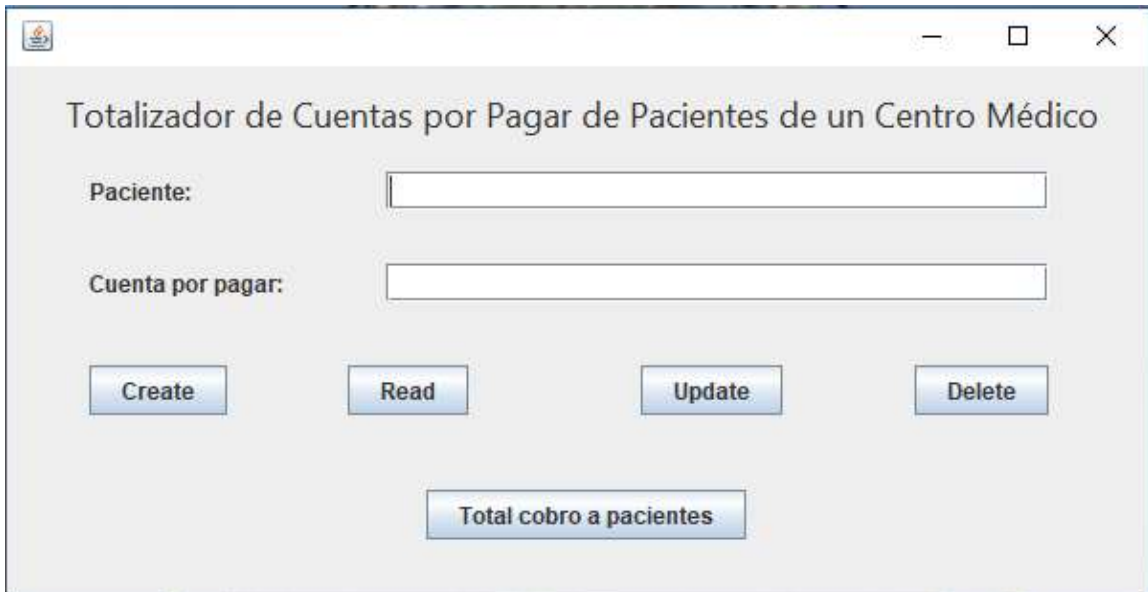


Propuesta desarrollada

Solución propuesta a partir de lo planteado para ésta actividad: Se desarrolla una aplicación sencilla para totalizar las cuentas por pagar de diferentes pacientes que pertenecen a un centro médico en general.



The screenshot shows a Windows application window with the title "Totalizador de Cuentas por Pagar de Pacientes de un Centro Médico". The window has a standard Windows title bar with minimize, maximize, and close buttons. Inside the window, there are two text input fields. The first is labeled "Paciente:" and the second is labeled "Cuenta por pagar:". Below these fields, there are four buttons arranged horizontally: "Create", "Read", "Update", and "Delete". At the bottom center of the window, there is a button labeled "Total cobro a pacientes".

Ejemplo con Create

Luis Rafael Diaz Anaya

71261313

Actividad 6. Programación Orientada a Objetos.


Enlace Github: <https://github.com/luisdico313/pooactividad6>

Totalizador de Cuentas por Pagar de Pacientes de un Centro Médico

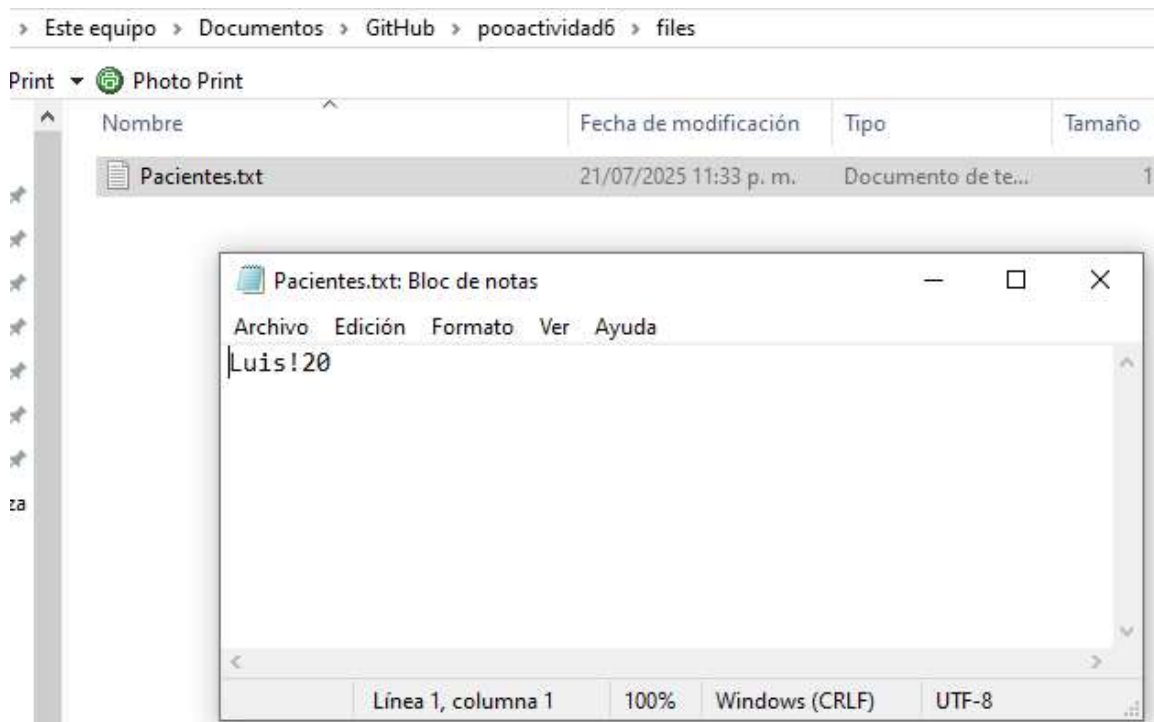
Paciente:

Cuenta por pagar:

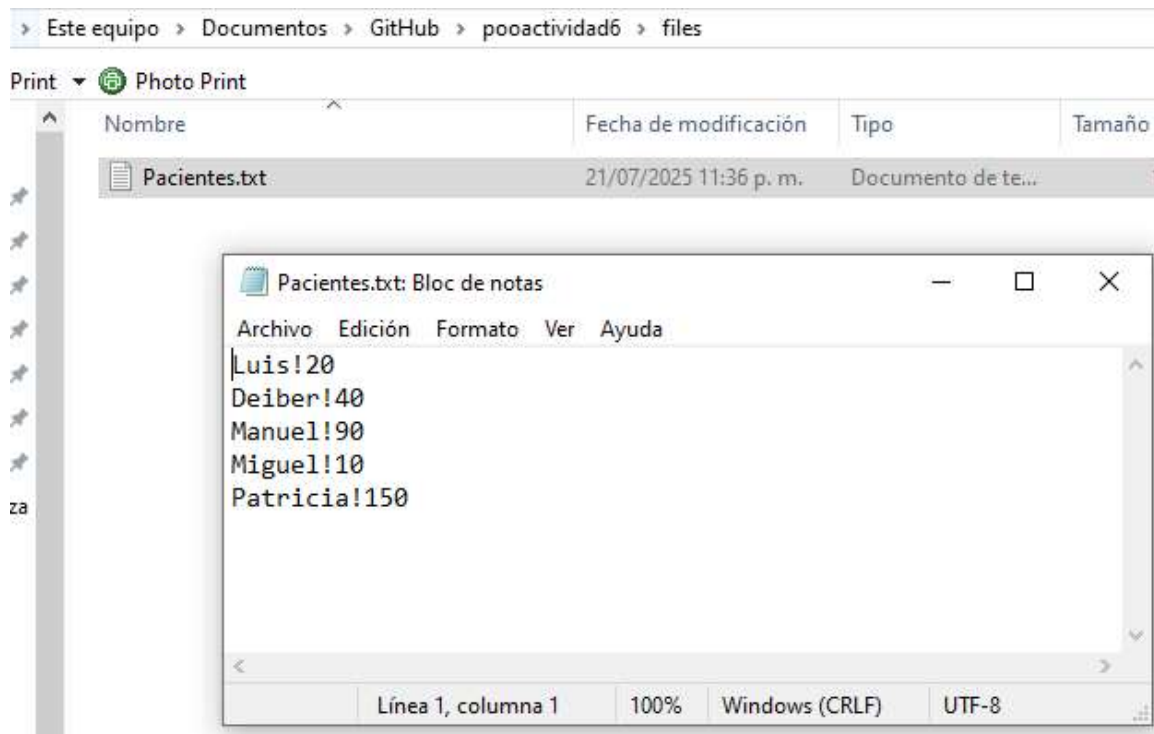
Message

 Paciente agregado.

En el archivo creado:

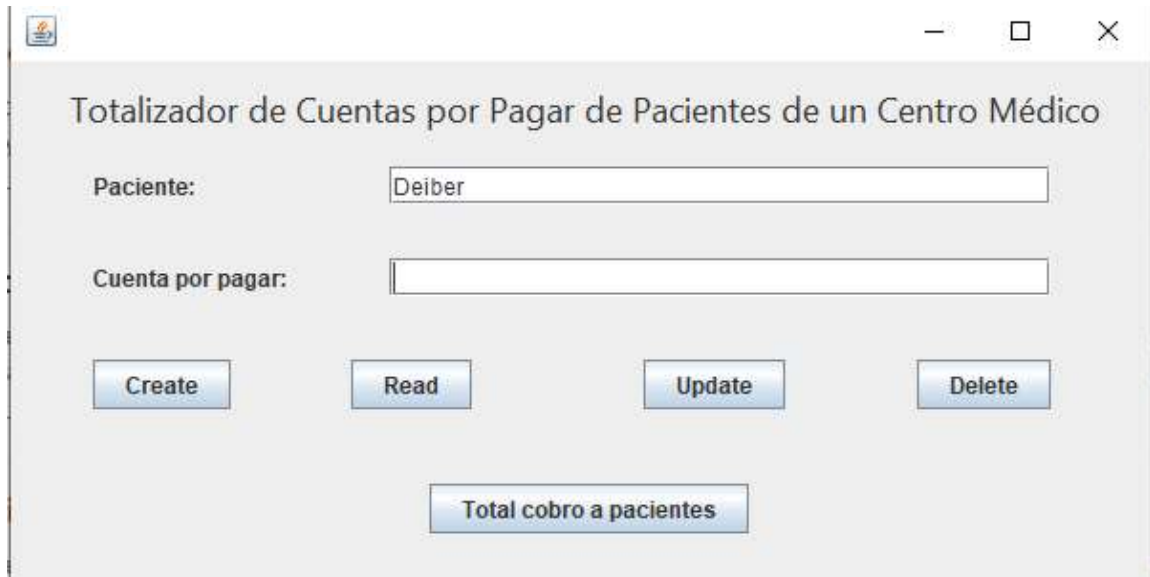


Se agregaron otros pacientes para quedar de la siguiente forma:



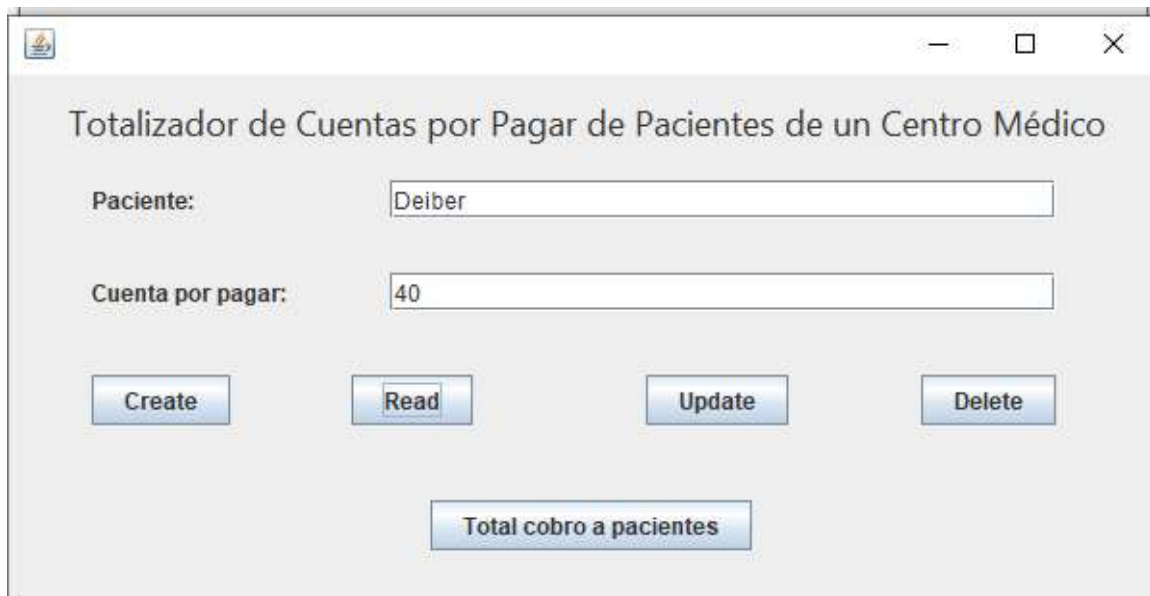
Esto es importante como insumo para comprender los resultados a partir del uso de los siguientes botones:

Botón Read



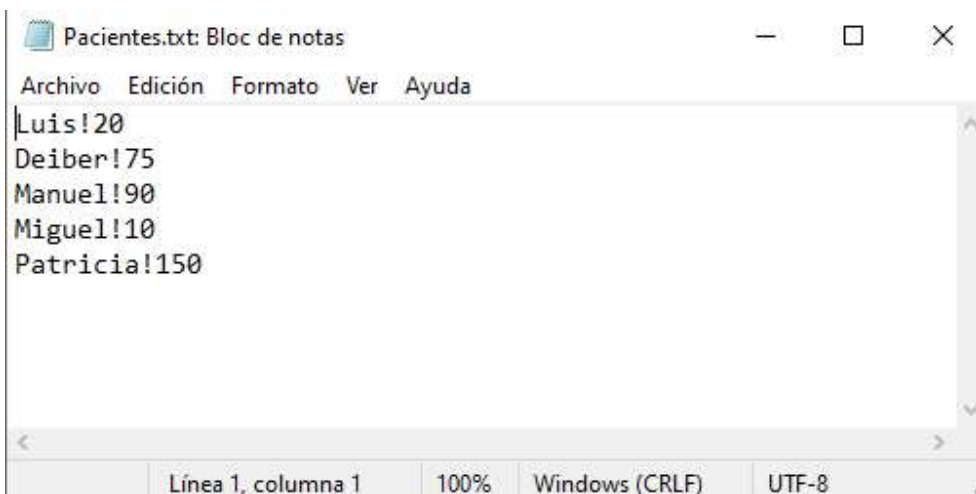
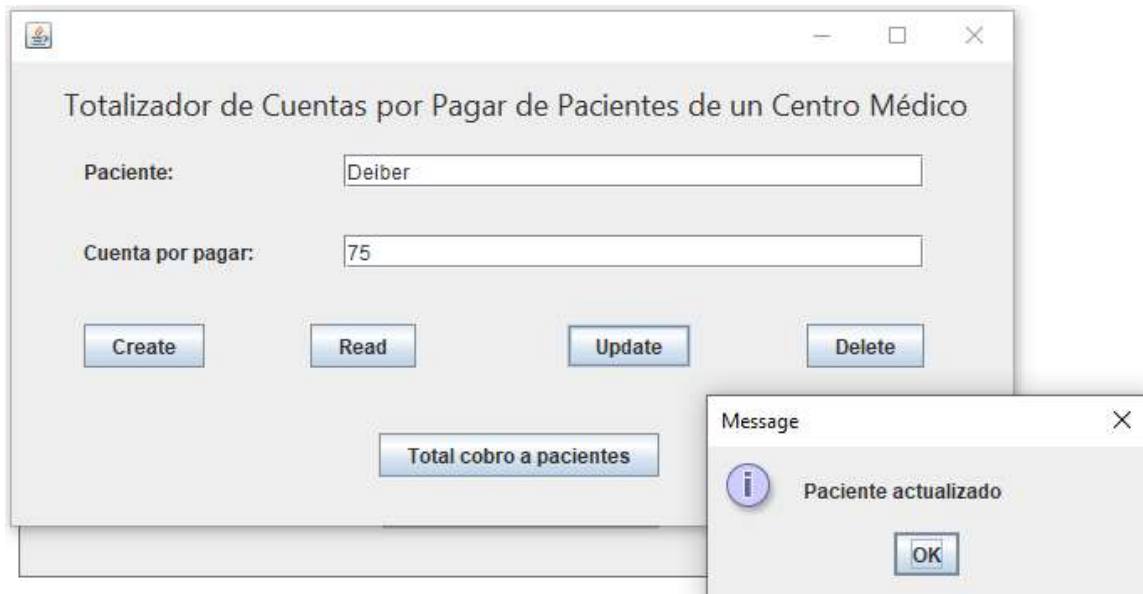
The screenshot shows a Java Swing window titled "Totalizador de Cuentas por Pagar de Pacientes de un Centro Médico". It contains two text input fields: "Paciente:" with the value "Deiber" and "Cuenta por pagar:" which is currently empty. Below the fields are four buttons: "Create", "Read", "Update", and "Delete". The "Read" button is highlighted with a blue border. At the bottom center is a button labeled "Total cobro a pacientes".

