



HITO PROGRAMACION

Seguimiento del hito

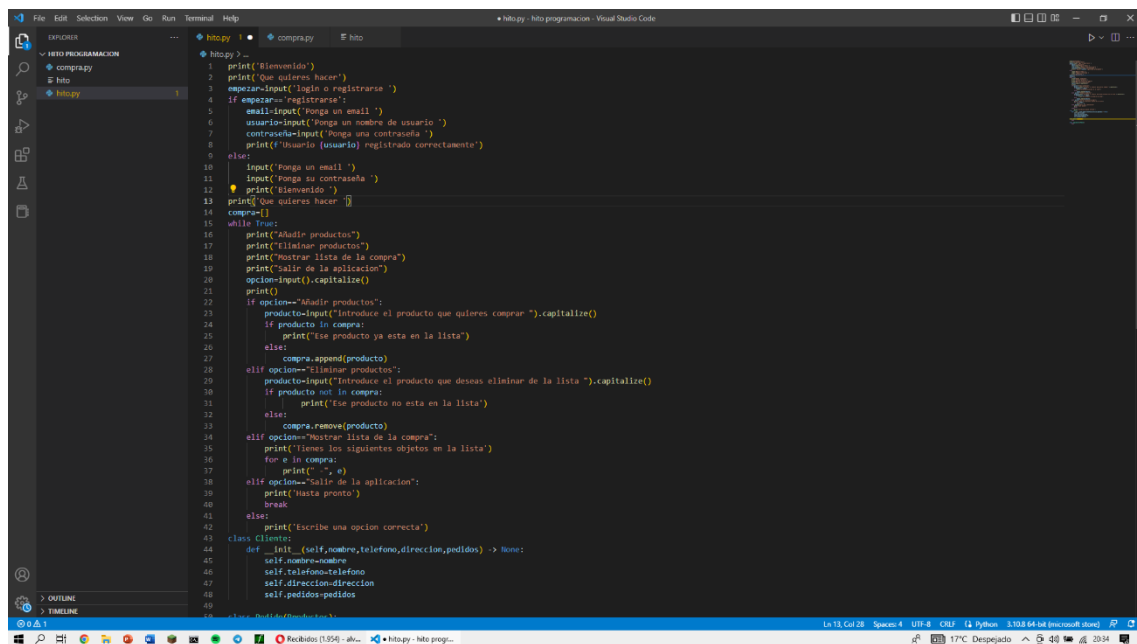
Fase 1: Diagrama de flujos ya entregado

Fase 2:

