

CODIGO

Class Interface

```
from colorama import *
class Interface:
    def Menu(self):
        print(Fore.GREEN
+ "*****"
+ "*****")
        print(Fore.YELLOW + "| Centro Deportivo                                || 9.-
Eliminar Producto                                ||")
        print("| 1.  Usuario nuevo                                || 10.- Reporte en especial
Por Producto                                ||")
        print("| 2.- Ver usuarios                                || 11.  Rentar producto
|")
        print("| 3.- Actualizar Usuario                                || 12.- Ver Los productos
rentados                                ||")
        print("| 4.- Eliminar Usuario                                || 13.- Eliminar productos
rentados                                ||")
        print("| 5.- Reporte en especial Por usuario            || 14  Actualizar productos
rentados                                ||")
        print("| 6.  Producto nuevo                                || 15  Consulta de la
persona o el producto de los pedidos    ||")
        print("| 7.- Ver Productos                                || 16 Salir
|")
        print("| 8.- Actualizar Producto                                ||
|")
        print( Fore.GREEN
+ "*****"
+ "*****")
```

Class Listadepedidos

```
import pandas as pd
from Class_MYSQL import Database as Mysql
from Class_Mongo import MongoClient
mongo=Mongodb()
mydb=Mysql()
class Listadepedidos:
    def __init__(self, id=None, nombre="", producto=""):
        self.idp = id
        self.nombre = nombre
        self.producto = producto
        self.list = []
        self.x = 0
        self.y = 0
        self.importacion = pd.read_csv('Listadepedidos.csv')
        self.id = self.importacion['idp']
        self.nombre2 = self.importacion['nombre']
        self.producto2 = self.importacion['producto']
        self.Listadepedidos = ''
    def agregarlistadepedido(self, nombre, producto):
        basedatos = Listadepedidos.regreso(self)
        for i in basedatos:
```

```

        bd = int(i)
self.Listadepedidos = open("Listadepedidos.csv", "a", newline='')
self.x += 1
self.idp = self.x
new = Listadepedidos(self.idp, nombre, producto)
self.Listadepedidos.write(str(self.idp) + "," + nombre + "," + producto +
'\n')

self.Listadepedidos.close()
if bd == 1:
    mydb.agregarpedidos(str(self.idp), nombre, producto)
#mongo
if bd == 2:
    mongo.ingresarpedido(str(self.idp), nombre, producto)
self.list.append(new)

def actualizarproducto(self, nombre, producto):
    basededatos = Listadepedidos.regreso(self)
    for i in basededatos:
        bd = int(i)
        c = 0
        e = 0
        s = 0
    for element in self.list:
        if (element.idp == nombre):
            g = element.idp
            s = 1
        else:
            if (element.idp != ""):
                if (s == 0):
                    c += 1
            if (s == 1):
                d = 0
                new = Listadepedidos(g, nombre, producto)
                self.Listadepedidos = open('Listadepedidos.csv', 'r+')
                texto = self.Listadepedidos.readlines()
                texto[c + 1] = (str(g) + "," + nombre + "," + producto + '\n')
                if bd == 1:
                    mydb.actualizarpedido(producto, g)
                # mongodb
                if bd == 2:
                    mongo.actualizarpedido(g, producto)
                self.Listadepedidos.seek(0)
                self.Listadepedidos.writelines(texto)
                self.Listadepedidos.close()
                self.list[c] = new
            e = 1
    if (e == 0):
        print("Lo sentimos la Persona no se encuentra Disponible")

def eliminarpedido(self, nombre=None):
    basededatos = Listadepedidos.regreso(self)
    for i in basededatos:
        bd = int(i)
        c = 0
    for element in self.list:

        if (element.nombre == nombre):

```

```

        self.Listadepedidos = open("Listadepedidos.csv", "r+")
        lineas = self.Listadepedidos.readlines()
        self.Listadepedidos.close()
        self.Listadepedidos = open("Listadepedidos.csv", "w")
        for linea in lineas:
            if linea != (str(element.idp) + "," + nombre + "," +
element.producto + '\n'):
                self.Listadepedidos.write(linea)
        self.list.pop(c)
        if bd == 1:
            mydb.eliminarpedido(element.idp)
        # mongodb
        if bd == 2:
            mongo.eliminarpedido(element.idp)
        self.Listadepedidos.close()

        c += 1

