MANUAL TÉCNICO

Biblioteca Facultad de Ingeniería USAC

Por: Mario Ernesto Marroquín Pérez

Carné: 202110509

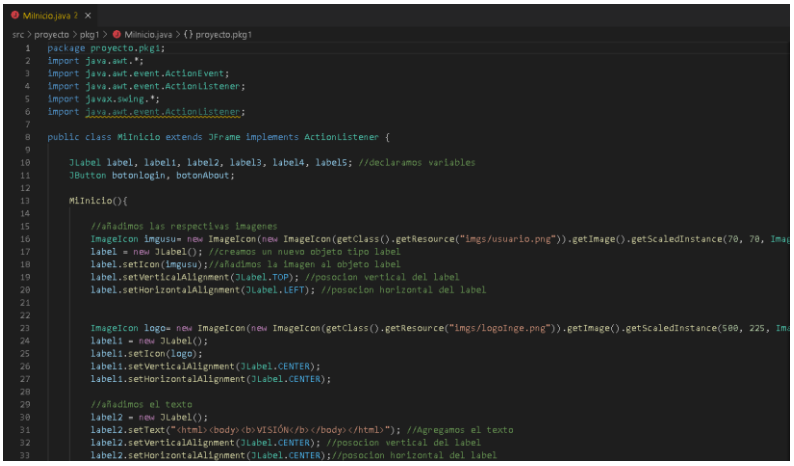
Observación: El software se encuentra escrito en Java.

Proyecto1.java 1

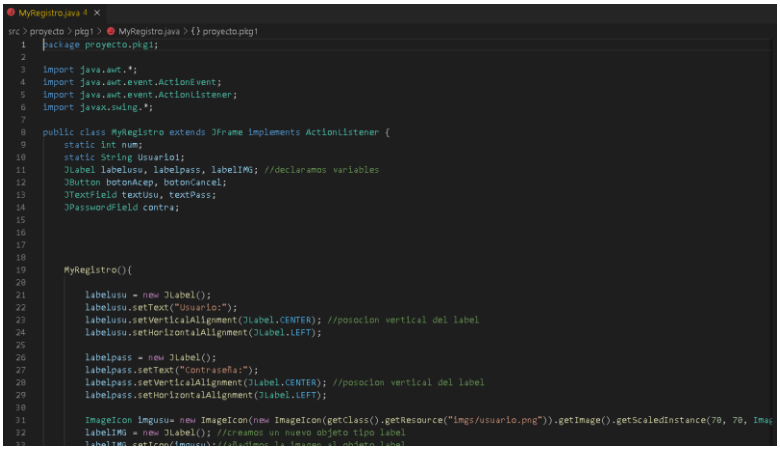
src > proyecto > pkg1 > Proyecto1.java > ...

```java
package proyecto.pkg1;

/**
 *
 * @author marroquin
 */
public class Proyecto1 {

    /**
     * @param args the command line arguments
     */
    Run | Debug
    public static void main(String[] args) {
        MiInicio frame = new MiInicio();
    }

}
```

Proyecto 1.java es nuestra clase principal en donde indicamos que se ejecutará primero, en este caso será la clase “MiInicio”, la cual nos muestra la visión de la biblioteca, así como nos da la opción de ingresar a la misma.

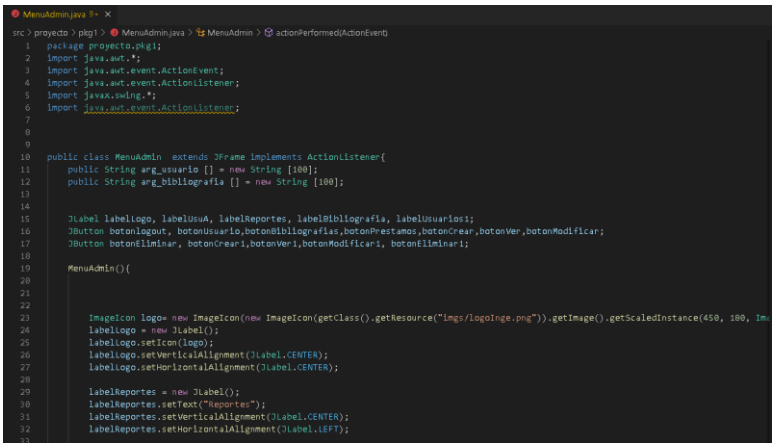


Al presionar iniciar sesión, vamos hacia la clase “MyRegistro” la cual valida nuestros datos para otorgarnos acceso a la biblioteca ya seamos un usuario normal o el administrador.



