Manual Técnico para Biblioteca

Proyecto 1 IPC1 – Lusvin Alexander Sicajá Ramírez
Carné: 201602630

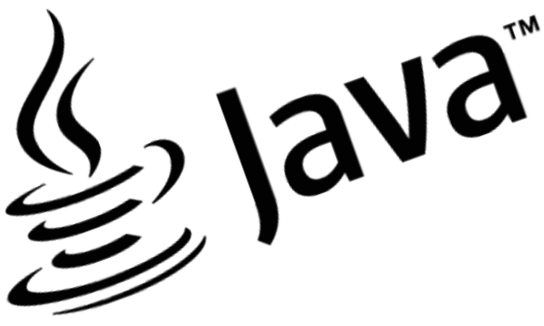
BibliotecApp

METODO PRINCIPAL

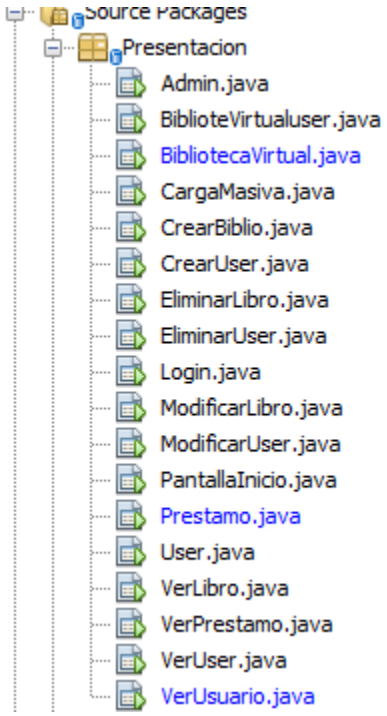
Se crea una clase principal con el método MAIN, el cual tiene como función única el llamado al frame de la clase Pantalla principal.

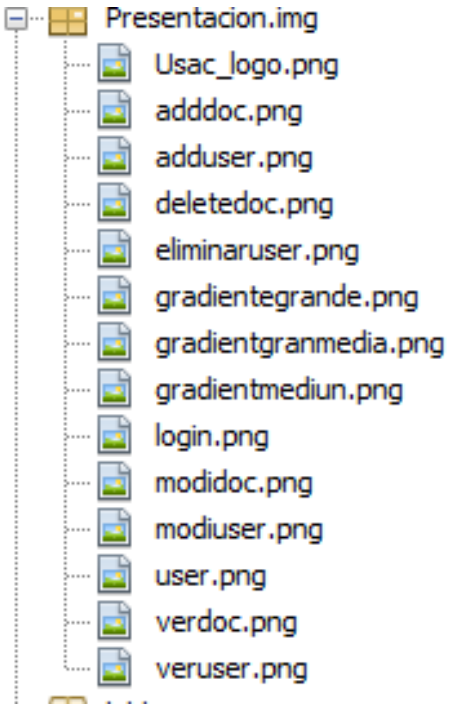
También se crean las variables principales con la palabra static para que éstas tengan siempre sus valores en las demás clases.

```
 8   */
 9   public class Biblioteca {
10
11       //Variables globales
12       //Usuario administrador
13       public static Administrador admin1 = new Administrador(1, "admin", "admin", "admin", "admin");
14       //Conteo de Usuario y vector de usuarios.
15       public static Integer contuser = 0;
16       public static Usuario users[] = new Usuario[10];
17
18       //Conteo de Documentos y vector de documentos.
19       public static Integer contlibros = 0;
20       public static Integer contrevistas = 0;
21       public static Integer conttesis = 0;
22       public static Integer contlibrodigital = 0;
23
24       public static Libro libros[] = new Libro[50];
25       public static Revista revistas[] = new Revista[50];
26       public static Tesis tesis[] = new Tesis[50];
27       public static LibroDigital librosDigital[] = new LibroDigital[30];
28
29       public static void main(String[] args) {
30
31           PantallaInicio pi = new PantallaInicio();
32           pi.setVisible(true);
```