```

```

def verlistadepedios(self, nombre=None):
    c = 0
    a = 0
    if (nombre != None):

        for element in self.list:

            if (element.nombre == nombre):
                return self.list[c]
                a = 1
            c += 1
        if (a == 0):
            nombre = ""
            idp = ""
            producto = ""
            new = Listadepedidos(idp, nombre, producto)

            return new

    else:
        return self.list
def verlistadepedios2(self, nombre=None):
    c = 0
    a = 0
    if (nombre != None):
        for element in self.list:
            if (element.producto == nombre):
                return self.list[c]
                a = 1
            c += 1
        if (a == 0):
            nombre = ""
            idp = ""
            producto = ""
            new = Listadepedidos(idp, nombre, producto)

            return new

```

```

        else:
            return self.list
def pasardatospedidos(self):
    basededatos = Listadepedidos.regreso(self)
    for i in basededatos:
        bd = int(i)
        if bd == 1:
            desigualdad = mydb.verificarpedidoserroneos()
        if bd == 2:
            mongodesigualdad=mongo.verificarpedido()
        cont = 0
        for j in range(len(self.nombre2)):
            id = self.id[j]
            nombre3 = str(self.nombre2[j])
            producto3 = str(self.producto2[j])
            self.x = id
            self.idp = self.x
            new = Listadepedidos(str(self.idp), nombre3, producto3)
            if bd == 1:
                for x in desigualdad:
                    for i in x:
                        if id == i:
                            cont = 1
            if cont == 0:

                mydb.agregarpedidos(str(self.idp),nombre3, producto3)
                cont = 0
        # condicion mongodb
        if bd == 2:
            for x in mongodesigualdad.find():
                d = int(x['_id'])
                if d == id:
                    cont = 1

            if cont == 0:
                mongo.ingresarpedido(str(self.idp), nombre3, producto3)
                cont = 0
            self.list.append(new)
def basededatos(self, opcion):
    self.y = opcion

def regreso(self):
    return self.y

```

Class Material

```

from datetime import datetime
from io import open
import pandas as pd
from Class_MYSQL import Database as Mysql
from Class_Mongo import Mongoddb
from Class_Persona import Persona
p=Persona()
mongo=Mongoddb()
mydb=Mysql()

```

```

class Material:
    def __init__(self, id=None, nombre="", fecha=""):
        self.idp = id
        self.nombre = nombre
        self.fechacaduacion = fecha
        self.list = []
        self.x = 0
        self.y=0
        self.importacion = pd.read_csv('Productos.csv')
        self.id = self.importacion['idp']
        self.nombre2 = self.importacion['nombre']
        self.fechadecaducacion2 = self.importacion['fehadecaducacion']
        self.Productos = ''

    def agregarproducto(self, nombre, fecha):
        basededatos = Material.regreso(self)
        for i in basededatos:
            bd = int(i)
            self.Productos = open("Productos.csv", "a", newline='')
            self.x += 1
            self.idp = self.x
            new = Material(self.idp, nombre, fecha)
            self.Productos.write(str(self.idp) + "," + nombre + "," + fecha + '\n')
            self.Productos.close()
            if bd == 1:
                mydb.agregarproducto(str(self.idp), nombre, fecha)
            #mongo
            if bd == 2:
                mongo.ingresarmaterial(str(self.idp), nombre, fecha)

        self.list.append(new)

    def actualizarproducto(self, nombre, fecha):
        basededatos = Material.regreso(self)
        for i in basededatos:
            bd = int(i)
            c = 0
            e = 0
            s = 0
            for element in self.list:

                if (element.nombre == nombre):

                    g = element.idp
                    s = 1
                else:
                    if (element.idp != ""):
                        if (s == 0):
                            c += 1
            if (s == 1):
                d = 0
                new = Material(g, nombre, fecha)
                self.personas = open('Productos.csv', 'r+')
                texto = self.personas.readlines()
                texto[c + 1] = (str(g) + "," + nombre + "," + fecha + '\n')
                if bd == 1:
                    mydb.actualizarproducto(fecha,g)

