

Actividad 01 (Repaso de Programación).

Hernandez Nieto Fernando

Seminario de Algoritmia I

1.1 Lineamientos de evaluación

Desarrolla en el lenguaje de programación que desees un programa de acuerdo a lo siguiente:

1. Crea la clase "Paquete" con los siguientes atributos:
 - id
 - origen
 - destino
 - peso
 - crea sus getter y setter (métodos de acceso).
2. Implementa la clase Paquetería (la cual hará uso de una lista simplemente ligada para administrar objetos de la clase Paquete) con los siguientes métodos:
 - Insertar al inicio
 - Eliminar al inicio
 - Mostrar
 - Guardar (archivo)
 - Recuperar (archivo)
3. Crea un menú con las siguientes opciones:
 - Agregar paquete: crea un objeto de la clase Paquete y lo inserta en la lista simplemente ligada (Paquetería).
 - Eliminar paquete: elimina el paquete que se encuentra al inicio de la lista de la Paquetería.
 - Mostrar: recorre la lista mostrando la información de cada paquete.
 - Guardar: crea un archivo .txt y vacía toda la información de todos los paquetes de la Paquetería (respaldar).
 - Recuperar: lee del archivo de la opción Guardar cada paquete y lo agrega a la Paquetería.

1.2 Lineamientos de evaluación

Se pondrá un X dentro de los corchetes de los lineamientos que se cumplieron.

- [X] El reporte está en formato Google Docs o PDF.
- [X] El reporte sigue las pautas del Formato de Actividades.
- [X] Se muestra código y captura de pantalla para agregar un objeto paquete dentro de la lista de la clase Paquetería.
- [X] Se muestra código y captura de pantalla para eliminar un objeto de la lista de paquete en la clase Paquetería.

- [X] Se muestra código y captura de pantalla para mostrar la información de toda la lista de paquetes en la clase Paquetería.
- [X] Se muestra código y captura de pantalla para guardar la lista de paquetes en un archivo (.txt).
- [X] Se muestra captura de pantalla del contenido del archivo .txt.
- [X] Se muestra código y captura de pantalla de la opción Recuperar.

Desarrollo

Para desarrollar este programa se utilizaron 2 archivos de python:

parcel.py

En este archivo se creó la clase “*Parcel*” donde creamos nuestra lista de paquetería y el menú para interactuar con las funciones: Agregar, Eliminar, Mostrar, Guardar, Recuperar.

package.py

En este archivo se creó la clase “*Package*” donde definimos los atributos de nuestro paquete: Id, Origen, Destino, Peso. También definimos los métodos *SET* y *GET*

Agregar paquete dentro de lista paquetería.

A continuación, se muestra el código donde pedimos al usuario que ingrese los datos que requerimos para llenar los atributos de nuestro paquete.

Tiene algunas restricciones en el caso del peso y del id del paquete, ya que estos son números se les aplico una validación para que no ingresen letras en lugar de números.

```
myPack = package.Package(0, "", "", 0)
myNum = "s"
while (isNum(myNum) == False):
    myNum = input("Ingrese el codigo del paquete: ")
    if (isNum(myNum)):
        myPack.set_Id(myNum)
    else:
        print("No se detecta numero valido...\n")
        print("Intente de nuevo...\n")
        input("Presione Enter para continuar...\n")

myNum = input("Ingrese el origen del paquete: ")
myPack.set_Origin(myNum)
myNum = input("Ingrese el destino del paquete: ")
myPack.set_Destination(myNum)

myNum = "s"
while (isNum(myNum) == False):
    myNum = input("Ingrese el peso del paquete: ")
    if (isNum(myNum)):
        myPack.set_Weight(myNum)
    else:
        print("No se detecta numero valido...\n")
```

```

        print("Intente de nuevo...\n")
        input("Presione Enter para continuar...\n")

    myNode = Node(myPack, None)
    ls.addItem(myPack)
    print(myPack)
    print("Ingresado con exito...\n")

```

Nuestra función “*addItem()*” es a la que enviamos nuestro objeto ya completo para ordenarlo en la lista.

El código de enseguida nos muestra como siempre que llegue un nuevo elemento este se colocara al inicio de nuestra lista.