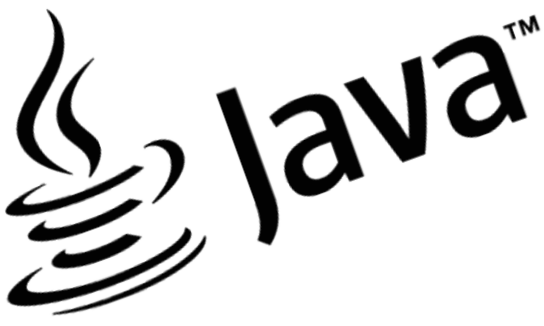
Se creo un paquete PRESENTACIÓN el cual contiene todos los Jframe de la aplicación, también dentro de este paquete se encuentra un paquete que contiene todo lo referente a gráficos que se utilizaron el los frame y demás etiquetas.





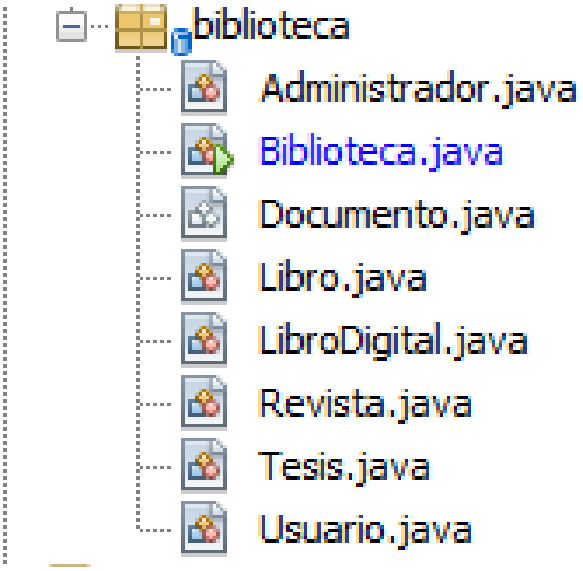
METODO DE INGRESO Y VALIDACIÓN DE CREDENCIALES:

```
private void jbtingresarlogMouseClicked(java.awt.event.MouseEvent evt) {

      for (int i = 0; i < biblioteca.Biblioteca.contuser; i++) {
         if (biblioteca.Biblioteca.users[i].getUser().equalsIgnoreCase(ftxtuser.getText())) {
            this.userlog = biblioteca.Biblioteca.users[i].getUser();
            this.userlogpass = biblioteca.Biblioteca.users[i].getPassword();
            this.userlogpos = i;
         } else {
            //this.userlog = null;
         }
      }

      if (ftxtuser.getText().length() == 0 && 0 == fpassuser.getPassword().length) {
         JOptionPane.showMessageDialog(null, "Ingrese Datos por favor...", "Error de Datos ",
JOptionPane.WARNING_MESSAGE);
      } else if (biblioteca.Biblioteca.admin1.getUser().equalsIgnoreCase(ftxtuser.getText()) &&
biblioteca.Biblioteca.admin1.getPassword().equals(new String(fpassuser.getPassword()))) {
         this.setVisible(false);
         Admin ad = new Admin();
         ad.setVisible(true);
         pi.setVisible(false);
      } else if (biblioteca.Biblioteca.admin1.getUser().equalsIgnoreCase(ftxtuser.getText()) &&
!biblioteca.Biblioteca.admin1.getPassword().equals(new String(fpassuser.getPassword()))) {
         JOptionPane.showMessageDialog(null, "Credencial de administrador incorrecta...", "Error de credenciales ",
JOptionPane.WARNING_MESSAGE);
      } else if (userlog == null) {
         JOptionPane.showMessageDialog(null, "El usuario no existe, ponerse en contacto con el Administrador...", "Error
de Usuario ", JOptionPane.WARNING_MESSAGE);
      } else if (userlog.equalsIgnoreCase(ftxtuser.getText()) && !userlogpass.equals(new String(fpassuser.getPassword())))
{
         JOptionPane.showMessageDialog(null, "El usuario y contraseña no coinciden revise sus datos...", "Error de
credenciales ", JOptionPane.WARNING_MESSAGE);
      } else if (userlog.equalsIgnoreCase(ftxtuser.getText()) && userlogpass.equals(new String(fpassuser.getPassword())))
{
         this.setVisible(false);
         User us = new User();
         us.setVisible(true);
         pi.setVisible(false);
      }

      System.out.print(userlog + userlogpass + userlogpos);


   }
```



