##############CLASE ADMINISTRACION##############

from Cliente import \*

from Producto import \*

from Promo import \*

from Tarjetas import \*

class Administracion :

def \_\_init\_\_(self):

self.\_\_ListaCliente = []

self.\_\_ListaPromocion = []

self.\_\_ListaProducto = []

self.\_\_ListaTarjeta = []

self.\_\_carrito = []

#Comprobar si existe el producto

def ExisteProducto(self,cod):

for x in self.\_\_ListaProducto:

if x.verCodP() == cod:

return True

else:

return False

#Comprobar si existe el cliente

def ExisteCliente(self,dni):

existe = "no"

for x in self.\_\_ListaCliente:

if x.verDni() == dni:

existe = "si"

return existe

#Comprobar si existe la tarjeta en el sistema

def ExisteTarjeta(self,nomt,banc):

for x in self.\_\_ListaTarjeta:

if x.vernomt() == nomt and x.verbanc() == banc:

return True

else:

return False

#Agregar producto

def AgregarProducto(self,producto):

if isinstance (producto,Producto):

self.\_\_ListaProducto.append(producto)

#Agregar cliente

def AgregarCliente(self,cliente):

self.\_\_ListaCliente.append(cliente)

#Mostrar los productos cargados en el sistema

def MostrarListaProd(self):

return self.\_\_ListaProducto

#Mostrar las promociones cargadas en el sistema

def MostrarListaProm(self):

return self.\_\_ListaPromocion

#Mostrar los clientes cargados en el sistema

def MostrarListaCliente(self):

return self.\_\_ListaCliente

#Mostrar las tarjetas cargadas al sistema

def verLisTar(self):

return self.\_\_ListaTarjeta

#Subir una promocion al sistema

def DarAltaPromo(self,prom):

if isinstance(prom,Promo):

self.\_\_ListaPromocion.append(prom)

#Agregar tarjeta

def AgregarTarjeta(self,tarjetas):

if isinstance(tarjetas,Tarjetas):

self.\_\_ListaTarjeta.append(tarjetas)

#Agregar un producto al carro del cliente

def AgregarAlCarro(self,producto):

self.\_\_carrito.append(producto)

#Sacar un producto del carro de cliente

def sacacar(self,producto):

if (producto,Producto):

for x in self.\_\_carrito:

if x.verCodP() == producto.verCodP():

self.\_\_carrito.remove(x)

#Muestra el carro del cliente

def vercar(self):

return self.\_\_carrito

def vaciarcarro(self):

self.\_\_carrito.clear()

#Sacar un producto de la lista de la tienda

#def baja(self,producto):

#if (producto,Producto):

#for x in self.\_\_carrito:

#if x.verCodP() == producto.verCodP():

#self.\_\_ListaProducto.remove(producto)

##############CLASE CLIENTE##############

class Cliente:

def \_\_init\_\_(self,dni,nomyape,fnac,monto):

self.\_\_dni = dni

self.\_\_nomyape = nomyape

self.\_\_fnac = fnac

self.\_\_monto = 0

#Muestra el nombre del cliente

def verNom(self):

return self.\_\_nomyape

#Muestra el D.N.I del cliente

def verDni(self):

return self.\_\_dni

#Muestra la fecha nacimiento del cliente

def verFnac(self):

return self.\_\_fnac

#Modifica el monto del cliente

def verMonto(self):

return self.\_\_monto

#Modifica el nombre del cliente

def modNom(self,otro):

self.\_\_nomyape = otro

#Modifica el D.N.I del cliente

def modDni(self,otro):

self.\_\_dni = otro

#Modifica la fecha nacimiento del cliente

def modFnac(self,otro):

self.\_\_fnac = otro

#Modifica el monto del cliente

def modMonto(self,otro):

self.\_\_monto += otro

##############CLASE PRODUCTO##############

class Producto:

def \_\_init\_\_(self,nom,cod,pre,cant,desc,porcen\_descu):

self.\_\_nom = nom

self.\_\_cod = cod

self.\_\_pre = pre

self.\_\_cant = cant

self.\_\_desc = desc

self.\_\_porcen\_descu = 0

def \_\_init\_\_(self,nom,cod,pre,cant,desc):

self.\_\_nom = nom

self.\_\_cod = cod

self.\_\_pre = pre

self.\_\_cant = cant

self.\_\_desc = desc

self.\_\_porcen\_descu = 0

#Muestra el nombre del producto

def verNom(self):

