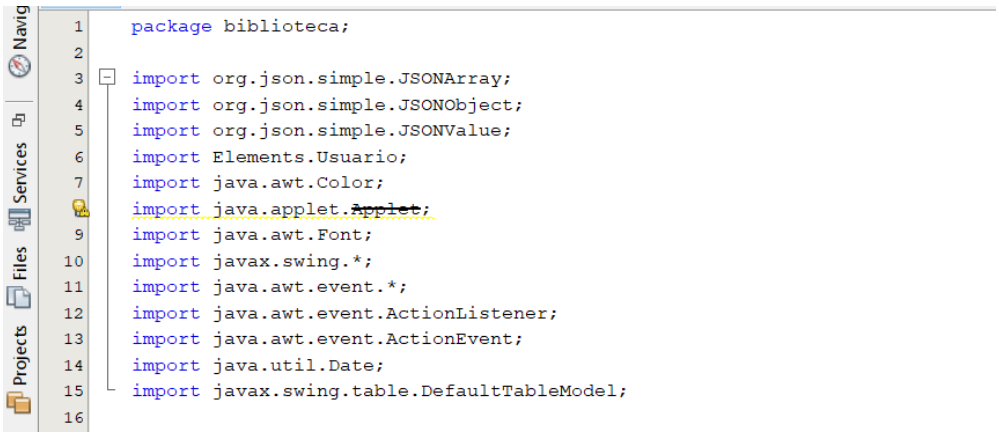
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

MANUAL TECNICO

LOGIN

Librerías a utilizar

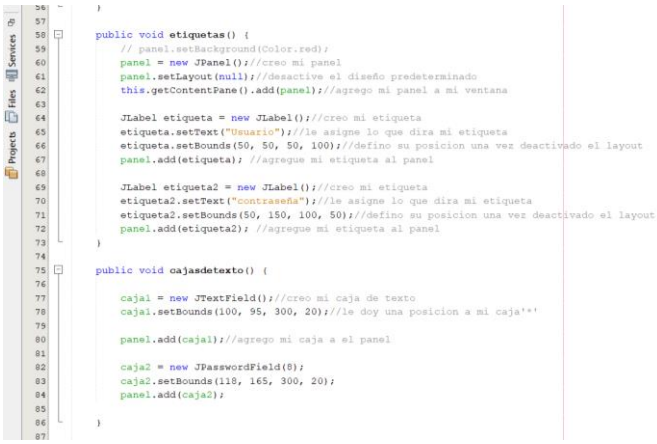


```
package biblioteca;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
import Elements.Usuario;
import java.awt.Color;
import java.applet.Applet;
import java.awt.Font;
import javax.swing.*;
import java.awt.event.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.util.Date;
import javax.swing.table.DefaultTableModel;
```

Ventana a utilizar donde creare la interfaz grafica

```
private void ventana() {

    this.setSize(500, 500);
    setTitle("LOGIN");//nombre de la ventana
    setLocationRelativeTo(null);//centre la ventana
    etiquetas();// made a llamar el metodo panel
    setDefaultCloseOperation(EXIT ON CLOSE);
}
```

Algunos métodos importantes



```
public void etiquetas() {
    // panel.setBackground(Color.red);
    panel = new JPanel();//creo mi panel
    panel.setLayout(null);//desactive el diseño predeterminado
    this.getContentPane().add(panel);//agrego mi panel a mi ventana

    JLabel etiqueta = new JLabel();//creo mi etiqueta
    etiqueta.setText("Usuario");//le asigne lo que dira mi etiqueta
    etiqueta.setBounds(50, 50, 50, 100);//defino su posicion una vez deactivado el layout
    panel.add(etiqueta); //agregue mi etiqueta al panel

    JLabel etiqueta2 = new JLabel();//creo mi etiqueta
    etiqueta2.setText("contraseña");//le asigne lo que dira mi etiqueta
    etiqueta2.setBounds(50, 150, 100, 50);//defino su posicion una vez deactivado el layout
    panel.add(etiqueta2); //agregue mi etiqueta al panel
}

public void cajasdetexto() {

    caja1 = new JTextField();//creo mi caja de texto
    caja1.setBounds(100, 95, 300, 20);//le doy una posicion a mi caja'*'

    panel.add(caja1);//agrego mi caja a el panel

    caja2 = new JPasswordField(8);
    caja2.setBounds(118, 165, 300, 20);
    panel.add(caja2);

}
```

Creacion del boton 1

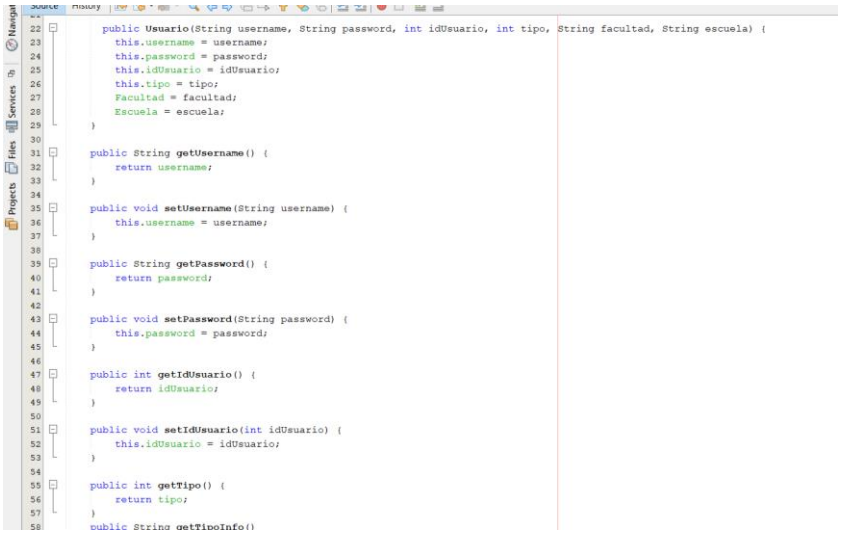
```
boton1 = new JButton();// creo mi boton
boton1.setText("Iniciar sesion");//le asigno el texto que tendra mi boton
boton1.setBounds(180, 250, 150, 30);//le doy posicion a mi boton
boton1.setBackground(Color.white);
boton1.setFont(new Font("Ubuntu", Font.BOLD, 15));
panel.add(boton1);//añado mi boton a el panel
boton1.addActionListener(this);//le permitimos al boton 1 er escuchado en el metodo
boton1.addActionListener(new ActionListener() {
```