PAQUETE BIBLIOTECA

En dicho paquete se encuentra las clases modelo que se utilizo para la funcionalidad del proyecto, se utiliza el paradigma orientación a objetos.



Se crea una clase Abstracta de la cual heredan los diferentes tipos de documentos:

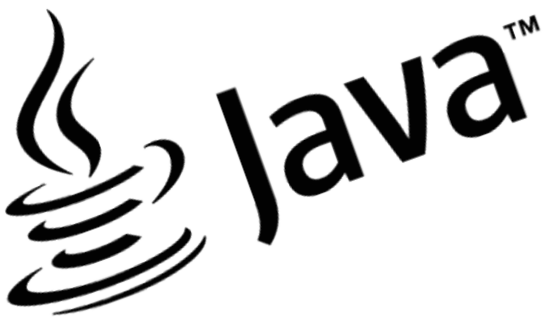
```
package biblioteca;


public abstract class Documento {

    private Integer tipo;
    private Integer anio_publica;
    private String autor;
    private String descripcion;
    private Integer edicion;
    private String[] palabraClave;
    private String[] temas;
    private String titulo;
```

Se crea una clase llamada Usuario que será nuestro objeto de logeo

```java
package biblioteca;

public class Usuario {

    private Integer id; //el DPI
    private String nombre;
    private String apellido;
    private String user; //nickename
    private String rol;
    private String password;
    private Libro[] libro;
    private Revista[] revista;
    private Tesis[] tesis;
    private LibroDigital[] libroDigital;

```

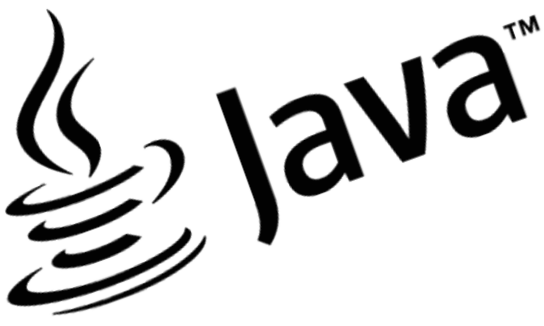
Adicionalmente se crea también una clase Administrador

```java

package biblioteca;

public class Administrador {

    private Integer id;
    private String nombre;
    private String user;
    private String rol;
    private String password;

    public Administrador(Integer id, String nombre, String user, String rol, String password) {
        this.id = id;
        this.nombre = nombre;
        this.user = user;
        this.rol = rol;
        this.password = password;
    }


```



METODO QUE VERIFICA EL INGRESO DE LOS USUARIOS AL SISTEMA

```
  //System.out.println("valor de contador "+biblioteca.Biblioteca.contuser);
      boolean permite = false;
     // System.out.println("user en posicion 1: " + biblioteca.Biblioteca.users[0]);

      if (jtxid.getText().length() == 0 || 0 == jtxuser.getText().length()) {
         JOptionPane.showMessageDialog(null, "Ingrese Datos por favor...", "Faltan Datos ",
JOptionPane.WARNING_MESSAGE);
      } else if (!(new String(jtxpass.getPassword()).equals(new String(jtxpassconfirm.getPassword())))) {
         JOptionPane.showMessageDialog(null, "Las contraseñas no coinciden...", "Error al crear usuario",
JOptionPane.WARNING_MESSAGE);
      } else if (biblioteca.Biblioteca.users[0]==null) {
         permite = true;
          System.out.println(biblioteca.Biblioteca.contuser);
      } else {
         for (int i = 0; i < biblioteca.Biblioteca.contuser; i++) {
            if (biblioteca.Biblioteca.users[i].getId() == Integer.parseInt(jtxid.getText())) {
               JOptionPane.showMessageDialog(null, "El usuario con éste ID ya existe...", "Error al crear usuario",
JOptionPane.WARNING_MESSAGE);
            } else {
               permite = true;
            }
         }
      }

JOptionPane.showMessageDialog(null, "El usuario: " +
biblioteca.Biblioteca.users[biblioteca.Biblioteca.contuser].getNombre() + " fue agregado exitosamente", "Usuario
agregado", JOptionPane.INFORMATION_MESSAGE);
         biblioteca.Biblioteca.contuser++;
         setVisible(false);
```



