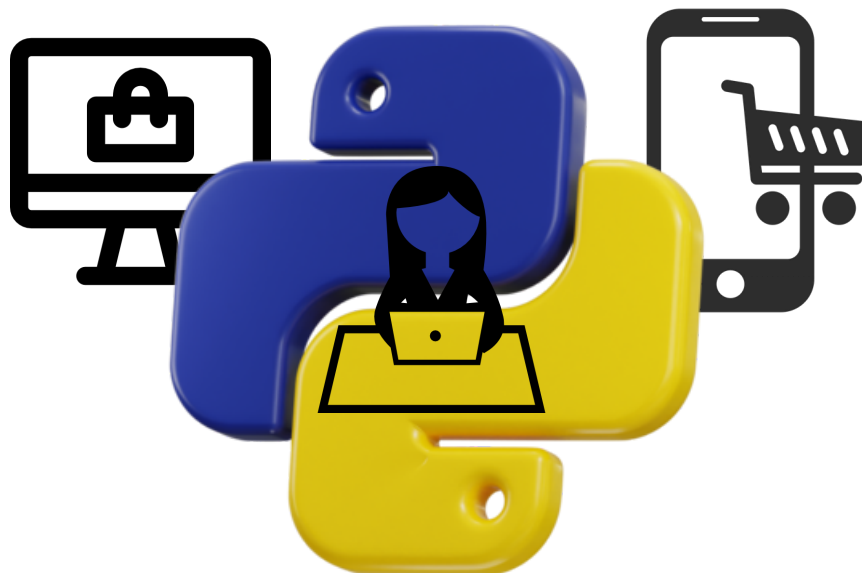


## **HITO 2 DEL 1º TRIMESTRE DE PROGRAMACIÓN**



**Iris Pérez Aparicio**  
**20/11/2024**

**CampusFP Humanes Emprende**  
**1º DAM**  
**Programación**

# ÍNDICE

★ ESPECIFICACIONES:	1
★ FASE 1	2
★ FASE 2	8
- Ejecución del programa:	16
★ BIBLIOGRAFÍA Y WEBGRAFÍA	25
- Tecnologías utilizadas	25

## [ENLACE GITHUB](#)



## **ESPECIFICACIONES:**

A nuestro equipo de desarrollo de software nos ha llegado la petición de un cliente para el diseño y desarrollo de una aplicación de gestión de pedidos.

El cliente necesita una aplicación en donde un posible cliente se registre en la aplicación para realizar un pedido de los productos que están disponibles en la tienda online.

La aplicación estará desarrollada en Python, aunque el análisis previo se realiza en pseudocódigo o diagramas de flujo.

Las principales operaciones que necesitaremos gestionar serán:

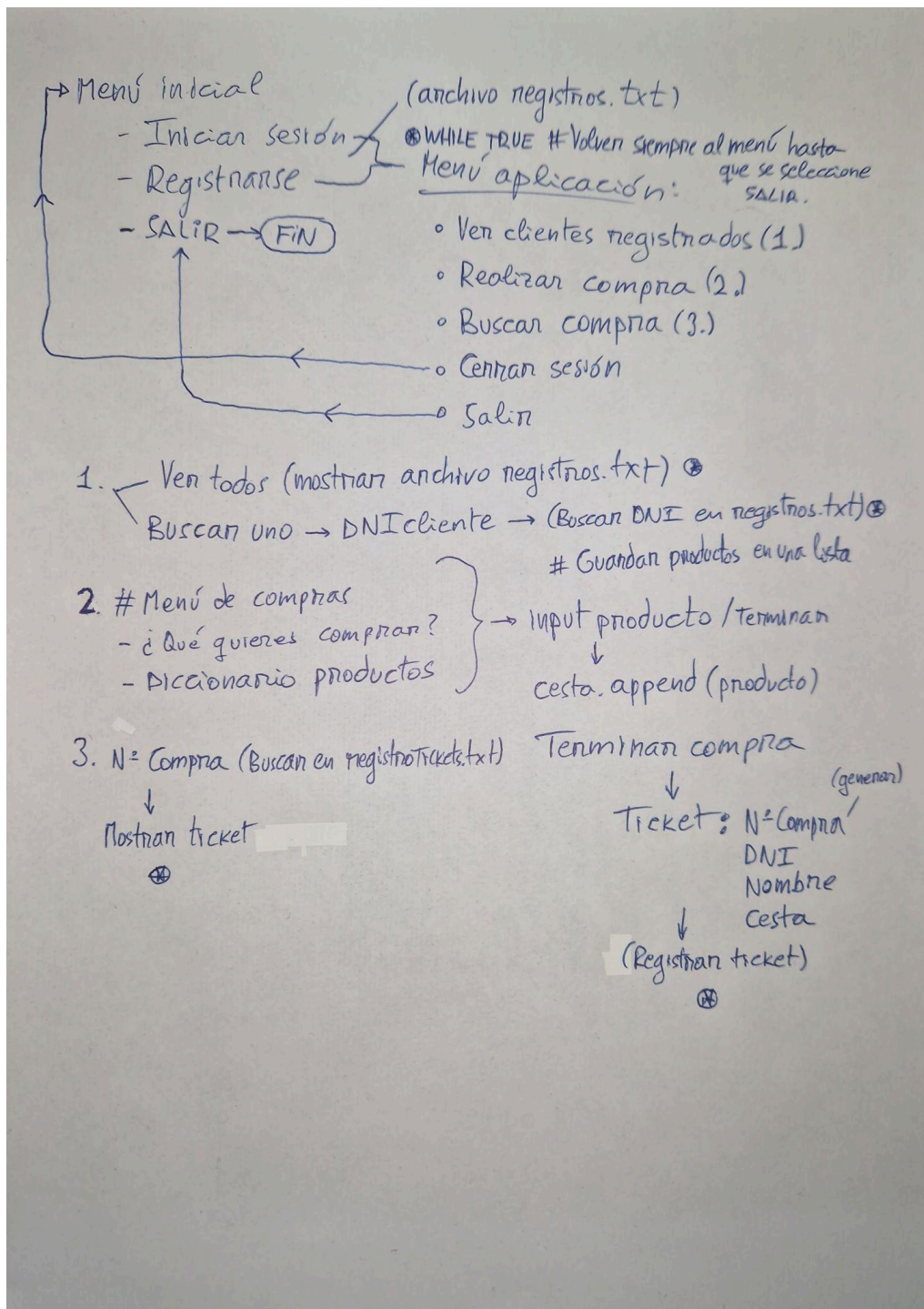
- Registro de cliente: se pedirán sus datos personales. Cada cliente debe tener un campo único.
- Visualizar todos los clientes registrados y realizar búsquedas de clientes a través de su campo único.
- Realizar una compra: cada compra estará asociada a un cliente y puede tener uno o varios productos. Los artículos estarán cargados previamente en la aplicación. Al finalizar cada compra se mostrará el número del pedido.
- Seguimiento de una compra: mediante el número de pedido se mostrarán todos los datos del cliente y del pedido.

El desarrollo de la aplicación consta de varias fases:

### **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.

Foto del **esquema** que hice a papel y boli sobre la idea inicial del programa, su estructura y su flujo. (Esto es un **esquema personal**, no es un diagrama de flujo)



## **PSEUDOCÓDIGO** del programa:

### **Función** menuInicial

Mientras True: #Para volver siempre al menú inicial hasta acceder a la aplicación.

Imprimir menú (iniciar sesión, registrarse, salir)

Pedir opción

Si opción == 1:

llamar a la función iniciar sesión

Si opción == 2:

llamar a la función registrarse

Si opción == 3:

cerrar el programa

sino:

imprimir "ERROR, opción no válida"

FinSi

FinMientras

FinFuncion

### **Función** registrarse

Pedir nombre

Pedir DNI

Mientras longitud(DNI) != 9

Pedir DNI

FinMientras

Abrir archivo "registros.txt" con r (leer)

Para linea en archivo

Si DNI en linea:

Imprimir "DNI ya registrado."

Llamar función menuInicial

FinSi

FinPara

CerrarArchivo

Llamar función registrarCliente

FinFuncion

### **Función** registrarCliente

Abrir archivo "registros.txt" con a (añadir)

Escribir "nombre:DNI" en archivo

Cerrar archivo

Imprimir ";Gracias por registrarte, {nombre}!"

Llamar función menu

FinFunción

**Función iniciarSesion**

```
Pedir nombre
Pedir DNI
Abrir archivo "registros.txt" con r (leer)
  Para linea en archivo
    Dividir linea con ":" en nombreArchivo, DNIarchivo
    Si nombre == nombreArchivo y DNI == DNIarchivo:
      Imprimir "Bienvenido de nuevo, {nombre}"
      Llamar función menu
    FinSi
  FinPara
Cerrar archivo
Imprimir "Nombre o DNI incorrectos."
Llamar función menuInicial
FinFuncion
```