Con este botón podremos iniciar sesión una vez comprobado que tenemos permiso de administrador, de lo contrario no se nos permitirá el ingreso.

```
nombre = caja1.getText();

reportes ventana = new reportes();

String usuario = caja1.getText();
String passwords = caja2.getText();
//        JOptionPane.showMessageDialog(null, "aqui añadire una accion");

usuario_logeado = new Usuario();
for (Usuario user : usuarios) {
    if (user != null) {
        System.out.println(user.getUsername().trim().equals(usuario.trim()));
        System.out.println(user.getPassword().trim().equals(passwords.trim()));
        if (user.getUsername().trim().equals(usuario.trim()) && user.getPassword().trim().equals(passwords.trim())) {
            usuario_logeado = user;
            break;
        }
    }
}
caja1.setText("");
caja2.setText("");
if (usuario_logeado.getIdUsuario() != -1) {
    //&& (caja1.getText().equals("D123") && caja2.getText().equals("123"))
    System.out.println("ID Usuario: " + usuario_logeado.getIdUsuario());
    if ((usuario_logeado.getTipo() == 1)) {
        clase3 obj2 = new clase3();
        obj2.setVisible(true);
        clase2.this.setVisible(false);

    } else {
        JOptionPane.showMessageDialog(null, "Usuario Normal - Acceso Restringido", "Su usuario no posee los permisos necesarios", JOptionPane.INFORMATION_MESSAGE
    }
} else {
    JOptionPane.showMessageDialog(null, "No Existe un usuario con las credenciales indicadas", "Error al Iniciar Sesion", JOptionPane.ERROR_MESSAGE);
}
```

Podemos observar el código que tendrá la acción al ingresar, lo sabremos mediante la clase usuarios que es donde se almacenaran nuestros usuarios ingresados en carga masiva

Clase usuarios:



```
public Usuario(String username, String password, int idUsuario, int tipo, String facultad, String escuela) {
    this.username = username;
    this.password = password;
    this.idUsuario = idUsuario;
    this.tipo = tipo;
    Facultad = facultad;
    Escuela = escuela;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public int getIdUsuario() {
    return idUsuario;
}

public void setIdUsuario(int idUsuario) {
    this.idUsuario = idUsuario;
}

public int getTipo() {
    return tipo;
}
public String getTipoInfo()
```

Carga masiva de login

Reconocera le formato json debido a la librería Json-simple-111, se desplegara un JTextarea en donde podremos ingresar los usuarios de manera masiva, dado así la facilidad al usuario de no ingresarlo manualmente, este posee un JButton que le permitirá ingresarlos.

