Universidad Rafael Landivar Campus Quetzaltenango

Programación Avanzada S2

Ing. Jorge Tello

Manual de Programador

José Castañeda – 1534422

Jason Gutiérrez – 1624622

Miguel Macario - 1597421

Quetzaltenango 10/11/2023

INTRODUCCIÓN

El presente manual, servirá para que el programador tenga conocimiento sobre el funcionamiento de cada línea de código y de las características acerca del programa, para así poder brindarle una guía exacta, la cual servirá para que sea de conocimiento para el programador sobre cada linea de codigo, por si en algún caso desea hacer algunos cambios en el programa, o bien simplemente comprender el código.

```
import queue
from threading import Timer
```

Primero importamos el queve que nos sirven para las colas, e importamos los archivos del Timer para realizar un controlador de tiempo.

```
1 usage  new *
class RepeatTimer(Timer):
    new *
    def run(self):
        while not self.finished.wait(self.interval):
            self.function(*self.args, **self.kwargs)
```

Creamos la clase RepeatTimer para controlar el tiempo para realizar una función

```
class Productos:
    new *
    def __init__(self, id, producto, precio, stock):
        self.id = id
        self.producto = producto
        self.precio = precio
        self.stock = stock
```

Se crea la clase Productos para poder guardar datos del producto

```
class Ofertas(Productos):
    new *
    def __init__(self, id, producto, precio, stock):
        super(Ofertas,self).__init__(id, producto, precio, stock)

9 usages  new *
class Pizza(Productos):
    new *
    def __init__(self, id, producto, precio, stock):
        super(Pizza, self).__init__(id, producto, precio, stock)

7 usages  new *
class Bebidas(Productos):
    new *
    def __init__(self, id, producto, precio, stock):
        super(Bebidas, self).__init__(id, producto, precio, stock)

4 usages  new *
class Complementos(Productos):
    new *
    def __init__(self, id, producto, precio, stock):
        super(Complementos, self).__init__(id, producto, precio, stock)
```

La clase productos hereda a 4 clases para que cada una tenga su propia creación de un nuevo producto

```
1 usage  new *
class ListaClientes:
    new *
    def __init__(self):
        self.pila_clientes = []

    1 usage  new *
    def agregar_clientes(self, clientes):
        self.pila_clientes.append(clientes)

    1 usage  new *
    def lista_vacia(self):
        vacia = len(self.pila_clientes)
        if vacia == 0:
            return 0

    1 usage  new *
    def mostrar(self):
        print(self.pila_clientes)
```

La clase ListaCliestes nos ayuda a guardar en una pila los clientes para futuras promociones además de mostrar luego los clientes guardados, si la pila está vacía mostrará que no hay clientes aun

```
class ModificacionOfertas:
    new *
    def __init__(self):
        self.pila_oferta=[]
    1 usage  new *
    def agregar_oferta(self,ofertas):
        self.pila_oferta.append(ofertas)
1 usage  new *
class ModificacionPizza:
    new *
    def __init__(self):
        self.pila_pizza = []
    8 usages  new *
    def agregar_pizza(self, pizzas):
        self.pila_pizza.append(pizzas)

1 usage  new *
class ModificacionBebida:
    new *
    def __init__(self):
        self.pila_bebida = []
    6 usages  new *
    def agregar_bebida(self, bebidas):
        self.pila_bebida.append(bebidas)
```

Las clases de ModificacionOferta, ModificacionPizza, ModificacionBebida, y ModificacionComplemento nos ayudan a crear una pila para cada producto de la tienda

```python
class Colas:
    new *
    def __init__(self):
        self.cola_pendiente = queue.Queue()
        self.cola_preparacion = queue.Queue()
        self.cola_listo_servir = queue.Queue()
    4 usages  new *
    def agregar_cola_pendiente(self, pedidos):
        self.cola_pendiente.put(pedidos)
    1 usage  new *
    def agregar_cola_preparacion(self):
        if not self.cola_pendiente.empty():
            pedido_preparar = self.cola_pendiente.get()
            self.cola_preparacion.put(pedido_preparar)
        else:
            print("No hay pedidos pendientes")
    1 usage  new *
    def agregar_cola_listo(self):
        if not self.cola_preparacion.empty():
            a = self.cola_preparacion.get()
            self.cola_listo_servir.put(a)
            print("El producto esta listo")
```

La clase Colas nos ayuda a hacer un seguimiento de los pedidos pendientes, los pedidos que se están preparando y el historial de pedidos que ya están listos, con esta clase utilizamos los métodos pasamos de la cola pedidos pendientes a pedidos a preparación, igualmente pasamos los pedidos de preparación a pedidos listos.

```python
def ordenamientoBurbuja(lista):
    n=len(lista)
    for i in range(n):
        for j in range(0,n-i-1):
            if lista[j]>lista[j+1]:
                lista[j],lista[j+1]=lista[j+1],lista[j]
```