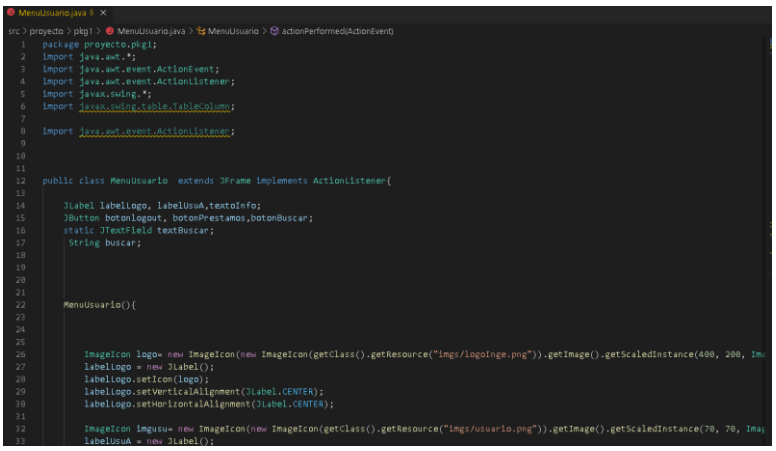
```java
package proyecto.pkg1;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class MyRegistro extends JFrame implements ActionListener {
    static int num;
    static String Usuario1;
    JLabel labelusu, labelpass, labelIMG; //declaramos variables
    JButton botonAcep, botonCancel;
    JTextField textUsu, textPass;
    JPasswordField contra;



    MyRegistro(){

        labelusu = new JLabel();
        labelusu.setText("Usuario:");
        labelusu.setVerticalAlignment(JLabel.CENTER); //posocion vertical del label
        labelusu.setHorizontalAlignment(JLabel.LEFT);

        labelpass = new JLabel();
        labelpass.setText("Contraseña:");
        labelpass.setVerticalAlignment(JLabel.CENTER); //posocion vertical del label
        labelpass.setHorizontalAlignment(JLabel.LEFT);

        ImageIcon imgusu= new ImageIcon(new ImageIcon(getClass().getResource("imgs/usuario.png")).getImage().getScaledInstance(70, 70, Ima
        labelIMG = new JLabel(); //creamos un nuevo objeto tipo label
```

Si somos administrador seremos enviados a la clase MenuAdmin, en donde se crean, modifican, eliminan y muestran los usuarios y bibliografías así como los préstamos.



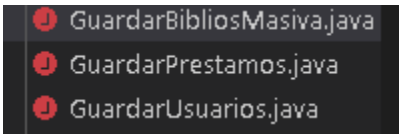
```java
package proyecto.pkg1;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import java.awt.event.ActionListener;



public class MenuAdmin  extends JFrame implements ActionListener{
    public String arg_usuario [] = new String [100];
    public String arg_bibliografia [] = new String [100];


    JLabel labelLogo, labelUsuA, labelReportes, labelBibliografia, labelUsuarios1;
    JButton botonlogout, botonUsuario,botonBibliografias,botonPrestamos,botonCrear,botonVer,botonModificar;
    JButton botonEliminar, botonCrear1,botonVer1,botonModificar1, botonEliminar1;

    MenuAdmin(){



        ImageIcon logo= new ImageIcon(new ImageIcon(getClass().getResource("imgs/logoInge.png")).getImage().getScaledInstance(450, 100, Ima
        labelLogo = new JLabel();
        labelLogo.setIcon(logo);
        labelLogo.setVerticalAlignment(JLabel.CENTER);
        labelLogo.setHorizontalAlignment(JLabel.CENTER);

        labelReportes = new JLabel();
        labelReportes.setText("Reportes");
        labelReportes.setVerticalAlignment(JLabel.CENTER);
        labelReportes.setHorizontalAlignment(JLabel.LEFT);
```