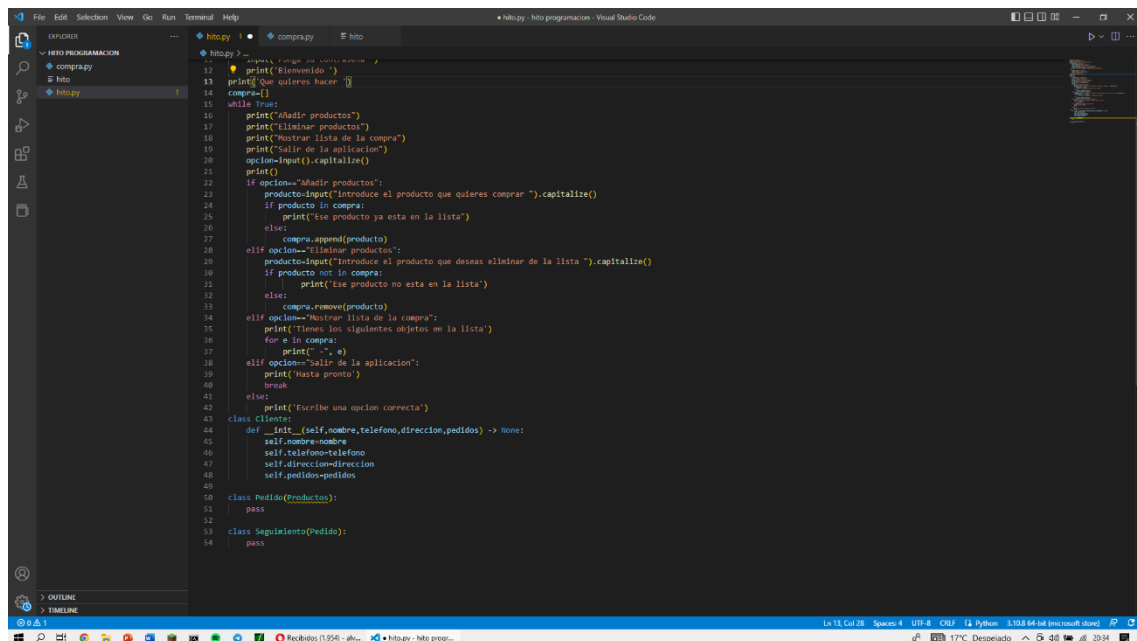
Dia 14 de Noviembre de 2022

He conseguido desarrollar que te pida un email para el registro, un nombre de usuario y su contraseña.

Te da 4 opciones para añadir, eliminar productos o ver la lista de los mismos y para salir de la aplicación cuando quieras.



```
1 print('Bienvenido')
2 print('Que quieres hacer')
3 empezar=input('login o registrarse ')
4 if empezar=="registrarse":
5     email=input('Ponga un email ')
6     usuario=input('Ponga un nombre de usuario ')
7     contraseña=input('Ponga una contraseña ')
8     print('Usuario (usuario) registrado correctamente')
9 else:
10    input('Ponga un email ')
11    input('Ponga su contraseña ')
12    print('Bienvenido ')
13    print('Que quieres hacer ')
14    compra=[]
15    while True:
16        print("Añadir productos")
17        print("Eliminar productos")
18        print("Mostrar lista de la compra")
19        print("Salir de la aplicacion")
20        opcion=input().capitalize()
21        print()
22        if opcion=="Añadir productos":
23            producto=input("Introduce el producto que quieres comprar ").capitalize()
24            if producto in compra:
25                print("Ese producto ya esta en la lista")
26            else:
27                compra.append(producto)
28        elif opcion=="Eliminar productos":
29            producto=input("Introduce el producto que deseas eliminar de la lista ").capitalize()
30            if producto not in compra:
31                print("Ese producto no esta en la lista")
32            else:
33                compra.remove(producto)
34        elif opcion=="Mostrar lista de la compra":
35            print("ienes los siguientes objetos en la lista")
36            for e in compra:
37                print(" ", e)
38        elif opcion=="Salir de la aplicacion":
39            print("hasta pronto")
40            break
41        else:
42            print("Escribe una opcion correcta")
43    class Cliente:
44        def __init__(self,nombre,telefono,direccion,pedidos) -> None:
45            self.nombre=nombre
46            self.telefono=telefono
47            self.direccion=direccion
48            self.pedidos=pedidos
49
```



```
50 class Producto:
51     pass
52
53 class Pedido:
54     pass
55
56 class Seguimiento(Pedido):
57     pass
58
```

Dia 17 de Noviembre de 2022

He incluido la primera clase llamada pedido en la que voy a pedir datos relacionados con el seguimiento del mismo. Para ello he tenido que crear un nuevo elemento en el menú llamado realizar pedido.