Con la función ordenamientoBurbuja hacemos que los pedidos están ordenados de el producto con menor precio, hasta el producto con mayor precio

```
modi_pizza = ModificacionPizza()
modi_bebida = ModificacionBebida()
modi_complemento = ModificacionComplemento()
modi_oferta=ModificacionOfertas()
agre_cliente = ListaClientes()
```

Agregamos unas variables para poder llamar las clases que utilizaremos en la interfaz del código

```
pizza = Pizza( id: 1, producto: "Orilla Queso", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)
pizza = Pizza( id: 2, producto: "5 Carnes", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)
pizza = Pizza( id: 3, producto: "Hawaiana", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)
pizza = Pizza( id: 4, producto: "Peperoni", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)
pizza = Pizza( id: 5, producto: "Vegetariano", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)
pizza = Pizza( id: 6, producto: "Jamon", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)
pizza = Pizza( id: 7, producto: "Queso", precio: 45, stock: 5)
modi_pizza.agregar_pizza(pizza)

bebida = Bebidas( id: 1, producto: "Coca Cola", precio: 5, stock: 5)
modi_bebida.agregar_bebida(bebida)
bebida = Bebidas( id: 2, producto: "Sprite", precio: 5, stock: 5)
modi_bebida.agregar_bebida(bebida)
bebida = Bebidas( id: 3, producto: "Fanta", precio: 5, stock: 5)
modi_bebida.agregar_bebida(bebida)
bebida = Bebidas( id: 4, producto: "Powerade", precio: 5, stock: 5)
modi_bebida.agregar_bebida(bebida)
bebida = Bebidas( id: 5, producto: "Te Lipton", precio: 5, stock: 5)
modi_bebida.agregar_bebida(bebida)
```

Agregamos unos productos al menú

```
timer = RepeatTimer( interval: 22, colas.agregar_cola_listo)
timer.start()
```

Llamamos la clase de RepearTimer para ponerla en funcionamiento y que repita el método de agregar_colas_listo de la clase Colas cada 22 segundos

```
id_pizza = 7
id_bebida = 5
id_complemento = 2
id_oferta=0
```

Se agregan unos id para poder seguir sumando luego si el administrador desea agregar más productos

```
hay_oferta=0
hay_name = 0
hay_nit = 0
hay_compra=0
```

Se agregan unas variables para saber que no hay ciertas cosas agregadas aún, por lo cual se bloquean algunas funciones o dirán que aún no hay nada agregado hasta que este valor cambie.

```
while True:
    Salir_Usuario = False
    Salir_Administrador = False
    usuario = input("\033[1;34mIngresar como: \n  1. Administrador\n  2. Usuario\n  3. Salir programa\nIngrese opcion: ")
    if usuario == "1":
        contrasena = input("Ingrese la contraseña: \033[0;m")
        if contrasena == "1234":
            print("\033[1;33mUsted a ingresado como administrador")
            while not Salir_Administrador:
                print("1. Agregar productos\n2. Modificar productos\n3. Ver clientes guardos\n4. Salir")
                opcion = input("Que desea escoger: ")
                print("_____")
                if opcion == "1":
                    seguir_agregando = False
                    while not seguir_agregando:
                        print("Opciones a agregar:\n1. Pizza\n2. Bebidas\n3. Complementos\n4. Ofertas")
                        opcion_agregar = input("Que desea agregar: ")
                        if opcion_agregar == "1":
                            id_pizza = id_pizza + 1
                            produc = input("Que clase de pizza quiere agregar: ")
                            exepcion = False
                            while not exepcion:
                                try:
                                    preci = int(input("Que precio va a tener la pizza: "))
                                except ValueError:
                                    print("Solo pueden ir numeros en este apartado")
```