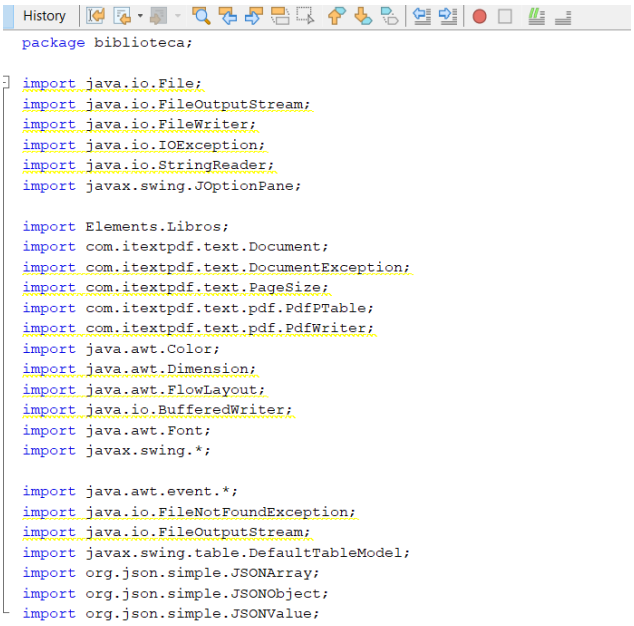
Posee un contador que controlara que no se puedan ingresar mas de 50 usuarios, esta inicializado en 0 y maneja su conteo de forma creciente con forme el numero de registros que se añaden al sistema.

```
@Override
public void actionPerformed(ActionEvent e) {
    JFrame pantalla_masiva = new JFrame("Carga Masiva");
    pantalla_masiva.setSize(300, 300);
    pantalla_masiva.setBackground(Color.white);
    pantalla_masiva.setLayout(null);
    pantalla_masiva.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent windowEvent) {
            pantalla_masiva.setVisible(false);
        }
    });

    JTextArea datos_c = new JTextArea(20, 20);
    datos_c.setEditable(true);
    datos_c.setVisible(true);
    JScrollPane scroll = new JScrollPane(datos_c);
    scroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
    scroll.setBounds(10, 10, 270, 200);
    scroll.setVisible(true);
//
    JButton cargar = new JButton("Cargar");
    cargar.setBounds(180, 220, 100, 30);
    cargar.setBackground(Color.white);
    cargar.setFont(new Font("Ubuntu", Font.BOLD, 15));
    cargar.setBorder(BorderFactory.createLineBorder(Color.black));
    cargar.setVisible(true);
    cargar.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            if (counter < usuarios.length) {
                String datos_carga = datos_c.getText();

                Object jsonObyeto = JSONValue.parse(datos_carga);
                JSONObject obyeto = (JSONObject) jsonObyeto;
```

```
Object jsonObyeto = JSONValue.parse(datos_carga);
JSONObject obyeto = (JSONObject) jsonObyeto;

Object jsonarrayobyeto = obyeto.get("Usuarios");
JSONArray arrayobyeto = (JSONArray) jsonarrayobyeto;
for (Object obyeto_inarray : arrayobyeto) {
    if (counter == usuarios.length) {
        break;
    }
    JSONObject obyeto_value = (JSONObject) obyeto_inarray;
    Usuario nuevo_usuario = new Usuario();

    nuevo_usuario.setUsername((String) obyeto_value.get("Usuario"));
    nuevo_usuario.setPassword((String) obyeto_value.get("Password"));
    nuevo_usuario.setEscuela((String) obyeto_value.get("Carrera"));
    nuevo_usuario.setFacultad((String) obyeto_value.get("Facultad"));
    Long value_tipo = (Long) obyeto_value.get("Tipo");
    nuevo_usuario.setTipo(value_tipo.intValue());
    Long value_id = (Long) obyeto_value.get("ID");
    nuevo_usuario.setIdUsuario(value_id.intValue());

    System.out.println("ID: " + (Long) obyeto_value.get("ID"));
    System.out.println("Usuario: " + (String) obyeto_value.get("Usuario"));
    System.out.println("Password: " + (String) obyeto_value.get("Password"));
    System.out.println("Tipo: " + (Long) obyeto_value.get("Tipo"));
    System.out.println("Facultad: " + (String) obyeto_value.get("Facultad"));
    System.out.println("Carrera: " + (String) obyeto_value.get("Carrera"));
    System.out.println("-----------------------------------");
    int idlu = Integer.parseInt(value_id.toString());//aqui convierto el strig a int y obten
    //  Object[] newr ={(Long) obyeto_value.get("ID"),(String) obyeto_value.get("Usuario"),(

    usuarios[counter] = nuevo_usuario;

    arrtu[counter] = idlu;

    counter++;
}
```

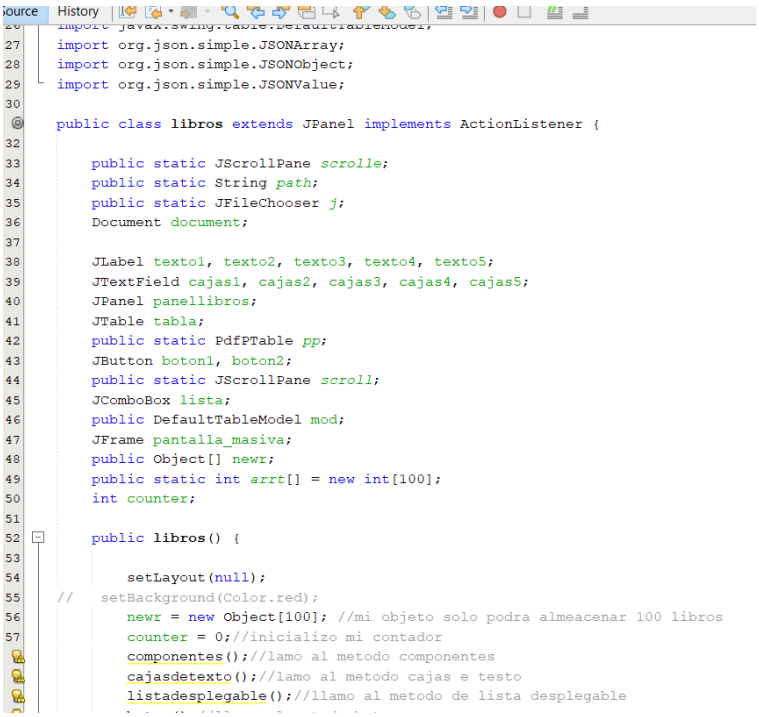
PESTAÑA LIBROS

Librerias utilizadas



```
History
package biblioteca;

import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.StringReader;
import javax.swing.JOptionPane;

import Elements.Libros;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.io.BufferedWriter;
import java.awt.Font;
import javax.swing.*;

import java.awt.event.*;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import javax.swing.table.DefaultTableModel;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
```

Atributos del la clase



```
ource  History
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;

public class libros extends JPanel implements ActionListener {

    public static JScrollPane scrolle;
    public static String path;
    public static JFileChooser j;
    Document document;

    JLabel texto1, texto2, texto3, texto4, texto5;
    JTextField cajas1, cajas2, cajas3, cajas4, cajas5;
    JPanel panellibros;
    JTable tabla;
    public static PdfPTable pp;
    JButton boton1, boton2;
    public static JScrollPane scroll;
    JComboBox lista;
    public DefaultTableModel mod;
    JFrame pantalla_masiva;
    public Object[] newr;
    public static int arrt[] = new int[100];
    int counter;

    public libros() {

        setLayout(null);
//   setBackground(Color.red);
        newr = new Object[100]; //mi objeto solo podra almeacenar 100 libros
        counter = 0;//inicializo mi contador
        componentes();//lamo al metodo componentes
        cajasdetexto();//lamo al metodo cajas e testo
        listadesplegable();//llamo al metodo de lista desplegable
```

