

**UNIVERSIDAD NACIONAL DE COLOMBIA – SEDE MEDELLÍN**

**Título**

Entrega Actividad # 6 – Programación Orientada a Objetos

**Estudiantes**

Jeffrey Santiago Navarro Espinosa ([jnavarro@unal.edu.co](mailto:jnavarro@unal.edu.co))

**Profesor Encargado**

Walter Hugo Arboleda Mazo ([walter.arboleda@iudigital.edu.co](mailto:walter.arboleda@iudigital.edu.co)) ([ia.walterarboleda@gmail.com](mailto:ia.walterarboleda@gmail.com))

**Grupo 3**

**Repositorio**

[Y3FF0/Actividad-6-POO-2023-1 \(github.com\)](#)

**Fecha de Entrega**

Jueves 29 de junio del 2023

Medellín, Antioquia, Colombia

### ***Clase Actividad\_6.java***

```
package actividad_6;

/**
 *
 * @author yeffer
 */
public class Actividad_6 {

    public static void main(String[] args) {
        // TODO code application logic here

        Interfaz_grafica formulario= new Interfaz_grafica();
        formulario.setVisible(true);
    }

}
```

### ***Clase Interfaz\_grafica.java***

```
package actividad_6;

import java.io.File;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.lang.NumberFormatException;
/**
 *
 * @author yeffo
 */
public class Interfaz_grafica extends
javax.swing.JFrame {

    /**
     * Creates new form Interfaz_grafica
     */
    public Interfaz_grafica() {
        initComponents();
    }

    /**
     * This method is called from within the
    constructor to initialize the form.
     * WARNING: Do NOT modify this code. The
    content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed"
    desc="Generated Code">
    private void initComponents() {

        Nombre = new javax.swing.JLabel();
        Numero = new javax.swing.JLabel();
        Nombrel = new javax.swing.JTextField();
        Numerol = new javax.swing.JTextField();
```

```

        agregar = new javax.swing.JButton();
        actualizar = new javax.swing.JButton();
        eliminar = new javax.swing.JButton();
        mostrar = new javax.swing.JButton();
        jScrollPane1 = new
javax.swing.JScrollPane();
        resultados = new javax.swing.JTextArea();

setDefaultCloseOperation(javax.swing.WindowConstant
s.EXIT_ON_CLOSE);

        Nombre.setFont(new java.awt.Font("Segoe
UI", 1, 12)); // NOI18N
        Nombre.setText("Nombre:");

        Numero.setFont(new java.awt.Font("Segoe
UI", 1, 12)); // NOI18N
        Numero.setText("Numero:");

        Nombre1.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                Nombre1ActionPerformed(evt);
            }
        });

        agregar.setFont(new java.awt.Font("Segoe
UI", 1, 12)); // NOI18N
        agregar.setForeground(new java.awt.Color(0,
0, 255));
        agregar.setText("AGREGAR");
        agregar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                agregarActionPerformed(evt);
            }
        });

        actualizar.setFont(new java.awt.Font("Segoe
UI", 1, 12)); // NOI18N

```

```

        actualizar.setForeground(new
java.awt.Color(0, 153, 0));
        actualizar.setText("ACTUALIZAR");
        actualizar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                actualizarActionPerformed(evt);
            }
        });

        eliminar.setFont(new java.awt.Font("Segoe
UI", 1, 12)); // NOI18N
        eliminar.setForeground(new
java.awt.Color(204, 0, 0));
        eliminar.setText("ELIMINAR");
        eliminar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                eliminarActionPerformed(evt);
            }
        });

        mostrar.setFont(new java.awt.Font("Segoe
UI", 1, 12)); // NOI18N
        mostrar.setForeground(new
java.awt.Color(102, 0, 102));
        mostrar.setText("MOSTRAR");
        mostrar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                mostrarActionPerformed(evt);
            }
        });

        resultados.setColumns(20);
        resultados.setFont(new
java.awt.Font("Arial", 1, 12)); // NOI18N
        resultados.setRows(5);
        jScrollPane1.setViewportViewView(resultados);