**Función menu**

```
Mientras True:
  Imprimir menú (Realizar compra, Buscar compra, Ver clientes, Cerrar sesión, Salir)
  Pedir opción
  Si opción == 1:
    Llamar función realizarCompra
  Si opción == 2:
    Llamar función buscarCompra
  Si opción == 3:
    Llamar función verClientes
  Si opción == 4:
    Imprimir "Has cerrado sesión."
    Llamar función menuInicial
  Si opción == 5:
    Llamar función salir
  sino:
    Imprimir "ERROR, opción no válida"
  FinSi
FinMientras
FinFunción
```

**Función realizarCompra**

```
Compra = False
Cesta = []
Diccionario productos con números y nombres de productos
Mientras True:
  Imprimir productos disponibles
  Pedir producto
```

```

Si producto == 7:
    Romper el bucle
Si producto >= 1 y <= 6:
    Compra = True
    Añadir producto a cesta
    Imprimir producto añadido
sino:
    Imprimir "ERROR, opción no válida"
FinSi
FinMientras
Si compra == True:
    Imprimir "¡Gracias por tu compra!"
    Llamar función asignarNumeroCompra
FinFuncion

```

**Función** asignarNumeroCompra

```

Generar numeroCompra aleatorio
Abrir archivo "compras.txt" con r (leer)
    Para linea en archivo:
        Dividir linea con "|"
        Si numeroCompra == numeroCompraArchivo:
            Incrementar numeroCompra
        FinSi
    FinPara
Cerrar archivo
FinFuncion

```

Función registrarCompra(numeroCompra, nombre, DNI, cesta)

```

Abrir archivo "compras.txt" con a (añadir)
    Escribir número de compra, nombre, DNI y cesta en archivo
Cerrar archivo
Llamar función imprimirTicket(numeroCompra)
FinFuncion

```

Función imprimirTicket(numeroCompra)

```

Inicializar compraEncontrada como False
Intentar:
    Abrir archivo "compras.txt" con r (leer)
    Para linea en archivo:
        Dividir linea con "|"
        Si numeroCompra == numeroCompraArchivo:
            Imprimir ticket con número de compra, nombre, DNI y productos comprados
            Cambiar compraEncontrada a True
        FinSi
    FinIntentar

```

```
    FinPara
    Cerrar archivo
    Si compraEncontrada == False:
        Imprimir "No existe ninguna compra con ese número asociado."
FinFuncion
```

```
Función buscarCompra
    Pedir número de compra
    Llamar función imprimirTicket
FinFuncion
```

```
Función verClientes
    Imprimir menú opciones (ver todos o buscar cliente)
    Pedir opción
    Si opción == 1:
        Abrir archivo "registros.txt" con r (leer)
        Leer e imprimir contenido
        Cerrar archivo
    Si opción == 2:
        Llamar función buscarCliente
    FinSi
FinFuncion
```

```
Función buscarCliente
    Encontrado = False
    Pedir DNI
    Abrir archivo "registros.txt" con r (leer)
    Para linea en archivo:
        Dividir linea con ":" en nombreArchivo, DNIarchivo
        Si DNI == DNIarchivo:
            Imprimir datos del cliente
            Cambiar encontrado a True
        FinSi
    FinPara
    Si encontrado == False:
        Imprimir "No existe ningún cliente con ese DNI."
    FinFuncion
```

```
Función salir
    Imprimir frase de despedida
    exit() # Para cerrar el programa
FinFuncion
```



## 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.

### CÓDIGO en PYTHON:

```
# Importo la librería random para poder generar números aleatorios.
import random

# Menú que aparece al inicio de la aplicación.
def menuInicial():
    while True: # Utilizo un bucle while True para que el menú le
vuelva a aparecer al usuario hasta que seleccione una opción válida.
        print("-----")
        print(";Hola! bienvenid@, ¿Qué quieres hacer?:\n1- Iniciar
Sesión\n2- Registrarme\n3- Salir")
        try:
            opcion = int(input("Selecciona una opción: "))
            if opcion == 1:
                iniciarSesion()
                break # Si la opción seleccionada es válida, rompo el
bucle while True con un break.
            elif opcion == 2:
                registrarse()
                break
            elif opcion == 3:
                salir() # Llamo a la función salir para cerrar el
programa.
                break
            else:
                print("ERROR, elige una de las opciones introduciendo
el número correspondiente.")
```

```

        except ValueError: # Uso try except para que mi programa no se
rompa si el usuario introduce una letra o un símbolo en vez de número
(ValueError).

            print("ERROR, elige una de las opciones introduciendo el
número correspondiente.")

# Función para registrar un cliente, solo si no está ya registrado.
# NOTA: utilizo el DNI como el campo único de cada cliente.
def registrarse():
    try:
        nombre = input("Introduce tu nombre: ").title() # .title()
para que la primera letra sea mayúscula.
        DNI = input("Introduce tu DNI: ").upper() # .upper para que la
letra del DNI sea mayúscula.
        while len(DNI) != 9: # Hago un bucle while para asegurarme de
que el DNI introducido tiene 9 dígitos.
            print("El DNI debe tener 9 caracteres.")
            DNI = input("Introduce tu DNI: ").upper()
        # Antes de registrar al cliente compruebo que no exista ningún
usuario registrado con el mismo DNI (campo único).
        with open("registros.txt", "r") as archivo:
            for linea in archivo:
                if DNI in linea:
                    print("-----")
                    print("Ya hay un cliente registrado con ese DNI,
por favor, prueba a iniciar sesión.")
                    menuInicial()

        # Si el DNI no está registrado (no aparece en el archivo
registros), registro al nuevo cliente.
        registrarCliente(nombre, DNI)
    except FileNotFoundError: # Si es el primer cliente en
registrarse no existe el archivo registros, por lo que daría error al
intentar leerlo.
        registrarCliente(nombre, DNI)

# Función que registra a los nuevos clientes (almacena los clientes
registrados en el archivo registros).