return self.\_\_nom

#Muestra el codigo del producto

def verCodP(self):

return self.\_\_cod

#Muestra el precio del producto

def verPre(self):

return self.\_\_pre

#Muestra la cantidad de unidades del producto

def verCant(self):

return self.\_\_cant

#Muestra una descripcion del producto

def verDesc(self):

return self.\_\_desc

#Ver % de descuent

def verporcen\_descu(self):

return self.\_\_porcen\_descu

#Modifica el nombre del producto

def modNom(self,otro):

self.\_\_nom = otro

#Modifica el codigo del producto

def modCod(self,otro):

self.\_\_cod = otro

#Modifica el precio del producto

def modPre(self,otro):

self.\_\_pre = otro

#Modifica la cantidad de unidades del producto

def modCant(self,otro):

self.\_\_cant = otro

#Modifica la descripcion del producto

def modDesc(self,otro):

self.\_\_desc = otro

#Modificar % de descuento

def modporcen\_descu(self,otro):

self.\_\_porcen\_descu = otro

##############CLASE PROMOCION##############

class Promo:

def \_\_init\_\_(self,cant\_desc,cod\_prod):

self.\_\_cant\_desc = cant\_desc

self.\_\_cod\_prod = cod\_prod

#Muestra el codigo del producto

def VerCod\_prod(self):

return self.\_\_cod\_prod

#Muestra el porcentaje de descueto

def VerCantDesc(self):

return self.\_\_cant\_desc

#Modifica el codigo del producto

def ModCodp(self,otro):

self.\_\_cod\_prod = otro

#Modifica el porcentaje de descueto

def ModCantDesc(self,otro):

self.\_\_cant\_desc = otro

##############CLASE TARJETAS##############

class Tarjetas():

def \_\_init\_\_(self,nomt,canfpag,recarpag,banc):

self.\_\_nomt = nomt

self.\_\_canfpag = canfpag

self.\_\_recarpag = recarpag

self.\_\_banc = banc

self.\_\_monto\_total = 0

def \_\_init\_\_(self,nomt,canfpag,recarpag,banc):

self.\_\_nomt = nomt

self.\_\_canfpag = canfpag

self.\_\_recarpag = recarpag

self.\_\_banc = banc

self.\_\_monto\_total = 0

#Muestra el nombre de la tarjeta

def vernomt(self):

return self.\_\_nomt

#Muestra las cantidad de formas de pago de la tarjeta( las Cuotas)

def vercanfpag(self):

return self.\_\_canfpag

#Mestra el costo del finaciamiento de las cuotas(interes)

def verrecarpag(self):

return self.\_\_recarpag

#Mustra el nombre del banco

def verbanc(self):

return self.\_\_banc

#Mustrar monto total

def vermonto\_total(self):

return self.\_\_monto\_total

#Modifica el nombre de la tarjeta

def modnomt(self,otro):

self.\_\_nomt = otro

#Modifica la cantidad de formas de pago de la tarjeta(las Cuotas)

def modcanfpag(self,otro):

self.\_\_canfpag = otro

#Modifica el costo del finaciamiento de las cuotas

def modrecarpag(self,otro):

self.\_\_recarpag = otro

#Modifica el nombre del banco

def modbanc(self,otro):

self.\_\_banc = otro

#Modificar monto total

def modmonto\_total(self,otro):

self.\_\_monto\_total += otro

##############ERRORES##############

class fuera\_de\_op(Exception):

def \_\_init\_\_(self,op):

self.op = op

def \_\_str\_\_(self):

return str (self.op)

##############LAUNCHER##############

from MenuTienda import \*

Tienda ()

##############MENU TIENDA##############

from Administracion import \*

from Cliente import \*

from Promo import \*

from Tarjetas import \*

from Errores import \*

import os

import sys

adm1 = Administracion ()

def BorrarPantalla():

return os.system("cls") #Windows

#return os.system("clear") #Linux

recaudacion\_tienda = 0

porcentaje\_de\_descuento = 0

def Tienda ():

while True :

BorrarPantalla()

print("==================================================")

print("Tienda de Articulos Gastronomicos")

print("==================================================")

print("1) Productos y Promociones\n")

print("2) Compras\n")

print("3) Tarjetas de Credito\n")

print("4) Administracion\n")

print("5) Salir del sistema\n")

try:

#con esta opcion se accede al modulo productos y promociones

opcionMenu = input("Ingrese una opcion: ")

if opcionMenu == "1":

BorrarPantalla()

TituloProdyPromo()

MenuProdyProm()

#con esta opcion se accede al modulo compras

if opcionMenu == "2":

BorrarPantalla()

MenuCompras()

#con esta opcion se accede al modulo tarjetas de credito

if opcionMenu == "3":

BorrarPantalla()

MenuTarjetas()

#con esta opcion se accede al modulo de administacion de la tienda

if opcionMenu == "4":

BorrarPantalla()

MenuAdministracion()

#con esta opcion se cierra el programa

if opcionMenu == "5":

sys.exit()

else:

raise fuera\_de\_op(opcionMenu)

except fuera\_de\_op as FI:

input("La opcion ingresada no esta contemplada en el menu ingrese un numero entre el 1 al 5.")

input()

BorrarPantalla()

Tienda()

#Titulo del modulo

def TituloProdyPromo():

print("==================================================")

print(" Modulo producto y promociones ")

print("==================================================")

def MenuProdyProm():

#Presentacion de las opciones del modulo

while True:

print("1) Dar de alta un producto\n")

print("2) Dar de alta una promocion\n")

print("3) Listar productos\n")

print("4) Listar promociones\n")

print("5) Volver al menu pricipal\n")

#Cargar datos del producto y darlo de alta(FUNCIONA)

opcionMenuP = input("Ingrese una opcion:")

if opcionMenuP == "1":

BorrarPantalla()

TituloProdyPromo()

while True:

try:

nom = input("Ingrese el nombre del producto: \n")

cod = int(input("Ingrese el codigo del producto: \n"))

desc = input("Ingrese una descripcion del producto: \n")

pre = float(input("Ingrese el precio del producto: \n"))

cant = int(input("Ingrese la cantidad del producto: \n"))

prod1 = Producto(nom,cod,pre,cant,desc)

rta = adm1.ExisteProducto(prod1.verCodP())

if rta == True:

print ("El producto que intenta cargar ya se encuentra cargado en el sistema.")

input()

BorrarPantalla()

TituloProdyPromo()

MenuProdyProm()

else :

adm1.AgregarProducto(prod1)

print("El producto con nombre:",prod1.verNom()," con el codigo: ",prod1.verCodP(),". Se cargo con exito en el sistema.")

input()

BorrarPantalla()

TituloProdyPromo()

MenuProdyProm()

break

except (ValueError):

input("Error la carga del producto no se pudo realizar, intente nuevamente.")

BorrarPantalla()

TituloProdyPromo()

MenuProdyProm()

BorrarPantalla()

MenuProdyProm()

#Dar de alta una promocion(FUNCIONA)

if opcionMenuP == "2":

BorrarPantalla()

TituloProdyPromo()

while True:

try:

cant\_desc = int(input("Ingrese porcentaje de descuento: \n"))

Lista\_prod = adm1.MostrarListaProd()

m = 1

for prod in Lista\_prod:

print(m,")","El producto con nombre:",prod.verNom()," con el codigo: ",prod.verCodP())

m +=1

cod\_prod = int(input("Ingrese una opcion para elegir el producto: \n"))

Lista\_prod[cod\_prod-1].modporcen\_descu(cant\_desc)

prom = Promo(cant\_desc,Lista\_prod[cod\_prod-1].verCodP())

adm1.DarAltaPromo(prom)

input("Se cargo con exito")

BorrarPantalla()

TituloProdyPromo()

MenuProdyProm()

break

except ValueError:

input("No se pudo cargar la promocion intente nuevamente.")

except IndexError:

input("La opcion ingresada no es valida")

MenuProdyProm()

BorrarPantalla()

#Mostrar una lista de los productos cargados en el sistema(FUNCIONA)

if opcionMenuP == "3":

BorrarPantalla()

TituloProdyPromo()

z = adm1.MostrarListaProd()

a = 0

for producto in z :

a+=1

print(a,") ","Nombre = ",producto.verNom(),"Codigo = ",producto.verCodP(),"Precio = ",producto.verPre(),"Cantidad de unidades = ",producto.verCant(),"Descripcion = ",producto.verDesc(),"Tiene un descuento del = ",producto.verporcen\_descu(),"%")

input("Presione enter para continuar")