Código de tabla donde se mostraran los registros de cada libro

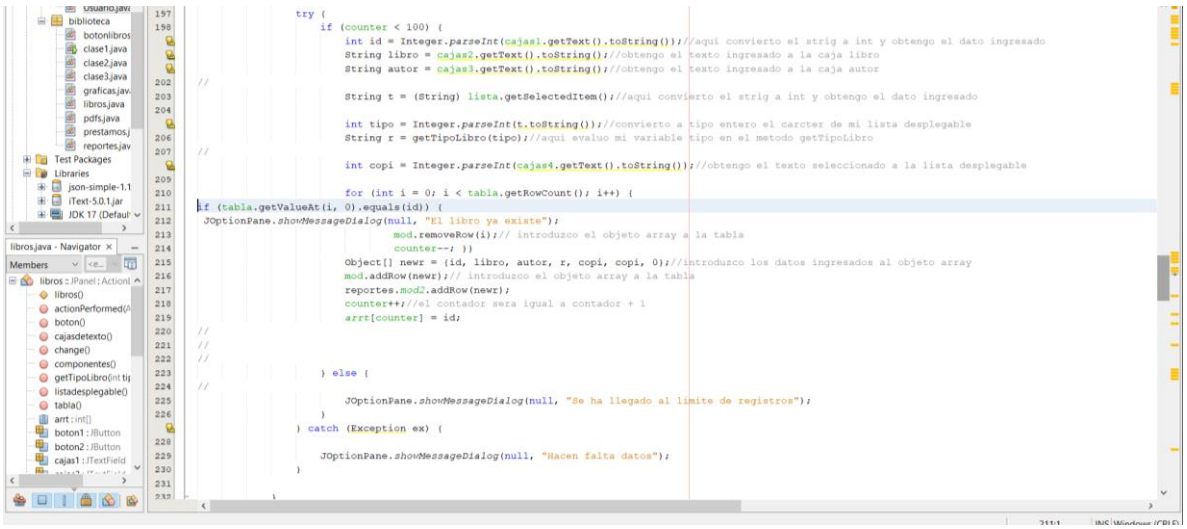
Aquí podremos visualizar cada uno de los datos ingresados para que se nos sea mas fácil poder llevar un control sobre ellos.

```
public void tabla() {

    mod = new DefaultTableModel(); // defino el modelo de mi tabla

    mod.addColumn("ID Libro");//agrego los titulos de mi tabla
    mod.addColumn("Nombre Libro");
    mod.addColumn("Autor");
    mod.addColumn("Tipo");
    mod.addColumn("Copias");
    mod.addColumn("Disponibles");
    mod.addColumn("Ocupados");

    tabla = new JTable(mod);

    scroll = new JScrollPane(tabla);
    tabla.setBounds(250, 40, 800, 400);
    scroll.setBounds(250, 40, 800, 400);

    add(scroll);
```

Código de registro

Podemos observar que cada registro ira aumentando un contador que se encuentra inicializado como 0, esto con el fin de que al momento de llegar a 100 el programa ya no acepte mas registros.

Se cuenta con un arreglo que va añadiendo cada uno de los ID registrados, esto con el fin de no permitir que el usuario pueda repetir algún ID de esta pestaña y poder llevar un mejor control.



Código de carga masiva de la pestaña libros

Es la misma metodología usada en la ventana login, se nos muestra un JTextArea en el que podremos añadir de forma masiva los datos de los libros que se desean registrar, siempre y cuando no exedan la cantidad establecida que permiten los registros del sistema, estos serán posteriormente enviados a la clase Libros donde estarán almacenados y poderlos utilizar en futuros procesos.

```
JTextArea datos_c = new JTextArea(30, 30);// creo mi ventana de texto
datos_c.setEditable(true);//le permitimos ser editable
datos_c.setVisible(true);//le permitimos ser visible
scrolle = new JScrollPane(datos_c);//creo mi panel scroll y se lo asigno a mi area de texto
scrolle.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);//definimos el estilo del scroll
scrolle.setBounds(10, 10, 270, 200);//le damos una posicion al scroll
scrolle.setVisible(true);//le permitimos ser visible al scrolle

JButton cargar = new JButton("Cargar");//creo mi boton el cual tendra el nombre "cargar "
cargar.setBounds(180, 220, 100, 30);//le damos una posicion al boton
cargar.setBackground(Color.white);//le damos un estilo y color al boton
cargar.setFont(new Font("Ubuntu", Font.BOLD, 15));
cargar.setBorder(BorderFactory.createLineBorder(Color.black));//le damos un interlineado mas grueso a el borde del boton y las lertras
cargar.setVisible(true);//le permitimos ser visible al boton
cargar.addActionListener(new ActionListener() {//le permitimos al boton compezar una accion
    @Override
    public void actionPerformed(ActionEvent e) {//compezamos a declarar la accion que tendra el boton
        if (counter < 100) {
            String datos_carga = datos_c.getText();

            Object jsonObyeto = JSONValue.parse(datos_carga); //obtengo la informacion
            JSONObject obyeto = (JSONObject) jsonObyeto; //se la doy a mi objeto

            Object jsonarrayobyeto = obyeto.get("Libros");//ccedo a 1 clave
            JSONArray arrayobyeto = (JSONArray) jsonarrayobyeto;
            for (Object obyeto_inarray : arrayobyeto) {
                JSONObject obyeto_value = (JSONObject) obyeto_inarray;
                Libros libro_nuevo = new Libros();
                Long id_libro = (Long) obyeto_value.get("ID");
                libro_nuevo.idLibro = id_libro.intValue();
                libro_nuevo.titulo = (String) obyeto_value.get("Titulo");
                libro_nuevo.autor = (String) obyeto_value.get("Autor");

                Long disponible = (Long) obyeto_value.get("Disponible");
                libro_nuevo.disponible = id_libro.intValue();
```

```
                libro_nuevo.copias = id_libro.intValue();

                Long ocupados = (Long) obyeto_value.get("Ocupados");
                libro_nuevo.ocupados = id_libro.intValue();

                Long Tipo = (Long) obyeto_value.get("Tipo");
                libro_nuevo.tipo = id_libro.intValue();

                Object[] values = (String.valueOf(libro_nuevo.idLibro), libro_nuevo.titulo, libro_nuevo.autor, getTipoLibro(libro_nuevo.tipo

                mod.addRow(values);
                reportes.mod2.addRow(values);

                counter++;//el contador sera igual a contador +1
                arrt[counter] = libro_nuevo.idLibro;

            }
            pantalla_masiva.setVisible(false);
            change();

        } else {

            JOptionPane.showMessageDialog(null, "LIMITE DE REGISTROS");//de lo contrario aparecera una ventana emergente con el siguiente me

        }

    }

));

pantalla_masiva.add(scrolle);//la ventana scrolle se añadira a la pantalla de carga masiva
pantalla_masiva.add(cargar);//el boton se añadira a la pantalla de carga masiva
pantalla_masiva.setLocationRelativeTo(null); //Center screen
pantalla_masiva.setVisible(true);//permitiremos que la pantalla aparezca

}
```