```

```

# Y los envía al menú de la aplicación, guardando su nombre y DNI.
def registrarCliente(nombre, DNI):
    with open("registros.txt", "a") as archivo:
        archivo.write(f"{nombre}:{DNI}\n")
    print("-----")
    print(f";Gracias por registrarte, {nombre}!")
    menu(nombre, DNI)

# Función para iniciar sesión. Concede acceso a la aplicación solo a
los clientes registrados.
def iniciarSesion():
    try:
        nombre = input("Introduce tu nombre: ").title()
        DNI = input("Introduce tu DNI: ").upper()
        with open("registros.txt", "r") as archivo: # Abro el archivo
registros para comprobar si el usuario está registrado.
            for linea in archivo:
                nombreArchivo, DNIarchivo = linea.strip().split(":")
# .split para separar cada campo en el archivo de texto.
                if DNIarchivo == DNI and nombreArchivo == nombre: #
Si el DNI y el nombre están registrados:
                    print("-----")
                    print(f"Bienvenid@ de nuevo, {nombre}")
                    menu(nombre, DNI) # Concedo el acceso a la
aplicación al usuario, guardando su nombre y DNI.
                    else: # Sinó, devuelvo al usuario al menú inicial
para que se registre.
                        print("-----")
                        print("Nombre o DNI incorrectos, prueba a
registrarte")

                        menuInicial()

    except FileNotFoundError: # Si no se ha registrado aun ningún
usuario no existe el archivo registros, por lo que daría error al
leerlo.
        print("ERROR, no hay ningún usuario registrado con ese
DNI.\nPor favor, prueba a registrarte.")
        menuInicial()

```

```

# Función que muestra el menú de la aplicación (con el nombre y DNI
del usuario guardados).
# NOTA: incluyo en el menú la opción ver clientes como como pide el
enunciado, aunque eso haga que la aplicación deje de ser realista.
def menu(nombre, DNI):
    while True: # Utilizo un bucle while True para devolver al
usuario al menú siempre, hasta que elija salir (break).
        print("-----")
        print("MENÚ:\n1- Realizar una compra\n2- Buscar una compra\n3-
Ver clientes\n4- Cerrar sesión\n5- Salir")
        try:
            opcion = int(input("Selecciona una opción: "))
            if opcion == 1:
                realizarCompra(nombre, DNI)
            elif opcion == 2:
                buscarCompra()
            elif opcion == 3:
                verClientes()
            elif opcion == 4:
                print(f"Has cerrado sesión, hasta pronto {nombre}.")
                menuInicial()
            elif opcion == 5:
                salir() # Llamo a la función salir para cerrar el
programa.
                break
            else:
                print("ERROR, elige una de las opciones introduciendo
el número correspondiente.")
        except ValueError:
            print("ERROR, elige una de las opciones introduciendo el
número correspondiente.")

# Función para hacer una compra.
def realizarCompra(nombre, DNI):