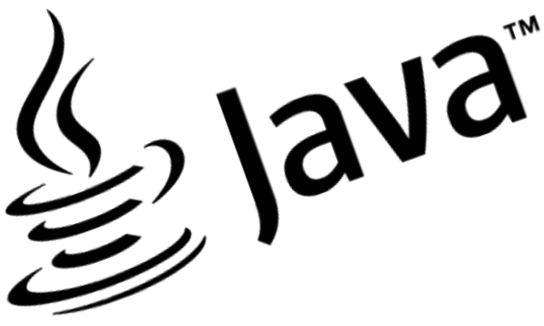
METODO QUE SE UTILIZA PARA LA CARGA DE DOCUMENTOS:

```
boolean permite = false;

      if (txtisbn.getText().length() == 0) {
         JOptionPane.showMessageDialog(null, "Ingrese Datos por favor...", "Faltan Datos ",
JOptionPane.WARNING_MESSAGE);
      } else if (((String) jcboxdoc.getSelectedItem()).equalsIgnoreCase("Libro")) {
         if (biblioteca.Biblioteca.libros[0] == null) {
            permite = true;
         } else {
            for (int i = 0; i < biblioteca.Biblioteca.contlibros; i++) {
               if (biblioteca.Biblioteca.libros[i].getIsbn() == Integer.parseInt(txtisbn.getText())) {
                  JOptionPane.showMessageDialog(null, "El Documento con este ID ya existe...", "Error al crear documento",
JOptionPane.WARNING_MESSAGE);
               } else {
                  permite = true;
               }
            }
         }
         if (permite) {
            String palabra[] = txtpalabra.getText().split(",");
            String tema[] = txttemas.getText().split(",");
            biblioteca.Biblioteca.libros[biblioteca.Biblioteca.contlibros] = new Libro(Integer.parseInt(txtcopias.getText()),
Integer.parseInt(txtdisponible.getText()), Integer.parseInt(txtisbn.getText()), 0, Integer.parseInt(txtaño.getText()),
txtautor.getText(), txtdescrip.getText(), Integer.parseInt(txtedicion.getText()), palabra, tema, txttitulo.getText());
            JOptionPane.showMessageDialog(null, "El Libro: " +
biblioteca.Biblioteca.libros[biblioteca.Biblioteca.contlibros].getTitulo() + " fue agregado exitosamente", "Libro
agregado", JOptionPane.INFORMATION_MESSAGE);
            biblioteca.Biblioteca.contlibros++;
            setVisible(false);
         }

      } else if (((String) jcboxdoc.getSelectedItem()).equalsIgnoreCase("Revista")) {
         if (biblioteca.Biblioteca.revistas[0] == null) {
            permite = true;
         } else {
            for (int i = 0; i < biblioteca.Biblioteca.contrevistas; i++) {
               if (biblioteca.Biblioteca.revistas[i].getId()== Integer.parseInt(txtisbn.getText())) {
                  JOptionPane.showMessageDialog(null, "El Documento con este ID ya existe...", "Error al crear documento",
JOptionPane.WARNING_MESSAGE);
               } else {
                  permite = true;
               }
            }
         }
         if (permite) {
            String palabra[] = txtpalabra.getText().split(",");
            String tema[] = txttemas.getText().split(",");
```

```
        biblioteca.Biblioteca.revistas[biblioteca.Biblioteca.contrevistas] = new
Revista(Integer.parseInt(txtisbn.getText()), txtcategoria.getText(), Integer.parseInt(txtcopias.getText()),
Integer.parseInt(txtdisponible.getText()), Integer.parseInt(txtejemplares.getText()) , 1,
Integer.parseInt(txtaño.getText()), txtautor.getText(), txtdescrip.getText(), Integer.parseInt(txtedicion.getText()),
palabra, tema, txttitulo.getText());
        JOptionPane.showMessageDialog(null, "La Revista: " +
biblioteca.Biblioteca.revistas[biblioteca.Biblioteca.contrevistas].getTitulo() + " fue agregada exitosamente", "Revista
agregada", JOptionPane.INFORMATION_MESSAGE);
        biblioteca.Biblioteca.contrevistas++;
        setVisible(false);
    }
```