Cuando recibimos nuestro objeto de tipo paquete enseguida se crea un nuevo nodo para nuestra lista, este es inicializado con nuestro objeto paquete y en su campo siguiente o nodo al que apunta lo configuramos en NULL o en este caso None.

```

def addItem(self, Value):
    myN = Node(Value, None)
    if self.First == None:
        self.First = myN
    else:
        myN.setNext(self.First)
        self.First = myN

```

Nuestro menú es muy simple, para ingresar un nuevo paquete teclearemos 1 y daremos ENTER.

```

....MENU....
1.- Agregar al inicio.
2.- Eliminar al inicio.
3.- Mostrar.
4.- Guardar Registro.
5.- Cargar Registro
0.- Salir.
Ingresa una opcion...
    Eleccion: 1

```

Después ingresamos los datos que se nos piden y como se puede ver se creo un nuevo paquete y fue ingresado a nuestra lista.

```

....MENU....
1.- Agregar al inicio.
2.- Eliminar al inicio.
3.- Mostrar.
4.- Guardar Registro.
5.- Cargar Registro
0.- Salir.
Ingrese una opcion...
    Eleccion: 1
Ingrese el codigo del paquete: 15453
Ingrese el origen del paquete: Mexico
Ingrese el destino del paquete: Canada
Ingrese el peso del paquete (Kg): 20
*****
Id-----: 15453
Origen--: Mexico
Destino-: Canada
Peso(Kg): 20

Ingresado con exito...

Presione Enter para continuar...

```

Eliminar paquete de la lista paquetería.

Se muestra el código utilizado para eliminar un elemento de la lista, en este caso siempre eliminará el primer elemento de la lista, antes de eso primero pondrá como primer elemento de la lista al siguiente que esta apuntando para no perder la referencia del resto de nodos.

En caso de que no existan elemento en nuestra lista solo nos mandara un mensaje diciéndonos que la lista está vacía.

```

def deleteItem(self):
    if self.First != None:
        aux = self.First
        self.First = aux.getNext()
        aux.__Next = None
        aux = None
        print("Eliminado con exito...\n")
    else:
        print("La lista esta vacia...")

```

Para esta prueba se ingresarán 3 paquetes y se mostrarán en pantalla.

```

----Lista de paquetes----
*****
Id-----: 6666
Origen--: Argentina
Destino-: Chile
Peso(Kg): 55

*****
Id-----: 15487
Origen--: EEUU
Destino-: Cuba
Peso(Kg): 5

*****
Id-----: 51274
Origen--: Mexico
Destino-: Venezuela
Peso(Kg): 3

ENTER para continuar.

```

Al eliminar con nuestra función nos daremos cuenta de que eliminara el último elemento en ingresar como si se tratara de una pila.

```

....MENU....
1.- Agregar al inicio.
2.- Eliminar al inicio.
3.- Mostrar.
4.- Guardar Registro.
5.- Cargar Registro
0.- Salir.
Ingrese una opcion...
      Eleccion: 2
Eliminado con exito...

ENTER para continuar...

```

```

----Lista de paquetes----
*****
Id-----: 15487
Origen--: EEUU
Destino-: Cuba
Peso(Kg): 5

*****
Id-----: 51274
Origen--: Mexico
Destino-: Venezuela
Peso(Kg): 3

ENTER para continuar.

```

Mostrar información de lista paquetería.

Como se ha podido ver en anteriores puntos estamos mostrando los elementos de la lista, esto es gracias a un simple recorrido gracias a la conexión de los nodos de nuestra lista. Mostramos el código a continuación:

```
def showItems(self):
    print("----Lista de paquetes----")
    if self.First != None:
        aux = self.First
        while aux != None:
            aux.showNode()
            aux = aux.getNext()
    else:
        print("La lista esta vacia...")
    input("ENTER para continuar.")
```

Para este ejemplo en pantalla ingresaremos 4 paquetes que se mostraran en pantalla.

```
----Lista de paquetes----
*****
Id-----: 15453
Origen--: Mexico
Destino-: Canada
Peso(Kg): 20

*****
Id-----: 6666
Origen--: Argentina
Destino-: Chile
Peso(Kg): 55

*****
Id-----: 15487
Origen--: EEUU
Destino-: Cuba
Peso(Kg): 5

*****
Id-----: 51274
Origen--: Mexico
Destino-: Venezuela
Peso(Kg): 3

ENTER para continuar.[]
```

Guardar información en archivo (.txt).

