDProducto: 24, Nombre: Refresco Guaraná Fantástica, Precio: 5
IDProducto: 25, Nombre: Crema de chocolate y nueces NuNuCa, Precio: 14
IDProducto: 26, Nombre: Ositos de goma Gumbär, Precio: 31

```

```

Cuántas unidades deseas comprar del producto al que echo un ojo?8
De nuestros productos disponibles, introduce el id de aquel que quiera comprar:
74
Se realizó la compra del producto con el id: 74 con el coste: 10, y la cantidad: 8
Deseas seguir comprando? Si/No
no
Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir

```

Detalles de la compra:

El cliente con el id:74 el nombre:Carlos,Realizo la compra del producto con el id:74 ,A un precio de:10 En la cantidad de :8

```

Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir1
Comprar producto
Se encuentra en la base de datos? Si/Nono
Ingresa su nombre: Alejandro
Ingresa su ciudad: Madrid
Ingresa su telefono723
Se te ha asignado el id:(12,)
Se insertó el cliente: Alejandro
con el idcliente: (12,)
la ciudad: Madrid
y el telefono 723
Iniciando compra...
Cual es su id de cliente?

```

```

■ (11,)
Luca
Palermo
987969
=== Lista de Clientes ===
Datos insertados:

(12,)
Alejandro
Madrid
723

```

```

#DEF FUNCION PARA SELECCIONAR TODOS LOS PRODUCTOS
def select_productos (cursor):
    try:
        #EJECUTO SELECT ALL DE PRODUCTOS
        consulta="select * from producto order by idproducto"

        cursor.execute(consulta)

```

```

#PRINTEA LOS RESULTADOS DE LA CONSULTA
resultado=cursor.fetchall()
for fila in resultado:
    print(f"IDProducto: {fila[0]}, Nombre: {fila[1]}, Precio: {fila[2]},")

#LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS POSIBLES)
except ValueError as e:
    print(f"Tipo de dato incorrecto, error {e}")
except Exception as unkown:
    print(f"No pudo ejecutarse la funcion, error {unkown}")

Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir2
Seleccionar todos los productos
IDProducto: 1, Nombre: Té Dharamsala, Precio: 18,
IDProducto: 2, Nombre: Cerveza tibetana Barley, Precio: 19,
IDProducto: 3, Nombre: Sirope de regaliz, Precio: 10,
IDProducto: 4, Nombre: Especias Cajun del chef Anton, Precio: 22,
IDProducto: 5, Nombre: Mezcla Gumbo del chef Anton, Precio: 21,
IDProducto: 6, Nombre: Mermelada de grosellas de la abuela, Precio: 25,
IDProducto: 7, Nombre: Peras secas orgánicas del tío Bob, Precio: 30,
IDProducto: 8, Nombre: Salsa de arándanos Northwoods, Precio: 40,
IDProducto: 9, Nombre: Buey Mishi Kobe, Precio: 97,
IDProducto: 10, Nombre: Pez espada, Precio: 31,
IDProducto: 11, Nombre: Queso Cabrales, Precio: 21,
IDProducto: 12, Nombre: Queso Manchego La Pastora, Precio: 38,
IDProducto: 13, Nombre: Algas Konbu, Precio: 6,
IDProducto: 14, Nombre: Cerveza Outback, Precio: 13,
IDProducto: 71, Nombre: Crema de queso Fløtemys, Precio: 22,
IDProducto: 72, Nombre: Queso Mozzarella Giovanni, Precio: 35,
IDProducto: 73, Nombre: Caviar rojo, Precio: 15,
IDProducto: 74, Nombre: Queso de soja Longlife, Precio: 10,
IDProducto: 75, Nombre: Cerveza Klosterbier Rhönbräu, Precio: 8,
IDProducto: 76, Nombre: Licor Cloudberry, Precio: 18,
IDProducto: 77, Nombre: Salsa verde original Frankfurter, Precio: 13,
Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir2

#DEF FUNCION PARA SELECCIONAR UNO SOLO
def select_producto (cursor):
    try:
        #EJECUTO SELECT PARA ENSEÑAR TODOS LOS PRODUCTOS Y POSTERIORMENTE ELEGIR Y EJECUTO
        lista= "select idproducto,nombre,precio from producto"

```

