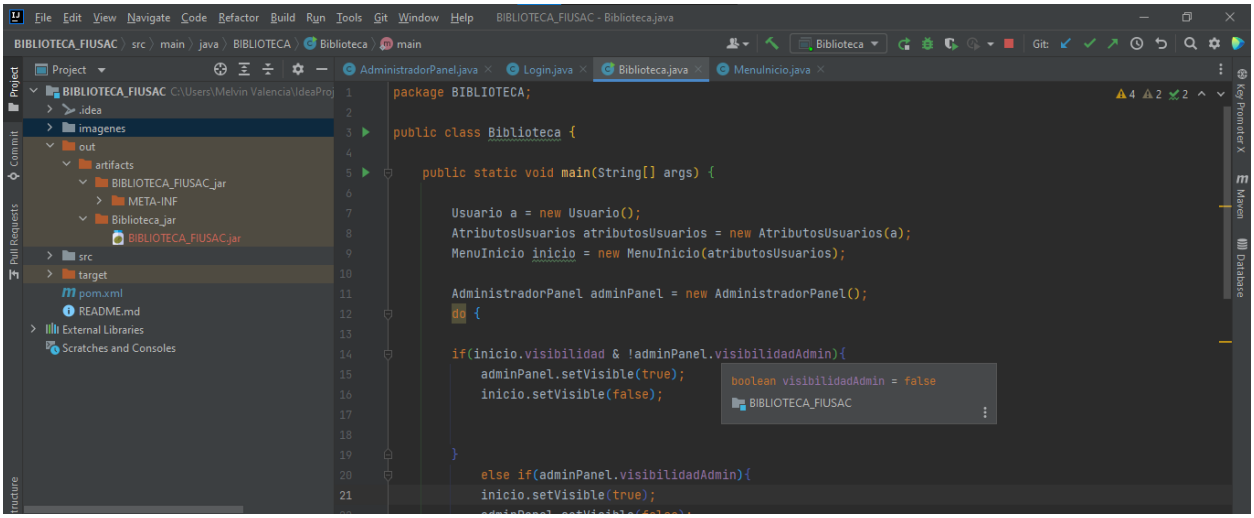
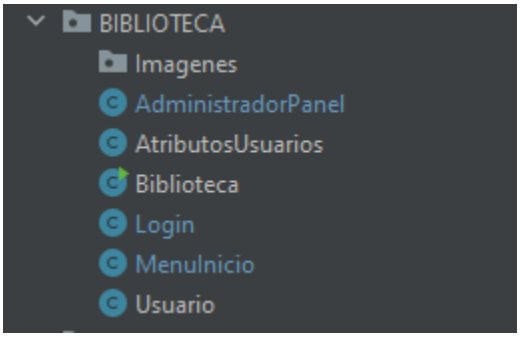


Existe una clase inicial, Biblioteca que crea todos los métodos y se controlan desde ahí pasando objetos y otros atributos.



La clase AdministradorPanel contiene la interfaz del administrador.