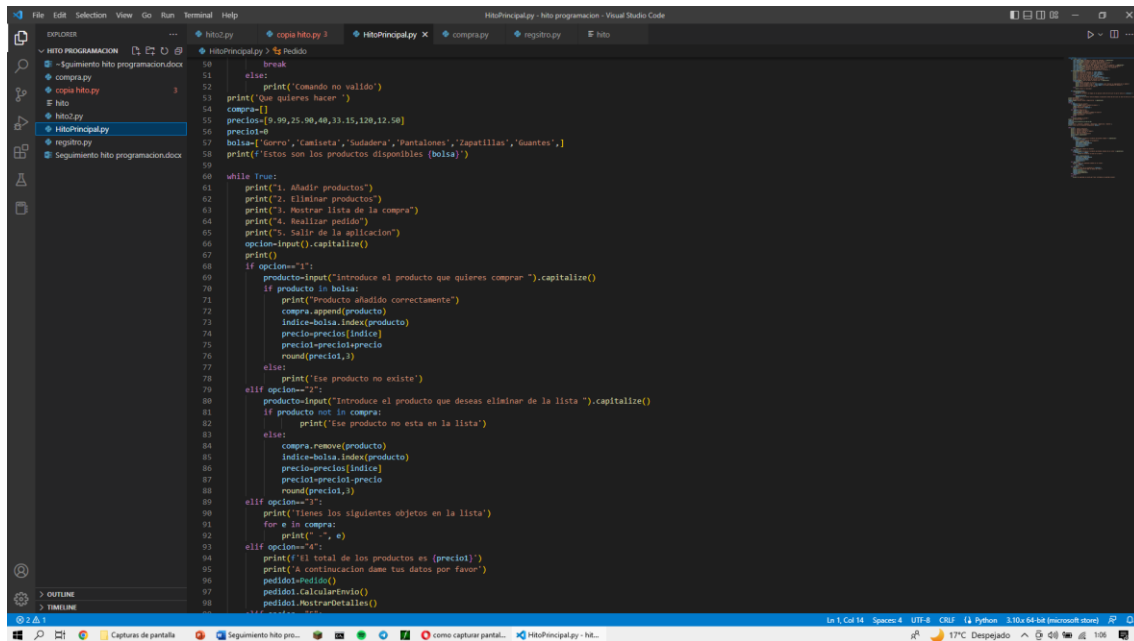
También he hecho que se pueda ver la bolsa de los productos que se venden en la tienda con su precio para una mejor elección del comprador y le he dado precios a los productos de la bolsa para realizar los cálculos al tramitar el pedido.

The image shows a Windows desktop environment with a code editor open. The editor window is titled 'hito.py' and contains Python code for a shopping cart application. The code is written in Spanish and includes comments. The code defines a class 'Pedido' and uses it to handle user input for products, quantities, and contact information. The code is organized into several sections, including a main loop for adding and removing products, and a section for saving the order details. The editor interface includes a sidebar on the left with a file explorer showing the project structure, a main editor area with line numbers, and a bottom status bar with system information. The status bar shows the current file is 'hito.py', the editor is using Python 3.7.3, and the system clock is 19:39. The taskbar at the bottom shows several open applications, including a web browser, a file explorer, and the code editor itself.