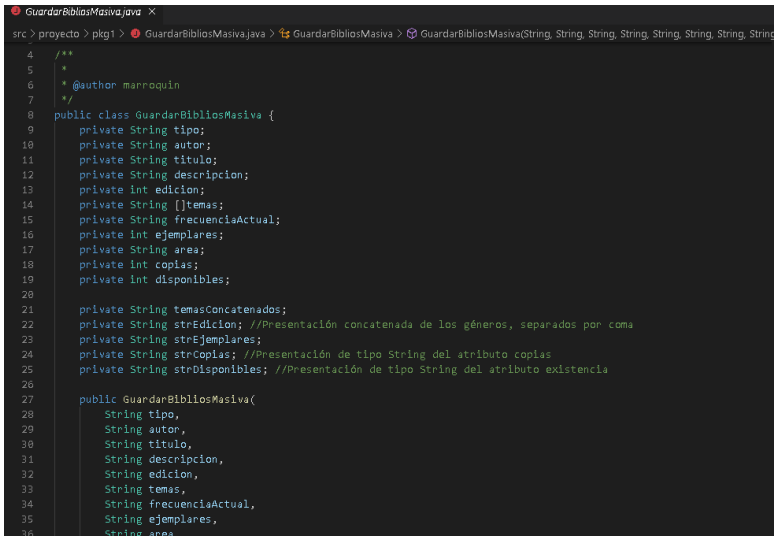
Si somos un usuario normal seremos enviados a la clase “MenuUsuario” en donde podemos visualizar las bibliografias de interés y ver los préstamos que hemos realizado.



```java
package proyecto.pkg1;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import javax.swing.table.TableColumn;

import java.awt.event.ActionListener;



public class MenuUsuario  extends JFrame implements ActionListener{

    JLabel labelLogo, labelUsuA,textoInfo;
    JButton botonlogout, botonPrestamos,botonBuscar;
    static JTextField textBuscar;
     String buscar;



    MenuUsuario(){



        ImageIcon logo= new ImageIcon(new ImageIcon(getClass().getResource("imgs/logoInge.png")).getImage().getScaledInstance(400, 200, Ima
        labelLogo = new JLabel();
        labelLogo.setIcon(logo);
        labelLogo.setVerticalAlignment(JLabel.CENTER);
        labelLogo.setHorizontalAlignment(JLabel.CENTER);

        ImageIcon imgusu= new ImageIcon(new ImageIcon(getClass().getResource("imgs/usuario.png")).getImage().getScaledInstance(70, 70, Ima
        labelUsuA = new JLabel();
```

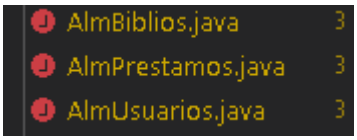
Estas clases se enlazan con otras clases que son donde se almacenan los textos de entradas, estas clases llevan por nombre “GuardarUsuarios”, “GuardarBiblios” y “GuardarUsuarios”, dentro de estas clases se encuentran los setters and getters.