```

```

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
        .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.TRAILING)
        .addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()
        .addComponent(Nombre)

.addPreferredGap(javax.swing.LayoutStyle.ComponentP
lacement.RELATED)
        .addComponent(Nombre1,
javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(Número)

.addPreferredGap(javax.swing.LayoutStyle.ComponentP
lacement.RELATED)
        .addComponent(Número1,
javax.swing.GroupLayout.PREFERRED_SIZE, 119,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(55, 55, 55)
        .addComponent(agregar)
        .addGap(18, 18, 18)
        .addComponent(actualizar)
        .addGap(18, 18, 18)
        .addComponent(mostrar)

.addPreferredGap(javax.swing.LayoutStyle.ComponentP
lacement.RELATED, 19, Short.MAX_VALUE)
        .addComponent(eliminar)))
        .addContainerGap())

```

```

        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE)
            .addComponent(Nombre)
            .addComponent(Numero)
            .addComponent(Nombre1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(Numero1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(agregar)
            .addComponent(actualizar)
            .addComponent(eliminar)
            .addComponent(mostrar))
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 207,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(21,
Short.MAX_VALUE))
        );

        pack();
    } // </editor-fold>

    public String StrToHtml(String texto){

        return "<html><p>" + texto + "</p></html>" ;
    }

```

```

    }
    private void
agregarActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:.
    try {

        // Get the name of the contact to be
updated
        // from the Command line argument
        String newName =
String.valueOf(Nombre1.getText());

        // Get the number to be updated
        // from the Command line argument
        long newNumber =
Long.parseLong(Numero1.getText());

        String nameNumberString;
        String name;
        long number;
        int index;

        // Using file pointer creating the
file.
        File file = new File("prueba.txt");

        if (!file.exists()) {

            // Create a new file if not exists.
            file.createNewFile();
        }

        // Opening file in reading and write
mode.

        RandomAccessFile raf
            = new RandomAccessFile(file, "rw");
        boolean found = false;

        // Checking whether the name
        // of contact already exists.

```



```

// getFilePointer() give the current
offset
// value from start of the file.
while (raf.getFilePointer() <
raf.length()) {

    // reading line from the file.
    nameNumberString = raf.readLine();

    // splitting the string to get name
and
    // number
    String[] lineSplit
        = nameNumberString.split("!");

    // separating name and number.
    name = lineSplit[0];
    number =
Long.parseLong(lineSplit[1]);

    // if condition to find existence
of record.
    if (name == newName
        || number == newNumber) {
        found = true;
        break;
    }
}

if (found == false) {

    // Enter the if block when a record
    // is not already present in the
file.
    nameNumberString
        = newName + "!"
        + String.valueOf(newNumber);

    // writeBytes function to write a
string
    // as a sequence of bytes.
    raf.writeBytes(nameNumberString);

```

```

        // To insert the next record in new
line.

raf.writeBytes(System.lineSeparator());

        // Print the message
        System.out.println(" Amigo
agregado. ");

        // Closing the resources.
        raf.close();
    }
    // The contact to be updated
    // could not be found
    else {

        // Closing the resources.
        raf.close();

        // Print the message
        System.out.println(" El contacto ya
existe ");
    }
}

catch (IOException ioe) {

    System.out.println(ioe);
}
catch (NumberFormatException nef) {

    System.out.println(nef);
}

// Java program to read from file
"friendsContact.txt"
// and display the contacts
try {

    String nameNumberString;

```

```

String name;
long number;
int index;

                                String acumulador;
                                acumulador="";

// Using file pointer creating the
file.
File file = new File("prueba.txt");

if (!file.exists()) {

    // Create a new file if not exists.
    file.createNewFile();
}

// Opening file in reading and write
mode.

RandomAccessFile raf
    = new RandomAccessFile(file, "rw");
boolean found = false;

// Traversing the file
// getFilePointer() give the current
offset
// value from start of the file.
while (raf.getFilePointer() <
raf.length()) {

    // reading line from the file.
    nameNumberString = raf.readLine();

    // splitting the string to get name
and
    // number
    String[] lineSplit
        = nameNumberString.split("!");

    // separating name and number.
    name = lineSplit[0];

```

```

        number =
Long.parseLong(lineSplit[1]);

        // Print the contact data
        acumulador=acumulador

+"\\n"+"Nombre de amigo: "
                                + name +
"\\n"
                                + "Numero

de amigo: " + number + "\\n";

resultados.setText(acumulador);
    }
    }
    catch (IOException ioe)
    {

        System.out.println(ioe);
    }
    catch (NumberFormatException nef)
    {

        System.out.println(nef);
    }

}

    private void
NombreActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
    }

    private void
actualizarActionPerformed(java.awt.event.ActionEven
t evt) {
// Java program to update in the file
"friendsContact.txt"