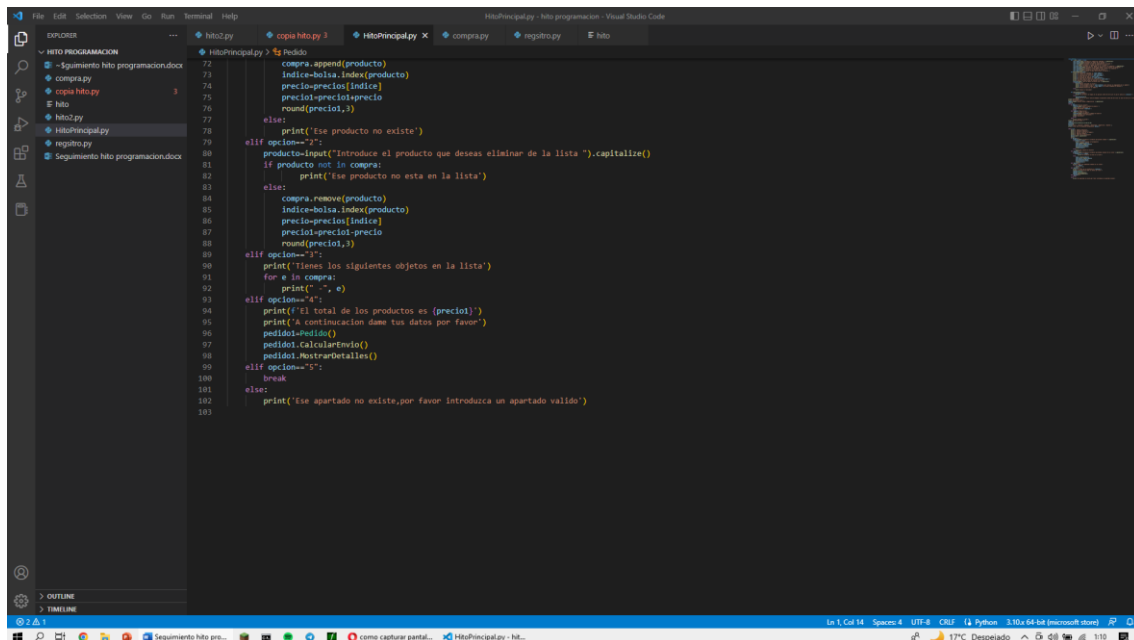
Dia 18 de Noviembre de 2022

The screenshot shows a Python IDE with a file named `hito.py` open. The code is a Flask application for an e-commerce site. It includes routes for product listing, cart management, and user authentication. The code is written in Python and uses Flask's `Flask` and `request` objects. The routes are defined using `@app.route` and the views are implemented as functions. The code is organized into several sections: a `def pedir` function for handling the 'Pedir' route, a `def calcularEnvio` function for calculating shipping costs, and a `def login` function for handling the 'login' route. The code also includes a `def logout` function for handling the 'logout' route. The code is written in a clear and concise style, using comments to explain the logic of the code. The IDE interface shows the file explorer on the left, the code editor in the center, and the output console on the right. The status bar at the bottom shows the current file is `hito.py` and the current line is 1.

```
1 class Pedido:
2     def __init__(self) -> None:
3         self.nombre=input('Introduce el nombre del comprador ').capitalize()
4         self.tarjeta=input('Introduce el numero de la tarjeta ')
5         self.fecha=input('Introduce la fecha de caducidad de la tarjeta ')
6         self.cvv=input('Introduce el codigo secreto de la tarjeta ')
7         self.pais=input('Introduce el pais donde se van a enviar los productos ').capitalize()
8         self.municipio=input('Introduce el municipio donde se van a enviar los productos ')
9         self.calle=input('Introduce la calle donde se van a enviar los productos ')
10        self.telefono=input('Introduce un telefono de contacto ')
11        self.email=input('Introduce un email para enviar la factura ')
12
13    def MostrarDetalle(self):
14        print("El nombre del comprador es (self.nombre):")
15        print("El número de la tarjeta es (self.tarjeta):")
16        print("El municipio del envío es (self.municipio):")
17        print("La calle de la casa es (self.calle):")
18        print("El teléfono de contacto es (self.telefono):")
19        print("El email para el envío de la factura es (self.email):")
20        print("Hecho los datos antes de realizar la compra")
21        eleccion=input("¿don los datos correctos? S/N ").capitalize()
22        if eleccion=="S":
23            print("Pedido realizado con éxito")
24            print("Se ha enviado un sms al (self.telefono) para realizar un seguimiento de su pedido")
25            print("Se ha enviado un email a (self.email) en el que se adjunta la factura")
26            print("Muchas gracias por su compra")
27        else:
28            print("Compra no realizada")
29
30    def CalcularEnvio(self):
31        if self.pais=="España":
32            print("Al ser envío en España no se aplican costes de envío por lo que el precio es (precio)€")
33        else:
34            precio=precio*1.15
35            print("Al ser un envío fuera de España se aplicaran costes de envío por un valor de 15€ por lo que el precio final será de (precio2)€")
36
37    def login(self):
38        print("Bienvenido")
39        print("Que quieres hacer")
40        eleccion=input("Iniciar sesion o Registrarse ").capitalize()
41        if eleccion=="Iniciar sesion":
42            usuario=input("Dime tu nombre de usuario ")
43            contraseña=input("Dime tu contraseña ")
44            print("Bienvenido (usuario)!")
45            break
46        elif eleccion=="Registrarse":
