

Nombre: Andrés Alejandro Cárdenas de la Cruz		Matricula: 2896424
Nombre del curso: Computación En Java	Nombre del profesor: Manuel Cruz Serrano	
Tema 12. Entrada y Salida de datos 2	Actividad 12. Manejo de archivos	
Fecha: 16/11/2020		

Solución Implementada

```
package actividad;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;

public class Gui extends JFrame implements ActionListener{
    //Elementos Graficos
    private JButton agenda, crear, eliminar;
    private JTextField valor1, valor2;
    public TextArea result;
    AddressBook addressBook = new AddressBook();
    ManejoAgenda manejoAgenda = new ManejoAgenda();
    public static void main(String[] args) {
        //Se establece los valores predeterminados de la ventana
        Gui Frame = new Gui();
        Frame.setSize(500,330);
        Frame.setResizable(false);
        Frame.setTitle("Area y Perimetro");
        Frame.interfaz();
        Frame.setVisible(true);
    }
    public void interfaz(){
        //Se indica cuando se cierra la aplicacion
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        //establece un panel a la ventana
        Container window = getContentPane();
        //establece el acomodo o layout de la ventana
        window.setLayout(new FlowLayout());
        //Establece los valores para los elementos graficos
        JLabel text = new JLabel("Numero");
        JLabel text2 = new JLabel("Nombre");
        JLabel aux = new JLabel("");
        result = new TextArea();
        valor1 = new JTextField(20);
        valor2 = new JTextField(20);
        agenda = new JButton("Agenda");
        crear = new JButton("Crear");
        eliminar = new JButton("Eliminar");
        agenda.addActionListener(this);
        crear.addActionListener(this);
        eliminar.addActionListener(this);
        //agrega los elementos al panel
        window.add(text);
        window.add(text2);
        window.add(valor1);
        window.add(valor2);
        window.add(agenda);
        window.add(crear);
        window.add(eliminar);
        window.add(aux);
        window.add(result);
        //Establece un tamaño a los elementos graficos indicados
        text.setPreferredSize(new Dimension(240,20));
    }
}
```

```

        agenda.setPreferredSize(new
Dimension(150,20));
        crear.setPreferredSize(new Dimension(150,20));
        eliminar.setPreferredSize(new Dimension(150,20));
        aux.setPreferredSize(new Dimension(500,0));
        load();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        //accion del boton agenda
        if (e.getSource().equals(agenda)) {
            load();
            result.setText(manejoAgenda.list(addressBook.directorio));
        }
        //accion del boton crear
        if (e.getSource().equals(crear)){
            if (!valor1.getText().isEmpty() && !valor2.getText().isEmpty()) {
                try {
                    result.setText(manejoAgenda.create(valor1.getText(),valor2.getText()));
                } catch (IOException ioException) {
                    JOptionPane.showMessageDialog(null,"No se pudo crear el
contacto");
                }
            }
        }
        //accion del boton eliminar
        if (e.getSource().equals(eliminar)){
            if (!valor1.getText().isEmpty()) {
                try {
                    result.setText(manejoAgenda.delete(valor1.getText(),
addressBook.directorio));
                } catch (IOException ioException) {
                    JOptionPane.showMessageDialog(null,"No se pudo eliminar el
contacto"+e.toString());
                }
            }
        }
    }

    //metodo para
    public void load(){
        try {
            result.setText(addressBook.load());
        } catch (IOException ioException) {
            result.setText(ioException.toString());
        }
    }
}

```

```

package actividad;

import javax.swing.*;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collection;

