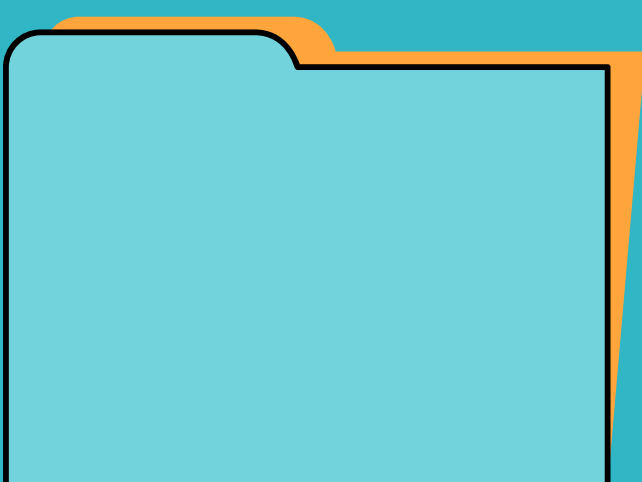


**BIBLIOTECA DIGITAL**

# MANUAL DE USUARIO



**Proyecto1 IPC-D**  
**Kevin Palacios 201902278**

# Indice

I. Introducción

II. Objetivos

III. Dirigido

IV. Especificación Técnica

1.Requisitos de Hardware

2. Requisitos de Software

V. Lógica del Programa

VI. Métodos Utilizados

# I. Introducción

El programa Biblioteca Digital esta diseñado de forma para aquellas personas que quieran analizar y estudiar la construcción del programa . Para poder cumplir con las especificaciones este fue desarrollado en el lenguaje de JAVA debido a que esté se adecuó a las necesidades que requería el uso de la Programación Orientada a Objetos, que sus Siglas en Ingles son POO.

## II. Objetivos

El objetivo principal de este manual es poder ayudar a diversos estudiantes y auxiliares a de catedra de Introducción a la Programación para así poder informarse y guiarse para poder programar, mediante la Programación Orientada a Objetos .

El objetivo de este programa es hacer que los administradores de dicha biblioteca puedan realizar diversas funciones para así facilitar su trabajo.

### **III. DIRIGIDO**

El manual va dirigido a todos aquellos programadores interesados en aprender este paradigma, para así poder diseñar aplicaciones con interfaz gráfica, también va dirigido a aquellos auxiliares que quieren cualificar el conocimiento que tenemos en este tipo de programación.

### **IV. ESPECIFICACIÓN TÉCNICA**

#### **1. Requisitos de Hardware**

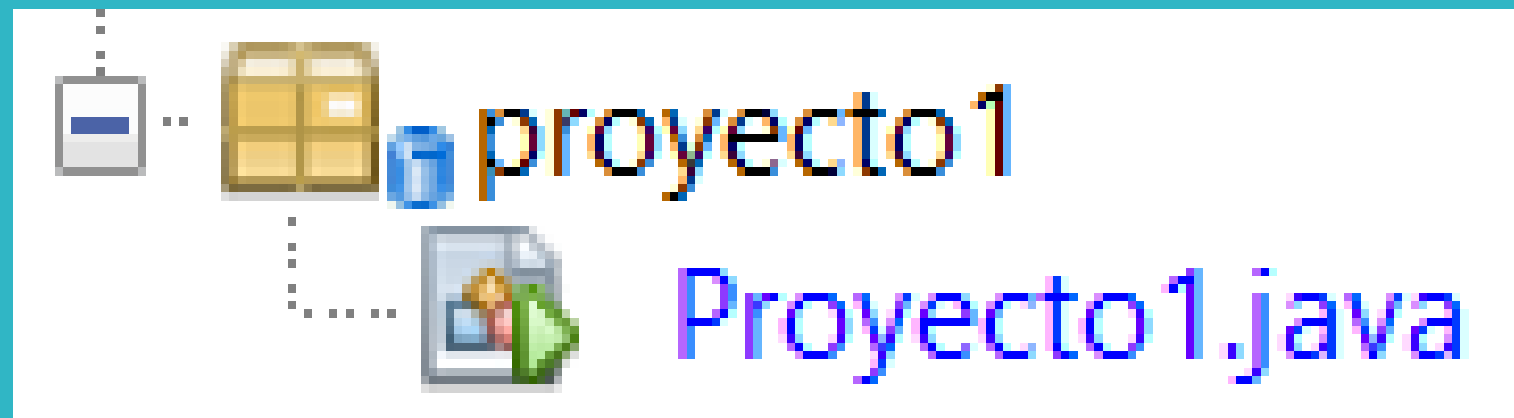
- Computadora de Escritorio o Portatil.
- Mínimo 8GB de Memoria RAM.
- 20 GB disponibles de Disco duro.
- Procesador Core i3 o Superior.
- Procesador a 64 bits
- Pantalla con Resolución Gráfica de 1024\*768 píxeles.

