

Actividad 01 - Clases

Repaso de Programación

JAIRO CAIN SANCHEZ ESTRADA//
Luis Angel Elisea Graciano

SEMINARIO DE SOLUCION DE PROBLEMAS DE
ALGORITMIA

Lineamientos de evaluación

- Al programa corre sin errores
- El programa ejecuta las funciones esperadas sin problema alguno
- Se utilizan clases para poder tener un control en los elementos creados
- Las clases cuentan con sus métodos de acceso getter y setter
- El programa Inserta al inicio
- Elimina al inicio
- Muestra los paquetes almacenados en la lista enlazada
- Guardar la información en un archivo txt

- Recuperar la información del archivo txt

Desarrollo

Al momento de ejecutar el programa se muestra este menú con las opciones de Agregar, Eliminar, Mostrar, Guardar, Recuperar.

```
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....
```

```
def Menu():
    New_paqueteria=Paqueteria()

    while True:
        print("Paqueteria...\n")
        print("1)Agregar paquete...\n")
        print("2)Eliminar paquete...\n")
        print("3)Mostrar paquetes...\n")
        print("4)Guardar paquetes...\n")
        print("5)Recuperar paquetes...\n")
        opcion=int(input("Ingrese una opcion...."))
        if opcion == 1:
            borrar()
            auxiliar_almacenapaquete=pedir_datos()
            New_paqueteria.Insertar(auxiliar_almacenapaquete)

        elif opcion ==2:
            borrar()
            New_paqueteria.Eliminar()

        elif opcion ==3:
            borrar()
            New_paqueteria.Mostrar()

        elif opcion ==4:
            borrar()
            New_paqueteria.guardar()

        elif opcion ==5:
            borrar()
            New_paqueteria.recuperar(New_paqueteria)

        else:
            print("Gracias por su visita...")
            exit()

Menu();
```

Se escoge la opción uno para insertar un paquete y se llenan los campos necesarios. Al llamar al menú se manda a llamar a la clase Paqueteria para poder utilizar la función insertar después de capturar los datos del paquete y en la función pedir datos se llama a la clase Paquete y se mandan los parámetros que la componen.

```
C:\Windows\py.exe
Ingrese el id del paquete...5696
Ingrese el origen del paquete...mexico
Ingrese el destino del paquete...peru
Ingrese el peso del paquete en kilos...200
```

```
def Menu():
    New_paqueteria=Paqueteria()

    while True:
        print("\nMenu")
        print("1) Agregar paquete")
        print("2) Eliminar paquete")
        print("3) Mostrar paquetes")
        print("4) Guardar paquetes")
        print("5) Recuperar paquetes")
        opcion = int(input("Ingrese una opcion..."))

        if opcion == 1:
            borrar()
            auxiliar_almacenapaquete=pedir_datos()
            New_paqueteria.Insertar(auxiliar_almacenapaquete)
```

```
def Insertar(self, data):
    self.head = nodo(data=data, next=self.head)
```

```
def pedir_datos():
    id=int(input("Ingrese el id del paquete..."))
    origen=str(input("Ingrese el origen del paquete..."))
    destino=str(input("Ingrese el destino del paquete..."))
    peso=float(input("Ingrese el peso del paquete en kilos..."))
    new_paquete=Paquete(id,origen,destino,peso)
    return new_paquete
```

Se escoge la opción tres para poder mostrar el paquete que se acaba de capturar, se recorren todos los nodos existentes y se accede a su paquete y mostramos los datos con su método getter.

```
C:\Windows\py.exe
Ingrese el id del paquete...5696
Ingrese el origen del paquete...mexico
Ingrese el destino del paquete...peru
Ingrese el peso del paquete en kilos...200
Paqueteria...

1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion...3_
```

```
elif opcion ==3:
    borrar()
    New_paqueteria.Mostrar()
```

```
def Mostrar( self ):
    node = self.head
    if self.head is None:
        print("No hay paquetes disponibles...")
    else:
        while node != None:
            print("Paquete: "+str(node.data.getter_Id()), end ="\n")
            print("Origen: "+str(node.data.getter_Origen()), end ="\n")
            print("Destino: "+str(node.data.getter_Destino()), end ="\n")
            print("Peso: "+str(node.data.getter_Peso()), end ="\n\n")
            node = node.next
```

```
C:\Windows\py.exe
Paquete: 5696
Origen: mexico
Destino: peru
Peso: 0.2

Paqueteria...

1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....
```

Se escoge la opción cuatro para poder guardar el paquete en el archivo txt el cual se abrió como lectura, se recorren todos los nodos en la lista y se accede a sus métodos getter para poder escribirlo en el archivo txt

```
C:\Windows\py.exe
Paquete: 5696
Origen: mexico
Destino: peru
Peso: 0.2

Paqueteria...

1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....4
```

```
C:\Windows\py.exe
Guardado concluido

Paqueteria...

1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....
```

```
elif opcion ==4:
    borrar()
    New_paqueteria.guardar()
```

```
def guardar(self):
    Archivo_deco = open("Archivo_paqueteria.txt", "a") # escritura
    node = self.head
    while node != None:
        Archivo_deco.write(str(node.data.getter_Id())+",")
        Archivo_deco.write(node.data.getter_Origen()+",")
        Archivo_deco.write(node.data.getter_Destino()+",")
        Archivo_deco.write(str(node.data.getter_Peso())+"\n")
        node = node.next
    print("Guardado concluido\n")
    Archivo_deco.close()
```