PRESTAMOS

Librerías a utilizar

```
import Elements.Prestamos;
import biblioteca.clase2;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Font;
import javax.swing.*;
import java.awt.event.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.swing.table.DefaultTableModel;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import biblioteca.libros;
public class prestamos extends JPanel implements ActionListener {
```

Atributos de la clase

```
public class prestamos extends JPanel implements ActionListener {
  Date fechaingresada;
    JLabel texto1 ;
         JLabel texto2;
         JScrollPane scrolle;
          JScrollPane scroll;
          JFrame pantalla_masiva;
         JLabel texto3;
         JLabel texto4;
         JComboBox lista1 ;
    JTextField caj1,caj2,caj3;
       JTable tablap;

    DefaultTableModel mod;
       JButton bot,boton2;
    Object[] newr;

    int counter;
```

Tabla de prestamos

Ene esta tabla podremos visualizar los prestamos realizados en el sistema, cuenta con un contador donde por cada registro hecho el contador ira aumentando hasta completar los 100 registros, esto con el fin de no saturar el sistema.

```
}
   public void componentespresta(){
         texto1 = new JLabel(); //creo el objeto de mi etiqueta
          texto2 = new JLabel();
          texto3 = new JLabel();

     texto4 = new JLabel();
texto1.setText("Usuario");//le asigno el texto que tendra mi etiqueta
texto1.setBounds(10, 10, 50, 20);//le asigno la posicion que tendra mi etiqueta
texto2.setText("Libro");//le asigno el texto que tendra mi etiqueta
texto2.setBounds(10, 70, 50, 20);//le asigno la posicion que tendra mi etiqueta
texto3.setText("Fecha ");//le asigno el texto que tendra mi etiqueta
texto3.setBounds(10, 130, 90, 20);//le asigno la posicion que tendra mi etiqueta
texto4.setText("Entrega ");//le asigno el texto que tendra mi etiqueta
texto4.setBounds(10, 144, 90, 20);//le asigno la posicion que tendra mi etiqueta

add(texto1);//le añado el texto 1 a mi etiqueta
add(texto2);//le añado el texto 1 a mi etiqueta
add(texto3);//le añado el texto 1 a mi etiqueta
add(texto4);
}
```