Después de usarse Read trae el valor actual de la Cuenta por Pagar



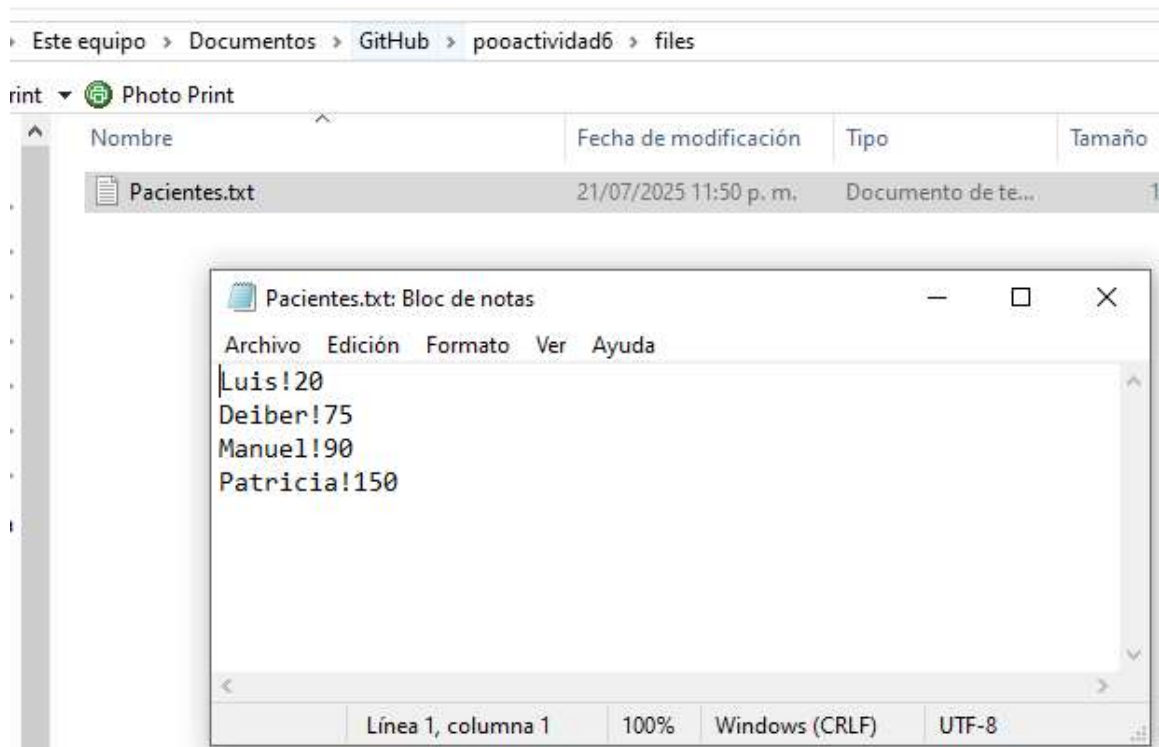
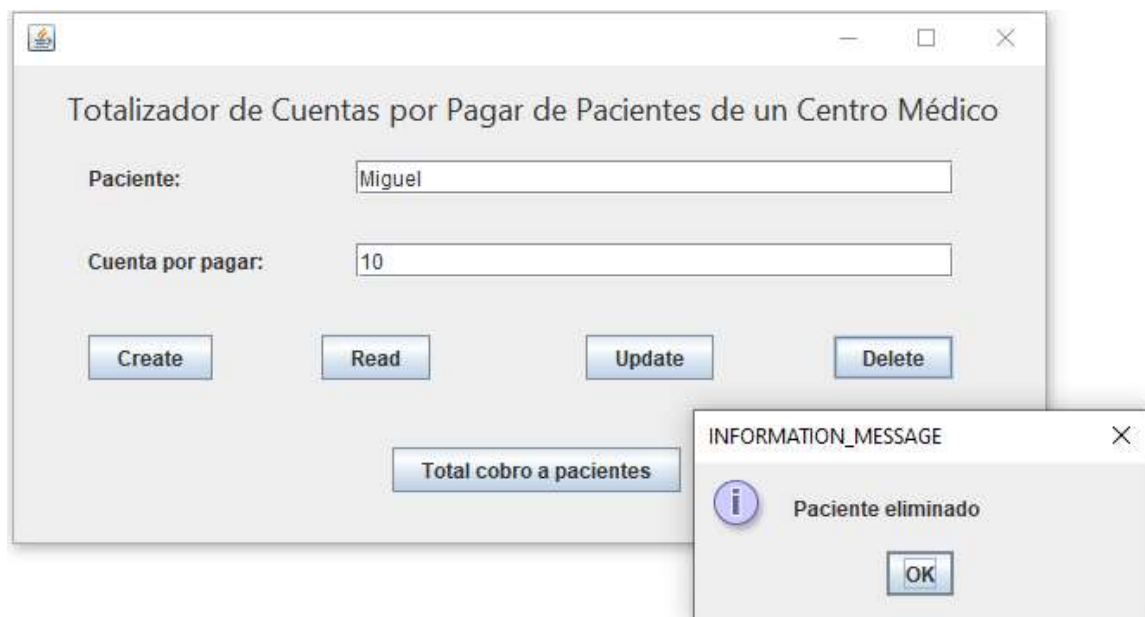
This screenshot shows the same application window after the "Read" button was clicked. The "Paciente" field still contains "Deiber". The "Cuenta por pagar:" field now displays the value "40". The "Read" button remains highlighted. The "Total cobro a pacientes" button is still at the bottom.