47            print("Introduce tu email")
48            usuario=input("Introduce tu nombre de usuario ")
49            contraseña=input("Introduce una contraseña ")
50            print("Bienvenido")
51            break
```



```
50         break
51     elif opcion == "2":
52         print("Comando no valido")
53     print("Que quieres hacer ")
54     compra=[]
55     precios=[99,25,99,49,33,15,120,12,58]
56     precios1=0
57     bolsa=["Gorro","Camiseta","Sudadera","Pantalones","Zapatillas","Guantes"]
58     print("Estos son los productos disponibles (bolsa)")
59
60 while True:
61     print("1. Agadir productos")
62     print("2. Eliminar productos")
63     print("3. Mostrar lista de la compra")
64     print("4. Realizar pedido")
65     print("5. Salir de la aplicacion")
66     opcion=input().capitalize()
67     print()
68     if opcion=="1":
69         producto=input("Introduce el producto que quieres comprar ").capitalize()
70         if producto in bolsa:
71             print("Producto añadido correctamente")
72             compra.append(producto)
73             indice=bolsa.index(producto)
74             precios1=precios[indice]
75             precio=precios1-precio
76             round(precio,1)
77         else:
78             print("Ese producto no existe")
79     elif opcion=="2":
80         producto=input("Introduce el producto que deseas eliminar de la lista ").capitalize()
81         if producto not in compra:
82             print("Ese producto no esta en la lista")
83         else:
84             compra.remove(producto)
85             indice=bolsa.index(producto)
86             precios1=precios[indice]
87             precio=precios1-precio
88             round(precio,1)
89     elif opcion=="3":
90         print("Muestrame los siguientes objetos en la lista")
91         for e in compra:
92             print(" ", e)
93     elif opcion=="4":
94         print("El total de los productos es (precio1)")
95         print("A continuacion dame tus datos por favor")
96         pedido=pedido()
97         pedido.calcularEnvio()
98         pedido.mostrarDetalles()
99         break
100     else:
101         print("Ese apartado no existe,por favor introduce un apartado valido")
102     print()
```



```
72     compra.append(producto)
73     indice=bolsa.index(producto)
74     precios1=precios[indice]
75     precio=precios1-precio
76     round(precio,1)
77 else:
78     print("Ese producto no existe")
79 elif opcion=="2":
80     producto=input("Introduce el producto que deseas eliminar de la lista ").capitalize()
81     if producto not in compra:
82         print("Ese producto no esta en la lista")
83     else:
84         compra.remove(producto)
85         indice=bolsa.index(producto)
86         precios1=precios[indice]
87         precio=precios1-precio
88         round(precio,1)
89 elif opcion=="3":
90     print("Muestrame los siguientes objetos en la lista")
91     for e in compra:
92         print(" ", e)
93 elif opcion=="4":
94     print("El total de los productos es (precio1)")
95     print("A continuacion dame tus datos por favor")
96     pedido=pedido()
97     pedido.calcularEnvio()
98     pedido.mostrarDetalles()
99 elif opcion=="5":
100     break
101 else:
102     print("Ese apartado no existe,por favor introduce un apartado valido")
103     print()
```

He acabado el hito. En el día de hoy solucioné el problema de los precios ya que no sabía cómo acumular los precios pero al final lo he conseguido. También he añadido mejoras como la del inicio de sesión o registro ya que si ponías cualquier cosa te dejaba acceder al programa.