```

```

    compra = False # Inicializo la variable compra como False para no
    registrar la compra ni generar un ticket si el usuario no selecciona
    ningún producto antes de terminar.

    cesta = [] # Inicializo la lista cesta en la que se almacenan los
    productos que selecciona el usuario.

    productos = {1: "Altavoces", 2: "Auriculares", 3: "Teclado", 4:
    "Raton", 5: "Alfombrilla", 6: "Monitor"} # Creo el diccionario
    productos en el que cada número corresponde a un producto.

    while True: # Inicio un bucle while True para pedir al usuario
    qué producto quiere comprar hasta que seleccione 7 (TERMINAR).
        print(f"\n¿Qué producto quieres comprar?\n{productos}")
        try:
            producto = int(input("Introduce el número de un producto
            para añadirlo a la cesta\n(Introduce 7 PARA TERMINAR la compra): "))
            if producto == 7: # Si el usuario selecciona 7, salgo del
            bucle (break).
                break
            if producto >= 1 and producto <= 6: # Si el usuario
            selecciona algún producto.
                compra = True # Cambio la variable compra a True.
                cesta.append(productos[producto]) # Añado el producto
            seleccionado a la cesta.
                print("-----")
                print(f"Has comprado: {productos[producto]}") #
            Muestro al usuario qué producto ha comprado.
                print("-----")
            else:
                print("ERROR, este número no corresponde a ningún
            producto.")
                continue
            except ValueError: # Por si el usuario introduce un caracter
            que no sea un número.
                print("ERROR")
                continue # Para seguir con la compra.

        if compra == True: # Si el usuario ha comprado al menos un
        producto, registro la compra y genero el ticket.
            print("-----")
            print("¡Gracias por tu compra!")

```

```

    asignarNumeroCompra(nombre, DNI, cesta)

# Asignamos un número de compra único, comprobando que no esté ya
creado en el archivo compras.
def asignarNumeroCompra(nombre, DNI, cesta):
    numeroCompra = random.randint(1,999)
    try:
        with open("compras.txt", "r") as archivo: # Compruebo que no
exista una compra registrada con el mismo número de compra generado.
            for linea in archivo:
                numeroCompraArchivo, _, _, _ =
linea.strip().split("|")
                numeroCompraArchivo = int(numeroCompraArchivo) #
Convierto de nuevo el número de compra a número (int) al sacarlo del
archivo.

                while numeroCompra == numeroCompraArchivo: # Mientras
el número de compra esté registrado en el archivo compras, le sumo 1.
De esta manera el número de compra será un campo único y no podrá
haber diferentes compras con el mismo número de compra.
                    numeroCompra = numeroCompra + 1
                print(f"Número de compra: {numeroCompra}") # Muestro el
número de pedido.

                registrarCompra(numeroCompra, nombre, DNI, cesta) # Registro
la compra en el archivo compras.
    except FileNotFoundError: # Si es la primera compra en
registrarse no existirá el archivo compras.
        registrarCompra(numeroCompra, nombre, DNI, cesta)

# Función para registrar la compra en el archivo compras.
def registrarCompra(numeroCompra, nombre, DNI, cesta):
    with open("compras.txt", "a") as archivo:
        compra = (f"{numeroCompra}|{nombre}|{DNI}|{cesta}\n")
        archivo.write(compra)

    # Una vez registrada la compra, muestro el ticket al usuario.
    imprimirTicket(numeroCompra)

# Función para imprimir un ticket mediante el número de compra.
def imprimirTicket(numeroCompra):

```

```

compraEncontrada = False
try:
    with open("compras.txt", "r") as archivo:
        for linea in archivo:
            numeroCompraArchivo, nombreArchivo, DNIarchivo,
cestaArchivo = linea.strip().split("|")
            numeroCompraArchivo = int(numeroCompraArchivo) #
Convierto el número de compra a int al sacarlo del archivo.
            if numeroCompra == numeroCompraArchivo: # Si el
numero de compra coincide con uno registrado, imprimo el ticket de esa
compra.

                compraEncontrada = True
                print("Aquí tienes el ticket de la compra ;).")
                print("-----")
                print(f"TICKET:\nNúmero de compra:
{numeroCompra}\nNombre: {nombreArchivo}\nDNI: {DNIarchivo}\nProductos
Comprados: {cestaArchivo}")
                if compraEncontrada == False:
                    print("Lo siento, no existe ninguna compra con ese número
asociado.")
            except FileNotFoundError: # Si aun no se ha registrado ninguna
compra y no existe el archivo compras.
                print("Lo siento, no existe ninguna compra con ese número
asociado.")

# Función para buscar una compra mediante el número de compra.
def buscarCompra():
    numeroCompra = int(input("\nIntroduce el número de la compra que
deseas buscar: "))
    print(f"Buscando la compra nº {numeroCompra}...")
    imprimirTicket(numeroCompra) # Llamo a la función imprimirTicket
para mostrar el ticket de la compra buscada.

# Función para visualizar los clientes registrados y realizar
búsquedas de clientes mediante su campo único (DNI).
def verClientes():