BorrarPantalla()

TituloProdyPromo()

#Mostrar una lista de las promociones cargadas en el sistema(FUNCIONA)

if opcionMenuP == "4" :

BorrarPantalla()

TituloProdyPromo()

l = adm1.MostrarListaProm()

p = 0

for promo in l :

p += 1

print(p, ") ","Codigo del producto",promo.VerCod\_prod(),"Tiene un descuento del",promo.VerCantDesc(),"%")

input("Presione enter para continuar")

BorrarPantalla()

TituloProdyPromo()

#Volver al menu principal

if opcionMenuP == "5" :

BorrarPantalla()

Tienda ()

#Titulo del modulo

def Titulocompras():

print("==================================================")

print(" Modulo de compras ")

print("==================================================")

def MenuCompras() :

#Presentacion de las opciones del modulo

while True :

Titulocompras()

print("1) Agregar producto al carro\n")

print("2) Sacar producto del carro\n")

print("3) Procesar compra\n")

print("4) Mostrar contenido de carro\n")

print("5) Volver al menu pricipal\n")

#Cargar productos al carro(FUNCIONA)

opcionMenuC = input("Ingrese una opcion:")

if opcionMenuC == "1":

BorrarPantalla()

Titulocompras()

while True:

try:

z = adm1.MostrarListaProd()

a = 1

if len(z) == 0:

input("El carro esta vacio, presione enter.")

BorrarPantalla()

MenuCompras()

else:

print("Para agregar productos al carro debe usar el codigo de producto:")

for producto in z :

print(a,") ","Nombre = ",producto.verNom(),"Codigo = ",producto.verCodP(),"Precio = ",producto.verPre(),"Cantidad = ",producto.verCant(),"Descripcion = ",producto.verDesc(),"Tiene un descuento del = ",producto.verporcen\_descu(),"%")#,"Descuento del",verporcen\_descu(),"%") #revisar

a+=1

buy=int(input("Ingrese el codigo de producto: "))

for x in z:

if x.verCodP() == buy:

adm1.AgregarAlCarro(x)

input("Se cargo con exito precione enter para continuar")

BorrarPantalla()

MenuCompras()

except ValueError:

print("Error no se pudo agregar del carro ")

BorrarPantalla()

MenuCompras()

except IndexError:

input("EL codigo de producto ingresado no esta cargado en el sistema intente con un codigo valido ")

BorrarPantalla()

MenuCompras()

#Sacar producto del carro(FUNCIONA)

if opcionMenuC == "2":

BorrarPantalla()

Titulocompras()

while True:

try:

sp = adm1.vercar()

cont = 0

if len(sp)==0:

input("No hay productos que mostrar.")

BorrarPantalla()

MenuCompras()

for producto in sp:

cont += 1

print(cont,")","Nombre:",producto.verNom(),"Codigo:",producto.verCodP())

num = int(input ("Seleccione una opccion: "))

adm1.sacacar(sp[num-1])

input("se saco con exito, presione enter.")

BorrarPantalla()

MenuCompras()

except ValueError:

print("Error no se pudo sacar del carro ")

BorrarPantalla()

MenuCompras()

except IndexError:

input("La opcion ingresada esta fuera de rango ")

#Procesar compra(FUNCIONA)

if opcionMenuC == "3":

BorrarPantalla()

Titulocompras()

listaCli = adm1.MostrarListaCliente()

global recaudacion\_tienda

global porcentaje\_de\_descuento

while True:

try:

car = adm1.vercar()

if len(car) == 0:

input("Debe haber almenos un producto en el carro para procesar la compra,precione enter para continuar")

BorrarPantalla()

MenuCompras()

else:

dni = int(input("Ingrese su D.N.I: "))

existe = adm1.ExisteCliente(dni)

if existe == "si":

input ("El cliente que intenta cargar ya se encuentra cargado en el sistema.\n")

else:

nomyape = input("Ingrese su nombre y apellido: ")

fnac = input("ingrese su fecha de nacimiento: dd/mm/aaaa")

monto = 0

client = Cliente(dni,nomyape,fnac,monto)

adm1.AgregarCliente(client)

print("El Cliente con nombre",client.verNom()," con el D.N.I: ",client.verDni(),". Se cargo con exito en el sistema.")