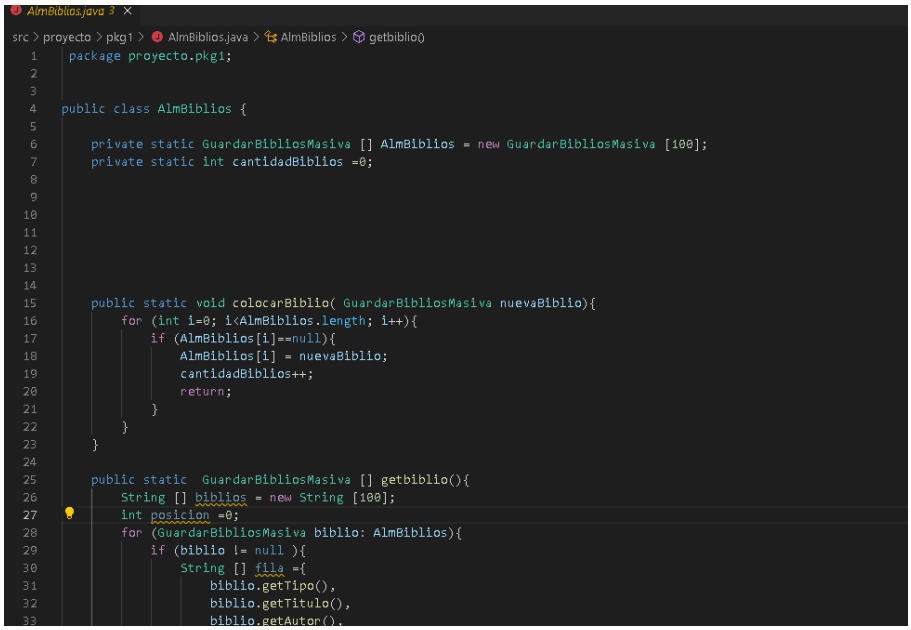



```
71      public String getTipo() {
72          return this.tipo;
73      }
74
75      public void setTipo(String tipo) {
76          this.tipo = tipo;
77      }
78
79      public String getAutor() {
80          return this.autor;
81      }
82
83      public void setAutor(String autor) {
84          this.autor = autor;
85      }
86
87      public String getTitulo() {
88          return this.titulo;
89      }
90
91      public void setTitulo(String titulo) {
92          this.titulo = titulo;
93      }
94
95      public String getDescripcion() {
96          return this.descripcion;
97      }
98
99      public void setDescripcion(String descripcion) {
100         this.descripcion = descripcion;
101     }
```

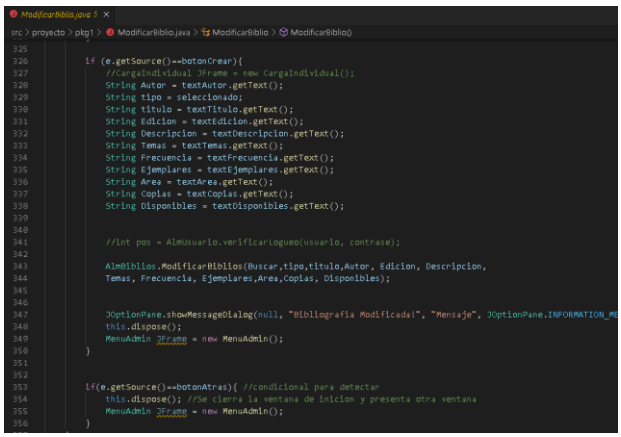
Los datos encapsulados de las clases anteriores se enlazan con una clase de almacenamiento, la de usuario se enlaza con la clase “AlmUsuario” y “AlmPrestamos”, mientras que la clase administrador se enlaza con “AlmBibliografias” y “AlmPrestamos”, todas las clases realizan el mismo proceso, retornar los datos ingresados por el administrador para su manipulación.



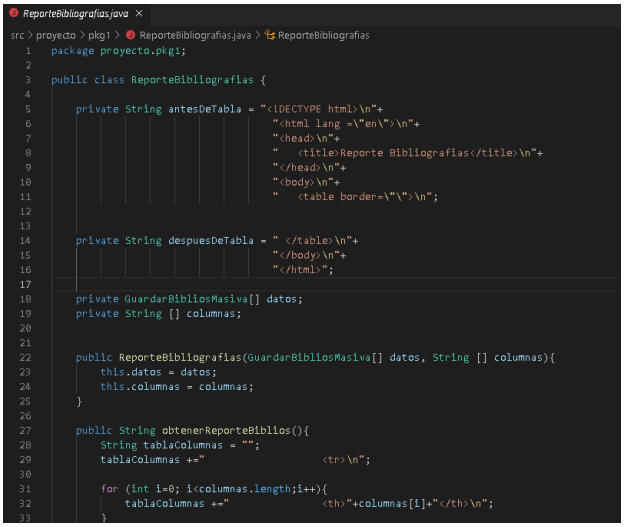
Dentro de estas clases se encuentran distintos métodos, por ejemplo “ColocarBibliografia”, “Obtener bibliografia”, métodos los cuales podemos obtener arreglos de los datos ingresados.