La parte del while True la usamos para así poder repetir el menú, usamos un if, para la parte del menú, si el usuario es igual a 1, se cumple esa línea de código, por lo tanto se procede a usar una variable contraseña, la cual servirá para que el usuario pueda ingresar. Si la contraseña es igual a 1234, se cumple esa línea de código, por lo cual se procede a imprimir el mensaje en pantalla. Seguimos con la línea de abajo while Not, que nos servirá para declarar que mientras Salir_Administrador no sea falso se seguirá con la siguiente línea de código. Imprimimos en pantalla las diferentes opciones que tendrá el Administrador. Se implementa una variable opción, para que el usuario pueda ingresar alguna de las opciones en pantalla. Seguimos con la línea de abajo, la cual declara que si la opción es igual a 1, se ejecuta la línea siguiente, en donde seguir_agregando será igual a False. La línea siguiente declara que si seguir agregando no es igual a False, se seguirán repitiendo

las líneas de código indentadas. Se implementa otra variable para la opción agregar, en la cual se le pregunta al usuario si desea agregar una pizza, bebida, complemento u oferta. Si la opción de agregar es igual a alguna de las anteriores opciones ya mencionadas, se procede a ejecutar dicha línea, de lo contrario se saliera de esa parte del programa.

```
elif opcion == "2":
    print("1. Pizzas\n2. Bebidas\n3. Complementos\n4. Ofertas")
    modificar = input("Que desea modificar: ")
    if modificar == "1":
        for pizza in modi_pizza.pila_pizza:
            print(f"{pizza.id}. Producto: {pizza.producto}")
        exepcion = False
        while not exepcion:
            try:
                pizza_modificar = int(input("Que pizza quiere modificar: "))
            except ValueError:
                print("Opcion invalida, solo se permien numeros")
            else:
                exepcion = True
        for pizza in modi_pizza.pila_pizza:
            if pizza_modificar == pizza.id:
                Modificar_nombre=input("1. Si\n2. No\n Desea modificar el nombre de la pizza: ")
                while Modificar_nombre≠"1" and Modificar_nombre≠"2":
                    Modificar_nombre=input("Opcion invalida\n1. Si\n2. No\n Desea modifica el nombre de la pizza: ")
                if Modificar_nombre=="1":
                    pizza.producto=input("Que nombre le desea poner a la pizza: ")
                exepcion = False
                while not exepcion:
                    try:
                        pizza.precio = float(input("Que precio desea ponerle: "))
                    except ValueError:
                        print("Solo se permiten numeros")
```

Administrador (Opción 2)

Para la segunda parte del código, tenemos una parte similar a la de la opción 1, la diferencia es que en esta parte se usa una variable para modificar alguna de las 4 opciones ya impresas. Usamos un for para así poder recorrer la variable de modificar en la pila de las pizzas, al momento de recorrer cada una, se imprime en pantalla el id de la pizza y el producto, media vez se termine de recorrer toda la pila, se procede a implementar una variable la cual sería excepción, que en este caso será False. Usamos nuevamente un while not, para declarar que mientras la excepción sea verdadera, se procederá a repetir las siguientes líneas de código. Pasamos con la línea de abajo, la cual sería Try, que en este caso se estaría usando para probar la línea de código indentada. Para continuar, se implementa la función except, que nos servirá para declarar, que si se encuentra el error indicado en pantalla, en este caso ValueError(Error de valor), se imprime en pantalla el mensaje en pantalla. Usamos el else para declarar que declarar que la variable excepción será verdadera (True). Seguimos con la línea de abajo, usando un for para la misma función, recorrer la pila. En caso de que la pizza a modificar sea igual al id de la pizza, se procede a usar una variable para preguntarle al usuario si desea modificar el nombre. En caso de que la opción sea diferente de 1 y diferente de 2, se procede a imprimir en pantalla un mensaje. De lo contrario se le pregunta al usuario que nombre desea ponerle a la pizza, el precio, y el stock.

```
        elif opcion == "3":
            numeros_clientes = agre_cliente.lista_vacia()
            if numeros_clientes == 0:
                print("No hay clientes aun")
            else:
                agre_cliente.mostrar()
            print("_____
        elif opcion == "4":
            print("Saliendo al menu principal...")
            Salir_Administrador = True
        else:
            print("Opcion no valida")
else:
    print("Contraseña incorrecta")
```

Administrador(Opción 3 y 4)

Para la opción número 3 de administrador, se usa una variable la cual será numeros_clientes que será igual a agregar cliente en la lista vacía, si el número de clientes es igual a 0, se procede a mostrar en pantalla 'No hay clientes aun', de lo contrario se muestran los clientes ya agregados.

En caso de la opción 4, se imprime el mensaje 'Saliendo al menú principal…' y nuestra variable salir administrador procede a ser verdadera. El else lo usamos por si el usuario ingresa algún número que no esté entre ese rango, se muestra opcion no valida, y el otro else se ejecuta si la contraseña ingresada no es correcta.