METODO QUE GENERA LA TABLA DE DOCUMENTOS

```
private void jbtBuscarMouseClicked(java.awt.event.MouseEvent evt) {

    String  tablaLibroDigital[][] = new String [biblioteca.Biblioteca.contlibrodigital][7];

    for (int i = 0; i < biblioteca.Biblioteca.contlibrodigital; i++) {
        Boolean prestado = false;
        tablaLibroDigital[i][0]= (Integer.toString(i));
        tablaLibroDigital[i][1]=biblioteca.Biblioteca.librosDigital[i].getId().toString();
        tablaLibroDigital[i][2]=biblioteca.Biblioteca.librosDigital[i].getTitulo();
        tablaLibroDigital[i][3]=biblioteca.Biblioteca.librosDigital[i].getEdicion().toString();
        tablaLibroDigital[i][4]=biblioteca.Biblioteca.librosDigital[i].getAutor();
        tablaLibroDigital[i][5]=Arrays.toString(biblioteca.Biblioteca.librosDigital[i].getTemas());
        tablaLibroDigital[i][6]=biblioteca.Biblioteca.librosDigital[i].getDescripcion();
        //tablaLibroDigital[i][7]= Boolean.toString(prestado);

    }

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        tablaLibroDigital,
        new String [] {
            "No.", "ID", "TITULO", "EDICION", "AUTOR", "TEMAS", "DESCRIPCION", "PRESTADO"
        }));
```