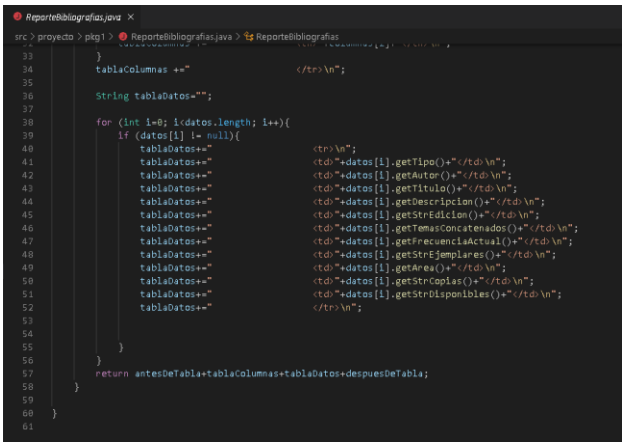
Dentro de las clases para modificar usuarios o librerías se encuentra la interfaz grafica de estas clases, estas clases se enlazan con las clases de almacenamiento y manipulación para retornar los datos que se desean eliminar o modificar.



Dentro de las clases “ReporteBiblio”, “ReporteUsuaros” y “ReportePrestamos” se encuentra los métodos para obtener los arreglos de las anteriores clases para elaborar el reporte en HTML, esta clase contiene código java así como lenguaje HTML, todas las clases hacen el mismo trabajo.

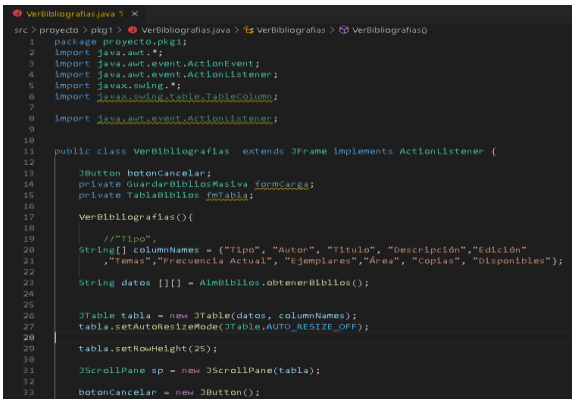


```java
package proyecto.pkg1;

public class ReporteBibliografias {

    private String antesDeTabla = "<!DOCTYPE html>\n"+
                                  "<html lang =\"en\">\n"+
                                  "<head>\n"+
                                  "    <title>Reporte Bibliografias</title>\n"+
                                  "</head>\n"+
                                  "<body>\n"+
                                  "    <table border=\"\">\n";


    private String despuesDeTabla = " </table>\n"+
                                    "</body>\n"+
                                    "</html>";

    private GuardarBibliosMasiva[] datos;
    private String [] columnas;


    public ReporteBibliografias(GuardarBibliosMasiva[] datos, String [] columnas){
        this.datos = datos;
        this.columnas = columnas;
    }

    public String obtenerReporteBiblios(){
        String tablaColumnas = "";
        tablaColumnas +="                <tr>\n";

        for (int i=0; i<columnas.length;i++){
            tablaColumnas +="                <th>"+columnas[i]+"</th>\n";
        }
```