```

```

        #mongo
        if bd == 2:
            mongo.actualizarmaterial(g, fecha)
        self.personas.seek(0)
        self.personas.writelines(texto)
        self.personas.close()
        self.list[c] = new
    e = 1
    if (e == 0):
        print("Lo sentimos la Persona no se encuentra Disponible")

def eliminarproducto(self, nombre=None):
    basededatos = Material.regreso(self)
    for i in basededatos:
        bd = int(i)
        c = 0
        for element in self.list:

            if (element.nombre == nombre):
                self.Productos = open("Productos.csv", "r+")
                lineas = self.Productos.readlines()
                self.Productos.close()
                self.Productos = open("Productos.csv", "w")
                for linea in lineas:
                    if linea != (str(element.idp) + "," + nombre + "," +
element.fechacaduacion+ '\n'):
                        self.Productos.write(linea)
                self.list.pop(c)
                if bd == 1:
                    mydb.eliminarproducto(element.idp)
                    #mongo
                if bd == 2:
                    mongo.eliminarmaterial(element.idp)
                self.Productos.close()
                c += 1

def verproductos(self, nombre=None):
    c = 0
    a=0
    if (nombre != None):

        for element in self.list:

            if (element.nombre == nombre):
                return self.list[c]
            a = 1
            c += 1
    if (a == 0):
        nombre = ""
        edad = ""
        idp = ""
        celular = ""
        new = Persona(idp, nombre, edad, celular)

        return new
    else:

```

```

        return self.list
def productos(self, idp=""):
    c = 0
    a = 0

    if (idp != ""):
        for element in self.list:
            if (element.idp == idp):
                return element.nombre
                a = 1
            c += 1
        if (a == 0):
            nombre = ""
            idp = ""
            fecha = ""

            return nombre

    else:
        return "fallido"
def Extraerdatos(self):
    basededatos = Material.regreso(self)
    for i in basededatos:
        bd = int(i)
        if bd == 2:
            mongodesigualdad=mongo.verificarmaterial()
        if bd == 1:
            desigualdad = mydb.verificarproductoserroneos()
        cont = 0
        for j in range(len(self.nombre2)):
            id = self.id[j]
            nombre3 = str(self.nombre2[j])
            fechadecaducacion3 = str(self.fechadecaducacion2[j])
            self.x = id
            self.idp = self.x
            new = Material(str(self.idp), nombre3, fechadecaducacion3)
            if bd == 1:
                for x in desigualdad:
                    for i in x:
                        if id == i:
                            cont=1
                if cont==0:
                    mydb.agregarproducto(str(self.idp),nombre3, fechadecaducacion3)
                    cont=0
            #mongodb
            if bd == 2:
                for x in mongodesigualdad.find():
                    d = int(x['_id'])
                    if d == id:
                        cont = 1

            if cont == 0:

                mongo.ingresarmaterial(str(self.idp), nombre3, fechadecaducacion3)
                cont = 0
            self.list.append(new)

```

```

def basededatos(self, opcion):
    self.y = opcion

def regreso(self):
    return self.y

```

Class Mongo

```

import pymongo
class Mongodb:
    def __init__(self):
        self.myclient = pymongo.MongoClient("mongodb://localhost:27017/")
        self.mydb = self.myclient["Prueba"]
        self.mycol = self.mydb["personas"]
        self.mycol2 = self.mydb["material"]
        self.mycol3 = self.mydb["pedidos"]
        self.list = []
    def comprobarcoleccion(self):
        collist = self.mydb.list_collection_names()
        for i in collist:
            print(i)
            if i=="personas":
                self.mycol.drop()
            if i == "material":
                self.mycol2.drop()
            if i=="pedidos":
                self.mycol3.drop()
    def ingresarpersona(self,id,nombre,edad,celular):
        self.myclient = pymongo.MongoClient("mongodb://localhost:27017/")
        self.mydb = self.myclient["Prueba"]
        self.mycol = self.mydb["personas"]
        self.mydict = {"_id": id, "nombre": nombre, "edad": edad, "celular": celular}
        self.mycol.insert_one(self.mydict)
    def actualizarpersona(self,id,edad,celular):
        self.myquery = {"_id":id}
        self.newvalues = {"$set": {"edad": edad, "celular": celular}}
        self.mycol.update_one( self.myquery, self.newvalues)
    def eliminarpersona(self,id):
        myquery = {"_id": id}
        self.mycol.delete_one(myquery)
    def verificardatos(self):
        return self.mycol
#materiales
    def ingresarmaterial(self, id, nombre, fechadecaducacion):
        self.mydict = {"_id": id, "nombre": nombre, "fechadecaducacion":
fechadecaducacion}
        self.mycol2.insert_one(self.mydict)
    def actualizarmaterial(self,id, fechadecaducacion):
        self.myquery = {"_id":id}
        self.newvalues = {"$set": { "fechadecaducacion": fechadecaducacion}}
        self.mycol2.update_one( self.myquery, self.newvalues)
    def eliminarmaterial(self,id):
        myquery = {"_id": id}
        self.mycol2.delete_one(myquery)
    def verificarmaterial(self):