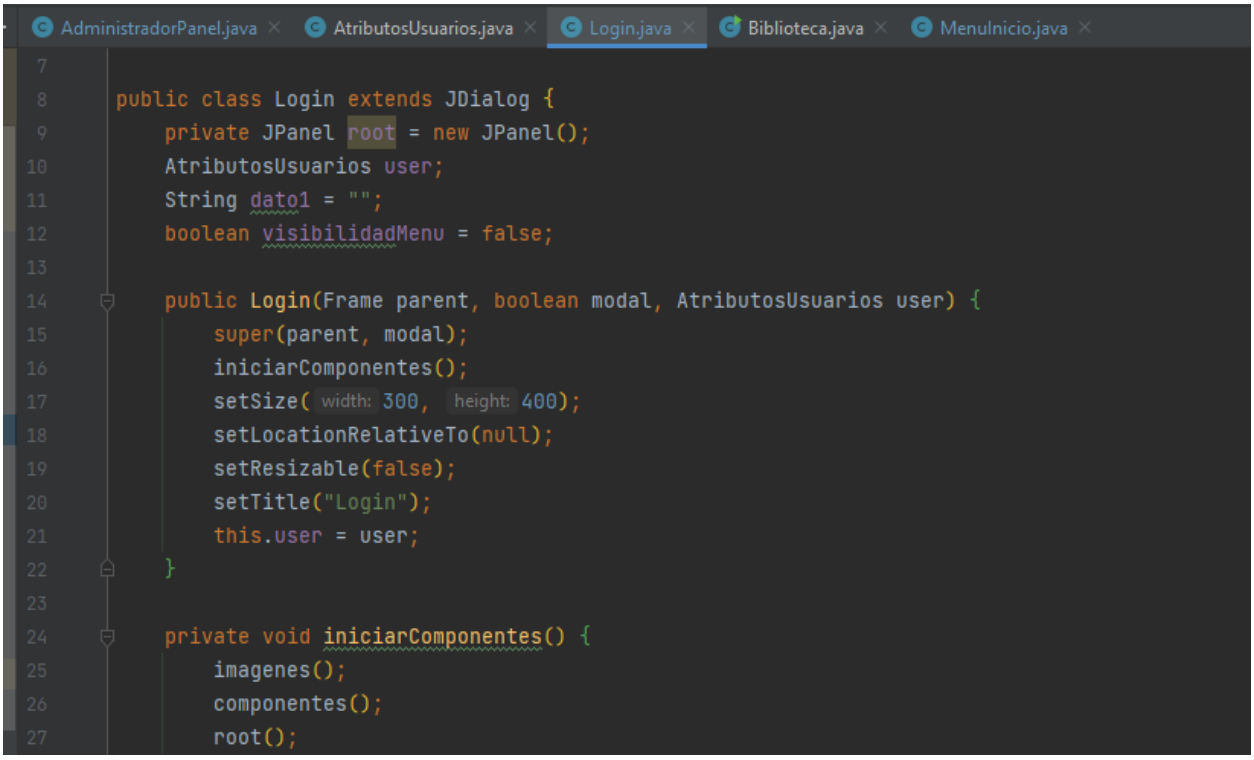
```java
package BIBLIOTECA;

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

public class AdministradorPanel extends JFrame {
    JPanel root = new JPanel();
    boolean visibilida

    public Administrad
        iniciarCompone
        setSize( width:
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        setTitle("ADMINISTRADOR");
    }

    public void iniciarComponentes(){...}
```

```
public JPanel()

Creates a new JPanel with a double buffer
and a flow layout.
< 1.8 > (rt.jar)
```

La clase AtributoUsuario sirve como intermediario entre la clase principal y Usuario.