#### **2. Requisitos de Software**

- Tener Instalado Windows 7 o superior
- Tener Instalado Netbeans 12.6
- Libreria Externa Gson 2.2.2
- Libreria Externa iText 5.0.1
- Libreria Interna Javax.AWT
- Libreria Interna Javax.SWING

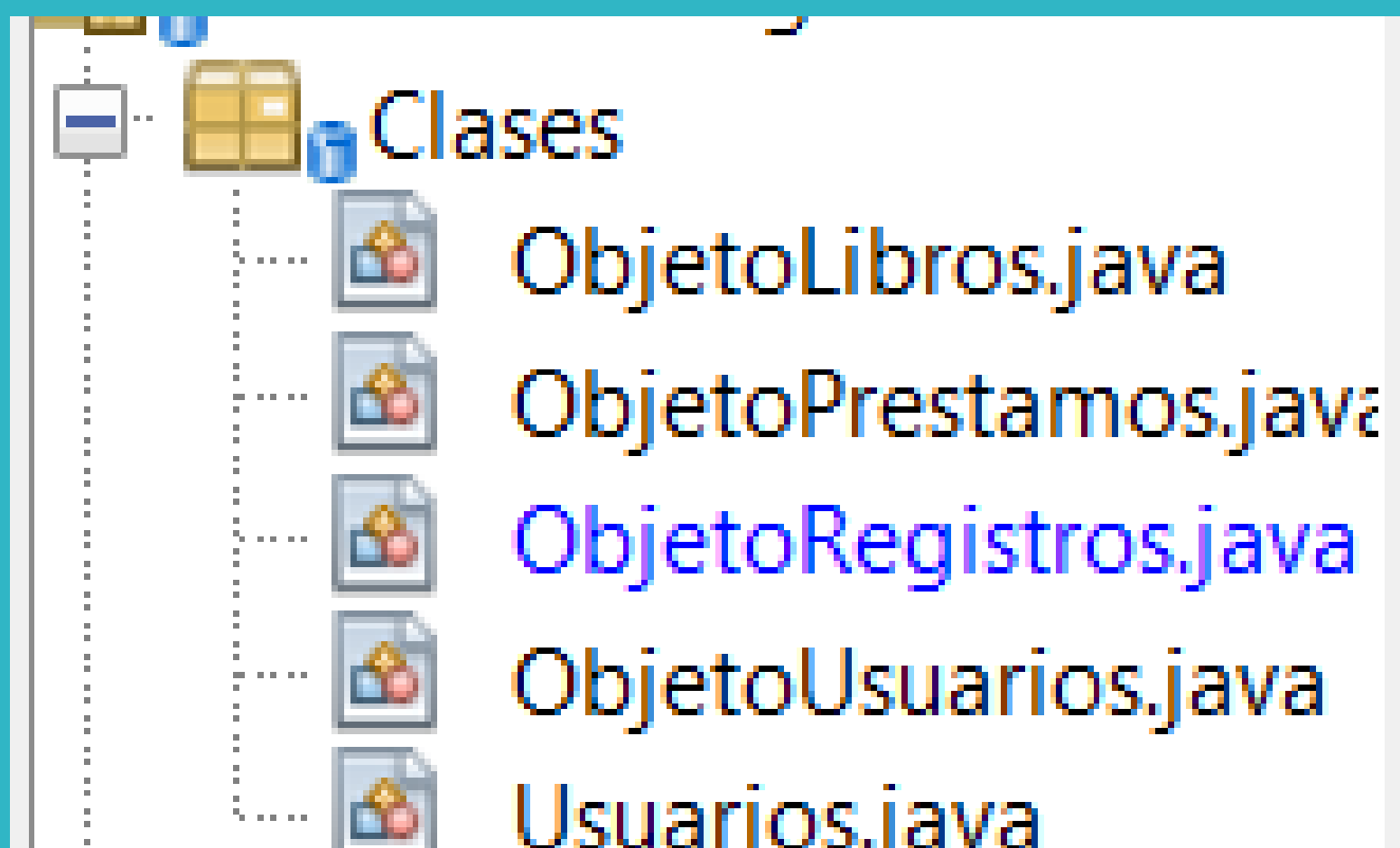
## V. LOGICA DEL PROGRAMA

### Paquete Proyecto 1 y Clase Proyecto1



En el paquete proyecto 1 se encuentra parte de la funcionalidad principal del programa, debido que acá es donde se manda a llamar a la interfaz gráfica del inicio del programa, todo esto sucede en Proyecto1. También se encuentran los arreglos diseñados para poder almacenar y crear las tablas a utilizar.

### Clases y Objetos



En el paquete clases se encuentran los constructores que son utilizados para almacenar y guardar las matrices.