```



```

        return self.mycol2
#pedidos
def ingresarpedido(self, id, nombre, producto):
    self.mydict = {"_id": id, "nombre": nombre, "producto": producto}
    self.mycol3.insert_one(self.mydict)
def actualizarpedido(self, id, producto):
    self.myquery = {"_id": id}
    self.newvalues = {"$set": { "producto": producto}}
    self.mycol3.update_one( self.myquery, self.newvalues)
def eliminarpedido(self, id):
    myquery = {"_id": id}
    self.mycol3.delete_one(myquery)
def verificarpedido(self):
    return self.mycol3

```

Class MySQL

```

import mysql.connector
class Database:
    def __init__(self):
        self.mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password=""
        )
        self.mycursor = self.mydb.cursor()
    def compruebalabasededatos(self):

        self.mycursor.execute("SHOW DATABASES")
        creardb = 0
        for x in self.mycursor:
            for i in x:
                if i == "prueba":
                    creardb = 1
        if creardb==1:
            self.mycursor.execute("DROP DATABASE Prueba")
            creardb = 0
        if creardb == 0:
            self.mycursor.execute("CREATE DATABASE Prueba")
        self.mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            db="Prueba"
        )
        self.mycursor = self.mydb.cursor()

        self.mycursor.execute("SHOW TABLES")
        tablapersonas = 0
        tablamaterial = 0
        tablapedidos = 0
        for x in self.mycursor:
            for i in x:
                if i == "personas":
                    tablapersonas = 1
                if i == "material":
                    tablamaterial = 1
                if i == "pedidos":

```

```

        tablapedidos = 1
    if tablapersonas == 0:
        self.mycursor.execute(
            "CREATE TABLE Personas (id INT PRIMARY KEY, nombre VARCHAR(255), edad
VARCHAR(255), "
            "celular VARCHAR(255))")
    if tablamaterial == 0:
        self.mycursor.execute(
            "CREATE TABLE Material (id INT PRIMARY KEY, nombre VARCHAR(255),
fechadecaducacion VARCHAR(255))")
    if tablapedidos == 0:
        self.mycursor.execute(
            "CREATE TABLE Pedidos (id INT PRIMARY KEY, nombre INT, producto
INT,FOREIGN KEY (nombre) REFERENCES Personas(id),FOREIGN KEY (producto) REFERENCES
Material(id))")
    def agregarapersona(self,id,nombre,edad,celular):
        self.mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            db="Prueba"
        )
        self.mycursor = self.mydb.cursor()
        sql = "INSERT INTO personas (id,nombre, edad,celular) VALUES (%s,%s, %s,%s)"
        val = [
            (id,nombre, edad, celular),
        ]
        self.mycursor.executemany(sql, val)
        self.mydb.commit()

    def verificardatoserroneos(self):
        self.mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            db="Prueba"
        )
        self.mycursor = self.mydb.cursor()
        self.mycursor.execute("SELECT id FROM personas")
        myresult = self.mycursor.fetchall()
        return myresult
    def actualizarpersona(self,edad,celular,id):
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            db="Prueba"
        )

        mycursor = mydb.cursor()

        sql = "UPDATE personas SET edad = %s,celular=%s WHERE id = %s"
        val = (edad,celular,id)

        mycursor.execute(sql, val)

        mydb.commit()