```

```

// and change the number of an old contact

    try {

        // Get the name of the contact to be
updated
        // from the Command line argument
        String newName =
String.valueOf(Nombre1.getText());

                                long newNumber =
Long.parseLong(Numero1.getText());

        String nameNumberString;
        String name;
        long number;
        int index;

        // Using file pointer creating the
file.
        File file = new File("prueba.txt");

        if (!file.exists()) {

            // Create a new file if not exists.
            file.createNewFile();
        }

        // Opening file in reading and write
mode.
        RandomAccessFile raf
            = new RandomAccessFile(file, "rw");
        boolean found = false;

        // Checking whether the name
        // of contact already exists.
        // getFilePointer() give the current
offset
        // value from start of the file.
        while (raf.getFilePointer() <
raf.length()) {

```

```

        // reading line from the file.
        nameNumberString = raf.readLine();

        // splitting the string to get name
and
        // number
        String[] lineSplit =
nameNumberString.split("!");

        // separating name and number.
        name = lineSplit[0];
        System.out.println(name);
        number =
Long.parseLong(lineSplit[1]);

        // if condition to find existence
of record.
        if (name == null ? newName == null
: name.equals(newName)) {
            found = true;
            break;
        }

        // Update the contact if record exists.
        if (found == true) {

            // Creating a temporary file
            // with file pointer as tmpFile.
            File tmpFile = new
File("temp.txt");

            // Opening this temporary file
            // in ReadWrite Mode
            RandomAccessFile tmpraf
            = new RandomAccessFile(tmpFile,
"rw");

            // Set file pointer to start
            raf.seek(0);

            // Traversing the

```

```

friendsContact.txt file
        while (raf.getFilePointer()
                < raf.length()) {

            // Reading the contact from the
file
            nameNumberString =
raf.readLine();

            index =
nameNumberString.indexOf('!');
            name =
nameNumberString.substring(
                0, index);

            // Check if the fetched contact
// is the one to be updated
            if (name == null ? newName ==
null : name.equals(newName)) {

                // Update the number of
this contact
                nameNumberString
                    = name + "!"
                    +
String.valueOf(Numero1.getText());
            }

            // Add this contact in the
temporary
            // file

tmpraf.writeBytes(nameNumberString);

            // Add the line separator in
the
            // temporary file
            tmpraf.writeBytes(
                System.lineSeparator());
        }

        // The contact has been updated now
        // So copy the updated content from

```

```

file.                                // the temporary file to original

// Set both files pointers to start
raf.seek(0);
tmpraf.seek(0);

// Copy the contents from
// the temporary file to original
file.
while (tmpraf.getFilePointer()
      < tmpraf.length()) {

raf.writeBytes(tmpraf.readLine());

raf.writeBytes(System.lineSeparator());
}

// Set the length of the original
file
// to that of temporary.
raf.setLength(tmpraf.length());

// Closing the resources.
tmpraf.close();
raf.close();

// Deleting the temporary file
tmpFile.delete();

System.out.println(" Amigo
Actualizado. ");
}

// The contact to be updated
// could not be found
else {

// Closing the resources.
raf.close();

// Print the message

```



```

        System.out.println(" Input name"
                            + " does not
exists. ");
    }
}

catch (IOException ioe) {
    System.out.println(ioe);
}

catch (NumberFormatException nef) {
    System.out.println(nef);
}

// TODO add your handling code here:
}

private void
eliminarActionPerformed(java.awt.event.ActionEvent
evt) {
    // Java program to delete a contact
    // from the file "friendsContact.txt"

    try {

        // Get the name of the contact to be
updated
        // from the Command line argument
        String newName =
String.valueOf(Nombre1.getText());

        String nameNumberString;
        String name = null;
        long number;
        int index;

        // Using file pointer creating the

```

```

file.
    File file = new File("prueba.txt");

    if (!file.exists()) {

        // Create a new file if not exists.
        file.createNewFile();
    }

    // Opening file in reading and write
mode.
    RandomAccessFile raf
    = new RandomAccessFile(file, "rw");
    boolean found = false;

    // Checking whether the name of contact
exists.
    // getFilePointer() give the current
offset
    // value from start of the file.
    while (raf.getFilePointer() <
raf.length()) {

        // reading line from the file.
        nameNumberString = raf.readLine();

        // splitting the string to get name
and
        // number
        String[] lineSplit =
nameNumberString.split("!");

        // separating name and number.
        name = lineSplit[0];
        System.out.println(name);
        number =
Long.parseLong(lineSplit[1]);

        // if condition to find existence
of record.
        if (name == null ? newName == null
: name.equals(newName)) {

