

Actividad 01 — Repaso de programación

Hernández Lomelí Diego Armando

Seminario de algoritmia

Lineamientos de actividad.

- El reporte está en formato Google Docs o PDF.
- El reporte sigue las pautas del [Formato de Actividades](<https://www.notion.so/Formato-de-Actividades-d098e1ce3a62423ca47c3661c5d93444>).
- Se muestra código y captura de pantalla para agregar un objeto `paquete` dentro de la lista de la clase `Paqueteria`.
- Se muestra código y captura de pantalla para eliminar un objeto de la lista de `paquete` en la clase `Paqueteria`.
- Se muestra código y captura de pantalla para mostrar la información de toda la lista de `paquete` en la clase `Paqueteria`.
- Se muestra código y captura de pantalla para guardar la lista de `paquete` en un archivo .txt.
- Se muestra captura de pantalla del contenido del archivo .txt.
- Se muestra código y captura de pantalla de la opción Recuperar.

Desarrollo

El desarrollo inicio desde una clase que funge como modelo llamada “**paquete**”, tiene 4 atributos

- Id
- Origen
- Destino
- Peso

El tipo de dato no se especificó por lo que todos son cadenas de caracteres a excepción del peso, que es un decimal doble.

La siguiente clase fue “**Paqueteria**” con 4 metodos únicos y 1 metodo sobrescrito (toString()), los métodos generaron fueron.

- void insertarInicio(p)- p:paquete
 - Agrega el paquete recibido al inicio de la lista
- void eliminarInicio
 - Elimina el paquete que este en el inicio de la lista
- void mostrar
 - Imprime en consola los paquetes almacenados en la lista actual
- void guardar
 - almacena en un .txt los paquetes guardados hasta el momento
- void recuperar

- lee un .txt previamente generado y genera paquetes en base a los atributos encontrados

Finalmente sigue la clase main, solo consta de un menú que lee las opciones a las que acceder y recopila los datos para almacenar nuevos paquetes, guardar o recuperar los paquetes previos.

1. Agregar un “**paquete**” a la lista.

Código dentro del menú.

```
case MENU_AGREGAR:{
    paquete = new Paquete();
    System.out.println("Ingrese el ID del paquete");
    String a = scanner.nextLine();
    paquete.setId(scanner.nextLine());
    System.out.println("Ingrese el origen del paquete");
    paquete.setOrigen(scanner.nextLine());
    System.out.println("Ingrese el destino del paquete");
    paquete.setDestino(scanner.nextLine());
    System.out.println("Ingrese el peso del paquete");
    paquete.setPeso(scanner.nextDouble());
    //Finalmente guardamos el paquete al inicio de la lista
    listaPaquetes.insertarInicio(paquete);
    System.out.println("Paquete generado");
    break;
}
```

```
=====
Agregar paquete
=====
-
Ingrese el ID del paquete
1100231
Ingrese el origen del paquete
Alemania
Ingrese el destino del paquete
Guatemala
Ingrese el peso del paquete
1
Paquete generado
```

Código dentro de paquetería

```
public void insertarInicio( Paquete p){
    paquetes.add(0,p);
}
```

2. Eliminar al inicio

Código dentro del menú.

```
case MENU_ELIMINAR:{
    listaPaquetes.eliminarInicio();
    break;
}
```

Código dentro de paquetería

```

public void eliminarInicio( ){
    if ( paquetes.isEmpty() ){
        System.out.println("No hay paquetes para eliminar");
        return ;
    }
    paquetes.remove(0);
}

```

```

=====
Eliminar primer paquete
=====
-

```

3. Mostrar

Código dentro del menú.

```

case MENU_MOSTRAR:{
    listaPaquetes.mostrar();
    break;
}

```

Captura previa a la eliminación

```

=====
Desplegando paquetes
=====
-

    {
id=1100231,
    origen=Alemania,
destino=Guatemala,
peso=1.0
    }

```

Código dentro de paquetería

```

public void mostrar(){
    System.out.println(this.toString());
}