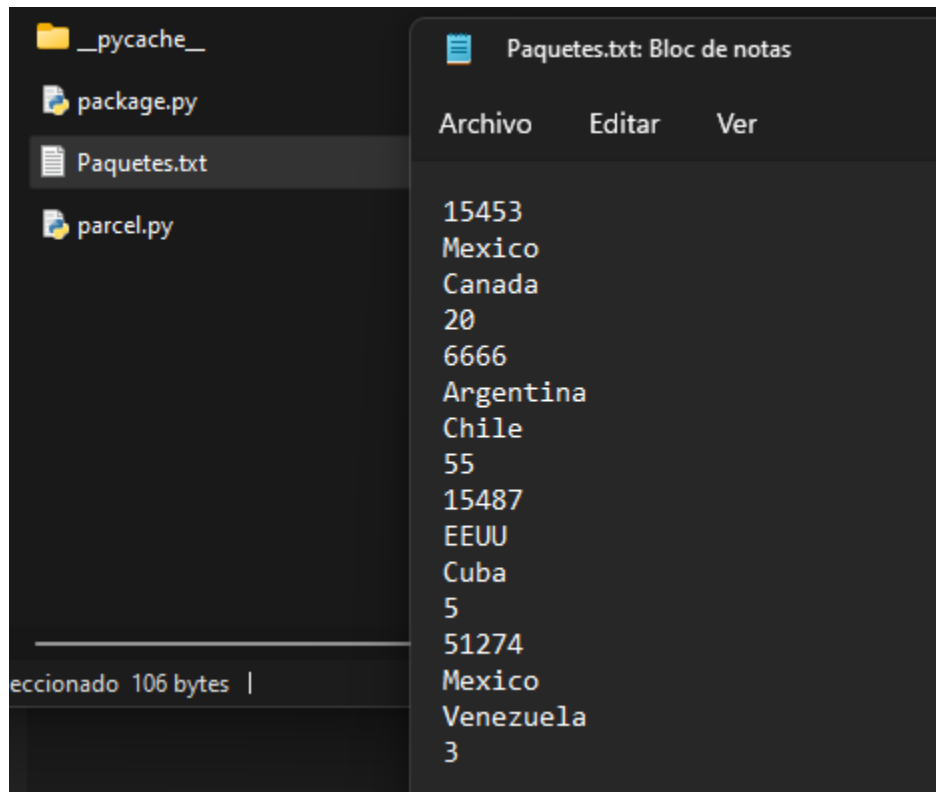
Para esto validamos si tenemos elementos en nuestra lista, después de esto creamos un nuevo archivo de texto donde guardaremos los campos de nuestro paquete por partes. Una vez que termina con toda la lista solo nos queda cerrar el archivo y nuestro respaldo quedara hecho.

```
def saveList(self):
    if ls.First != None:
        file = open('Paquetes.txt', 'w')
        aux = self.First
        while aux != None:
            file.write(aux.getData().get_Id())
            file.write("\n")
            file.write(aux.getData().get_Origin())
            file.write("\n")
            file.write(aux.getData().get_Destination())
            file.write("\n")
            file.write(aux.getData().get_Weight())
            file.write("\n")
            aux = aux.getNext()
        file.close()
        print("Archivo Guardado correctamente...")
    else:
        print("La lista esta vacia...")
```

Guardaremos los anteriores 4 paquetes en nuestro archivo, para poder mostrar el archivo creado y su contenido.

```
....MENU....
1.- Agregar al inicio.
2.- Eliminar al inicio.
3.- Mostrar.
4.- Guardar Registro.
5.- Cargar Registro
0.- Salir.
Ingrese una opcion...
    Eleccion: 4
Archivo Guardado correctamente...
Presione Enter para continuar...
█
```

Archivo creado:



Recuperar información de archivo (.txt).

Para cargar la información de nuevo a nuestra aplicación tenemos que leer las líneas y comenzar a guardar la información en variables, dependiendo de la línea en la que se encuentra se guardara como un atributo de nuestro paquete, una vez que llene todos sus atributos crearemos el objeto paquete y después lo ingresaremos a nuestra lista.

```
def loadList(self):  
    file = open("Paquetes.txt", "r")  
    i = 0  
    aux = package.Package(0, "", "", 0)  
    for line in file:  
        if i == 0:  
            myId = line.rstrip()  
        if i == 1:  
            myOri = line.rstrip()  
        if i == 2:  
            myDet = line.rstrip()  
        if i == 3:  
            myW = line.rstrip()  
        i += 1  
        if i == 4:
```



```
aux = package.Package(myId, myOri, myDet, myW)
self.addItem(aux)
i = 0
```

Volveremos a correr el programa con la lista vacía y la llenaremos con nuestro archivo anterior mente creado “*Paquetes.txt*”.