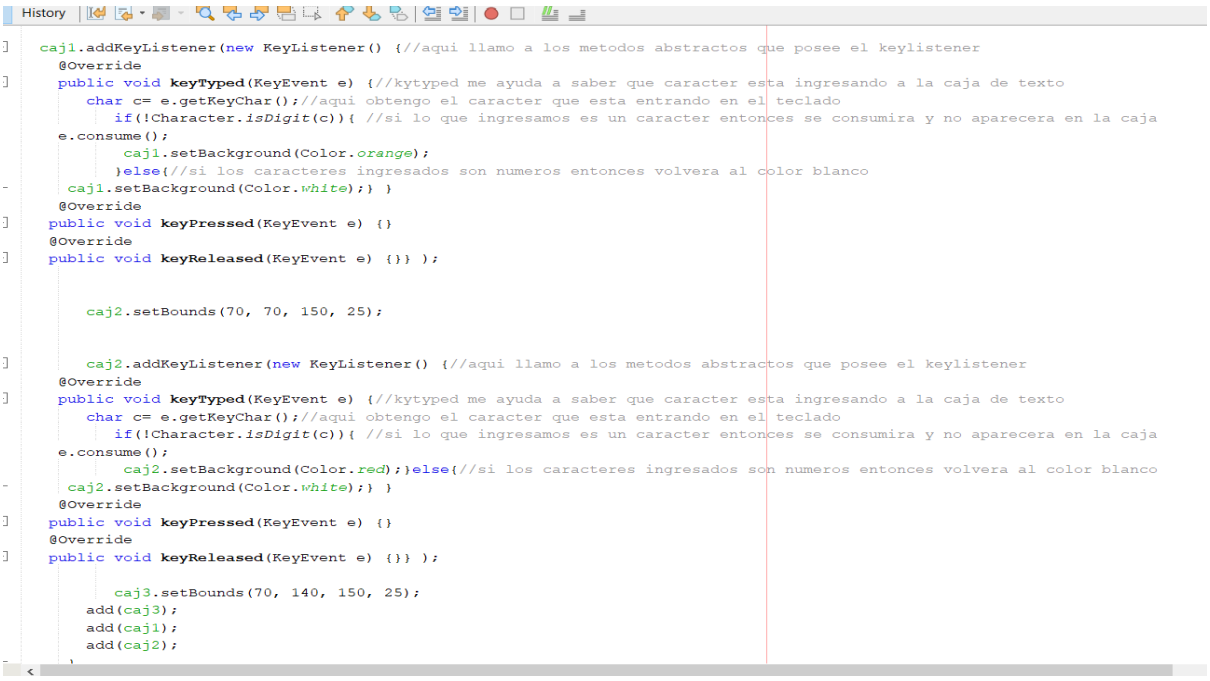
Código del botón registra préstamo

El código se encargara de enviar la información dada en el JtextFile a la tabla, se ha utilizado un vector donde se almacenan los ID de usuarios y libros, si estos no han sido registrados anteriormente el programa no hará el registro de préstamo, también se cuenta con un con un formato Date, esto verificara si el libro a sido entregado o prestado, utilizando un try y catch se cuidara que el usuario no ingrese un formato de fecha errónea, para que no hayan complicaciones en el programa al momento de analizar la fecha ingresada.

```
          try{
      if(counter<100){
  int idl = Integer.parseInt(caj1.getText().toString());//aqui convierto el strig a int y obtengo el dato ingresado

    int ide = Integer.parseInt(caj2.getText().toString());//obtengo el texto seleccionado a la lista desplegable
    String fecha= caj3.getText().toString();//obtengo el texto ingresado a la caja autor


     SimpleDateFormat fechh = new SimpleDateFormat("dd/MM/YYYY");
             Date fechahoy = new Date();
             libros obg = new libros();
             clase2 usu = new clase2();
    Date fechaconv = conversordate(fecha);
     for(int i=0; i<obg.arrt.length;i++){
                                    for(int j=0; j<usu.arrtu.length;j++){
                                   if(obg.arrt[i]==ide){
                                        if(usu.arrtu[j]==idl){
   if(fechaconv != null){
      if(fechahoy.compareTo(fechaconv)<= 0){
       Object[] newr ={idl,ide,stringadate(fechaconv),"prestado"};//introduzco los datos ingresados al objeto array
mod.addRow(newr);// introduzco el objeto array a la tabla
 reportes.mod3.addRow(newr);
counter++;//el contador sera igual a contador + 1
     }else{ Object[] newr ={idl,ide,stringadate(fechaconv),"entregado"};//introduzco los datos ingresados al objeto ar
mod.addRow(newr);// introduzco el objeto array a la tabla
 reportes.mod3.addRow(newr);
counter++;//el contador sera igual a contador + 1
     } }else{
      JOptionPane.showMessageDialog(null, "El formato es dd/mm/yyyy");
 }

                                    }) }  }  }else {
  JOptionPane.showMessageDialog(null, "LIMITE DE REGISTROS");
 }}catch(Exception ex){JOptionPane.showMessageDialog(null, "Hacen falta datos");} } });
```

No se pueden ingresar caracteres en algunos JTextfile.

Con ayuda de KeyTyped podemos asegurarnos que el usuario no ingresara un carácter en los JtextFile de el programa como en cantidad de libros o ID, esto nos ayudara a que el programa no cometa ningún error si se diera un caso como ese.



```
History
caj1.addKeyListener(new KeyListener() {//aqui llamo a los metodos abstractos que posee el keylistener
  @Override
  public void keyTyped(KeyEvent e) {//kytyped me ayuda a saber que caracter esta ingresando a la caja de texto
     char c= e.getKeyChar();//aqui obtengo el caracter que esta entrando en el teclado
       if(!Character.isDigit(c)){ //si lo que ingresamos es un caracter entonces se consumira y no aparecera en la caja
  e.consume();
        caj1.setBackground(Color.orange);
       }else{//si los caracteres ingresados son numeros entonces volvera al color blanco
   caj1.setBackground(Color.white);} }
  @Override
  public void keyPressed(KeyEvent e) {}
  @Override
  public void keyReleased(KeyEvent e) {}} );


     caj2.setBounds(70, 70, 150, 25);


     caj2.addKeyListener(new KeyListener() {//aqui llamo a los metodos abstractos que posee el keylistener
  @Override
  public void keyTyped(KeyEvent e) {//kytyped me ayuda a saber que caracter esta ingresando a la caja de texto
     char c= e.getKeyChar();//aqui obtengo el caracter que esta entrando en el teclado
       if(!Character.isDigit(c)){ //si lo que ingresamos es un caracter entonces se consumira y no aparecera en la caja
  e.consume();
        caj2.setBackground(Color.red);}else{//si los caracteres ingresados son numeros entonces volvera al color blanco
   caj2.setBackground(Color.white);} }
  @Override
  public void keyPressed(KeyEvent e) {}
  @Override
  public void keyReleased(KeyEvent e) {}} );

       caj3.setBounds(70, 140, 150, 25);
     add(caj3);
     add(caj1);
     add(caj2);
  }
```