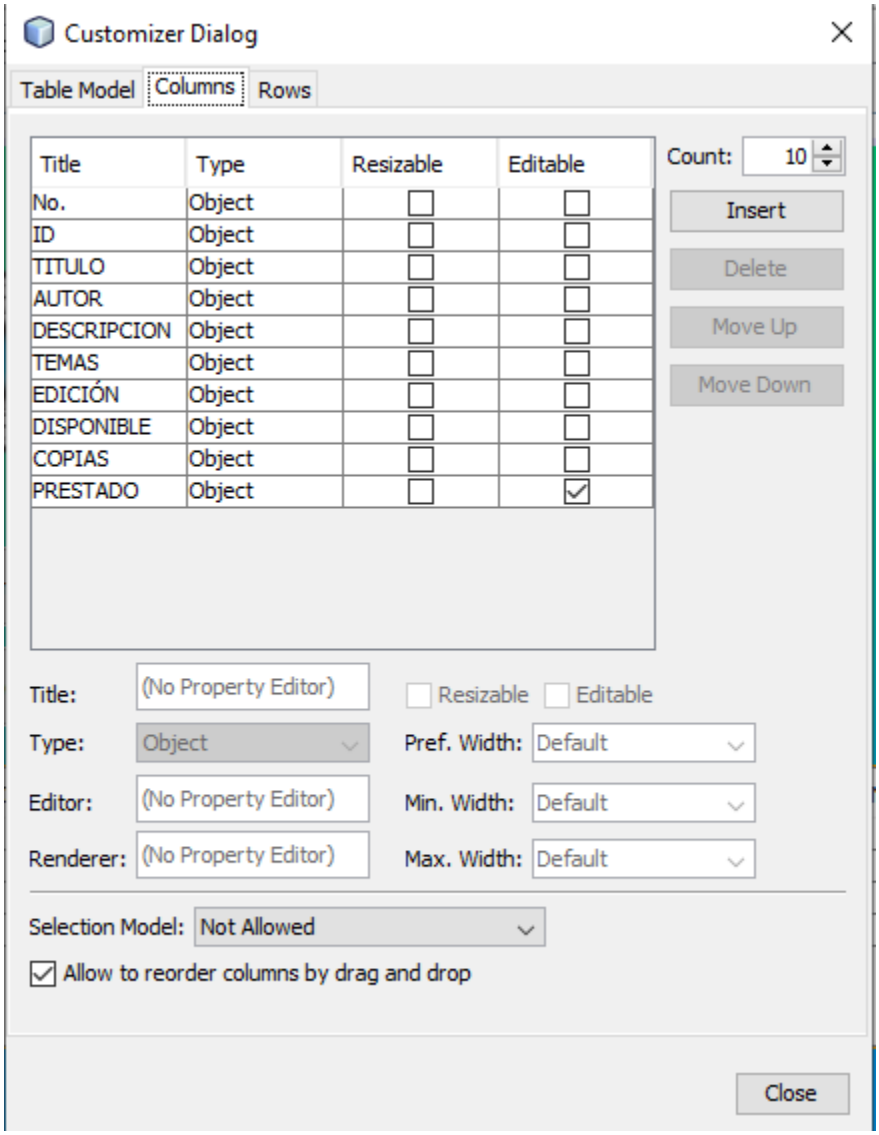


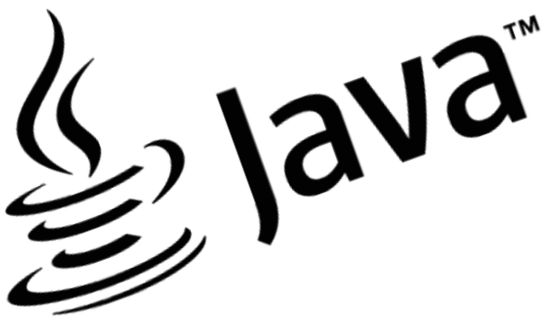
METODO QUE SE UTILIZA PARA LA CARGA MASIVA

```
private void jbtncargaMouseClicked(java.awt.event.MouseEvent evt) {
      //Separando las filas
      String filas[] = jtxareaDatos.getText().split("\n");
      int columnas = filas[0].split(";").length;
      //String datos[] = filas[columnas].split(";");

      for (int i = 0; i < filas.length; i++) {

         String datos[] = filas[i].split(";");

         for (int j = 0; j < 1; j++) {
            if (Integer.parseInt(datos[0]) == 0) {
               System.out.println("Esto es un libro");
               String palabra[] = datos[6].split(",");
               String tema[] = datos[8].split(",");
               biblioteca.Biblioteca.libros[biblioteca.Biblioteca.contlibros] = new Libro(Integer.parseInt(datos[9]),
Integer.parseInt(datos[13]), Integer.parseInt(datos[3]), 0, Integer.parseInt(datos[2]), datos[1], datos[7],
Integer.parseInt(datos[5]), palabra, tema, datos[4]);
               biblioteca.Biblioteca.contlibros++;

            } else if (Integer.parseInt(datos[0]) == 1) {
               System.out.println("Esto es una revista");
               String palabra[] = datos[6].split(",");
               String tema[] = datos[8].split(",");
               biblioteca.Biblioteca.revistas[biblioteca.Biblioteca.contrevistas] = new
Revista(/*Integer.parseInt(datos[3])*/Biblioteca.contrevistas, datos[10], Integer.parseInt(datos[9]),
Integer.parseInt(datos[13]), Integer.parseInt(datos[11]), 1, Integer.parseInt(datos[2]), datos[1], datos[7],
Integer.parseInt(datos[5]), palabra, tema, datos[4]);
               biblioteca.Biblioteca.contrevistas++;

            } else if (Integer.parseInt(datos[0]) == 2) {
               System.out.println("Esto en una tesis");
               String palabra[] = datos[6].split(",");
            String tema[] = datos[8].split(",");
            biblioteca.Biblioteca.tesis[biblioteca.Biblioteca.conttesis] = new
Tesis(/*Integer.parseInt(datos[3])*/Biblioteca.conttesis,  Integer.parseInt(datos[9]),
Integer.parseInt(datos[13]),datos[12], 2, Integer.parseInt(datos[2]), datos[1], datos[7], Integer.parseInt(datos[5]),
palabra, tema, datos[4]);
            biblioteca.Biblioteca.conttesis++;
            }

         }
         System.out.println();
      }
```

```
        JOptionPane.showMessageDialog(null, "Se cargaron: \t \n"+ "\t\t " + biblioteca.Biblioteca.contlibros + " Libros "
+"\n" + "\t\t "+biblioteca.Biblioteca.contrevistas+ " Revistas " +"\n" + "\t\t "+biblioteca.Biblioteca.conttesis+ "  Tesis " ,
"Documentos agregados correctamente...", JOptionPane.INFORMATION_MESSAGE);
            this.setVisible(false);

        // TODO add your handling code here:
    }
```