```

```

try:
    print("\nElige qué quieres consultar:\n1- Ver todos los
clientes registrados\n2- Buscar un cliente")
    opcion = int(input("Selecciona una opción: "))
    if opcion == 1:
        with open("registros.txt", "r") as archivo: # Asumo que
existe el archivo registros ya que el cliente que consulta esta opción
ha tenido que registrarse antes.
            contenido = archivo.read()
            print(contenido)
    else:
        buscarCliente()
except ValueError: # Por si el cliente introduce un caracter que
no sea un número.
    print("ERROR, elige una de las opciones introduciendo el
número correspondiente.")
    verClientes()

# Función para buscar un cliente registrado a través de su DNI.
def buscarCliente():
    encontrado = False
    DNIBuscado = input("Introduce el DNI del cliente que quieres
buscar: ").upper()
    with open("registros.txt", "r") as archivo: # Busco el DNI en el
archivo registros.
        for linea in archivo:
            nombreArchivo, DNIarchivo = linea.strip().split(":")
            if DNIBuscado == DNIarchivo: # Si el DNI buscado coincide
con uno registrado, muestro los datos del cliente.
                encontrado = True
                print("-----")
                print(f"Cliente encontrado:\nNombre del cliente:
{nombreArchivo} (DNI: {DNIarchivo})")
            if encontrado == False: # Si el DNI buscado no está registrado.
                print("-----")
                print("Lo siento, no existe ningún cliente registrado con ese
DNI.")

```



```

# Función para cerrar el programa.
def salir():
    print("Gracias por todo, ¡Hasta pronto! :)")
    exit() # La función exit cierra directamente el programa.

#####
# Ejecución del menú inicial como función principal.
# El resto de funciones se ejecutarán en cadena dependiendo de lo que
seleccione el usuario.
menuInicial()

```

## EJECUCIÓN del programa:

NOTA: Voy a intentar mostrar todas las variantes del código con ejemplos de casos realistas, de todas formas, el archivo python está en el repositorio de github para poder probar el código en cualquier momento.

Inicio el programa y registro el primer usuario “Pepe”.

```

-----
¡Hola! bienvenid@, ¿Qué quieres hacer?:
1- Iniciar Sesión
2- Registrarme
3- Salir
Selecciona una opción: 2
Introduce tu nombre: pepe
Introduce tu DNI: 12345678A
-----
¡Gracias por registrarte, Pepe!
-----
MENÚ:
1- Realizar una compra
2- Buscar una compra
3- Ver clientes
4- Cerrar sesión
5- Salir
Selecciona una opción: 

```

Pepe ha escrito su nombre en minúsculas pero se guardará con la primera letra en mayúsculas gracias a la función .title().

Tras registrarse, pepe accede a la aplicación y se le muestra el menú.

Pepe va a hacer la primera compra

```
-----
MENÚ:
1- Realizar una compra
2- Buscar una compra
3- Ver clientes
4- Cerrar sesión
5- Salir
Selecciona una opción: 1

¿Qué producto quieres comprar?
{1: 'Altavoces', 2: 'Auriculares', 3: 'Teclado', 4: 'Raton', 5: 'Alfombrilla', 6: 'Monitor'}
Introduce el número de un producto para añadirlo a la cesta
(Introduce 7 PARA TERMINAR la compra): 1
-----
Has comprado: Altavoces
-----

¿Qué producto quieres comprar?
{1: 'Altavoces', 2: 'Auriculares', 3: 'Teclado', 4: 'Raton', 5: 'Alfombrilla', 6: 'Monitor'}
Introduce el número de un producto para añadirlo a la cesta
(Introduce 7 PARA TERMINAR la compra): 4
-----
Has comprado: Raton
-----

¿Qué producto quieres comprar?
{1: 'Altavoces', 2: 'Auriculares', 3: 'Teclado', 4: 'Raton', 5: 'Alfombrilla', 6: 'Monitor'}
Introduce el número de un producto para añadirlo a la cesta
(Introduce 7 PARA TERMINAR la compra): 7
-----
¡Gracias por tu compra!
Aquí tienes el ticket de la compra ;).
-----
TICKET:
Número de compra: 566
Nombre: Pepe
DNI: 12345678A
Productos Comprados: ['Altavoces', 'Raton']
-----
```

Al finalizar la compra, se muestra el ticket con los datos del cliente, un número de compra único (nunca se repetirá) y los productos comprados. Después, el programa vuelve al menú.