```

4. Guardar (archivo)

Código dentro del menú.

```

case MENU_GUARDAR:{
    System.out.println("Seleccionando ruta de guardado");
    JFileChooser f = new JFileChooser();
    f.setDialogType(JFileChooser.OPEN_DIALOG);
    f.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    f.showDialog(null, "Seleccione la ruta de guardado");
    String ruta = f.getSelectedFile().getAbsolutePath();
    listaPaquetes.guardar(ruta);
    System.out.println("Paquetes guardados");
    break;
}

```

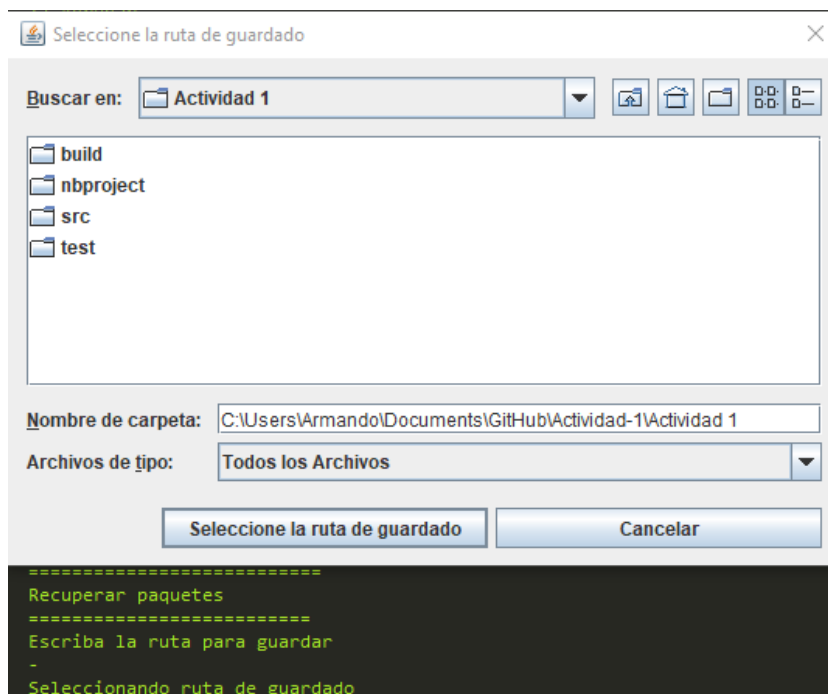
Código dentro de paquetería

```
public void guardar( String ruta ) throws FileNotFoundException,
IOException{

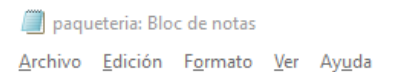
    try (FileOutputStream rutaSalida = new FileOutputStream(
        ruta + "/paqueteria.txt"
    )) {
        rutaSalida.write( this.toString().getBytes() );
    }

}
```

Captura selección ruta (previo a eliminación)



Archivo generado.



```
{
id=1100231,
origen=Alemania,
destino=Guatemala,
peso=1.0
}
```

5. Recuperar (archivo)

Código dentro del menú.

```
case MENU_RECUPERAR:{
    System.out.println("Seleccionando ruta de recuperación");
    JFileChooser f = new JFileChooser();
    f.setDialogType(JFileChooser.OPEN_DIALOG);
    f.setFileSelectionMode(JFileChooser.FILES_ONLY);
    f.showDialog(null, "Seleccione la ruta de guardado");
    String ruta = f.getSelectedFile().getAbsolutePath();
    listaPaquetes.recuperar(new File (ruta));
    System.out.println("Paquetes recuperado");
    break;
}
```