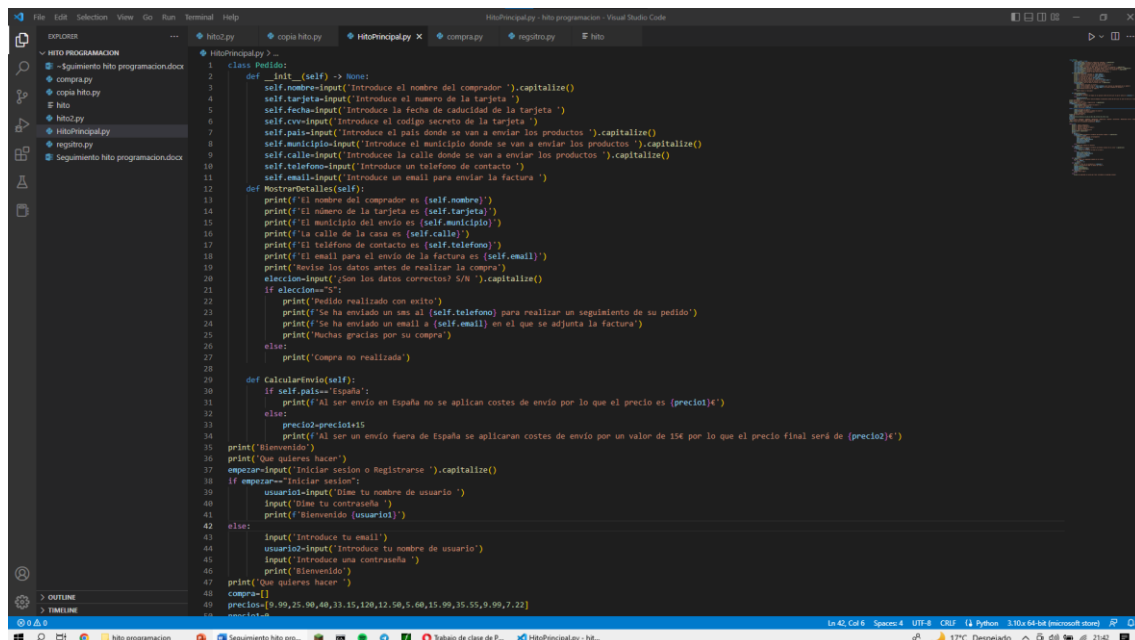
He incluido la clase llamada productos en la cual nos encontramos que pide por consola detalles al usuario usando el parámetro 'self' y más abajo nos encontramos el 'MostrarDetalles' el cual nos muestra los detalles pedidos al usuario para que al momento de realizar la compra los pueda comprobar y asegurarse de que los datos que ha introducido son

correctos. También he añadido la mejora de que solo puedas añadir a la lista de la compra productos que estén en la bolsa.

Día 20 de Noviembre de 2022.

Realizo las ultimas comprobaciones para vigilar que todo el programa funcione bien y no tenga ningún punto flaco. La idea principal que tenia que era la de entrar en un bucle en el momento del inicio de sesión o registro no la voy a llevar a cabo debido a que se la manera en la que tengo que implementar el while para entrar en un bucle pero una vez que fallo no me saca del bucle por lo que a pesar de eso he tenido que volver atrás e implementar un if simple el cual se que si pones cualquier otra cosa que no sea login o registrarse te va a dar acceso al siguiente menú programado. Ahora voy a realizar las ultimas comprobaciones y voy a añadir nuevos productos, nuevos precios y voy a testear el programa hasta ponerlo en las ocasiones más extremas de muchísimos productos añadidos a la cesta, muchos añadir y eliminar productos para ver como reacciona y que puedo mejorar al respecto.