Carga masiva de la pestaña prestamos

Al presionar el botón nombrado como carga masiva, se pondrá en pantalla un JTextArea en donde podremos ingresar de foma masiva la cantidad de libros que se desea prestar, todo esto siempre en formato Json identificando así cada uno de los datos que se requieren para realizar el préstamo, se cuenta con un objeto que conecta con la clase Prestamos en donde se enviaran los datos almacenados con el Json.

```
                    public void windowClosing(WindowEvent windowEvent){
                       pantalla_masiva.setVisible(false);
                    }
                 });

                JTextArea datos_c = new JTextArea(30,30);
                datos_c.setEditable(true);
                datos_c.setVisible(true);
                 scrolle = new JScrollPane(datos_c);
                scrolle.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
                scrolle.setBounds(10,10,270,200);
                scrolle.setVisible(true);

    JButton cargar = new JButton("Cargar");
                cargar.setBounds(180, 220, 100, 30);
                cargar.setBackground(Color.white);
                cargar.setFont(new Font("Ubuntu", Font.BOLD, 15));
                cargar.setBorder(BorderFactory.createLineBorder(Color.black));
                cargar.setVisible(true);
  cargar.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e) {
                        if(counter<100){
                            String datos_carga = datos_c.getText();

                            Object jsonObyeto = JSONValue.parse(datos_carga); //obtengo la informacion
                            JSONObject obyeto = (JSONObject) jsonObyeto; //se la doy a mi objeto

                            Object jsonarrayobyeto = obyeto.get("Prestamos");//accedo a l clave
                            JSONArray arrayobyeto = (JSONArray) jsonarrayobyeto;
                            for(Object obyeto_inarray: arrayobyeto)
                            {
                                JSONObject obyeto_value = (JSONObject) obyeto_inarray;
Prestamos prestamo_nuevo=new Prestamos();
```

Verificador de fecha

Utilizando un if podremos verificar si la fecha que se desea almacenar en la tabla es de distinto formato al que hemos establecido, si se cumple con esto se procede a verificar si la fecha ya establecida como formato Date es mayor o menor a la fecha actual, si es menor en el objeto que se enviara a la tabla, introduciremos el “entregado ” en la columna status y la fila que se utilizara, de lo contrario enviaremos un “prestado”.

```
if(fechaconv != null){
    if(fechahoy.compareTo(fechaconv)<= 0){
     Object[] newr ={String.valueOf(prestamo_nuevo.idLibro),String.valueOf(prestamo_nuevo.idUsuario),stringadate(fechaconv),"prestado"};//introduzco
mod.addRow(newr);// introduzco el objeto array a la tabla
 reportes.mod3.addRow(newr);
counter++;//el contador sera igual a contador + 1
    }else{ Object[] newr ={String.valueOf(prestamo_nuevo.idLibro),String.valueOf(prestamo_nuevo.idUsuario),stringadate(fechaconv),"entregado"};//intr
mod.addRow(newr);// introduzco el objeto array a la tabla
 reportes.mod3.addRow(newr);
counter++;//el contador sera igual a contador + 1
    } }else{
     JOptionPane.showMessageDialog(null, "El formato es dd/mm/yyyy");
 } }
                        pantalla_masiva.setVisible(false);
                        change();}else {
          JOptionPane.showMessageDialog(null, "LIMITE DE REGISTROS");//de lo contrario aparecera una ventana emergente con el siguiente mensaje
 }) });
 pantalla_masiva.add(scrolle);//la ventana scrolle se añadira a la pantalla de carga masiva
            pantalla_masiva.add(cargar);//el boton se añadira a la pantalla de carga masiva
 pantalla_masiva.setLocationRelativeTo(null); //Center screen
            pantalla_masiva.setVisible(true);//permitiremos que la pantalla aparezca
```

Conversor de String a Fecha tipo Date

Si el formato de la fecha introducido en el JTextfile cumple con el formato establecido, lo enviaremos a un conversor donde lo convertiremos a tipo Date, retornando así el valor que introduciremos en nuestro objeto que se enviara a la tabla

```
15
16      public static Date conversordate(String fecha){
17        SimpleDateFormat fech = new SimpleDateFormat("dd/MM/yyyy");
18              Date f =null;
19         try{
20      //
21         f=fech.parse(fecha);
22
23
24         }catch(Exception e){
25      //
26         JOptionPane.showMessageDialog(null, "Formato de fecha no valido");
27         }      return f;
28      }
29      //
30      public static String stringadate(Date fecha){
31       SimpleDateFormat fech = new SimpleDateFormat("dd/MM/yyyy");
32          return fech.format(fecha); }
33      public void change()
34       {
35
36           JTable tabla = new JTable(mod);
37          tabla.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
38           scrolle = new JScrollPane(tabla);
39           scrolle.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
40           scrolle.setBounds(300,10,670,400);
41           scrolle.setVisible(true);
42
43       }
44
45       @Override
46       public void actionPerformed(ActionEvent ent) {
47
48       }
49
```

REPORTES

Librerías a utilizar

```
package biblioteca;

import Elements.Usuario;
import biblioteca.clase2;
import biblioteca.libros;
import static biblioteca.libros.pp;

import biblioteca.prestamos;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileOutputStream;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JButton;
import javax.swing.JComboBox;

import Elements.Libros;
import static biblioteca.libros.scroll;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.io.BufferedWriter;
import java.awt.Font;
import javax.swing.*;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

Atributos de la clase