Pepe cierra sesión:

```
-----
MENÚ:
1- Realizar una compra
2- Buscar una compra
3- Ver clientes
4- Cerrar sesión
5- Salir
Selecciona una opción: 4
Has cerrado sesión, hasta pronto Pepe.
-----
¡Hola! bienvenid@, ¿Qué quieres hacer?:
1- Iniciar Sesión
2- Registrarme
3- Salir
Selecciona una opción: █
```

Pepe selecciona salir y cierra el programa:

```
-----  
¡Hola! bienvenid@, ¿Qué quieres hacer?:  
1- Iniciar Sesión  
2- Registrarme  
3- Salir  
Selecciona una opción: 3  
Gracias por todo, ¡Hasta pronto! :)
```

Vuelvo a ejecutar el programa

Marta se registra introduciendo su nombre y DNI:

```
-----  
¡Hola! bienvenid@, ¿Qué quieres hacer?:  
1- Iniciar Sesión  
2- Registrarme  
3- Salir  
Selecciona una opción: 2  
Introduce tu nombre: Marta  
Introduce tu DNI: 87654321b  
-----  
¡Gracias por registrarte, Marta!  
-----  
MENÚ:  
1- Realizar una compra  
2- Buscar una compra  
3- Ver clientes  
4- Cerrar sesión  
5- Salir  
Selecciona una opción: █
```

Marta hace una compra:

```
MENÚ:
1- Realizar una compra
2- Buscar una compra
3- Ver clientes
4- Cerrar sesión
5- Salir
Selecciona una opción: 1

¿Qué producto quieres comprar?
{1: 'Altavoces', 2: 'Auriculares', 3: 'Teclado', 4: 'Raton', 5: 'Alfombrilla', 6: 'Monitor'}
Introduce el número de un producto para añadirlo a la cesta
(Introduce 7 PARA TERMINAR la compra): 5
-----
Has comprado: Alfombrilla
-----

¿Qué producto quieres comprar?
{1: 'Altavoces', 2: 'Auriculares', 3: 'Teclado', 4: 'Raton', 5: 'Alfombrilla', 6: 'Monitor'}
Introduce el número de un producto para añadirlo a la cesta
(Introduce 7 PARA TERMINAR la compra): 6
-----
Has comprado: Monitor
-----

¿Qué producto quieres comprar?
{1: 'Altavoces', 2: 'Auriculares', 3: 'Teclado', 4: 'Raton', 5: 'Alfombrilla', 6: 'Monitor'}
Introduce el número de un producto para añadirlo a la cesta
(Introduce 7 PARA TERMINAR la compra): 7
-----
¡Gracias por tu compra!
Número de compra: 941
Aquí tienes el ticket de la compra ;).
-----
TICKET:
Número de compra: 941
Nombre: Marta
DNI: 87654321B
Productos Comprados: ['Alfombrilla', 'Monitor']
-----
MENÚ:
1- Realizar una compra
2- Buscar una compra
3- Ver clientes
4- Cerrar sesión
```

Al terminar la compra obtiene su ticket y vuelve al menú.

Marta selecciona salir y cierra el programa:

```
MENÚ:  
1- Realizar una compra  
2- Buscar una compra  
3- Ver clientes  
4- Cerrar sesión  
5- Salir  
Selecciona una opción: 5  
Gracias por todo, ¡Hasta pronto! :)
```

Un nuevo cliente (Juan) se registra, pero intenta introducir un número que no es su DNI:

```
-----  
¡Hola! bienvenid@, ¿Qué quieres hacer?:  
1- Iniciar Sesión  
2- Registrarme  
3- Salir  
Selecciona una opción: 2  
Introduce tu nombre: juan  
Introduce tu DNI: 1234  
El DNI debe tener 9 caracteres.  
Introduce tu DNI: 4321  
El DNI debe tener 9 caracteres.  
Introduce tu DNI: 66787563c  
-----  
¡Gracias por registrarte, Juan!  
-----  
MENÚ:  
1- Realizar una compra  
2- Buscar una compra  
3- Ver clientes  
4- Cerrar sesión  
5- Salir  
Selecciona una opción: █
```

Finalmente Juan se registra, pero se tiene que ir y cierra sesión.

```
Selecciona una opción: 4  
Has cerrado sesión, hasta pronto Juan.  
-----  
¡Hola! bienvenid@, ¿Qué quieres hacer?:  
1- Iniciar Sesión  
2- Registrarme  
3- Salir  
Selecciona una opción: █
```

Llega de nuevo Pepe. Como ya es usuario del programa, inicia sesión.

```
Selecciona una opción: 1
Introduce tu nombre: pepe
Introduce tu DNI: 13245678A
-----
Nombre o DNI incorrectos, prueba a registrarte
-----
¡Hola! bienvenid@, ¿Qué quieres hacer?:
1- Iniciar Sesión
2- Registrarme
3- Salir
Selecciona una opción: █
```