```

```

def eliminarpersona(self,id):
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )
    mycursor = mydb.cursor()
    sql = "DELETE FROM Personas WHERE id = %s"
    adr = (id,)
    mycursor.execute(sql, adr)

    mydb.commit()

def agregarproducto(self, id,nombre, fechadecaducacion):
    self.mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )
    self.mycursor = self.mydb.cursor()
    sql = "INSERT INTO material (id,nombre, fechadecaducacion) VALUES (%s,%s, %s)"
    val = [
        (id,nombre, fechadecaducacion),
    ]
    self.mycursor.executemany(sql, val)
    self.mydb.commit()

def verificarproductoserroneos(self):
    self.mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )
    self.mycursor = self.mydb.cursor()
    self.mycursor.execute("SELECT id FROM material")
    myresult = self.mycursor.fetchall()
    return myresult

def actualizarproducto(self,fechadecaducacion,id):
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )

    mycursor = mydb.cursor()

    sql = "UPDATE material SET fechadecaducacion = %s WHERE id = %s"
    val = (fechadecaducacion,id)

    mycursor.execute(sql, val)

    mydb.commit()

```

```

def eliminarproducto(self,id):
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )
    mycursor = mydb.cursor()
    sql = "DELETE FROM material WHERE id = %s"
    adr = (id,)
    mycursor.execute(sql, adr)

    mydb.commit()
def agregarpedidos(self,id, nombre, producto):
    self.mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )
    self.mycursor = self.mydb.cursor()
    sql = "INSERT INTO pedidos (id,nombre, producto) VALUES (%s,%s, %s)"
    val = [
        (id,nombre, producto),
    ]
    self.mycursor.executemany(sql, val)
    self.mydb.commit()

def verificarpedidoserroneos(self):
    self.mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )
    self.mycursor = self.mydb.cursor()
    self.mycursor.execute("SELECT id FROM pedidos")
    myresult = self.mycursor.fetchall()
    return myresult
def actualizarpedido(self,producto,id):
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        db="Prueba"
    )

    mycursor = mydb.cursor()

    sql = "UPDATE pedidos SET producto = %s WHERE id = %s"
    val = (producto,id)

    mycursor.execute(sql, val)

    mydb.commit()
def eliminarpedido(self,id):

```

```

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    db="Prueba"
)
mycursor = mydb.cursor()
sql = "DELETE FROM pedidos WHERE id = %s"
adr = (id,)
mycursor.execute(sql, adr)

mydb.commit()

```

Class Persona

```

from datetime import datetime
from io import open
import pandas as pd
from Class_MYSQL import Database as Mysql
from Class_Mongo import MongoDB
import pymongo
mydb=Mysql()
mongo=Mongodb()

```

```

class Persona:

```

```

    def __init__(self,id=None,nombre="", edad="",celular=""):
        self.idp=id
        self.nombre = nombre
        self.edad = edad
        self.celular=celular
        self.list = []
        self.x=0
        self.y=0
        self.importacion = pd.read_csv('Personas.csv')
        self.id = self.importacion['idp']
        self.nombre2 = self.importacion['nombre']
        self.edad2= self.importacion['edad']
        self.celular2 = self.importacion['celular']
        self.personas=''

```

```

    def agregarpersona(self,nombre,edad,celular):
        basededatos = Persona.regreso(self)
        for i in basededatos:
            bd = int(i)
            self.personas=open("Personas.csv", "a",newline='')
            self.x+=1
            self.idp=self.x
            new=Persona(self.idp,nombre,edad,celular)

```

```

self.personas.write(str(self.idp)+","+nombre+","+edad+","+celular+'\n')
self.personas.close()
#mysql
if bd==1:
    mydb.agregarpersona(str(self.idp),nombre,edad,celular)
#mongodb
if bd==2:

```

```

        mongo.ingresarpersona(str(self.idp),nombre, edad, celular)
    return self.list.append(new)

def actualizarpersona(self,nombre,edad,celular):
    basededatos = Persona.regreso(self)
    for i in basededatos:
        bd = int(i)
        c=0
        e=0
        s=0
    for element in self.list:

        if(element.nombre==nombre):

            g=element.idp
            s=1
        else:
            if (element.idp != ""):
                if(s==0):
                    c+=1
    if (s == 1):
        d = 0
        print(g)
        new = Persona(g, nombre, edad, celular)
        self.personas = open('Personas.csv', 'r+')
        texto = self.personas.readlines()
        texto[c+1] =(str(g)+","+nombre+","+edad+","+celular+'\n')
        #mysql
        if bd == 1:
            mydb.actualizarpersona(edad,celular,g)
        #mongodb
        if bd == 2:
            mongo.actualizarpersona(g,edad,celular)
        self.personas.seek(0)
        self.personas.writelines(texto)
        self.personas.close()
        self.list[c] = new
    e=1
    if (e == 0):
        print("Lo sentimos la Persona no se encuentra Disponible")

def eliminarpersona(self, nombre=None):
    basededatos = Persona.regreso(self)
    for i in basededatos:
        bd = int(i)
        c = 0
    for element in self.list:

        if (element.nombre == nombre):
            self.personas = open("Personas.csv", "r+")
            lineas =self.personas.readlines()
            self.personas.close()
            self.personas= open("Personas.csv", "w")
            for linea in lineas:

```

```

        if linea!=
(str(element.idp)+","+nombre+","+element.edad+","+element.celular+'\n'):
            self.personas.write(linea)
        self.list.pop(c)
        #mysql
        if bd==1:

            mydb.eliminarpersona(element.idp)

        #mongodb
        if bd == 2:
            mongo.eliminarpersona(element.idp)

        self.personas.close()
        c += 1

def verpersonas(self, nombre=None):
    c = 0
    a=0
    if(nombre!=None):

        for element in self.list:

            if (element.nombre == nombre):
                return self.list[c]
                a=1
            c += 1
        if(a==0):
            nombre = ""
            edad=""
            idp=""
            celular=""
            new = Persona(idp, nombre, edad, celular)

            return new
        else :
            return self.list

def pasarpersona(self, id=None):
    c = 0
    a = 0
    if (id != None):
        for element in self.list:

            if (element.idp == id):
                return self.list[c]
                a = 1
            c += 1
        if (a == 0):
            nombre = ""
            edad = ""
            idp = ""
            celular = ""
            new = Persona(idp, nombre, edad, celular)

            return new

def pasardatospersonas(self):

```

```

basededatos = Persona.regreso(self)
for i in basededatos:
    bd=int(i)
    if bd == 1:
        desigualdad=mydb.verificardatoserroneos()
    if bd == 2:
        mongodesigualdad=mongo.verificardatos()
    cont=0
    for j in range(len(self.nombre2)):
        id=self.id[j]
        nombre3 = str(self.nombre2[j])
        edad3 = str(self.edad2[j])
        celular3 = str(self.celular2[j])
        self.x =id
        self.idp = self.x
        new = Persona(str(self.idp), nombre3, edad3, celular3)
    #condicion mysql
    if bd == 1:
        for x in desigualdad:
            for i in x:
                if id == i:
                    cont=1
            if cont==0:
                mydb.agregarapersona(str(self.idp),nombre3, edad3, celular3)
            cont=0
    #condicion mongodb
    if bd == 2:
        for x in mongodesigualdad.find():
            d =int(x['_id'])
            if d == id:
                cont=1

        if cont==0:

            mongo.ingresarpersona(str(self.idp),nombre3, edad3, celular3)
            cont=0
            self.list.append(new)
def basededatos(self,opcion):
    self.y=opcion
def regreso(self):
    return self.y
def persona(self, idp=""):
    c = +0
    a = 0

    if (idp != ""):
        for element in self.list:
            if (element.idp == idp):
                return element.nombre
                a = 1
            c += 1
        if (a == 0):
            nombre = ""
            idp = ""
            fecha = ""

            return nombre