He añadido un mensaje de despedida para cuando salgas de programa y cambiado el método de inicio de sesión.



```
1 class Pedido:
2     def __init__(self) -> None:
3         self.nombre=input('Introduce el nombre del comprador ').capitalize()
4         self.tarjeta=input('Introduce el numero de la tarjeta ')
5         self.fecha=input('Introduce la fecha de caducidad de la tarjeta ')
6         self.cvv=input('Introduce el codigo secreto de la tarjeta ')
7         self.pais=input('Introduce el pais donde se van a enviar los productos ').capitalize()
8         self.municipio=input('Introduce el municipio donde se van a enviar los productos ').capitalize()
9         self.calle=input('Introduce la calle donde se van a enviar los productos ').capitalize()
10        self.telefono=input('Introduce un telefono de contacto ')
11        self.email=input('Introduce un email para enviar la factura ')
12
13    def MostrarDetalles(self):
14        print(f'El nombre del comprador es {self.nombre}')
15        print(f'El numero de la tarjeta es {self.tarjeta}')
16        print(f'El municipio del envio es {self.municipio}')
17        print(f'La calle de la casa es {self.calle}')
18        print(f'El telefono de contacto es {self.telefono}')
19        print(f'El email para el envio de la factura es {self.email}')
20        print('Revisa los datos antes de realizar la compra')
21        eleccion=input('Son los datos correctos? S/N ').capitalize()
22        if eleccion=="S":
23            print('Pedido realizado con exito')
24            print(f'Se ha enviado un sms al {self.telefono} para realizar un seguimiento de su pedido')
25            print(f'Se ha enviado un email a {self.email} en el que se adjunta la factura')
26            print('Muchas gracias por su compra')
27        else:
28            print('Compra no realizada')
29
30    def CalcularEnvio(self):
31        if self.pais=="España":
32            print('Al ser envio en España no se aplican costes de envio por lo que el precio es {precios1}€')
33        else:
34            precios2=precios1+5
35            print('Al ser un envio fuera de España se aplicaran costes de envio por un valor de 15€ por lo que el precio final será de {precios2}€')
36
37    print("Bienvenido")
38    print("Que quieres hacer")
39    esperar=input('Iniciar sesion o Registrarse ').capitalize()
40    if esperar=="Iniciar sesion":
41        usuario=input('Dime tu nombre de usuario ')
42        input('Dime tu contraseña ')
43        print("Bienvenido {usuario}")
44    else:
45        input('Introduce tu email ')
46        usuario=input('Introduce tu nombre de usuario ')
47        input('Introduce una contraseña ')
48        print("Bienvenido ")
49    print("Que quieres hacer ")
50    compra=[]
51    precios=[9.99,25.90,40,33.15,120,12.50,5.60,15.99,35.55,9.99,7.22]
```


primera letra la va a pasar a mayúscula. De esta forma queda la interfaz más limpia y con una imagen de profesionalidad.

También he creado otro método el cual se llama "CalcularEnvío" el cual se encarga de ir al método self.pais y compararlo con la palabra España para que si la palabra es España en el self.pais no se nos apliquen gastos de envío al ser un envío nacional y que si es al contrario que se nos calcule un segundo precio sumándole 15 euros fijos de gastos de envío al precio final de los productos. He decidido poner la clase Pedido al inicio del desarrollo para que así pueda tener acceso a todos los datos del self en cualquier momento.

Una vez esto el programa arranca y lo primero que nos comenta es la bienvenida y que queremos hacer, si iniciar sesión o registrarnos. He tenido que usar un if. Se que no es la mejor opción ya que ponga lo que ponga siempre voy a tener acceso al menú pero el bucle no lo he conseguido implementar de una manera que me gustara así que tuve que dar un paso atrás en ese sentido. He creado la variable empezar para poder utilizar el if. Si dices iniciar sesión te va a pedir un nombre de usuario y una contraseña y en el caso contrario te va a pedir un email, un nombre de usuario y una contraseña.