Ahora con Update

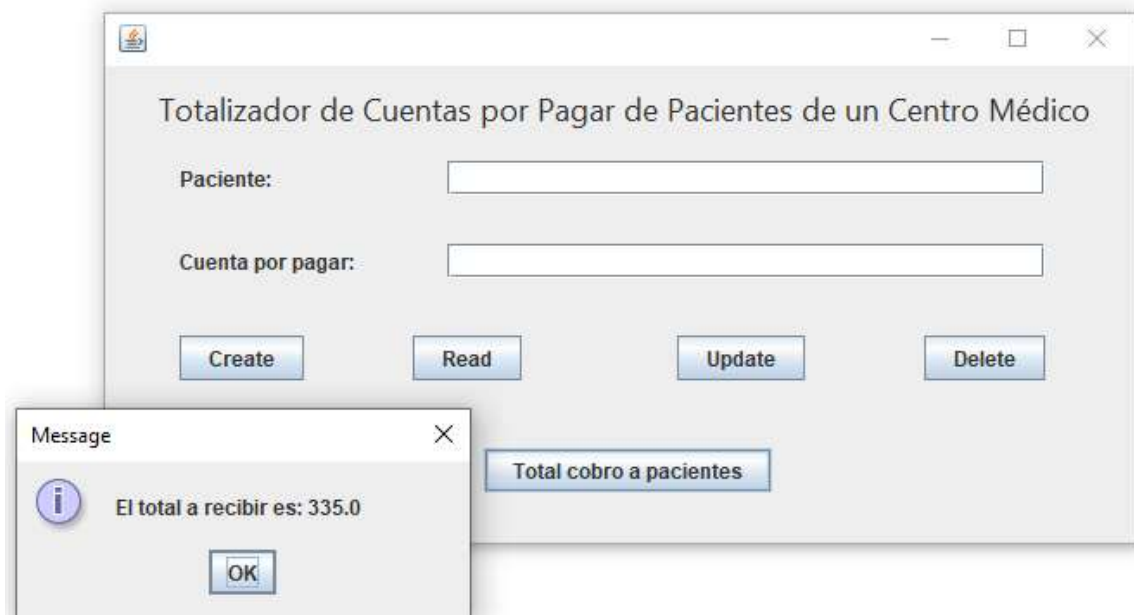


Efectivamente, en el archivo se observa que actualizó el valor de la cuenta por pagar

Ahora con el botón Delete



Y último lugar el botón que totaliza las cuentas por pagar, así:



## Código

Además de la interfaz creada desde NetBeans, se implementó el siguiente código en el archivo **Pacientes.java**:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * change this license
 */

package PaquetePacientes;

/**
 *
 * @author Usuario
 */
public class Pacientes {

    public static void main(String[] args) {
        Principal formulario = new Principal();
        formulario.setVisible(true);
    }
}
```

```
}  
}
```

Y en el archivo que administra la interfaz gráfica, osea **Principal.java**:

```
/*  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to  
change this license  
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit  
this template  
 */  
  
package PaquetePacientes;  
  
import java.io.File;  
import java.io.IOException;  
import java.io.RandomAccessFile;  
import javax.swing.JOptionPane;  
  
/**  
 *  
 * @author Usuario  
 */  
public class Principal extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Principal  
     */  
    public Principal() {  
        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() {  
  
    lbNombre = new javax.swing.JLabel();  
    lbNumero = new javax.swing.JLabel();  
    txtNombre = new javax.swing.JTextField();  
    txtNumero = new javax.swing.JTextField();  
    btnCreate = new javax.swing.JButton();  
    btnUpdate = new javax.swing.JButton();  
    btnRead = new javax.swing.JButton();  
    btnDelete = new javax.swing.JButton();  
    btnTotal = new javax.swing.JButton();  
    jLabel1 = new javax.swing.JLabel();  
  
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);  
  
    lbNombre.setText("Paciente:");  
  
    lbNumero.setText("Cuenta por pagar:");  
  
    txtNombre.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            txtNombreActionPerformed(evt);  
        }  
    });  
  
    txtNumero.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            txtNumeroActionPerformed(evt);  
        }  
    });  
  
    btnCreate.setText("Create");  
    btnCreate.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            btnCreateActionPerformed(evt);  
        }  
    });  
  
    btnUpdate.setText("Update");  
    btnUpdate.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            btnUpdateActionPerformed(evt);  
        }  
    });  
}
```

```
    }  
    });  
  
    btnRead.setText("Read");  
    btnRead.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            btnReadActionPerformed(evt);  
        }  
    });  
  
    btnDelete.setText("Delete");  
    btnDelete.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            btnDeleteActionPerformed(evt);  
        }  
    });  
  
    btnTotal.setText("Total cobro a pacientes");  
    btnTotal.addActionListener(new java.awt.event.ActionListener() {  
        public void actionPerformed(java.awt.event.ActionEvent evt) {  
            btnTotalActionPerformed(evt);  
        }  
    });  
  
    jLabel1.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N  
    jLabel1.setText("Totalizador de Cuentas por Pagar de Pacientes de un Centro  
Médico");  
  
    javax.swing.GroupLayout layout = new  
    javax.swing.GroupLayout(getContentPane());  
    getContentPane().setLayout(layout);  
    layout.setHorizontalGroup(  
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addGroup(layout.createSequentialGroup()  
                .addGap(43, 43, 43)  
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                    .addComponent(btnCreate)  
                    .addComponent(btnRead)  
                )  
            )  
    );
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnUpdate)
    .addGap(69, 69, 69)
    .addComponent(btnDelete))
    .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    G)
        .addComponent(lbNumero)
        .addComponent(lbNombre))
        .addGap(54, 54, 54)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    G)
        .addComponent(txtNombre)
        .addComponent(txtNumero))))
        .addGap(53, 53, 53))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addGroup(layout.createSequentialGroup()
        .addContainerGap(31, Short.MAX_VALUE)
        .addComponent(jLabel1)
        .addGap(27, 27, 27))
        .addGroup(layout.createSequentialGroup()
            .addGap(219, 219, 219)
            .addComponent(btnTotal)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createSequentialGroup()
                .addGap(12, 12, 12)
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(lbNombre)
```