input()

car = adm1.vercar()

ctotal = 0 #Sin descuento

ctotaldesc = 0 #Con descuento

TotalPagar = 0

promocion = adm1.MostrarListaProd()

#Calcular precio del producto

for x in car:

precio =(x.verPre() \* x.verCant()) #PRECIO PACK

ctotal+=precio

#Calcular precio del producto con la promocion

if x.verporcen\_descu() > 0 :

for y in promocion:

if x.verCodP() == y.verCodP():

desc = (x.verporcen\_descu()/ 100) # CONVERSION DE % A DECIMAL

predesc = (precio \* desc) #MULTIPLICAR LA CONVERCION A DECIMAL POR EL PRECIO DEL PRODUCTO

final = (precio - predesc) #RESTAR EL DESCUENTO AL PRECIO DEL PRODUCTO

print("\*\*\*\* Costo total del producto",x.verNom(),"Sin promocion es de: $",("{:,.2f}".format(precio)) ,"con la promocion es de:$",("{:,.2f}".format(final)),"\*\*\*\*")

TotalPagar += final

else:

print("Costo total del producto",x.verNom()," es:$",("{:,.2f}".format(precio)))

TotalPagar += precio

print("\*\*\*\* El costo total de los productos es es de: $", ("{:,.2f}".format(TotalPagar)),"\*\*\*\*")

#Elejir tarjeta y calcular el monto de las cuotas y el costo de financiacion

t = adm1.verLisTar()

y = 1

for tarjeta in t :

print(y,") ","Tarjeta",tarjeta.vernomt(),"Banco",tarjeta.verbanc(),"Cuotas",tarjeta.vercanfpag())

y+=1

numt = int(input("Seleccione una opcion: "))

cuotas = TotalPagar / (t[numt-1].vercanfpag()) #CALCULAR EL MONTO DE LAS CUOTAS

interes = ((cuotas\*(t[numt-1].verrecarpag())) / 100) #CALCULAR EL INTERES

PrecioFinal = cuotas + interes

PrecioTotalFinanciado = PrecioFinal \* t[numt-1].vercanfpag()

print ("El total de la compra es :",("{:,.2f}".format(PrecioTotalFinanciado)),"En",t[numt-1].vercanfpag(),"cuotas de $",("{:,.2f}".format(PrecioFinal)))

rta = input ("Desea confirmar la compra: (S/N)")

if rta.lower() == "s":

#Mandar el monto de la compra del cliente

for x in listaCli:

if x.verDni() == dni:

x.modMonto(PrecioTotalFinanciado)

t[numt-1].modmonto\_total(PrecioTotalFinanciado)

#Mandar el monto de la compra a la administracion

recaudacion\_tienda += PrecioTotalFinanciado

vc = adm1.vercar()

vlp = adm1.MostrarListaProd()

#Mandar el monto del descuento a la administracion

for c in vc:

if c.verporcen\_descu() > 0 :

porcentaje\_de\_descuento += predesc

#Sacar producto de la tienda

#for p in vlp:

# if c.verCodP() == p.verCodP():

# adm1.baja(p)

input("La transaccion se llevo a cabo con exito")

## adm1.vaciarcarro()

BorrarPantalla()

MenuCompras()

else:

BorrarPantalla()

MenuCompras()

except ValueError :

input("Error no se pudo procesar la compra")

#Mostrar el carro(FUNCIONA)

if opcionMenuC == "4":

BorrarPantalla()

Titulocompras()

c = adm1.vercar()

contcar = 0

if len(c) == 0:

input("El carro esta vacio")

BorrarPantalla()

MenuCompras()

else:

for producto in c:

contcar += 1

print(contcar,")","Nombre = ",producto.verNom(),"Codigo = ",producto.verCodP(),"Precio = ",producto.verPre(),"Cantidad = ",producto.verCant(),"Descripcion = ",producto.verDesc(),"Tiene un descuento del = ",producto.verporcen\_descu(),"%")

input("Precione enter para continuar")

BorrarPantalla()

MenuCompras()

#Volver al menu pricipal(FUNCIONA)

if opcionMenuC == "5":

BorrarPantalla()

Tienda()

#Titulo del modulo

def TituloTarjetas() :

print("==================================================")

print(" Modulo de Tarjetas de credito ")

print("==================================================")

def MenuTarjetas() :