```

```
import java.util.Set;
import java.util.HashMap;

public class AddressBook {
    //hash de la agenda
    public HashMap<String,String> directorio = new HashMap<>();
    public String load() throws IOException {
        //variable de las llaves y valores
        StringBuilder key = new StringBuilder();
        StringBuilder value = new StringBuilder();
        boolean llenado = true;
        //crea el txt si no esta
        int aux=1;
        String result = "Se accedio correctamente a la agenda";
        File file = new File("Contactos.txt");
        if (!file.exists()){
            file.createNewFile();
            result = "Se creo una nueva agenda";
        }
        //leer el txt y pasa el contenido a la hashmap
        FileReader leer = null;
        try {
            leer = new FileReader("Contactos.txt");
            while (aux!=-1){
                aux = leer.read();
                if (aux!=-1) {
                    if ((char)aux=='\n'){
                        if (llenado) {
                            llenado = false;
                        }
                    }
                    else {
                        directorio.put(key.toString(), value.toString());
                        key.setLength(0);
                        value.setLength(0);
                        llenado = true;
                    }
                }
                else {
                    if (llenado)
                        key.append(((char) aux));
                    else
                        value.append(((char) aux));
                }
            }
            return result;
        } catch (IOException e) {
            return "No se pudo acceder a laa agenda";
        } finally {
            assert leer != null;
            leer.close();
        }
    }

    public void save(String llave,String valor,boolean isnew) throws IOException
    {
        File file = new File("Contactos.txt");
        FileWriter fileWriter = null;
        //ingresa el valor y la llave al txt
        if (isnew){
```

```

        try {
            fileWriter = new FileWriter (file.getAbsolutePath(),true);
            for (int cont = 0; cont<llave.length();cont++){
                fileWriter.append(llave.charAt(cont));
                if (cont==llave.length()-1)
                    fileWriter.append("\n");
            }
            for (int cont = 0; cont<valor.length();cont++){
                fileWriter.append(valor.charAt(cont));
                if (cont==valor.length()-1)
                    fileWriter.append("\n");
            }
        } catch (IOException e) {
            JOptionPane.showMessageDialog(null,"No se pudo crear el
contacto");
        }finally {
            assert fileWriter != null;
            fileWriter.close();
        }
    }
    else {
        //elimina el txt
        try {
            valor = directorio.get(llave);
            fileWriter = new FileWriter (file.getAbsolutePath(),false);
            fileWriter.write("");
            Set <String> keyset = directorio.keySet();
            Collection<String> values = directorio.values();
            ArrayList<String> llaves = new ArrayList<String>(keyset);
            ArrayList<String> valores = new ArrayList<String>(values);

            for (int cont = 0;cont<llaves.size();cont++){
                fileWriter = new FileWriter (file.getAbsolutePath(),true);
                if (!llaves.get(cont).equals(llave)) {
                    for (int cont2 = 0; cont2 < llaves.get(cont).length();
cont2++) {
                        fileWriter.append(llaves.get(cont).charAt(cont2));
                        if (cont2 == llaves.get(cont).length() - 1)
                            fileWriter.append("\n");
                    }
                }
                if (!valores.get(cont).equals(valor)) {
                    for (int cont2 = 0; cont2 < valores.get(cont).length();
cont2++) {
                        fileWriter.append(valores.get(cont).charAt(cont2));
                        if (cont2 == valores.get(cont).length() - 1)
                            fileWriter.append("\n");
                    }
                }
            }
        } catch (IOException e) {
            JOptionPane.showMessageDialog(null,"No se pudo crear el
contacto");
        }finally {
            assert fileWriter != null;
            fileWriter.close();
        }
    }
}

```

```

package actividad;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class ManejoAgenda {
    //metodo para mostrar la agenda
    public String list(HashMap<String,String> hashMap) {
        StringBuilder result = new StringBuilder();
        for (Map.Entry<String, String> entry : hashMap.entrySet()) {
            String llave = entry.getKey();
            String valor = entry.getValue();
            result.append("Numero: ").append(llave).append("    Nombre: ")
                .append(valor).append("\n");
        }
        return result.toString();
    }
    //metodo para crear contacto
    public String create(String numero,String nombre) throws IOException {
        AddressBook addressBook = new AddressBook();
        addressBook.save(numero,nombre,true);
        return "Se creo correctamente el contacto "+nombre;
    }
    //metodo para borrar contacto
    public String delete(String contacto,HashMap<String,String> hashMap) throws
        IOException {
        AddressBook addressBook = new AddressBook();
        addressBook.save(contacto,null,false);
        return "Se elimino correctamente el contacto "+contacto;
    }
}

```