Usuario (Opción 2):

```
elif usuario == "2":
    print("\033[0;m")
    while not Salir_Usuario:
        print("\033[1;32m--Interfaz--")
        print("1. Mostar el menu\n2. Comprar\n3. Pedidos\n4. Salir ")
        opcion = input("Que desea escoger: ")
        print("_____________________________________________________________________")
        if opcion == "1":
            print("1. Pizzas: ")
            for pizza in modi_pizza.pila_pizza:
                print(f"    {pizza.id}. {pizza.producto}   Precio: {pizza.precio}   Stock: {pizza.stock}")
            print("2. Bebidas: ")
            for bebida in modi_bebida.pila_bebida:
                print(f"    {bebida.id}. {bebida.producto}   Precio: {bebida.precio}   Stock: {bebida.stock}")
            print("3. Complementos: ")
            for complemento in modi_complemento.pila_complemento:
                print(
                    f"    {complemento.id}. {complemento.producto}   Precio: {complemento.precio}   Stock: {complemento.stock}")
            print("4. Ofertas: ")
            if hay_oferta==1:
                for oferta in modi_oferta.pila_oferta:
                    print(f"{oferta.id}. {oferta.producto}   Precio: {oferta.precio}   Stock: {oferta.stock}")
            else:
                print("No hay ofertas disponibles aun")
            print("_____________________________________________________________________")
```

Opción 2 en el menú principal:
Nos da acceso a cuatro opciones. En la primera opción accedemos a las pilas donde se almacenaron los datos de los productos a vender (id, nombre, precio, stock) el menú.


```
elif usuario == "2":
    print("\033[0;m")
    while not Salir_Usuario:
        print("\033[1;32m--Interfaz--")
        print("1. Mostar el menu\n2. Comprar\n3. Pedidos\n4. Salir ")
        opcion = input("Que desea escoger: ")
        print("_____________________________________________________________________")
        if opcion == "1":
            print("1. Pizzas: ")
            for pizza in modi_pizza.pila_pizza:
                print(f"    {pizza.id}. {pizza.producto}   Precio: {pizza.precio}   Stock: {pizza.stock}")
            print("2. Bebidas: ")
            for bebida in modi_bebida.pila_bebida:
                print(f"    {bebida.id}. {bebida.producto}   Precio: {bebida.precio}   Stock: {bebida.stock}")
            print("3. Complementos: ")
            for complemento in modi_complemento.pila_complemento:
                print(
                    f"    {complemento.id}. {complemento.producto}   Precio: {complemento.precio}   Stock: {complemento.stock}")
            print("4. Ofertas: ")
            if hay_oferta==1:
                for oferta in modi_oferta.pila_oferta:
                    print(f"{oferta.id}. {oferta.producto}   Precio: {oferta.precio}   Stock: {oferta.stock}")
            else:
                print("No hay ofertas disponibles aun")
            print("_____________________________________________________________________")
```

En la segunda opción se va a realizar lo que es la “compra de productos”, la función que va a tener esta opción es que según se vaya seleccionando este busque la ubicación donde se encuentra el producto seleccionado permitiendo reducir el número de stock, agregarlo a la lista pedidos, agregarlo a la lista de factura y agregarlo a la cola donde se registran como pendientes los pedidos, lo mismo para las bebidas (opcion 2), complementos (opción 3) y ofertas (opcion 4), en caso estos no cuenten con stock no se agregara ningún dato a la cola de pedidos pendientes y regresará al menú del usuario.

```
seguir_comprando = input("1. Si\n2. No\nDesea seguir comprando algun pedido: ")
while seguir_comprando != "1" and seguir_comprando != "2":
    print("Opcion invalida")
    seguir_comprando = input("1. Si\n2. No\nDesea seguir comprando algun pedido: ")
print("_____________________________________________________________________")
if seguir_comprando == "2":
    Seguir_comprando = True
    if hay_compra == 0:
        cf_nit = input('1.- NIT \n 2.- C/F \n Por favor, ingrese una opcion: ')
        if cf_nit == '1':
            nit = input('Ingrese el NIT: ')
            hay_nit = 1
            modo = input(
                '1.- Pago con tarjeta \n 2.- Pago en efectivo \n Por favor, elija su forma de pago: ')
            if modo == '1':
                name = input('Ingrese nombre para realizar la factura: ')
                hay_name = 1
                iva = (total * 0.12) + total

                ordenamientoBurbuja(compra_factura)
```