#Presentacion de las opciones del modulo

while True :

TituloTarjetas()

print("1) Dar de alta una tarjeta de credito\n")

print("2) Dar de alta beneficios para las tarjetas\n")

print("3) Listar tarjetas de credito\n")

print("4) Listar tarjetas de credito con beneficios\n")

print("5) Volver al menu pricipal\n")

#Dar de alta una tarjeta de credito (Funciona )

opcionMenuT = input("Ingrese una opcion:")

if opcionMenuT == "1":

BorrarPantalla()

TituloTarjetas()

while True :

try:

nomt = input("Ingrese el nombre de la tarjeta:")

banc =input("Ingrese el nombre del banco:")

canfpag = int(input("Ingrese la cantidad de cuotas:"))

recarpag = int(input("Ingrese el interes por cuota:"))

tar1 = Tarjetas(nomt,canfpag,recarpag,banc)

rta2 = adm1.ExisteTarjeta(tar1.vernomt(),tar1.verbanc())

if rta2 == True:

input("La tarjeta que intenta cargar ya se encuentra dada de alta")

BorrarPantalla()

MenuTarjetas()

else:

adm1.AgregarTarjeta(tar1)

print("[","Tarjeta = ",tar1.vernomt(),"Cantidad de cuotas = ",tar1.vercanfpag(),"Costo del finaciamiento de las cuotas = ",tar1.verrecarpag(),"%","Banco = ",tar1.verbanc(),"Total gastado = ",tar1.vermonto\_total(),"]","\n se cargo con exito.")

input()

BorrarPantalla()

MenuTarjetas()

except ValueError :

input("Error no se pudo cargar la tarjeta")

BorrarPantalla()

MenuTarjetas()

# Dar de alta beneficios para las tarjetas (FUNCIONA)

if opcionMenuT == "2":

BorrarPantalla()

TituloTarjetas()

while True:

try:

vt = adm1.verLisTar()

cont\_tar = 1

if len(vt) == 0:

input("No hay tarjetas que mostrar, pulse enter para continuar.")

BorrarPantalla()

TituloTarjetas()

else:

for x in vt :

print (cont\_tar,")","Nombre:",x.vernomt(),"Cuotas:",x.vercanfpag(),"Costo del finaciamiento de las cuotas",x.verrecarpag(),"%","Nombre del banco ",x.verbanc(),"Total gastado",("{:,.2f}".format(x.vermonto\_total())))

cont\_tar += 1

select = int(input("Ingrese una opcion:"))

mod\_cuot = int(input("Ingrese las nuevas cuotas:"))

mod\_inter = int(input("Ingrese el nuevo interes:"))

vt[select - 1].modcanfpag(mod\_cuot)

vt[select - 1].modrecarpag(mod\_inter)

input("Beneficios actualizados con exito.")

BorrarPantalla()

MenuTarjetas()

except ValueError:

input("No se pudo actualizar los beneficios intente nuevamente.")

BorrarPantalla()

MenuTarjetas()

except IndexError:

input("La opcion ingresada esta fuera de rango ")

BorrarPantalla()

MenuTarjetas()

#Listar tarjetas de credito(FUNCIONA)

if opcionMenuT == "3":

BorrarPantalla()

TituloTarjetas()

t = adm1.verLisTar()

y = 1

if len(t) == 0:

input("No hay tarjetas que mostrar, pulse enter para continuar.")

BorrarPantalla()

MenuTarjetas()

else:

for tarjeta in t :

print(y,") ","Tarjeta:",tarjeta.vernomt(),"Cuotas:",tarjeta.vercanfpag(),"Interes de las cuotas:",tarjeta.verrecarpag(),"%","Banco:",tarjeta.verbanc(),"Total gastado:",("{:,.2f}".format(tarjeta.vermonto\_total())))

y+=1

input("Presione enter para continuar")

BorrarPantalla()

MenuTarjetas()

# Listar tarjetas de credito con beneficios( FUNCIONA)

if opcionMenuT == "4":

BorrarPantalla()

TituloTarjetas()

tb = adm1.verLisTar()

b = 1

if len(tb) == 0:

input("No hay tarjetas cargadas en el sistema, pulse enter para continuar.")