```
    public static Calendar calendar = new GregorianCalendar();
    Calendar fecha = Calendar.getInstance();
    int segundo = fecha.get(Calendar.SECOND);
      int minuto = fecha.get(Calendar.MINUTE);
        boolean verdad=true;
      int hora = fecha.get(Calendar.HOUR);
  String h = Integer.toString(hora);
  String m = Integer.toString(minuto);
  String s = Integer.toString(segundo);
    public static int getDia(){
          return calendar.get(calendar.DATE);     }
    public static int getMes(){
  return calendar.get(calendar.MONTH);}
    public static int getAño(){

  return calendar.get(calendar.YEAR);
      }
      public static int getHora(){

  return calendar.get(calendar.HOUR);
      }
       public static int getMinuto(){
  return calendar.get(calendar.MINUTE);   }
    JButton boton1;
    JLabel etiqueta;
    JScrollPane scrol;
    JTable tablet;
    JComboBox lista;
    public static PdfPTable pp;
     public static PdfPTable ppu;
    public static PdfPTable pp3;
    public static JFileChooser j;
    public static JFileChooser j3;
public static JFileChooser ju;
    public static DefaultTableModel mod2;
    JTable tabla2;
```

Envió de datos de las tablas de las clases libros y prestamos

Declarando como public static las tablas en reportes, podremos almacenar los datos intrpoducidos en las tablas de las distintas clases que reportaremos, esto con el fin de generar un pdf cuando lo requiramos con los datos actualizados.

```
        mod2.addColumn("ID Libro");//agrego los titulos de mi tabla
        mod2.addColumn("Nombre Libro");
        mod2.addColumn("Autor");
        mod2.addColumn("Tipo");
        mod2.addColumn("Copias");
        mod2.addColumn("Disponibles");
        mod2.addColumn("Ocupados");
        tabla2 = new JTable();
        tabla2.setModel(mod2);
        scroll = new JScrollPane(tabla2);
        tabla2.setBounds(250, 40, 800, 400);
        scroll.setBounds(250, 40, 800, 400);


        mod3 = new DefaultTableModel(); // defino el modelo de mi tabla

        mod3.addColumn("Nombre usuario");//agrego los titulos de mi tabla
        mod3.addColumn("Libro");
        mod3.addColumn("Fecha enrega");
        mod3.addColumn("Status");

        tabla3 = new JTable(mod3);
        scroll = new JScrollPane(tabla3);// añado mi tabla al scroll(permitira que el scrool se ajuste a las nesecidades de la tabla)
        tabla3.setBounds(250, 40, 800, 400);//defino la posicion de mi tabla
        scroll.setBounds(250, 40, 800, 400);//defino la posicion de mi scroll

//        add(scroll);
```

Boton generar

En este botón enviaremos los datos ingresados a un switch donde verificaremos que es lo que se quiere generar

```
    public void botoness() {

        boton1 = new JButton();// creo mi boton
        boton1.setText("Generar");//le asigno el texto que tendra mi boton
        boton1.setBounds(270, 140, 100, 30);//le doy posicion a mi boton
        boton1.setBackground(Color.white);//le damos un estilo y color al boton
        boton1.setFont(new Font("Ubuntu", Font.BOLD, 15));
        add(boton1);
        boton1.addActionListener(this);//le permitimos al boton 1 er escuchado en el metodo
        boton1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent en) {

                String tipo = lista.getSelectedItem().toString();//obtengo el texto ingresado a la caja autor
                String r = getTipoLibro(tipo);



                clase2 object = new clase2();
                SimpleDateFormat fech = new SimpleDateFormat("dd/MM/YYYY");
                Date fecha = new Date();
//
                Object[] newr = (fech.format(fecha), object.nombre, r);//introduzco los datos ingresados al objeto array


                model.addRow(newr);// introduzco el objeto array a la tabla

            }
        });
    }
```

Método Switch

```
libros lll = new libros();

public String getTipoLibro(String tipo) {
    switch (tipo) {
        case "usuarios registrados":

            return "usuarios registrados";

        case "libros registrados":

           fff();
/

            return "libros registrados";

        case "prestamos realizados":
           fgf();
            //JOptionPane.showMessageDialog(null, "ee");
            return "prestamos realizados";
        default:

            return "";

    }

}

public void fff() {
```

Generador de pdf

```
        String ruta = System.getProperty("user.home");
        PdfWriter.getInstance(document, new FileOutputStream(ruta+"/Desktop/"+getDia()+getMes()+getAño()+h+m+s+".pdf"));
        document.open();
        pp = new PdfPTable(7);

        pp.addCell("ID Libro");
        pp.addCell("Nombre Libro");
        pp.addCell("Autor");
        pp.addCell("Tipo");
        pp.addCell("Copias");
        pp.addCell("Disponibles");
        pp.addCell("Ocupados");
        for (int i = 0; i < tabla2.getRowCount(); i++) {

            String id = tabla2.getValueAt(i, 0).toString();
            String nom = tabla2.getValueAt(i, 1).toString();
            String au = tabla2.getValueAt(i, 2).toString();
            String tip = tabla2.getValueAt(i, 3).toString();
            String cop = tabla2.getValueAt(i, 4).toString();
            String dis = tabla2.getValueAt(i, 5).toString();
            String ocu = tabla2.getValueAt(i, 6).toString();


            pp.addCell(id);
            pp.addCell(nom);
            pp.addCell(au);
            pp.addCell(tip);
            pp.addCell(cop);
            pp.addCell(dis);
            pp.addCell(ocu);
        }document.add(pp);
    } catch (Exception epp) {
!OptionPane.showMessageDialog(null, "error");
```