Recien iniciado el programa:

```
----Lista de paquetes----
La lista esta vacia...
ENTER para continuar.[]
```

Despues de recuperar la informacion:

```
----Lista de paquetes----
*****
Id-----: 51274
Origen--: Mexico
Destino-: Venezuela
Peso(Kg): 3

*****
Id-----: 15487
Origen--: EEUU
Destino-: Cuba
Peso(Kg): 5

*****
Id-----: 6666
Origen--: Argentina
Destino-: Chile
Peso(Kg): 55

*****
Id-----: 15453
Origen--: Mexico
Destino-: Canada
Peso(Kg): 20

ENTER para continuar.[]
```

Conclusiones

Crear esta actividad en Python fue el mayor inconveniente que me topé en esta actividad, ya que nunca había programado en este lenguaje, me decidí a hacerlo en este lenguaje para irme familiarizando en las siguientes actividades no llegar en cero,

Más que nada el hecho de la sintaxis fue lo que más me demoró al crear la actividad, ya que buscaba funciones similares a las que utilizaría en C o C++, pero no encontré algunas cosas como lo fue el SWITCH(), esta estructura de control no se encuentra como tal en Python.

Tampoco encontré como definir los prototipos para poder utilizar métodos o clases que se encuentren más abajo en el archivo.


Este lenguaje se me hizo extrañamente limpio, aunque la costumbre de utilizar C me resultó algo raro de programar al principio.

Referencias

Se vieron algunos videos para poder crear la actividad 1, a continuación estarán los nombres de los videos y sus respectivas URL.

¿Cómo usar CLASES en PYTHON?

<https://www.youtube.com/watch?v=9x7RK6mb1uA>

Métodos Accesos (GET - SET) En Python | Curso Python 3  # 33

https://www.youtube.com/watch?v=6xoo1Ya_uQ0

25. Programación en Python | Colecciones | Listas

<https://www.youtube.com/watch?v=xdCJa2QXmJ8>

★ Listas enlazadas en Python

https://www.youtube.com/watch?v=asO724_WIJ0

Importar archivos propios (módulos) en Python

<https://www.youtube.com/watch?v=rYcluou5gEo>

31.- Curso Python - Clases y Objetos

<https://www.youtube.com/watch?v=vdz9HGfFguc>

 PYTHON POO: CÓMO IMPRIMIR OBJETOS DE CUALQUIER CLASE

https://www.youtube.com/watch?v=qn-cc5_vaQc

Python - Ejercicio 113: Validar si un Número Ingresado por el Usuario es Válido

<https://www.youtube.com/watch?v=Gv7nHbRVk9s>

Python: ¿Cómo grabar la información en un archivo TXT? (Básico)

https://www.youtube.com/watch?v=DR_G1EF7C1Y

Python 3 Tutorial - 10 Lectura y Escritura de Archivos
<https://www.youtube.com/watch?v=5svp6GldzzA>

Código

package.py

```
from re import S
from typing import get_origin

class Package():
    def __init__(self, id, origin, destination, weight):
        self.__Id = id
        self.__Origin = origin
        self.__Destination = destination
        self.__Weight = weight

    #Getters
    def get_Id(self):
        return self.__Id

    def get_Origin(self):
        return self.__Origin

    def get_Destination(self):
        return self.__Destination

    def get_Weight(self):
        return self.__Weight

    #Setters
    def set_Id(self, myId):
        self.__Id = myId

    def set_Origin(self, myO):
        self.__Origin = myO

    def set_Destination(self, myDes):
        self.__Destination = myDes

    def set_Weight(self, myW):
        self.__Weight = myW
```

```

#Impresion en pantalla del objeto
def __str__(self):
    return( f"*****\n" +
            f"Id-----: {self.__Id}\n" +
            f"Origen--: {self.__Origin}\n" +
            f"Destino-: {self.__Destination}\n" +
            f"Peso(Kg): {self.__Weight}\n")

