HITO 2 DEL 1º TRIMESTRE DE Programación

Rodrigo García Ortiz

20 - 11 – 2024

Índice

Contenido

[Fase 1 2](#_Toc183085508)

[Fase 2 4](#_Toc183085509)

[Webgrafía 10](#_Toc183085510)

1.

# Fase 1

El primer paso es crear la base de datos en la que se guardaran los datos del programa,

creando tablas para los clientes, los productos y los pedidos.

Luego pasamos al código. Empezamos por crear la conexión a la base de datos.

Después definimos las funciones: para registrar clientes, para mostrar los clientes registrados, para buscar un cliente en concreto, para mostrar los productos, para elegir los productos, para hacer un pedido y para guardar los detalles del pedido.

Después creamos un menú y definimos una función para escoger la opción deseada.

Por último, creamos una función que nos permita ver el menú y escoger opciones todas las veces que queramos.

· Importamos la conexión a la base de datos

· Definimos la función conectar para conectarnos a la base de datos

· Importamos la función a otro archivo que usaremos para las funciones del programa

· Definimos la función que registra clientes usando un insert into en la tabla clientes

· Definimos la función que muestra los clientes usando un select from la tabla clientes

· Definimos la función que permite buscar un cliente concreto usando un select from clientes

· Definimos la función que muestra los productos usando un select from productos

· Definimos la función que permite elegir productos usando un select from productos

· Definimos la función que permite hacer pedidos usando un insert into pedidos

· Definimos la función que muestra los detalles de un pedido usando select from pedidos inner join clientes inner join detalles

· Exportamos las funciones a otro archivo en el crearemos el menú

· Definimos la función que crea el menú usando print()

· Definimos la función que permite escoger opciones:  
· Una opción nos pide los datos para registrar un cliente

· Una opción muestra a los clientes y nos pide el id del cliente que queremos buscar y nos lo muestra

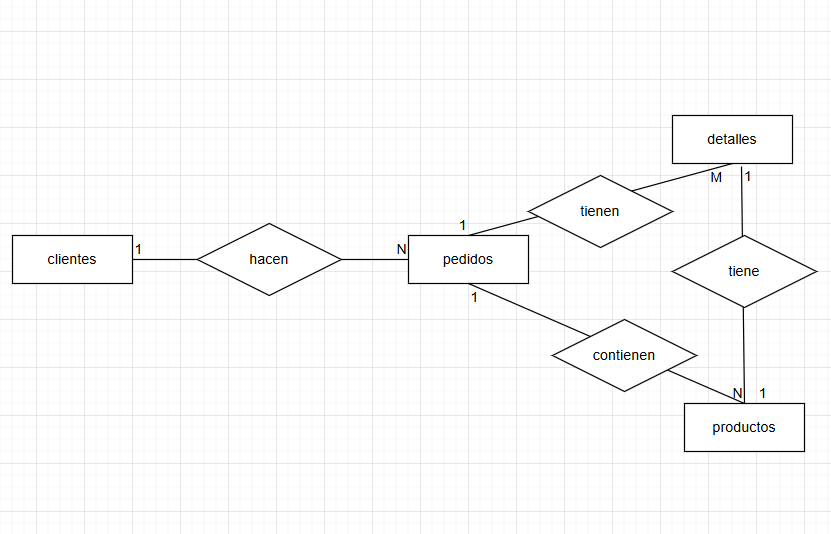
· Una opción nos muestra los productos y nos permite hacer un pedido usando un bucle while para que introduzcamos todos los productos deseados al pedido

· Una opción nos muestra los datos del pedido que busquemos 2.

· Una opción nos permite salir del programa

· Exportamos las funciones de menú a otro archivo

· Definimos una función que usa un bucle while para que hagamos todas las consultas que queramos



3.