```



```
    else:
        return "fallido"
```

Main

```
from Class_Persona import Persona
from Class_Material import Material as Pe
from Class_Listadepedidos import Listadepedidos as L
from Class_interface import Inteface
from colorama import *
from Class_MYSQL import Database as Mysql
from Class_Mongo import Mongodb
import random

if __name__ == "__main__":
    my=Mysql()
    mongo = Mongodb()
    opcion=0

    print("En que base de datos quieres guardar"
          "\n1)Mysql"
          "\n2)Mongo DB")
    opcion = (input(Fore.LIGHTRED_EX + "Seleccione una opcion:"))
#SE SELCCIONA BD ALETAORIAMENTE
    opcion = str(opcion)
    if opcion == "1":
        print("-----BD FUNCIONANDO CON MYSQL-----")
    elif opcion == "2":
        mongo.comprobarcolleccion()
        print("-----BD FUNCIONANDO CON MongoDB-----")

    if opcion == "1":
        my.compruebalabasededatos()
        l=L()
        P=Pe()
        lista = Persona()
        lista.basededatos(opcion)
        P.basededatos(opcion)
        l.basededatos(opcion)
        accion=0
        lista.pasardatospersonas()
        P.Extraerdatos()
        l.pasardatospedidos()
        while accion >=0 and accion <=16:
            i = Inteface()
            i.Menu()
            accion = int(input( Fore.LIGHTRED_EX+"Seleccione una opcion:"))
            if accion == 1:
                nombre = input("Ingrese su nombre: ")
                edad = input("su edad: ")
                celular = input("ingrese su celular:")
                lista.agregarpersona(nombre,edad,celular)
            elif accion == 2:
                print("<-----LISTA DE USUARIOS GUARDADOS----->")
                objeto = lista.verpersonas()
```

```

        for i in objeto:
            print("id|", "nombre|", "edad|", "celular")
            print(str(i.idp) + "|" + i.nombre + "|" + i.edad + "|" + i.celular)
elif accion == 3:
    p=0
    objeto = lista.verpersonas()
    for i in objeto:
        print("id|", "nombre|", "edad|", "celular")
        print(str(i.idp) + " " + i.nombre + " " + i.edad + " " + i.celular)
    numero = input("ingrese su nombre:")
    edad = input("su nueva edad: ")
    celular = input("su nuevo celular:")
    lista.actualizarpersona(numero, edad, celular)
elif accion == 4:
    objeto = lista.verpersonas()
    for i in objeto:
        print("id|", "nombre|", "edad|", "celular")
        print(str(i.idp) + " " + i.nombre + " " + i.edad + " " + i.celular)
    nombre = input("Ingresa el nombre de usuario a eliminar:")
    objeto = lista.verpersonas(nombre)
    if (objeto.nombre != ""):
        condicion=l.verlistadepedidos(objeto.idp)
        if condicion.nombre!="":
            print("No se puede eliminar la persona por que se encuentra el la
tabla de pedidos")
            if condicion.nombre=="":
                lista.eliminarpersona(nombre)
                print("Usuario eliminado")
    else:
        print("No existe el usuario")
elif accion == 5:
    nombre = input("Ingresa al usuario que deseas ver:")
    i = lista.verpersonas(nombre)
    if(i.nombre!=""):
        print("id|", "nombre|", "edad|", "celular")
        print(str(i.idp) + " " + i.nombre + " " + i.edad + " " + i.celular)
elif accion == 6:
    nombre = input("Ingresa el Producto: ")
    fecha = input("fecha de caducacion: ")
    P.agregarproducto(nombre, fecha)
elif accion == 7:
    print("<-----PRODCUTOS DISPONIBLES----->")
    objeto2 = P.verproductos()
    for i in objeto2:
        if (i.nombre != ""):
            print("id|", "nombre|", "fecha de caducidad|")
            print(str(i.idp) + " " + i.nombre + " " + i.fechacaducacion)
elif accion == 8:
    objeto2 = P.verproductos()
    for i in objeto2:
        print("id|", "nombre|", "fecha de caducidad|")
        print(str(i.idp) + " " + i.nombre + " " + i.fechacaducacion)
    nombre = input("Ingresa el Producto: ")
    fecha = input("fecha de caducacion: ")
    P.actualizarproducto(nombre, fecha)
elif accion == 9:
    objeto2 = P.verproductos()
    for i in objeto2:

```

```

        print("id|","nombre|","fecha de caducidad|")
        print(str(i.idp) + " " + i.nombre + " " + i.fechacaducacion)
    nombre = input("Ingrese el Producto a eliminar: ")
    i = P.verproductos(nombre)
    if (i.nombre != ""):
        condicion=l.verlistadepedios2(i.idp)
        if condicion.nombre != "":
            print("No se puede eliminar el material por que se encuentra en la
tabla de pedidos")
        if condicion.nombre == "":
            P.eliminarproducto(nombre)
            print("Material eliminado")
    else:
        print("No existe ")
elif accion == 10:
    nombre = input("Ingrese el Producto a ver: ")
    i = P.verproductos(nombre)
    if (i.nombre != ""):
        print("id|","nombre|","fecha de caducidad|")
        print(str(i.idp) + " " + i.nombre + " " + i.fechacaducacion)
elif accion == 11:
    s=0
    f=0
    print("-----USUARIOS A SELECCIONAR-----")
    objeto = lista.verpersonas()
    for i in objeto:
        print("||" + i.nombre + "||")
    print("-----PRODUCTOS A SELECCIONAR-----")
    objeto2 = P.verproductos()
    for i in objeto2:
        print("||"+ i.nombre +"||")
    print("*****")
    name = input("Ingresa al usuario de la lista:")
    objeto = lista.verpersonas()
    for i in objeto:
        if (i.nombre==name):
            s=1
            persona=i.idp
    if(s==1):
        producto = input("Ingrese el Producto: ")
        i = P.verproductos(producto)
        if (i.nombre != ""):
            l.agregarlistadepedido(persona,i.idp)
        else:
            f=1
    if(f==1):
        print("no se encuentra el producto")
    if(s==0):
        print("no existe el usuario")
elif accion==12:
    print("<-----PRODUCTOS RENTADOS----->")
    objeto3=l.verlistadepedios()
    for i in objeto3:
        if (i.producto != ""):
            pro=str(i.producto)
            nombre=str(i.nombre)
            nombre=lista.persona(nombre)
            producto=P.productos(pro)

```

```

        print("nombre||producto")
        print(nombre + "||" + producto)

elif accion==13:
    objeto3 = l.verlistadepedios()
    for i in objeto3:
        if (i.producto != ""):
            pro = str(i.producto)
            nombre = str(i.nombre)
            nombre = lista.persona(nombre)
            producto = P.productos(pro)
            print("nombre||producto")
            print(nombre + "||" + producto)
    name = input("Ingresa al usuario de la lista:")
    objeto = lista.verpersonas()
    for i in objeto:
        if (i.nombre == name):
            s = 1
            persona = i.idp
    if (s == 1):
        if (i.nombre != ""):
            l.eliminarpedido(persona)
    if (s == 0):
        print("no existe el usuario")
elif accion == 14:
    s=0
    objeto3 = l.verlistadepedios()
    for i in objeto3:
        if (i.producto != ""):
            pro = str(i.producto)
            nombre = str(i.nombre)
            nombre = lista.persona(nombre)
            producto = P.productos(pro)
            print("nombre||producto")
            print(nombre + "||" + producto)
    name = input("Ingresa al usuario de la lista:")
    i = lista.verpersonas(name)
    print(i.nombre)
    if (i.nombre == name):
        persona=i.idp
        s = 1
    if (s == 1):
        print("-----PRODUCTOS A SELECCIONAR-----")
        objeto2 = P.verproductos()
        for i in objeto2:
            print("||nombre||fecha de caducidad||")
            print("||" + i.nombre + "||" + i.fechacaducacion + "||")
        print("*****")
        producto = input("Ingresa el Producto nuevo: ")
        i = P.verproductos(producto)
        if(i.nombre!=""):
            l.actualizarproducto(persona,i.idp)
    if (s == 0):
        print("no existe el usuario")
elif accion == 15:
    print("1. Búsqueda por persona")
    print("2.- Búsqueda por producto")
    seleccion = int(input("Seleccione una opcion:"))

```

```

if seleccion==1:
    s=0
    objeto3 = l.verlistadepedidos()
    for i in objeto3:
        if (i.nombre != ""):
            nombre = str(i.nombre)
            nombre = lista.persona(nombre)
            producto=P.productos(i.producto)
            print("nombre|producto")
            print(nombre + "||" + producto)
    name = input("Ingresa al usuario a ver:")
    i = lista.verpersonas(name)
    if (i.nombre == name):
        print("id|nombre|edad|celular")
        print(str(i.idp) + "||" + i.nombre + "||" + i.edad + "||" + i.celular)
        s = 1
    if (s == 0):
        print("no existe el usuario")
elif seleccion==2:
    s = 0
    objeto3 = l.verlistadepedidos()
    for i in objeto3:
        if (i.nombre != ""):
            nombre = str(i.nombre)
            nombre = lista.persona(nombre)
            producto = P.productos(i.producto)
            print("nombre|producto")
            print(nombre + "||" + producto)
    name = input("Ingresa al Producto a ver:")
    i = P.verproductos(name)
    if (i.nombre == name):
        print("id|nombre|fecha de caducidad|")
        print(str(i.idp) + " " + i.nombre + " " + i.fechacaducacion)
        s = 1
    if (s == 0):
        print("no existe el producto")
elif accion == 16:

    accion=20

```