Se ingresa la opcion dos para eliminar el primer paquete que esta en la lista y despues se escoge la opcion tres para mostrar que ya no hay archivos en la lista.

```

C:\Windows\py.exe
Guardado concluido
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....2_

```

```

C:\Windows\py.exe
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....3_

```

```

C:\Windows\py.exe
No hay paquetes disponibles....
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....

```

```

def Eliminar(self):
    if self.head is None:
        print("No hay paquetes disponibles....")
    else:
        curr = self.head
        self.head = curr.next

```

```

elif opcion ==2:
    borrar()
    New_paqueteria.Eliminar()

```

Se escoge la opcion cinco para poder recuperar los paquetes guardados en el archivo txt y guardarlos en la lista del programa, creando un paquete para ellos y guardandolos en la paqueteria, se lee el archivo .txt y se recorre linea por linea para almacenar los valores y guardarlos en variables para después utilizarlos como parametros en la funcion insertar y guardar el paquete.

```

elif opcion ==5:
    borrar()
    New_paqueteria.recuperar(New_paqueteria)

```

```

C:\Windows\py.exe
No hay paquetes disponibles....
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....5_

```

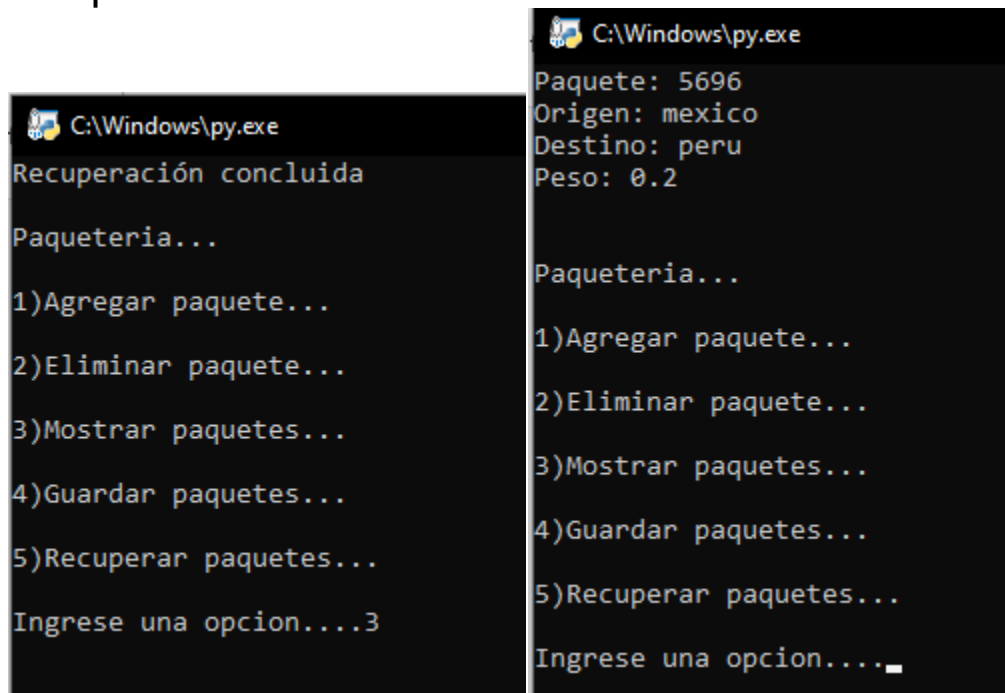
```

C:\Windows\py.exe
Recuperación concluida
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....

```

```
def recuperar(self, New_paqueteria):
    Archivo = open("Archivo_paqueteria.txt", "r") #Modo lectura
    node = self.head
    for linea in Archivo.readlines():
        lista = linea.split(',')
        id, origen, destino, peso = lista[0], lista[1], lista[2], lista[3]
        new_paquete = Paquete(id, origen, destino, peso)
        New_paqueteria.Insertar(new_paquete)
    print("Recuperación concluida\n")
    Archivo.close()
```

Por ultimo se escoge la opción tres para poder mostrar que se recuperaron los datos correctamente.



```
C:\Windows\py.exe
Recuperación concluida
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....3

C:\Windows\py.exe
Paquete: 5696
Origen: mexico
Destino: peru
Peso: 0.2
Paqueteria...
1)Agregar paquete...
2)Eliminar paquete...
3)Mostrar paquetes...
4)Guardar paquetes...
5)Recuperar paquetes...
Ingrese una opcion....
```

Conclusiones

Esta actividad me ayudo bastante a retomar conceptos de la POO y a recordar como si implementan ciertos métodos, ya que tenía bastante tiempo sin programar en Python, me ayudo a recordar todo lo importante como clases, el trabajar con listas enlazadas.