```

```

        found = true;
        break;
    }
}
System.out.println(raf.length());
System.out.println(name==newName);
// Delete the contact if record exists.
if (found == true) {

    // Creating a temporary file
    // with file pointer as tmpFile.
    File tmpFile = new
File("temp.txt");

    // Set file pointer to start
    try ( // Opening this temporary
file
        // in ReadWrite Mode
            RandomAccessFile tmpraf =
new RandomAccessFile(tmpFile, "rw")) {
        // Set file pointer to start
        raf.seek(0);
        // Traversing the
friendsContact.txt file
        while (raf.getFilePointer()
            < raf.length()) {

            // Reading the contact from
the file
            nameNumberString =
raf.readLine();

            index =
nameNumberString.indexOf('!');
            name =
nameNumberString.substring(
                0, index);

            // Check if the fetched
contact
            // is the one to be deleted
            if (name == null ? newName
== null : name.equals(newName)) {

```

```

// Skip inserting this
contact
// into the temporary
file
continue;
}

// Add this contact in the
temporary
// file

tmpraf.writeBytes(nameNumberString);

// Add the line separator
in the
// temporary file
tmpraf.writeBytes(

System.lineSeparator());
} // The contact has been
deleted now
// So copy the updated content
from
// the temporary file to
original file.

// Set both files pointers to
start
raf.seek(0);
tmpraf.seek(0);
// Copy the contents from
// the temporary file to
original file.
while (tmpraf.getFilePointer()
    < tmpraf.length()) {

raf.writeBytes(tmpraf.readLine());

raf.writeBytes(System.lineSeparator());
} // Set the length of the
original file
// to that of temporary.

```

```

        raf.setLength(tmpraf.length());
        // Closing the resources.
    }
    raf.close();

    // Deleting the temporary file
    tmpFile.delete();

    System.out.println("Contacto
Eliminado. ");
}

// The contact to be deleted
// could not be found
else {

    // Closing the resources.
    raf.close();

    // Print the message
    System.out.println(" No se encontro
el contacto. ");
}

}

catch (IOException ioe) {
    System.out.println(ioe);
}

// TODO add your handling code here:
}

private void
mostrarActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    try {

        String nameNumberString;
        String name;
        long number;
        int index;

```

```

        String acumulador;
        acumulador="";

        // Using file pointer creating the
file.
        File file = new File("prueba.txt");

        if (!file.exists()) {

            // Create a new file if not exists.
            file.createNewFile();
        }

        // Opening file in reading and write
mode.

        RandomAccessFile raf
            = new RandomAccessFile(file, "rw");
        boolean found = false;

        // Traversing the file
        // getFilePointer() give the current
offset
        // value from start of the file.
        while (raf.getFilePointer() <
raf.length()) {

            // reading line from the file.
            nameNumberString = raf.readLine();

            // splitting the string to get name
and
            // number
            String[] lineSplit
                = nameNumberString.split("!");

            // separating name and number.
            name = lineSplit[0];
            number =
Long.parseLong(lineSplit[1]);

```

```

        // Print the contact data
        acumulador=acumulador

        +"\n"+"Nombre de amigo: "
                                                    + name +
        "\n"
                                                    + "Numero
de amigo: " + number + "\n";

resultados.setText(acumulador);
    }
    }
    catch (IOException ioe)
    {

        System.out.println(ioe);
    }
    catch (NumberFormatException nef)
    {

        System.out.println(nef);
    }

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed"
desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is
not available, stay with the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/
lookandfeel/plaf.html
    */
    try {

```

```

        for
(javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if
("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassN
ame());

                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Interfaz_grafica
.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Interfaz_grafica
.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Interfaz_grafica
.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch
(javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Interfaz_grafica
.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
    } //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new
Runnable() {
        public void run() {
            new
Interfaz_grafica().setVisible(true);
        }
    });

```



```
}

// Variables declaration - do not modify
private javax.swing.JLabel Nombre;
private javax.swing.JTextField Nombre1;
private javax.swing.JLabel Numero;
private javax.swing.JTextField Numero1;
private javax.swing.JButton actualizar;
private javax.swing.JButton agregar;
private javax.swing.JButton eliminar;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton mostrar;
private javax.swing.JTextArea resultados;
// End of variables declaration
}
```