**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).

1. 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)
2. 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 Paqueteria.
3. Reporta en un *Google Docs* tu desarrollo usando las pautas (guidelines) del [Formato de Actividades](https://www.notion.so/Formato-de-Actividades-d098e1ce3a62423ca47c3661c5d93444).

**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](https://www.notion.so/Formato-de-Actividades-d098e1ce3a62423ca47c3661c5d93444).
* [ ***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.

***Texto

Descripción generada automáticamente***

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

***Texto

Descripción generada automáticamente***

***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.

Texto

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

***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.

Texto

Descripción generada automáticamente

***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()

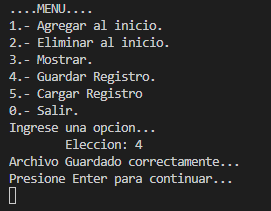
            file.close()

            print("Archivo Guardado correctamente...")

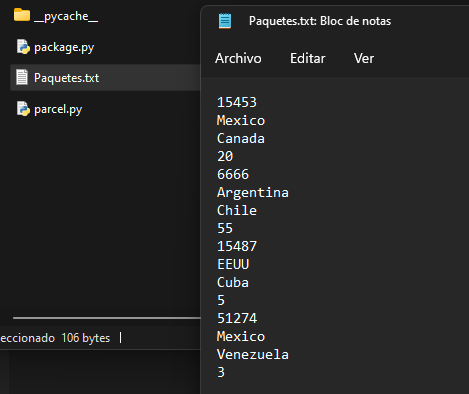
        else:

            print("La lista esta vacia...")

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



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”*.

**Conclusiones**

En esta sección escribirás los problemas con los que te enfrentaste, que aprendiste,

que no entendiste desde el inicio y como hiciste para entenderlo (o sigues sin entender),

puntos de vista con otras cosas que conoces o visto de otros programadores, etc.

**Referencias**

En esta sección colocarás las páginas web que consultaste, ya sea desde los recursos proveídos por el profesor, como aquellos que te sirvieron para realizar la actividad, como videos, libros, tutorial, cursos, etc

.

Para referencias  web, agregar la url completa, título (del video, tutorial o curso) y autor (de existir el autor).

Para libros habrá que incluir título del libro, año, autor(es)

**Código**

En esta sección agregarás el código fuente de tu actividad.

Es posible que la actividad requiera incluir todo el código o solamente el código relevante.

Trata que el código sea legible, tenga formato y tratando de imitar el siguiente resultado:

#include <stdio.h>

    int main() {

        char word[] = "Hola Mundo!";

        printf("%s", word);

        return 0;

    }

No usar capturas de pantalla para mostrar el código.