Mis problemas al principio fue retomar el programar en Python ya que tenía muy

deteriorados mis conocimientos en programación en Python y más en POO ya que tengo aproximadamente un año sin retomar POO, pero al revisar practicas más de semestres pasados y reforzar algunos conocimientos básicos en páginas web, rápidamente retome la actividad y la realice sin mayores problemas.

Referencias

Tagliaferri, L. (2020, December 15). *Cómo convertir tipos de datos en Python 3*. Digitalocean.com; DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-convert-data-types-in-python-3-es>

Importar datos de un fichero .txt a multiples variables en Python. (n.d.). Stack Overflow en español. Retrieved September 17, 2022, from <https://es.stackoverflow.com/questions/514136/importar-datos-de-un-fichero-txt-a-multiples-variables-en-python>

González, J. D. M. (2020, September 7). *Listas (arreglos o vectores) en Python: Uso y creación*. Programarya.com; ProgramarYa. <https://www.programarya.com/Cursos/Python/estructuras-de-datos/listas>

5. Estructuras de datos – documentación de Python - 3.10.7. (n.d.). Python.org. Retrieved September 17, 2022, from <https://docs.python.org/es/3/tutorial/datastructures.html>

Canepa, G. (n.d.). Uso de las funciones split y join en Python. Com.ar. Retrieved September 17, 2022, from <https://blog.carreralinux.com.ar/2017/07/uso-split-y-join-python/>

Código

```
import os

class Paquete:
    Id=0
    Origen=0
    Destino=0
    Peso=0
    def __init__(self,id,origen,destino,peso):
        self.Id=id
        self.Origen=origen
        self.Destino=destino
        self.Peso=peso

    def getter_Id(self):
        return self.Id

    def getter_Origen(self):
        return self.Origen

    def getter_Destino(self):
        return self.Destino

    def getter_Peso(self):
        return self.Peso

class nodo:
    def __init__(self, data = None, next = None):
        self.data = data
        self.next = next

class Paqueteria:
    def __init__(self):
        self.head = None

    def Insertar(self, data):
        self.head = nodo(data=data, next=self.head)

    def Eliminar(self):
        if self.head is None:
            print("No hay paquetes disponibles....")
        else:
```



```

        curr = self.head
        self.head = curr.next

def Mostrar( self ):
    node = self.head
    if self.head is None:
        print("No hay paquetes disponibles....")
    else:
        while node != None:
            print("Paquete: "+str(node.data.getter_Id()), end ="\n")
            print("Origen: "+str(node.data.getter_Origen()), end ="\n")
            print("Destino: "+str(node.data.getter_Destino()), end ="\n")
            print("Peso: "+str(node.data.getter_Peso()), end ="\n\n")
            node = node.next

def guardar(self):
    Archivo_deco = open("Archivo_paqueteria.txt", "a") # escritura
    node = self.head
    while node != None:
        Archivo_deco.write(str(node.data.getter_Id())+",")
        Archivo_deco.write(node.data.getter_Origen()+",")
        Archivo_deco.write(node.data.getter_Destino()+",")
        Archivo_deco.write(str(node.data.getter_Peso())+"\n")
        node = node.next
    print("Guardado concluido\n")
    Archivo_deco.close()

def recuperar(self,New_paqueteria):
    Archivo = open("Archivo_paqueteria.txt","r") #Modo lectura
    node = self.head
    for linea in Archivo.readlines():
        lista = linea.split(',')
        id, origen, destino, peso = lista[0], lista[1], lista[2], lista[3]
        new_paquete=Paquete(id,origen,destino,peso)
        New_paqueteria.Insertar(new_paquete)
    print("Recuperación concluida\n")
    Archivo.close()

def borrar():
    os.system("cls")

def pedir_datos():
    id=int(input("Ingrese el id del paquete..."))
    origen=str(input("Ingrese el origen del paquete..."))

```

```

destino=str(input("Ingrese el destino del paquete..."))
peso=float(input("Ingrese el peso del paquete en kilos..."))
new_paquete=Paquete(id,origen,destino,peso)
return new_paquete

def Menu():
    New_paqueteria=Paqueteria()

    while True:
        print("Paqueteria...\n")
        print("1)Agregar paquete...\n")
        print("2)Eliminar paquete...\n")
        print("3)Mostrar paquetes...\n")
        print("4)Guardar paquetes...\n")
        print("5)Recuperar paquetes...\n")
        opcion=int(input("Ingrese una opcion...."))
        if opcion == 1:
            borrar()
            auxiliar_almacenapaquete=pedir_datos()
            New_paqueteria.Insertar(auxiliar_almacenapaquete)

        elif opcion ==2:
            borrar()
            New_paqueteria.Eliminar()

        elif opcion ==3:
            borrar()
            New_paqueteria.Mostrar()

        elif opcion ==4:
            borrar()
            New_paqueteria.guardar()

        elif opcion ==5:
            borrar()
            New_paqueteria.recuperar(New_paqueteria)

        else:
            print("Gracias por su visita...")
            exit()
Menu();

```