```

parcel.py

```

from email.policy import default
from os import system
import pickle
import package

class Node:          # Clase Nodo
    def __init__(self, myData, next):
        self.__Data = myData
        self.__Next = next

    def showNode(self):
        print(self.__Data)

    def getData(self):
        return self.__Data

    def setNext(self, Value):
        self.__Next = Value

    def getNext(self):
        return self.__Next

class Parcel:        # Clase paqueteria (Lista)
    def __init__(self):
        self.First = None

    def addItem(self, Value):
        myN = Node(Value, None)
        if self.First == None:
            self.First = myN

        else:
            myN.setNext(self.First)

```

```

        self.First = myN

def saveList(self):
    if ls.First != None:
        file = open('Paquetes.txt', 'w')
        aux = self.First
        while aux != None:
            file.write(aux.getData().get_Id())
            file.write("\n")
            file.write(aux.getData().get_Origin())
            file.write("\n")
            file.write(aux.getData().get_Destination())
            file.write("\n")
            file.write(aux.getData().get_Weight())
            file.write("\n")
            aux = aux.getNext()
        file.close()
        print("Archivo Guardado correctamente...")
    else:
        print("La lista esta vacia...")

def loadList(self):
    file = open("Paquetes.txt", "r")
    i = 0
    aux = package.Package(0, "", "", 0)
    for line in file:
        if i == 0:
            myId = line.rstrip()
        if i == 1:
            myOri = line.rstrip()
        if i == 2:
            myDet = line.rstrip()
        if i == 3:
            myW = line.rstrip()
        i += 1
        if i == 4:
            aux = package.Package(myId, myOri, myDet, myW)
            self.addItem(aux)
            i = 0

def deleteItem(self):
    if self.First != None:
        aux = self.First
        self.First = aux.getNext()
        aux.__Next = None

```

```

        aux = None
        print("Eliminado con exito...\n")
    else:
        print("La lista esta vacia...")

def showItems(self):
    print("----Lista de paquetes----")
    if self.First != None:
        aux = self.First
        while aux != None:
            aux.showNode()
            aux = aux.getNext()
    else:
        print("La lista esta vacia...")
    input("ENTER para continuar.")

def isNum(myNum):
    try:
        int(myNum)
        return True
    except ValueError:
        return False

def mainMenu():
    print("....MENU....")
    print("1.- Agregar al inicio.")
    print("2.- Eliminar al inicio.")
    print("3.- Mostrar.")
    print("4.- Guardar Registro.")
    print("5.- Cargar Registro")
    print("0.- Salir.")
    print("Ingresa una opcion... ")

if __name__ == "__main__":
    ls = Parcel()
    while(True):
        system("cls")
        mainMenu()
        num = input("\tEleccion: ")
        if num == "1":
            myPack = package.Package(0, "", "", 0)
            myNum = "s"

            while(isNum(myNum) == False):
                myNum = input("Ingresa el codigo del paquete: ")

```

```

        if(isNum(myNum)):
            myPack.set_Id(myNum)
        else:
            print("No se detecta numero valido...\n")
            print("Intente de nuevo...\n")
            input("Presione Enter para continuar...\n")

myNum = input("Ingrese el origen del paquete: ")
myPack.set_Origin(myNum)
myNum = input("Ingrese el destino del paquete: ")
myPack.set_Destination(myNum)

myNum = "s"
while(isNum(myNum) == False):
    myNum = input("Ingrese el peso del paquete (Kg): ")
    if(isNum(myNum)):
        myPack.set_Weight(myNum)
    else:
        print("No se detecta numero valido...\n")
        print("Intente de nuevo...\n")
        input("Presione Enter para continuar...\n")

myNode = Node(myPack, None)
ls.addItem(myPack)
print(myPack)
print("Ingresado con exito...\n")
input("Presione Enter para continuar...\n")

if num == "2":
    ls.deleteItem()
    input("ENTER para continuar...\n")

if num == "3":
    system("cls")
    ls.showItems()

if num == "4":
    ls.saveList()
    input("Presione Enter para continuar...\n")

if num == "5":
    ls.loadList()
    print("Se cargo correctamente la lista\n")

if num == "0":

```

```
print("Saliendo del sistema...")  
print("Hasta la proxima. ")  
  
exit()
```