else:

for x in tb:

if x.verrecarpag() == 0:

print(b,")","Nombre",x.vernomt(),"Cuotas",x.vercanfpag(),"Interes de las cuotas:",x.verrecarpag(),"%","Banco",x.verbanc(),"Total gastado",("{:,.2f}".format(x.vermonto\_total())))

b += 1

input("Presione enter para continuar")

BorrarPantalla()

MenuTarjetas()

#Volver al menu principal( FUNCIONA)

if opcionMenuT == "5":

BorrarPantalla()

Tienda()

#Titulo del modulo

def TituloAdministracion():

print("==================================================")

print(" Modulo de Administracion ")

print("==================================================")

def MenuAdministracion() :

#Presentacion de las opciones del modulo

while True :

TituloAdministracion()

print("1) Recaudacion total de la tienda\n")

print("2) Ver total comprado por un cliente\n")

print("3) Ver total Vendido con cada tarjeta\n")

print("4) Ver total de descuento por promociones\n")

print("5) Volver al menu pricipal\n")

#Mostrar el recaudado total por la tienda(FUNCIONA)

opcionMenuA = input("Ingrese una opcion:")

if opcionMenuA == "1":

BorrarPantalla()

TituloAdministracion()

print("\*\*\*\*\*\* La recaudacion total de la tienda es: $", ("{:,.2f}".format(recaudacion\_tienda)),"\*\*\*\*\*\*")

input("Presione enter para continuar")

BorrarPantalla()

MenuAdministracion()

#Mostrar el total comprado por un cliente(FUNCIONA FALTA ORDENA DE MENOR A MAYOR )

if opcionMenuA == "2":

BorrarPantalla()

TituloAdministracion()

list\_cli = adm1.MostrarListaCliente()

rec = (len(list\_cli)-1)

if len(list\_cli) > 0 :

listar\_clientes (list\_cli,rec)

input("Precione enter para continuar")

BorrarPantalla()

MenuAdministracion()

else :

input("Todavia no se han realizado ventas, precione enter para continuar")

BorrarPantalla()

MenuAdministracion()

#Mostrar el total Vendido con cada tarjeta(FUNCIONA ORDENA DE MAYOR A MENOR)

if opcionMenuA == "3":

BorrarPantalla()

TituloAdministracion()

vtt = adm1.verLisTar()

recaudado = []

if len(vtt) > 0 :

for r in vtt:

if len(vtt) > 0 :

recaudado.append(r.vermonto\_total())

w = ordenar(recaudado)

w.reverse()

for f in w:

for h in vtt:

if h.vermonto\_total() == f:

print("Tarjeta:",h.vernomt(),"del banco",h.verbanc(),"abono $",("{:,.2f}".format(h.vermonto\_total())))

input("Precionde enter para continuar")

BorrarPantalla()

MenuAdministracion()

else:

input("Todavia no se han realizado ventas, precione enter para continuar")

BorrarPantalla()

MenuAdministracion()

#Mostrar el Ver total de descuento por promociones(FUNCIONA)

if opcionMenuA =="4":

BorrarPantalla()

TituloAdministracion()

print("\*\*\*\*\*\* Monto total de descuentos por promociones = $",("{:,.2f}".format(porcentaje\_de\_descuento)),"\*\*\*\*\*\*")

input("Precionde enter para continuar")

BorrarPantalla()

MenuAdministracion()

#Volver al menu pricipa(FUNCIONA)

if opcionMenuA == "5":

BorrarPantalla()

Tienda ()

#ordenamiento burbuja de las tarjetas

def ordenar(lt):

num = len(lt)

p = 0

while p < num:

k = p

while k < num:

if lt[p] > lt[k]:

ax = lt[p]

lt[p] = lt[k]

lt[k] = ax

k = k +1

p += 1

return (lt)

#Recurcion

def listar\_clientes (list\_cli,rec):

if rec == 0:

print("\n","Nombre",list\_cli[rec].verNom(),"D.N.I",list\_cli[rec].verDni(),"Fecha de nacimiento",list\_cli[rec].verFnac(),"Gasto $",("{:,.2f}".format(list\_cli[rec].verMonto())))

else:

print("\n","Nombre",list\_cli[rec].verNom(),"D.N.I",list\_cli[rec].verDni(),"Fecha de nacimiento",list\_cli[rec].verFnac(),"Gasto$",("{:,.2f}".format(list\_cli[rec].verMonto())))

listar\_clientes(list\_cli,rec-1)