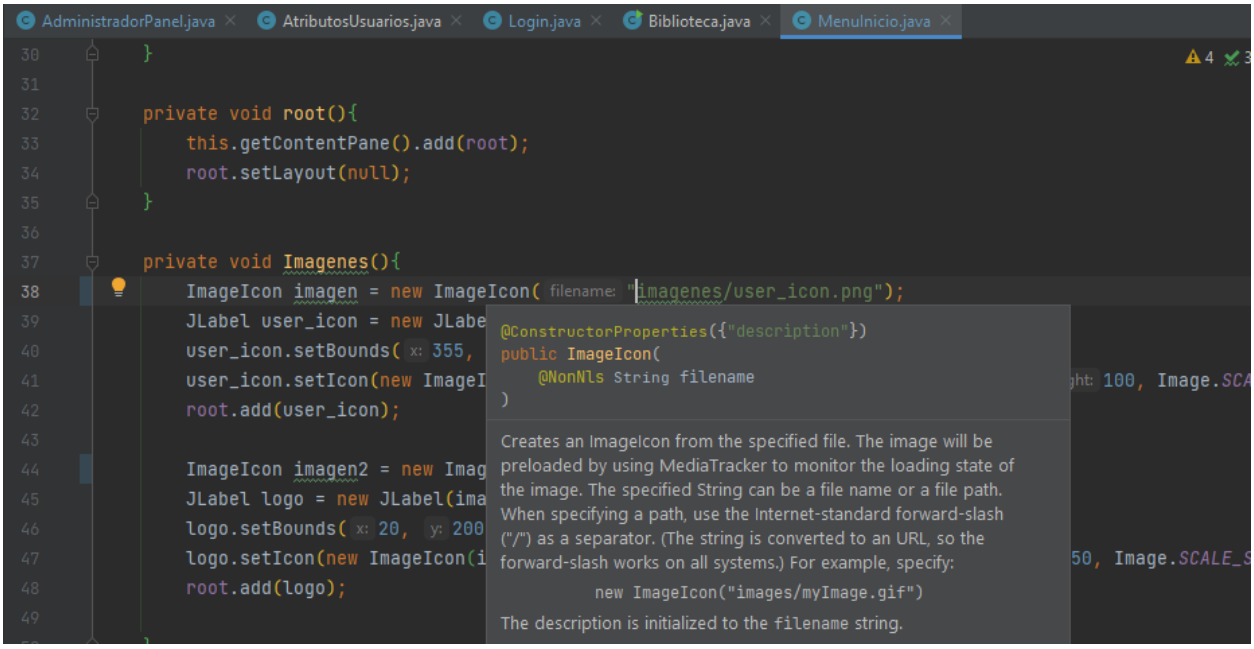
```java
package BIBLIOTECA;

public class AtributosUsuarios{
    private static Usuario[] usuario = new Usuario[100];
    private static int contador = 1;

    public AtributosUsuarios(Usuario a){
        colocarUsuario(a);
    }

    public static void colocarUsuario(Usuario nUsuario) {
        for(int i = 0; i < usuario.length; i++) {
            if(usuario[i] == null) {
                usuario[i] = nUsuario;
                contador++; //Incrementar la cantidad de comics con cada Comic agregado
                return;
            }
        }
    }

    public String getId(int i) { return usuario[i].getId(); }
```

La clase Login es una ventana emergente para validar credenciales.



```java
public class Login extends JDialog {
    private JPanel root = new JPanel();
    AtributosUsuarios user;
    String dato1 = "";
    boolean visibilidadMenu = false;

    public Login(Frame parent, boolean modal, AtributosUsuarios user) {
        super(parent, modal);
        iniciarComponentes();
        setSize( width: 300,  height: 400);
        setLocationRelativeTo(null);
        setResizable(false);
        setTitle("Login");
        this.user = user;
    }

    private void iniciarComponentes() {
        imagenes();
        componentes();
        root();
```

La clase MenuInicio es la pagina principal para acceder al login.



```java
    }

    private void root(){
        this.getContentPane().add(root);
        root.setLayout(null);
    }

    private void Imagenes(){
        ImageIcon imagen = new ImageIcon( filename: "imagenes/user_icon.png");
        JLabel user_icon = new JLabe
        user_icon.setBounds( x: 355,
        user_icon.setIcon(new ImageI
        root.add(user_icon);

        ImageIcon imagen2 = new Imag
        JLabel logo = new JLabel(ima
        logo.setBounds( x: 20,  y: 200
        logo.setIcon(new ImageIcon(i
        root.add(logo);
```

```
@ConstructorProperties({"description"})
public ImageIcon(
    @NonNls String filename
)
```

Creates an ImageIcon from the specified file. The image will be preloaded by using MediaTracker to monitor the loading state of the image. The specified String can be a file name or a file path. When specifying a path, use the Internet-standard forward-slash ("/") as a separator. (The string is converted to an URL, so the forward-slash works on all systems.) For example, specify:

```
new ImageIcon("images/myImage.gif")
```

The description is initialized to the filename string.

La clase Usuario contiene los atributos de los usuarios

AdministradorPanel.java | AtributosUsuarios.java | Login.java | Biblioteca.java | MenuInicio.java | Usuario.java

```java
package BIBLIOTECA;

public class Usuario {
private String id, nombre, apellido, usuario, rol, contraseña;


    public String getId() {
        return id;
    }

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
```