Código dentro de paquetería

```
public void recuperar( File archivo) throws IOException,
ClassNotFoundException{
    try (FileInputStream rutaSalida = new FileInputStream(
        archivo
    )) {
        archivo.createNewFile();
        String cadenaFinal = "";
        int caracterPrevio;
        do{
            caracterPrevio = rutaSalida.read();
            if( caracterPrevio > 0){
                char caracter = (char)caracterPrevio;
                cadenaFinal += caracter;
            }
        } while( caracterPrevio > 0 );
        //Limpiamos todos los paquetes encontrados
        paquetes.removeAll(paquetes);
        Paquete paquete = null;
        for( int i = 0 ; i < cadenaFinal.length(); i++){
            //Saber si termino un nuevo objeto para no asignar nada
            if( cadenaFinal.charAt( i ) == '}' ) {
                //Agregamos el paquete a la lista
                paquetes.add(paquete);
                paquete = null;
            }
            //Saber si empieza un nuevo objeto
            if( paquete == null ){
                //Se inicio un nuevo objeto
            }
        }
    }
}
```

```

        if( cadenaFinal.charAt( i ) == '{' ) {
            paquete = new Paquete();
            i++;
        }
    } else if( paquete != null){
        String identificador = "";
        String valorEncontrado = "";
        //Intentamos obtener los parametros
        //Hasta que encontremos un "=" detenemos la obtencion del
identificador
        while( cadenaFinal.charAt( i ) != '='){
            identificador += cadenaFinal.charAt( i ) == ' '?
                "":cadenaFinal.charAt( i );
            i++;
            //Si llegamos al limite de la cadena cortamos el proceso
            if( i >= cadenaFinal.length() ){
                break;
            }
        }
        i++;
        //Intentamos obtener el valor
        while( cadenaFinal.charAt( i ) != '\n' ){
            valorEncontrado += cadenaFinal.charAt( i ) == ','?
                "":cadenaFinal.charAt( i );
            i++;
            //Si llegamos al limite de la cadena cortamos el proceso
            if( i >= cadenaFinal.length() ){
                break;
            }
        }
        //Asignación de las valores por identificador
        if( "id".equals(identificador)){
            paquete.setId(valorEncontrado);
        }
        if( "origen".equals(identificador)){
            paquete.setOrigen(valorEncontrado);
        }
        if( "destino".equals(identificador)){
            paquete.setDestino(valorEncontrado);
        }
        if( "peso".equals(identificador)){
            paquete.setPeso(Double.parseDouble(valorEncontrado));
        }
    }
}

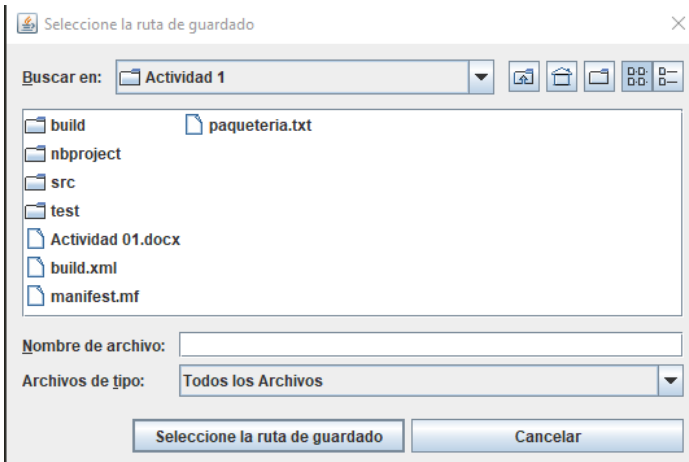
```

```

    };
}
}

```

Captura de selección del archivo anterior (posterior a eliminación)



Captura de los paquetes recuperados

```

=====
Desplegando paquetes
=====
-
    {
    id=1100231,
    origen=Alemania ,
    destino=Guatemala,
    peso=1.0
    }

```

Conclusiones

Los problemas enfrentados fueron al almacenar y recuperar los datos de los archivos, sobre todo al recuperar los datos, pues se debía discriminar los caracteres en el archivo para obtener los datos específicamente o para generar un nuevo paquete en base a los caracteres leídos. Útil para repasar bases para lectura de documentos.

Código

Código clase “**paquete**”

```

package models;

import java.io.Serializable;

/**
 *
 * @author Armando
 */
public class Paquete {

    private String id, origen, destino;
    private double peso;

    public String getId() {

```

```

        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getOrigen() {
        return origen;
    }

    public void setOrigen(String origen) {
        this.origen = origen;
    }

    public String getDestino() {
        return destino;
    }

    public void setDestino(String destino) {
        this.destino = destino;
    }

    public double getPeso() {
        return peso;
    }

    public void setPeso(double peso) {
        this.peso = peso;
    }

    @Override
    public String toString() {
        return "{\nid=" + id + ",\n origen=" + origen + ", \ndestino=" +
destino + ",\n peso=" + peso + "\n}";
    }
}