```

cursor.execute(lista)

#PRINTEO RESULTADOS
resultado=cursor.fetchall()
for idproducto, nombre,precio in resultado:
    print(f"IDProducto: {idproducto}, Nombre: {nombre}")

#PREGUNTO EL ID DEL PRODUCTO DEL QUE QUIERE SABER EL PRECIO Y
EJECUTO CON EL DATO
idproducto=int(input("Que id tiene el producto del que deseas
informacion?"))

consulta="select * from producto where idproducto=%s"

cursor.execute(consulta,(idproducto,))

#PRINTEO LOS RESULTADOS ASOCIADOS A LA CONSULTA SQL PREVIA
resultado=cursor.fetchall()
for idproducto, nombre,precio in resultado:
    print(f"IDProducto: {idproducto}, Nombre: {nombre}, Precio:
{precio}")

#LOS MISMOS ERRORES (DATA TYPE ERRONEO Y OTROS ERRORES VARIOS
POSIBLES)
except ValueError as e:
    print(f"Tipo de dato incorrecto, error {e}")
except Exception as unkown:
    print(f"No pudo ejecutarse la funcion, error {unkown}")

```

```
Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir3
Seleccionar producto
IDProducto: 1, Nombre: Té Dharamsala
IDProducto: 2, Nombre: Cerveza tibetana Barley
IDProducto: 3, Nombre: Sirope de regaliz
IDProducto: 4, Nombre: Especias Cajun del chef Anton
IDProducto: 5, Nombre: Mezcla Gumbo del chef Anton
IDProducto: 6, Nombre: Mermelada de grosellas de la abuela
IDProducto: 7, Nombre: Peras secas orgánicas del tío Bob
IDProducto: 8, Nombre: Salsa de arándanos Northwoods
IDProducto: 9, Nombre: Buey Mishi Kobe
IDProducto: 10, Nombre: Pez espada
IDProducto: 11, Nombre: Queso Cabrales
IDProducto: 12, Nombre: Queso Manchego La Pastora
IDProducto: 13, Nombre: Algas Konbu
IDProducto: 14, Nombre: Cuajada de judías
IDProducto: 15, Nombre: Salsa de soja baja en sodio
IDProducto: 16, Nombre: Postre de merengue Pavlova
IDProducto: 17, Nombre: Cordero Alice Springs
IDProducto: 18. Nombre: Langostinos tigre Carnarvon
IDProducto: //, nombre: Salsa verde original Frankfurter
Que id tiene el producto del que deseas informacion?66
IDProducto: 66, Nombre: Especias picantes de Luisiana, Precio: 17
Menú de opciones
1/Comprar producto
2/Seleccionar todos los productos
3/Seleccionar un producto en particular
4/Salir
```

## Bibliografía

(s/f). El Libro De Python. Recuperado el 14 de noviembre de 2024, de <https://ellibrodepython.com/>

Bravo, B. I. (2023, octubre 2). Lectura y escritura de archivos en Python: Como crear, leer y escribir archivos. freecodecamp.org. <https://www.freecodecamp.org/espanol/news/lectura-y-escritura-de-archivos-en-python-como-crear-leer-y-escribir-archivos/>

Brugués, A. (2019, mayo 12). Leer y Escribir Ficheros de Texto. Programa en Python; Albert Brugués. <https://www.programaenpython.com/fundamentos/leer-y-escribir-ficheros-de-texto/>

ChatGPT. (s/f). Chatgpt.com. Recuperado el 14 de noviembre de 2024, de <https://chatgpt.com/>

Moisset, D. (s/f). Archivos de texto: creación, escritura y lectura. Tutorialesprogramacionya.com. Recuperado el 14 de noviembre de 2024, de <https://www.tutorialesprogramacionya.com/pythonya/detalleconcepto.php?punto=79&codigo=79&inicio=75>

(S/f). Programminghistorian.org. Recuperado el 14 de noviembre de 2024, de <https://programminghistorian.org/es/lecciones/trabajar-con-archivos-de-texto>