# Fase 2

bdd.py

import mysql.connector

def conectar(): #Conecta con la base de datos

    conexion = mysql.connector.connect(

        host="localhost",    # Cambia según tu configuración

        user="root",      # Cambia según tu configuración

        password="curso",  # Cambia según tu configuración

        database="gestion\_pedidos"

    )

    return conexion

funciones.py

from bdd import conectar

def crear\_cliente(nombre, telefono, direccion): #Registra un cliente

    conexion = conectar()

    cursor = conexion.cursor()

    consulta = "INSERT INTO clientes (nombre, telefono, direccion) VALUES (%s, %s, %s)"

    cursor.execute(consulta, (nombre, telefono, direccion))

    conexion.commit()

    cursor.close()

    conexion.close()

def leer\_clientes(): #Muestra los clientes registrados

    conexion = conectar()

    cursor = conexion.cursor()

    cursor.execute("select \* from clientes")

    clientes = cursor.fetchall()

    cursor.close()

    conexion.close()

    return clientes

def buscar\_cliente(busqueda): # Muestra un cliente concreto

    conexion = conectar()

    cursor = conexion.cursor()

    consulta = """select nombre, telefono, direccion from clientes where id\_cliente = %s"""

    cursor.execute(consulta, (busqueda,))

    resultado = cursor.fetchall()

    conexion.commit()

    cursor.close() 4.

    conexion.close()

    return resultado

def mostrar\_productos(): # Muestra los productos

    conexion = conectar()

    cursor = conexion.cursor()

    cursor.execute("select \* from productos")

    productos = cursor.fetchall()

    conexion.commit()

    cursor.close()

    conexion.close()

    return productos

def comprar(producto): #Devuelve el precio del producto escogido

    conexion = conectar()

    cursor = conexion.cursor()

    consulta = "select precio\_producto from productos where id\_producto = %s"

    cursor.execute(consulta, (producto,))

    precio = cursor.fetchall()

    precio\_producto = float(precio[0][0])

    conexion.commit()

    cursor.close()

    conexion.close()

    return precio\_producto

def crear\_pedido(cliente, precio, numero\_objetos): #Registra un pedido

    conexion = conectar()

    cursor = conexion.cursor()

    consulta = "INSERT INTO pedidos (cliente, precio, numero\_objetos) VALUES (%s, %s, %s)"

    cursor.execute(consulta, (cliente, precio, numero\_objetos))

    id\_pedido = cursor.lastrowid

    conexion.commit()

    cursor.close()

    conexion.close()

    return id\_pedido

def seleccionar\_pedido(id\_pedido): #Muestra los datos de un pedido, su cliente y sus detalles

    conexion = conectar()

    cursor = conexion.cursor()

    consulta = "select pedidos.\*, clientes.\* from pedidos inner join clientes on id\_cliente = cliente where id\_pedido = %s"

    cursor.execute(consulta, (id\_pedido,))

    pedido = cursor.fetchall()

    conexion.commit()

    cursor.close()

   conexion.close() 5.

    return pedido

menus.py

from funciones import crear\_cliente, leer\_clientes, buscar\_cliente, mostrar\_productos, comprar, crear\_pedido, seleccionar\_pedido

def mostrar\_menu(): #Muestra el menú

    print("\n=== Menú de opciones ===")

    print("1. Registrar clientes")

    print("2. Ver y buscar clientes")

    print("3. Realizar compra")

    print("4. Seguimiento de compra")

    print("5. Salir")

def ejecutar\_opcion(opcion):#Realiza la opción asignada

    match opcion:

        case 1: #Registra un cliente con los datos introducidos

            nombre = input("Ingrese el nombre del nuevo cliente: ")

            telefono = int(input("Ingrese el teléfono del nuevo cliente: "))

            direccion = input("Ingrese la dirección del nuevo cliente: ")

            crear\_cliente(nombre, telefono, direccion)

            print("Cliente registrado exitosamente.")

        case 2: #Muestra los clientes

            clientes = leer\_clientes()

            print("\nListado de Clientes:")

            for id\_cliente, nombre, telefono, direccion in clientes:

                print(f"ID: {id\_cliente}, Nombre: {nombre}, Teléfono: {telefono} Dirección: {direccion}")

            busqueda = (input("Introduzca el ID del cliente que quiere buscar: "))

            print(buscar\_cliente(busqueda))

        case 3: #Registra un pedido asignado al cliente escogido con los productos escogidos

            id\_cliente = int(input("Indique el cliente que realizará la compra: ")) #Escoge el cliente

            productos = mostrar\_productos() #Selecciona los datos del producto

            print("\nLista de Productos:")

            for id\_producto, nombre\_producto, precio\_producto in productos: #Muestra los productos

                print(f"ID: {id\_producto} Nombre: {nombre\_producto} Precio: {precio\_producto}€")

            precio\_total = 0

            numero\_objetos = 0 6.