Pepe se equivoca al escribir su DNI, pero sabe que está registrado.  
Vuelve a seleccionar iniciar sesión, introduce los datos correctamente y entra en la aplicación.

```
-----
Nombre o DNI incorrectos, prueba a registrarte
-----
¡Hola! bienvenid@, ¿Qué quieres hacer?:
1- Iniciar Sesión
2- Registrarme
3- Salir
Selecciona una opción: 1
Introduce tu nombre: Pepe
Introduce tu DNI: 12345678a
-----
Bienvenid@ de nuevo, Pepe
-----
MENÚ:
1- Realizar una compra
2- Buscar una compra
3- Ver clientes
4- Cerrar sesión
5- Salir
Selecciona una opción: █
```

Pepe quiere buscar la compra que hizo.

Selecciona buscar una compra y busca su compra mediante el número de compra.

```
-----  
MENÚ:
```

```
1- Realizar una compra
```

```
2- Buscar una compra
```

```
3- Ver clientes
```

```
4- Cerrar sesión
```

```
5- Salir
```

```
Selecciona una opción: 2
```

```
Introduce el número de la compra que deseas buscar: 566
```

```
Buscando la compra nº 566...
```

```
Aquí tienes el ticket de la compra ;).
```

```
-----  
TICKET:
```

```
Número de compra: 566
```

```
Nombre: Pepe
```

```
DNI: 12345678A
```

```
Productos Comprados: ['Altavoces', 'Raton']
```

```
-----  
MENÚ:
```

```
1- Realizar una compra
```

```
2- Buscar una compra
```

```
3- Ver clientes
```

```
4- Cerrar sesión
```

```
5- Salir
```

```
Selecciona una opción: 
```

Pepe selecciona ver clientes y decide ver todos los clientes registrados (esta opción es muy poco realista, pero pongamos que pepe trabaja para la empresa y por ello tiene acceso a esta función)

```
-----  
MENÚ:  
1- Realizar una compra  
2- Buscar una compra  
3- Ver clientes  
4- Cerrar sesión  
5- Salir  
Selecciona una opción: 3  
  
Elige qué quieres consultar:  
1- Ver todos los clientes registrados  
2- Buscar un cliente  
Selecciona una opción: 1  
Pepe:12345678A  
Marta:87654321B  
Juan:66787563C  
-----
```

Después, selecciona ver clientes y busca un cliente por su campo único (DNI).

```
-----  
MENÚ:  
1- Realizar una compra  
2- Buscar una compra  
3- Ver clientes  
4- Cerrar sesión  
5- Salir  
Selecciona una opción: 3  
  
Elige qué quieres consultar:  
1- Ver todos los clientes registrados  
2- Buscar un cliente  
Selecciona una opción: 2  
Introduce el DNI del cliente que quieres buscar: 87654321B  
-----  
Cliente encontrado:  
Nombre del cliente: Marta (DNI: 87654321B)  
-----
```



NOTA: hay muchos controles de errores que no aparecen en las capturas de la ejecución pero están en el código, no muestro la comprobación de todos porque el pdf sería demasiado extenso.

Adjunto capturas de los archivos de registros de datos que se han creado durante la ejecución:

```
≡ registros.txt ×  
PROGRAMACION > ≡ registros.txt  
1  Pepe:12345678A  
2  Marta:87654321B  
3  Juan:66787563C  
4
```

```
≡ compras.txt ×  
PROGRAMACION > ≡ compras.txt  
1  566|Pepe|12345678A|['Altavoces', 'Raton']  
2  941|Marta|87654321B|['Alfombrilla', 'Monitor']  
3
```

Dejo aquí de nuevo el enlace a github:

<https://github.com/IrisCampusFP/Programacion.git>

## BIBLIOGRAFÍA Y WEBGRAFÍA

### References

*3.13.0 documentation*. (n.d.). Python.org. Retrieved November 20, 2024, from <https://docs.python.org/3/>

*Python tutorial*. (n.d.). W3schools.com. Retrieved November 20, 2024, from <https://www.w3schools.com/python/default.asp>

### Tecnologías utilizadas

- **Visual Studio Code (VS Code):** 

Extensions, L. M. A. (n.d.). *Visual Studio Code - code editing. Redefined*.

Visualstudio.com. Retrieved November 20, 2024, from

<https://code.visualstudio.com/>

- **Python:** 

*Python 3.11: descarga e instalación gratuitas en Windows*. (n.d.). Microsoft

Apps. Retrieved November 20, 2024, from

<https://apps.microsoft.com/detail/9nrwmjp3717k?hl=es-es&gl=US>