## Paquete ParteGráfica



En este paquete se le da vida a la interfaz gráfica, debido a que dentro de cada uno hay partes muy importantes del programa

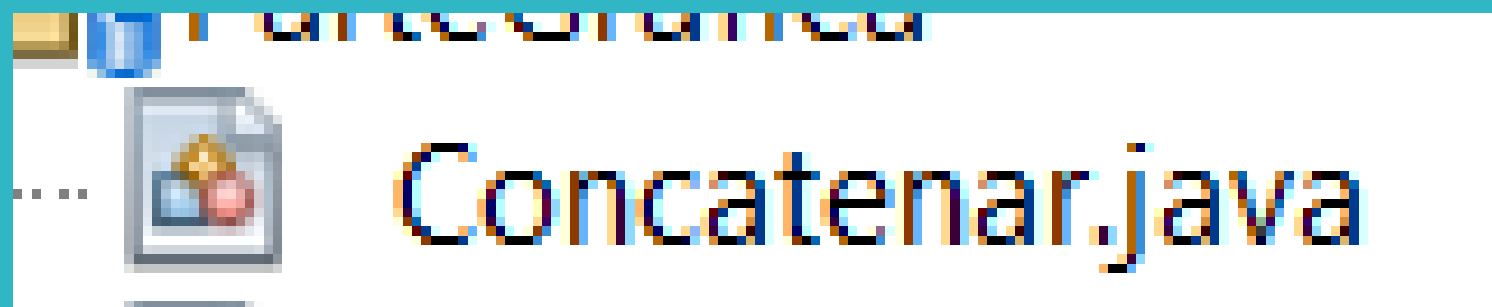
## Clase Registro



En esta parte del programa comienza nuestra interfaz, también hacemos carga masiva en un documento de extension Json, para luego poder confirmar las credenciales cargadas al sistema.



## Clase Concatenar



Acá juntamos todas las pestañas de las demás clases para que se muestren todas juntas en un JPanel.

## Imagenes Utilizadas en la interfaz gráfica



Para hacer que el la interfaz gráfica sea mas agradable al usuario destinado a darle uso al programa.

## Métodos Utilizados

Dentro del Código del programa contamos con diversos métodos para cumplir diversas funciones. A continuación daremos un breve ejemplo de algunos métodos que fueron utilizados para dicha realización

# Métodos Convertir Datos

```
public static Object[][] convertirDatos() {  
    int filas = cUsuarios;  
    Object[][] arreglo = new Object[filas][6];  
    for (int i = 0; i < filas; i++) {  
        arreglo[i][0] = usuarios[i].getId();  
        arreglo[i][1] = usuarios[i].getNombre();  
        arreglo[i][2] = usuarios[i].getContrasena();  
        arreglo[i][3] = usuarios[i].getTipo();  
        arreglo[i][4] = usuarios[i].getFacultad();  
        arreglo[i][5] = usuarios[i].getCarrera();  
    }  
    return arreglo;  
}
```

Utilizado para convertir el arreglo de Usuarios en una matriz de las dimensiones antes mencionadas, para así poder imprimirse en una tabla.

## Método Agregar Usuarios

```
public static void AgregarUsuario(Usuarios usuario) {  
    if (cUsuarios < usuarios.length) {  
        usuarios[cUsuarios] = usuario;  
        cUsuarios++;  
    } else {  
        System.out.println("Ya llegaste al limite de Usuarios");  
    }  
}
```

Este método es el encargado de agregar un objeto de tipo Usuario para añadirlo al arreglo de Usuarios, y se llama al ordenamiento de usuarios.



# Método Iniciar Sesión

```
public void iniciarsesion() {  
    usuarioss = usuarios.getText();  
    contraseñass = contraseñas.getText();  
    if (Proyecto1.verificarusuario(usuarioss)==true) {  
        if (Proyecto1.verificarcontraseña(usuarioss, contraseñass)) {  
            if (Proyecto1.retornarusuarios(usuarioss).getTipo()==1){  
                JOptionPane.showMessageDialog(this, "Bienvenido Admin");  
                this.setVisible(false);  
                Concatenar a = new Concatenar();  
            }else if(Proyecto1.retornarusuarios(usuarioss).getTipo()==2){  
                JOptionPane.showMessageDialog(this, "Usted no tiene permisos de administrador");  
            }  
        }else{  
            JOptionPane.showMessageDialog(this, "Por favor ingresa la contraseña bien");  
        }  
    }else{  
        JOptionPane.showMessageDialog(this, "Usuario no valido :c");  
    }  
}
```

Método encargado de verificar luego de realizar la carga masiva el tipo de usuario con el que se ingresa y si tiene las credenciales correctas

# actionPerformed Iniciar sesion

```
public void actionPerformed(ActionEvent ae) {  
    if (ae.getSource() == inicio) {  
        iniciarsesion();  
    }  
  
    if (ae.getSource() == carga) {  
        // TODO LO QUE ESTE AQUI, HARA EL BOTON1  
        System.out.println("Presionaste al Boton 1.");  
        leerArchivos();  
  
        Proyecto1.LeerUsuarios();  
    } else if (ae.getSource() == boton2) {  
  
        Tabla nueva = new Tabla();  
  
        this.dispose();  
    }  
}
```

Se encarga de dar vida al inicio del programa, al presionar el botón lee el json ingresado.

Boton 2: Se encarga de mostrar la tabla de usuarios .