            while True:

                producto = int(input("Introduce el ID del producto que quieres añadir al pedido: ")) #Introduce un producto al pedido

                if producto == 0:

                    break

                else:

                    precio\_total = precio\_total + comprar(producto)

                    numero\_objetos = numero\_objetos + 1

            print(f"El precio total del pedido es: {precio\_total}€")

            crear\_pedido(id\_cliente, precio\_total, numero\_objetos) #Registra un pedido

            print(f"El número del pedido es: {crear\_pedido(id\_cliente, precio\_total, numero\_objetos)}")

        case 4: #Muestra el número del pedido creado

            pedido = int(input("Introduce el ID del pedido deseado: "))

            seleccionar\_pedido(pedido)

        case \_: #Introduces un número sin opción asignada

            print("Opción inválida")

main.py

import menus

def main():

    continuar = True

    while continuar:

        menus.mostrar\_menu()  #Muestra el menú

        try:

            opcion = int(input("Seleccione una opción: ")) #Elije una opción

            if opcion == 5:

                break

            else:

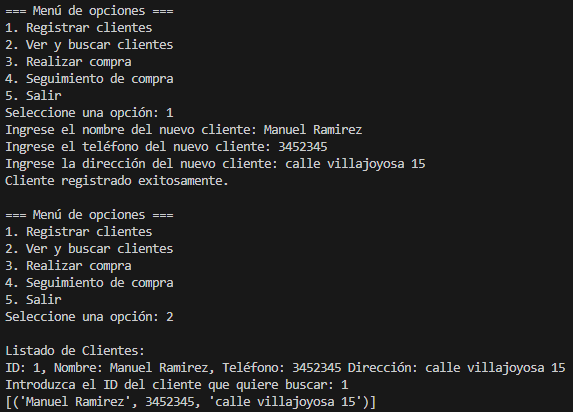
                menus.ejecutar\_opcion(opcion) #Ejecuta la opción escogida

        except ValueError:

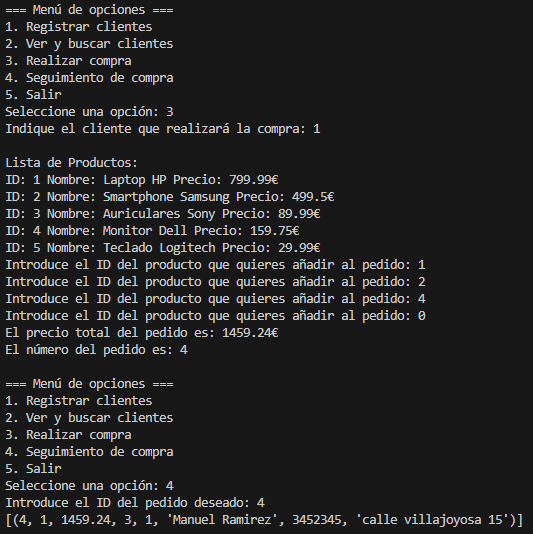
            print("Error: Debe ingresar un número válido.")

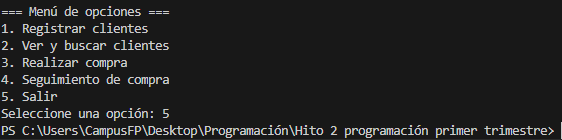
if \_\_name\_\_ == "\_\_main\_\_":

    main() 7.



8.





9.

# Webgrafía

*Get the id after INSERT into MySQL database using Python*. (2020, diciembre 10). GeeksforGeeks. <https://www.geeksforgeeks.org/get-the-id-after-insert-into-mysql-database-using-python/>

*MySQL INNER JOIN keyword*. (s/f). W3schools.com. Recuperado el 21 de noviembre de 2024, de <https://www.w3schools.com/mysql/mysql_join_inner.asp>

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>

10.