```

Código clase “Paqueteria”

```

package controlers;

import java.io.File;
import java.io.FileInputStream;

```



```

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.Serializable;
import java.util.ArrayList;
import models.Paquete;

/**
 *
 * @author Armando
 */
public class Paqueteria implements Serializable{
    private final long serialVersionUID = 1L;
    private ArrayList<Paquete> paquetes = new ArrayList();

    public void insertarInicio( Paquete p){
        paquetes.add(0,p);
    }
    public void eliminarInicio( ){
        if ( paquetes.isEmpty() ){
            System.out.println("No hay paquetes para eliminar");
            return ;
        }
        paquetes.remove(0);
    }

    public void mostrar(){
        System.out.println(this.toString());
    }

    public void guardar( String ruta ) throws FileNotFoundException,
IOException{

        try (FileOutputStream rutaSalida = new FileOutputStream(
            ruta +"/paqueteria.txt"
        )) {
            rutaSalida.write( this.toString().getBytes() );
        }

    }

    public void recuperar( File archivo) throws IOException,
ClassNotFoundException{
        try (FileInputStream rutaSalida = new FileInputStream(
            archivo

```

```

)) {
    archivo.createNewFile();
    String cadenaFinal = "";
    int caracterPrevio;
    do{
        caracterPrevio = rutaSalida.read();
        if( caracterPrevio > 0){
            char caracter = (char)caracterPrevio;
            cadenaFinal += caracter;
        }
    } while( caracterPrevio > 0 );
    //Limpiamos todos los paquetes encontrados
    paquetes.removeAll(paquetes);
    Paquete paquete = null;
    for( int i = 0 ; i < cadenaFinal.length(); i++){
        //Saber si termino un nuevo objeto para no asignar nada
        if( cadenaFinal.charAt( i ) == '}' ) {
            //Agregamos el paquete a la lista
            paquetes.add(paquete);
            paquete = null;
        }
        //Saber si empieza un nuevo objeto
        if( paquete == null ){
            //Se inicio un nuevo objeto
            if( cadenaFinal.charAt( i ) == '{' ) {
                paquete = new Paquete();
                i++;
            }
        } else if( paquete != null){
            String identificador = "";
            String valorEncontrado = "";
            //Intentamos obtener los parametros
            //Hasta que encontremos un "=" detenemos la obtencion
            del identificador
            while( cadenaFinal.charAt( i ) != '='){
                identificador += cadenaFinal.charAt( i ) == ' '?
                    "":cadenaFinal.charAt( i );
                i++;
            }
            //Si llegamos al limite de la cadena cortamos el
            proceso
            if( i >= cadenaFinal.length() ){
                break;
            }
        }
        i++;
    }
}

```

```

        //Intentamos obtener el valor
        while( cadenaFinal.charAt( i ) != '\n' ){
            valorEncontrado += cadenaFinal.charAt( i ) == ','?
                                "":cadenaFinal.charAt( i );

            i++;
            //Si llegamos al limite de la cadena cortamos el
proceso

            if( i >= cadenaFinal.length() ){
                break;
            }
        }
        //Asignación de las valores por identificador
        if( "id".equals(identificador)){
            paquete.setId(valorEncontrado);
        }
        if( "origen".equals(identificador)){
            paquete.setOrigen(valorEncontrado);
        }
        if( "destino".equals(identificador)){
            paquete.setDestino(valorEncontrado);
        }
        if( "peso".equals(identificador)){
            paquete.setPeso(Double.parseDouble(valorEncontrado))
;
        }

    }

};

}

@Override
public String toString() {
    String paquetesString = "";
    for (Paquete paquete : paquetes) {
        paquetesString += "\n\t" + paquete.toString() + "\n";
    }
    return paquetesString;
}
}