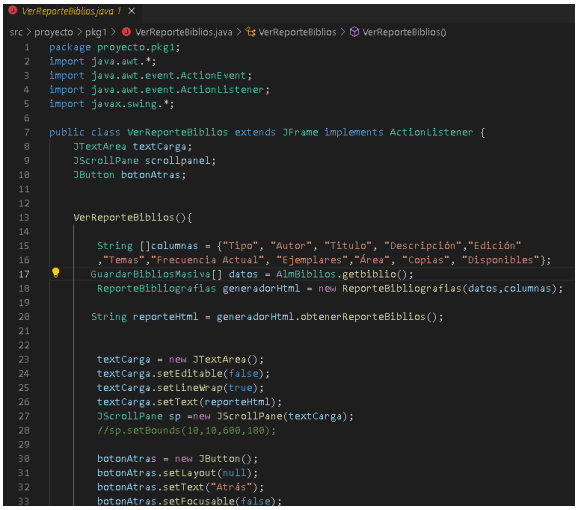
```java
        }
        tablaColumnas +="                </tr>\n";

        String tablaDatos="";

        for (int i=0; i<datos.length; i++){
            if (datos[i] != null){
                tablaDatos+="                <tr>\n";
                tablaDatos+="                <td>"+datos[i].getTipo()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getAutor()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getTitulo()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getDescripcion()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getStrEdicion()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getTemasConcatenados()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getFrecuenciaActual()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getStrEjemplares()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getArea()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getStrCopias()+"</td>\n";
                tablaDatos+="                <td>"+datos[i].getStrDisponibles()+"</td>\n";
                tablaDatos+="                </tr>\n";


            }
        }
        return antesDeTabla+tablaColumnas+tablaDatos+despuesDeTabla;
    }

}
```

Las clases “VerBibliografia”, “VerPrestamos” y “VerUsuarios” son las encargadas de retornarnos los datos de los arreglos en formas de tablas, estas clases entran en funcionamiento cuando el administrador da clic sobre la opción “ver” ya sea para bibliografias o usuarios.



```java
package proyecto.pkg1;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import javax.swing.table.TableColumn;

import java.awt.event.ActionListener;


public class VerBibliografias  extends JFrame implements ActionListener {

    JButton botonCancelar;
    private GuardarBibliosMasiva formCarga;
    private TablaBiblios fmTabla;

    VerBibliografias(){

        //"Tipo",
    String[] columnNames = {"Tipo", "Autor", "Titulo", "Descripción","Edición"
        ,"Temas","Frecuencia Actual", "Ejemplares","Área", "Copias", "Disponibles"};

    String datos [][] = AlmBiblios.obtenerBiblios();


    JTable tabla = new JTable(datos, columnNames);
    tabla.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

    tabla.setRowHeight(25);

    JScrollPane sp = new JScrollPane(tabla);

    botonCancelar = new JButton();
```

Las clases “VerReporteBibliografia”, “VerReportePrestamos” y “VerReporteUsuarios” son las encargadas de estructuras las tablas que se mostraran en lenguaje HTML, las tablas obtienen los arreglos de las clases “ReporteBiblio”, “ReporteUsuaros” y “ReportePrestamos”, luego de obtener los arreglos se hace llamado al método “reporteHTML” el cual se definió en las clases de Reportes y se imprime el HTML generado en un textArea.



```
VerReporteBiblios.java 1
src > proyecto > pkg1 > VerReporteBiblios.java > VerReporteBiblios > VerReporteBiblios()
1   package proyecto.pkg1;
2   import java.awt.*;
3   import java.awt.event.ActionEvent;
4   import java.awt.event.ActionListener;
5   import javax.swing.*;
6
7   public class VerReporteBiblios extends JFrame implements ActionListener {
8       JTextArea textCarga;
9       JScrollPane scrollpanel;
10      JButton botonAtras;
11
12
13      VerReporteBiblios(){
14
15          String []columnas = {"Tipo", "Autor", "Titulo", "Descripción","Edición"
16          ,"Temas","Frecuencia Actual", "Ejemplares","Área", "Copias", "Disponibles"};
17          GuardarBibliosMasiva[] datos = AlmBiblios.getbiblio();
18          ReporteBibliograflas generadorHtml = new ReporteBibliograflas(datos,columnas);
19
20          String reporteHtml = generadorHtml.obtenerReporteBiblios();
21
22
23          textCarga = new JTextArea();
24          textCarga.setEditable(false);
25          textCarga.setLineWrap(true);
26          textCarga.setText(reporteHtml);
27          JScrollPane sp =new JScrollPane(textCarga);
28          //sp.setBounds(10,10,600,180);
29
30          botonAtras = new JButton();
31          botonAtras.setLayout(null);
32          botonAtras.setText("Atrás");
33          botonAtras.setFocusable(false);
```

Dentro de la carpeta “scr” del proyecto se encuentra otra carpeta llamada “imgs” la cual contiene todas las imágenes utilizadas durante la ejecución del programa.