```
.addComponent(txtNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(28, 28, 28)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(lbNumero)
    .addComponent(txtNumero, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnCreate)
    .addComponent(btnRead)
    .addComponent(btnUpdate)
    .addComponent(btnDelete))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 39,
Short.MAX_VALUE)
    .addComponent(btnTotal)
    .addGap(29, 29, 29))
);

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

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

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

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

    try{
        String newName = String.valueOf(txtNombre.getText());
```

```
        long newNumber = Long.parseLong(txtNumero.getText());

        String nameNumberString;
        String name;
        long number;

        File file = new
File("C:\\Users\\Usuario\\Documents\\Git\\Hub\\pooactividad6\\files\\Pacientes.txt"
);

        if(file.exists() == false){
            file.createNewFile();
        }

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

        // Si el paciente existe
        while (raf.getFilePointer() < raf.length()){
            nameNumberString = raf.readLine();

            String[] lineSplit = nameNumberString.split("!");

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

            if(name.equals(newName)||number==newNumber){
                found = true;
                break;
            }
        }
        // Termina si el paciente existe

        if (found == false){

            //Ingresar al bloque If cuando un registro no está presente en el archivo
            nameNumberString = newName + "!" + String.valueOf(newNumber);

            // Para escribir un String como una secuencia de bytes
            raf.writeBytes(nameNumberString);

            // Para insertar el siguiente registro en una nueva línea
```

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

// Imprimir mensaje
// System.out.println(" Paciente agregado. ");
JOptionPane.showMessageDialog(null, " Paciente agregado. ");
// Cerrando los recursos
raf.close();

}
// El contacto para ser actualizado podría no haber sido encontrado
else{
    //Cerrando los recursos
    raf.close();

    //System.out.println(" El paciente ya existe. ");
    JOptionPane.showMessageDialog(null, "El paciente ya existe");
}

}

catch (IOException ioe){
    //System.out.println(ioe);
    JOptionPane.showMessageDialog(null, "Error");
    txtNombre.setText("");
    txtNumero.setText("");

}

catch (NumberFormatException nef){
    //System.out.println(nef);
    JOptionPane.showMessageDialog(null, "Error");
    txtNombre.setText("");
    txtNumero.setText("");
}
}

private void btnReadActionPerformed(java.awt.event.ActionEvent evt) {
    try{

        String nameNumberString;
        String name;
```

```
String newName = String.valueOf(txtNombre.getText());

long number;
//int index;

File file = new
File("C:\\Users\\Usuario\\Documents\\GitHub\\pooactividad6\\files\\Pacientes.txt"
);

if(!file.exists()){
    //Crea el archivo si no existe
    file.createNewFile();
}
//Se abre el archivo en modo de Lectura y Escritura

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

while(raf.getFilePointer() < raf.length()){
    //leyendo línea desde el archivo
    nameNumberString = raf.readLine();

    String[] lineSplit = nameNumberString.split("!");

    //Se separan las columnas
    name = lineSplit[0];
    number = Long.parseLong(lineSplit[1]);

    if(name.equals(newName)){

        txtNombre.setText(String.valueOf(name));
        txtNumero.setText(String.valueOf(number));

        raf.close();
        found = true;

    }
}

if(found == false){
    raf.close();
    txtNombre.setText("");
}
```

```
        txtNumero.setText("");
        JOptionPane.showMessageDialog(null, "El paciente no existe");

    }

}

catch(IOException ioe){
    //System.out.println(ioe);
    JOptionPane.showMessageDialog(null, "Error");
}

catch(NumberFormatException nef){
    //System.out.println(nef);
    JOptionPane.showMessageDialog(null, "Error");
}
// TODO add your handling code here:
}

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

    try{
        String newName = txtNombre.getText();
        long newNumber = Long.parseLong(txtNumero.getText());

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

        File file = new
File("C:\\Users\\Usuario\\Documents\\Git\\Hub\\pooactividad6\\files\\Pacientes.txt"
);
        if(!file.exists()){
            file.createNewFile();
        }

        RandomAccessFile raf = new RandomAccessFile(file,"rw");
```



```
boolean found = false;

while (raf.getFilePointer() < raf.length()){
    nameNumberString = raf.readLine();
    String[] lineSplit = nameNumberString.split("!");
    name = lineSplit[0];
    number = Long.parseLong(lineSplit[1]);

    if(name.equals(newName) && number == newNumber){
        found = true;
        break;
    }
}

if(found == true){
    //Se utiliza un archivo temporal
    File tmpFile = new File("temp.txt");

    //Tal archivo se abre en modo lectura y escritura
    RandomAccessFile tmpraf = new RandomAccessFile(tmpFile,"rw");

    // Se inicia el apuntador para comenzar
    raf.seek(0);

    while(raf.getFilePointer()<raf.length()){
        nameNumberString = raf.readLine();

        String[] lineSplit2 = nameNumberString.split("!");

        name = lineSplit2[0];
        number = Long.parseLong(lineSplit2[1]);

        if(name.equals(newName)&&number==newNumber){
            continue;
        }

        tmpraf.writeBytes(nameNumberString);
        tmpraf.writeBytes(System.lineSeparator());
    }
    raf.seek(0);
    tmpraf.seek(0);
}
```

```
        while(tmpraf.getFilePointer()<tmpraf.length()){
            raf.writeBytes(tmpraf.readLine());
            raf.writeBytes(System.lineSeparator());
        }
        raf.setLength(tmpraf.length());

        tmpraf.close();
        raf.close();

        tmpFile.delete();
        JOptionPane.showMessageDialog(null,"Paciente eliminado",
            "INFORMATION_MESSAGE",JOptionPane.INFORMATION_MESSAGE);
        txtNombre.setText("");
        txtNumero.setText("");
    }

    else{
        raf.close();

        JOptionPane.showMessageDialog(null,"El paciente no existe",
            "INFORMATION_MESSAGE",JOptionPane.INFORMATION_MESSAGE);

    }

}

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

}

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

}
```

```
private void btnUpdateActionPerformed(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 = txtNombre.getText();  
  
        // Get the number to be updated  
        // from the Command line argument  
        long newNumber = Long.parseLong(txtNumero.getText());  
  
        String nameNumberString;  
        String name;  
        long number;  
        int index;  
  
        // Using file pointer creating the file.  
        File file = new  
File("C:\\Users\\Usuario\\Documents\\Git\\Hub\\pooactividad6\\files\\Pacientes.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.equals(newName) || number == newNumber) {
    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.equals(newName)) {
```