```

Código clase “main”

```

import controlers.Paqueteria;
import java.io.File;
import java.io.IOException;
import java.util.Scanner;
import javax.swing.JFileChooser;
import models.Paquete;

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author Armando
 */
public class main {

    private static final byte MENU_PRINCIPAL = 0;
    private static final byte MENU_AGREGAR = 1;
    private static final byte MENU_ELIMINAR = 2;
    private static final byte MENU_MOSTRAR = 3;
    private static final byte MENU_GUARDAR = 4;
    private static final byte MENU_RECUPERAR = 5;
    private static final byte SALIR = 6;

    private static String menu( byte opcion ){
        switch( opcion ){
            case MENU_PRINCIPAL:{
                return "=====\n" +
                    "Paquetería\n" +
                    "=====\n" +
                    "1) Agregar\n" +
                    "2) Eliminar primero\n" +
                    "3) Mostrar paquetes\n" +
                    "4) Guardar \n" +
                    "5) Recuperar (guardar al inicio de la lista)\n-";
            }
            case MENU_AGREGAR:{
                return "=====\n" +
                    "Agregar paquete\n" +
                    "=====\n-";
            }
        }
    }
}

```

```

        case MENU_ELIMINAR:{
            return "=====\n" +
                "Eliminar primer paquete\n" +
                "=====\n-";
        }
        case MENU_MOSTRAR:{
            return "=====\n" +
                "Desplegando paquetes\n" +
                "=====\n-";
        }
        case MENU_GUARDAR:{
            return "=====\n" +
                "Recuperar paquetes\n" +
                "=====\n"+
                "Escriba la ruta para guardar\n-";
        }
        case MENU_RECUPERAR:{
            return "=====\n" +
                "Recuperar paquetes\n" +
                "=====\n"+
                "Escriba la ruta para recuperar\n-";
        }
        default :{
            return "";
        }
    }
}

```

```

    public static void main(String[] args) throws IOException,
ClassNotFoundException {
        //Opcion que seleccionará el usuario en el menú
        byte opcion ;
        //Escaner para leer las salidas en consola
        Scanner scanner ;
        //Paquete auxiliar para cada nuevo registro de paquete
        Paquete paquete ;
        //Nueva lista para gestionar los paquetes
        Paqueteria listaPaquetes = new Paqueteria();
        do{
            scanner = new Scanner(System.in);
            opcion = MENU_PRINCIPAL;
            //Imprimimos el menu segun el parametro recibido
            System.out.println(menu(opcion));
            opcion = scanner.nextByte();
            System.out.println(menu(opcion));

```

```

switch( opcion ){
    case MENU_AGREGAR:{
        paquete = new Paquete();
        System.out.println("Ingrese el ID del paquete");
        String a = scanner.nextLine();
        paquete.setId(scanner.nextLine());
        System.out.println("Ingrese el origen del paquete");
        paquete.setOrigen(scanner.nextLine());
        System.out.println("Ingrese el destino del paquete");
        paquete.setDestino(scanner.nextLine());
        System.out.println("Ingrese el peso del paquete");
        paquete.setPeso(scanner.nextDouble());
        //Finalmente guardamos el paquete al inicio de la lista
        listaPaquetes.insertarInicio(paquete);
        System.out.println("Paquete generado");
        break;
    }
    case MENU_ELIMINAR:{
        listaPaquetes.eliminarInicio();
        break;
    }
    case MENU_MOSTRAR:{
        listaPaquetes.mostrar();
        break;
    }
    case MENU_GUARDAR:{
        System.out.println("Seleccionando ruta de guardado");
        JFileChooser f = new JFileChooser();
        f.setDialogType(JFileChooser.OPEN_DIALOG);
        f.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
        f.showDialog(null, "Seleccione la ruta de guardado");
        String ruta = f.getSelectedFile().getAbsolutePath();
        // System.out.println("Nombre del archivo");
        // scanner.next();
        // String name = scanner.nextLine();
        listaPaquetes.guardar(ruta);
        System.out.println("Paquetes guardados");
        break;
    }
    case MENU_RECUPERAR:{
        System.out.println("Seleccionando ruta de
recuperación");
        JFileChooser f = new JFileChooser();
        f.setDialogType(JFileChooser.OPEN_DIALOG);
        f.setFileSelectionMode(JFileChooser.FILES_ONLY);

```

```
        f.showDialog(null, "Seleccione la ruta de guardado");
        String ruta = f.getSelectedFile().getAbsolutePath();
        listaPaquetes.recuperar(new File (ruta));
        System.out.println("Paquetes recuperado");
        break;
    }
    default :{
        break;
    }
}
}while( opcion != SALIR );
}
```