Terminando la “compra de productos”, siempre se preguntará si desea seguir comprando en caso sea 1 (si) este repetirá los procesos antes mencionados hasta que se seleccione 2 (no), donde según la pila donde se encuentran los precios serán sumados para obtener el valor total y el iva de la compra, además preguntará si se desea agregar nit o consumidor final (c/f), en caso deseara agregar nit éste será agregado a una pila para almacenar los datos del cliente para en un futuro acceder a ofertas, al igual que preguntara el método de pago en caso de ser efectivo el programa continuará sin pedir ningún dato, en caso sea con tarjeta se pedirá el nombre del propietario de la tarjeta para guardar sus datos (nombre).

```
                print(
                    f'Factura \n Nombre: {name} \n Numero de identificacion tributaria (NIT): {nit} \n '
                    f'{compra_factura} \n Subtotal a pagar: Q.{total} \n Total a pagar (con IVA incluido): '
                    f'Q.{iva} \n Pago con tarjeta')
            elif modo == '2':
                name = input('Ingrese nombre para realizar la factura: ')
                iva = (total * 0.12) + total
                print(
                    f'Factura \n Nombre: {name} \n Numero de identificacion tributaria (NIT): {nit} \n '
                    f'{compra_factura} \n Subtotal a pagar: Q.{total} \n Total a pagar (con IVA incluido): '
                    f'Q.{iva} \n Pago con efectivo')
        elif cf_nit == '2':
            modo = input(
                '1.- Pago con tarjeta \n 2.- Pago en efectivo \n Por favor, elija su forma de pago: ')
            if modo == '1':
                iva = (total * 0.12) + total
                print(
                    f'{compra_factura} \n Subtotal a pagar: Q.{total} \n Total a pagar (con IVA incluido): '
                    f'Q.{iva} \n Pago con tarjeta')
            elif modo == '2':
                iva = (total * 0.12) + total
                print(
                    f'{compra_factura} \n'
                    f'Subtotal a pagar: Q.{total} \n Total a pagar (con IVA incluido): '
                    f'Q.{iva} \n Pago con efectivo')
        if hay_nit == 1 and hay_name == 1:
            cliente = nit, name
            agre_cliente.agregar_clientes(cliente)
print("_____________________________________________________________________")
```

Terminando de pagar (efectivo o tarjeta), se preguntará si desea nit o consumidor final para su factura, en caso sea consumidor final no se le pedirá ningún dato, pero se mostrará los pedidos que realizó obtenidos de la pila de los productos (pizza, bebida, complemento u oferta) y el subtotal en este caso la suma de los datos obtenidos de la pila precio y el total que es la suma del iva al subtotal. En caso pida nit se le solicitarán los siguientes datos: nit, nombre y método de pago, estos se almacenarán en una pila.

```
            elif opcion == "3":
                print("Pedidos pendientes")
                pedido_pendiente = list(colas.cola_pendiente.queue)
                for i, pedido in enumerate(pedido_pendiente):
                    print(f"    {i + 1}. {pedido[1]} Q{pedido[0]}")
                print("Pedidos en preparacion")
                pedido_preparacion = list(colas.cola_preparacion.queue)
                for i, pedido in enumerate(pedido_preparacion):
                    print(f"    {i + 1}. {pedido[1]} Q{pedido[0]}")
                print("Pedidos listos")
                pedido_listo = list(colas.cola_listo_servir.queue)
                for i, pedido in enumerate(pedido_listo):
                    print(f"    {i + 1}. {pedido[1]} Q{pedido[0]}")
                print("_____________________________________________________________________")
                preparar = input("1. Si\n2. No\nDesea preparar algun pedido: ")
                if preparar == "1":
                    colas.agregar_cola_preparacion()
                elif preparar == "2":
                    print("No se prepararan pedidos")
                else:
                    print("Opcion invalida")
                print("_____________________________________________________________________")
            elif opcion == "4":
                print("Saliendo al menu principal...")
                Salir_Usuario = True
            else:
                print("Opcion invalida")
        print("_____________________________________________________________________\033[0;m")
    elif usuario == "3":
        break
    else:
        print("\033[1;34m_____________________________________________________________________")
        print("Opcion no valida\nUsuario no identificado")
print("\033[1;31mCerrando el programa...")
timer.cancel()
```

En la tercera opción se mostrarán las colas donde previamente en la “compra de pedidos” se agregaron a la misma, teniendo tres colas: 1. Pedido Pendiente, 2. Pedido en Preparación, 3. Pedido Listo.
En este apartado se encuentran dos opciones para preparar el pedido (si , no) en caso se seleccione “si”, el la cola “pedido pendiente” se eliminara el primer pedido en la cola y se agregara en la cola “pedido en preparación”, una vez agregado en la cola con un timer este de manera automática será agregado a la cola “pedido listo”, cada cola con el ciclo for se irán enumerando para llevar un orden en los pedidos. En la segunda opción (no) se mostrará el mensaje “no se prepararan pedidos” y regresará al menú de usuario.

Cuarta opción y última se saldrá del menú de usuario y regresará al menú de inicio (selección de usuarios).