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

// 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
// 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.
tmpraf.close();
raf.close();

// Deleting the temporary file
```

```
        tmpFile.delete();

        //System.out.println(" Friend updated. ");
        JOptionPane.showMessageDialog(null, "Paciente 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. ");
        JOptionPane.showMessageDialog(null, "El paciente "+newName+" no
existe");
    }
}

catch (IOException ioe) {
    //System.out.println(ioe);
    JOptionPane.showMessageDialog(null, "Error "+ioe);
}

catch (NumberFormatException nef) {
    //System.out.println(nef);
    JOptionPane.showMessageDialog(null, "Error "+nef);
}

}

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

    try{
        String nameNumberString;
```

```
String name;
long number;
int index;
float total;

total = 0;

File file = new
File("C:\\Users\\Usuario\\Documents\\GitHub\\pooactividad6\\files\\Pacientes.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]);

    total += number;

}
```

```
        JOptionPane.showMessageDialog(null, "El total a recibir es: "+total);
    }

    catch (IOException ioe) {
        //System.out.println(ioe);
        JOptionPane.showMessageDialog(null, "Error "+ioe);
    }

    catch (NumberFormatException nef) {
        //System.out.println(nef);
        JOptionPane.showMessageDialog(null, "Error "+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.getClassName());
                break;
            }
        }
    }
    catch (ClassNotFoundException ex) {
```



```
java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

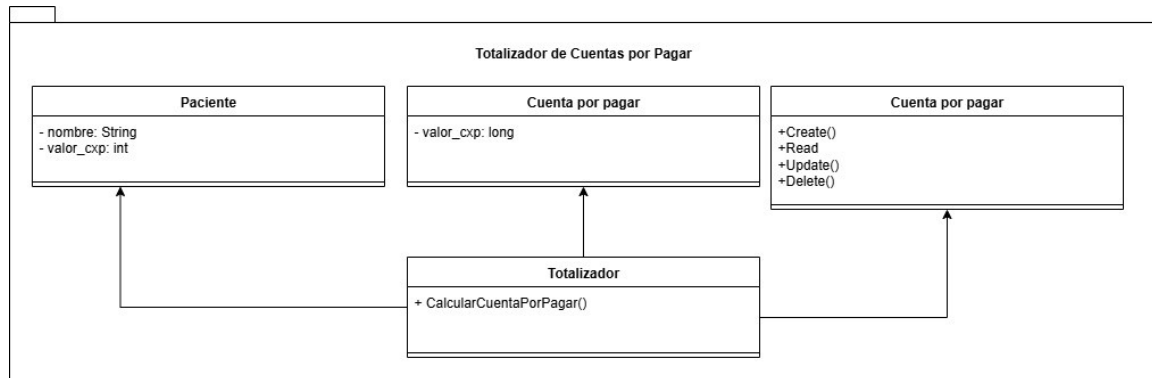
java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

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

// Variables declaration - do not modify
private javax.swing.JButton btnCreate;
private javax.swing.JButton btnDelete;
private javax.swing.JButton btnRead;
private javax.swing.JButton btnTotal;
private javax.swing.JButton btnUpdate;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel lbNombre;
private javax.swing.JLabel lbNumero;
private javax.swing.JTextField txtNombre;
private javax.swing.JTextField txtNumero;
// End of variables declaration
}
```

## Diagramas de Clases



## Diagramas de Casos de Uso