A continuación nos muestra la lista de la compra con los objetos que tenemos disponible para comprar y se nos despliega un menú en el cual tenemos 5 opciones. Este menú lo he creado con el método while. Si pones cualquier otra cosa que no sea un menú que este disponible te va a decir que ese apartado no existe y te va a volver a enseñar el menú. Para acceder a estos apartados no es necesario escribir de forma exacta el nombre del menú sino que mediante el uso del número del apartado del menú y un punto conseguimos que con solo escribir el número del apartado del menú al cual queremos ir nos lleve. En el primer apartado el cual tengo para añadir productos a la lista lo he creado de manera que solo puedas añadir productos que estén en la lista y que si no están en la lista te diga que esos productos no están en la lista. La suma del precio de estos productos es lo que más me ha costado ya que no daba con la clave la cual era sobrescribir el valor de precio 1 y no tener que crear otro variable de precio.

El método append lo que he conseguido es que teniendo dos listas, productos y precios, ordenadas, es decir, sudadera producto 1 y el precio de la sudadera es 1, así de forma seguida. Lo que he conseguido que es de la variable producto la cual se lo decimos nosotros vaya a la lista bolsa, lo busque y lo almacene en la variable índice. Luego de ahí se va a la lista de los precios y según la posición donde allá encontrado el producto escogido va al mismo apartado en la lista de los precios. De ahí lo guarda en la variable precio y luego tengo un acumulador el cual tengo llamado precio1 y lo tengo con valor 0. Lo que hago es que cuando tenga el precio del producto lo sume al valor 0 que tiene precio1 y se acumule. Eso lo hace cada vez que añadamos un producto nuevo acumulando el precio en la variable 1. Para luego no tener muchos decimales en la suma he puesto que redondee el precio a 3 decimales.

En el segundo apartado tengo la opción para eliminar los productos guardados en la variable compra. Con el método remove elimino la variable producto de la bolsa y hago lo mismo que cuando añado productos lo único que cuando los elimino de la lista pues le resto el precio.

La opción 3 tenemos la posibilidad de ver de forma estilizada los productos que tenemos en la lista.

La opción 4 es la opción de completar el pedido y aquí es cuando entra en acción la clase del principio que cree la cual te pide los datos como el nombre, el número de la tarjeta, el país, el teléfono o un email entre otros muchos datos para poder llevar a cabo una compra.

Los datos como el nombre, el país, el municipio y la calle le he implementado la extensión `.capitalize()` para que la primera letra de la primera palabra que escriba se ponga automáticamente en mayúscula y la impresión sea mas agradable para el usuario.

Cuando le enseño los datos le comento al usuario que dependiendo de su país se le van a aplicar gastos de envío. Para esto uso otra vez el `self.pais` el cual lo comparo con la palabra España.

A continuación cuando el usuario ha puesto sus datos en la clase constructora los muestro en el siguiente apartado para que así el usuario se pueda asegurar de que los datos que ha introducido son los correctos. Para ello le doy la posibilidad de elegir si quiere continuar con la compra preguntando si los datos que ha introducido son los correctos dando como posibilidad para responder una `s` mayúscula o una `n` mayúscula. Para ello creo la variable `elección`, con un `input` y al final poniendo la extensión `.capitalize()` para dar esa sensación de profesionalidad y limpieza. Si la elección del usuario es que `si` pues entonces se le muestra al usuario el precio final de la compra y dos mensajes mostrando que se le va a enviar un SMS al número de teléfono que puso en el `self.telefono` para que así pueda realizar un seguimiento del pedido. A su vez le muestro por pantalla que se le ha enviado un correo electrónico al email que nos proporcionó al inicio del pedido. Para esto me baso en el `self.email` el cual lo imprimo. Por último muestro por pantalla que muchas gracias por su compra y en el caso de que haya seleccionado otro valor que no sea la `s` que muestre compra no realizada. Una vez que hayamos o no hecho la compra nos devuelve al menú en el cual podemos volver a hacer una compra o salir. Por último tenemos la opción 5 la cual nos saca del bucle con el uso del `break` mostrando por último un mensaje diciéndonos hasta luego.