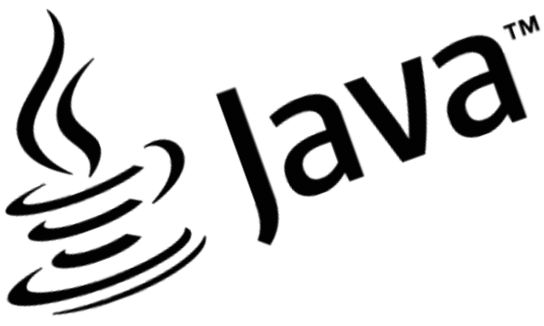
CONFIGURACIÓN GENERAL EN TABLAS





METODO PARA EL FILTRO DE DOCUMENTOS EN LA TABLA

```
private void jbtfiltroMouseClicked(java.awt.event.MouseEvent evt) {

     if (((String) jcbxfiltro.getSelectedItem()).equalsIgnoreCase("-")){
    JOptionPane.showMessageDialog(null, "Seleccione un filtro de documento...", "Error documento", JOptionPane.WARNING_MESSAGE);

     }else if (((String) jcbxfiltro.getSelectedItem()).equalsIgnoreCase("Libro")) {

         String  tablaLibro[][] = new String [biblioteca.Biblioteca.contlibros][10];

    for (int i = 0; i < biblioteca.Biblioteca.contlibros; i++) {
        Boolean prestado = false;
        tablaLibro[i][0]= (Integer.toString(i));
        tablaLibro[i][1]=biblioteca.Biblioteca.libros[i].getIsbn().toString();
        tablaLibro[i][2]=biblioteca.Biblioteca.libros[i].getTitulo();
        tablaLibro[i][3]=biblioteca.Biblioteca.libros[i].getAutor();
        tablaLibro[i][4]=biblioteca.Biblioteca.libros[i].getDescripcion();
        tablaLibro[i][5]=Arrays.toString(biblioteca.Biblioteca.libros[i].getTemas());
        tablaLibro[i][6]=biblioteca.Biblioteca.libros[i].getEdicion().toString();
        tablaLibro[i][7]=biblioteca.Biblioteca.libros[i].getDisponibles().toString();
        tablaLibro[i][8]=biblioteca.Biblioteca.libros[i].getCopias().toString();
    }
    jTablel.setModel(new javax.swing.table.DefaultTableModel(
        tablaLibro,
        new String [] {
            "No.", "ID", "TITULO", "AUTOR", "DESCRIPCION", "TEMAS", "EDICIÓN", "DISPONIBLE", "COPIAS", "PRESTADO"
        }));
```



ENLACE DEL REPOSITORIO

https://github.com/lasicajar/IPC1-Proyecto1_201602630