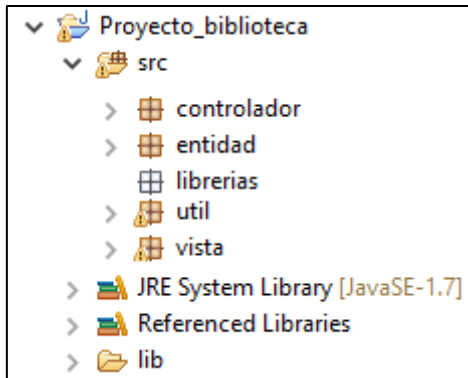


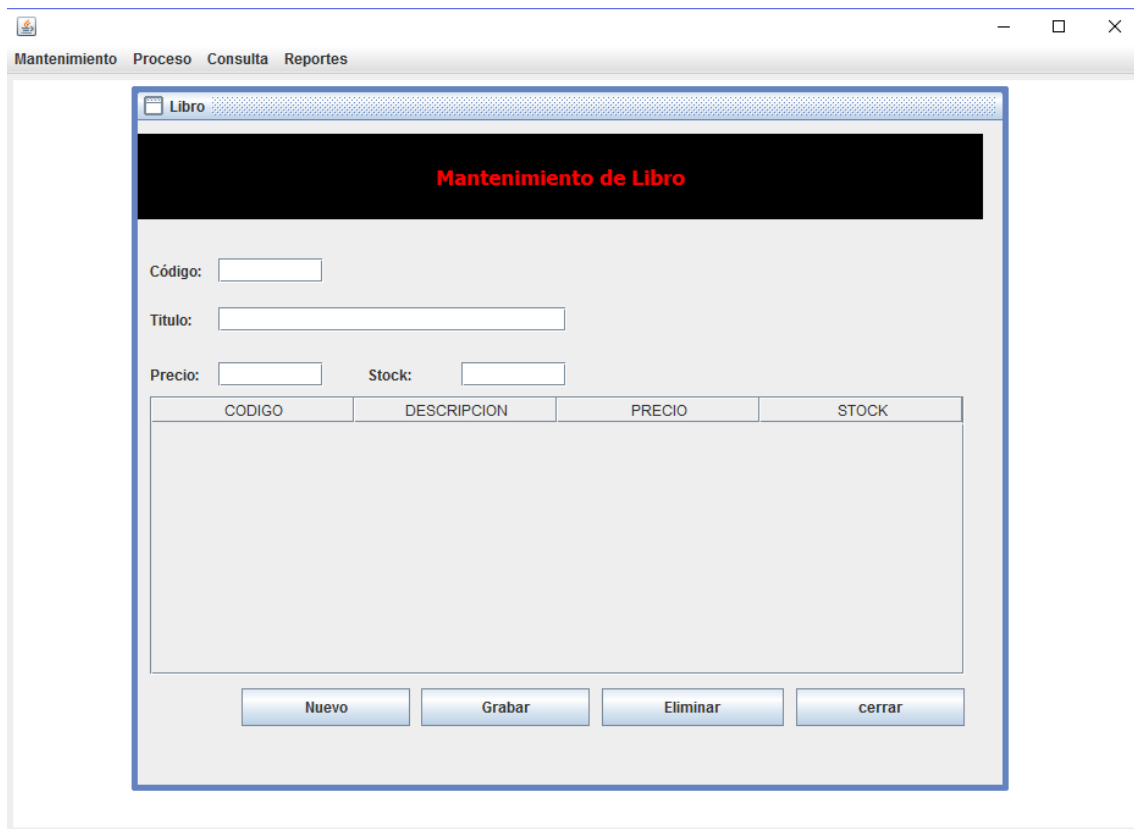
Proyecto LPI - Biblioteca

Pasos:

1. Trabajar con el proyecto: **Proyecto_Biblioteca**:



frmLector(JInternalframe)



2. Dentro del paquete **entidad** crear la siguiente clase:

LibroDTO

```
package entidad;
public class LibroDTO {
    private int codigo;
    private String descripcion;
    private double precio;
    private int cantidad;
    public int getCodigo() {
    }
    public void setCodigo(int codigo) {
    }
    public String getDescripcion() {
    }
    public void setDescripcion(String descripcion) {
    }
    public double getPrecio() {
    }
    public void setPrecio(double precio) {
    }
    public int getCantidad() {
    }
    public void setCantidad(int cantidad) {
    }
    public LibroDTO(int codigo, String descripcion, double precio, int cantidad) {
        super();
        this.codigo = codigo;
        this.descripcion = descripcion;
        this.precio = precio;
        this.cantidad = cantidad;
    }
}
```

3. Dentro del paquete **controlador** crear la siguiente clase:
MySQLLibroDAO

```
package controlador;
import java.sql.Connection;
public class MySQLLibroDAO {
    //crear un método que retorne un valor entero
    //1 se registro
    //-1 error
    public int registrarLibro(LibroDTO x){
        int valor=-1;
        Connection cn=null;
        PreparedStatement pstmt=null;
        try {
            //Paso 1: establecer conexión
            cn=new MySQLConectar().getConectar();
            //Paso 2: establecer sentencia SQL
            String sql="insert into tb_libro values(null,?,?,?)";
            //Paso 3: indicar la sentencia SQL al objeto "pstmt"
            pstmt=cn.prepareStatement(sql);
            //Paso 4: establecer los parametros
            pstmt.setString(1, x.getDescripcion());
            pstmt.setDouble(2, x.getPrecio());
            pstmt.setInt(3, x.getCantidad());
            //Paso 5: ejecutar sentencia SQL
            valor=pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        finally{
            try {
                if(pstmt!=null)pstmt.close();
                if(cn!=null) cn.close();
            } catch (Exception e2) {
                e2.printStackTrace();
            }
        }
        return valor;
    }
    public int actualizarLibro(LibroDTO x){
        int valor=-1;
        Connection cn=null;
        PreparedStatement pstmt=null;
        try {
            //Paso 1: conexión
            cn=new MySQLConectar().getConectar();
            //Paso 2: Sentencia SQL
            String sql="update tb_libro set des_lib=?,pre_lib=?,"+
                "can_lib=? where cod_lib=?";
            //Paso 3: enviar sentencia SQL al Objeto pstmt
            pstmt=cn.prepareStatement(sql);
            //Paso 4: para.
            pstmt.setString(1, x.getDescripcion());
            pstmt.setDouble(2, x.getPrecio());
            pstmt.setInt(3, x.getCantidad());
            pstmt.setInt(4, x.getCodigo());
            //Paso 5: ejecutar sql
            valor=pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        finally{
            try {
                if(pstmt!=null)pstmt.close();
                if(cn!=null)cn.close();
            } catch (Exception e2) {
                e2.printStackTrace();
            }
        }
        return valor;
    }
}
```

```

public int eliminarLibro(int cod){
    int valor=-1;
    Connection cn=null;
    PreparedStatement pstmt=null;
    try {
        //1
        cn=new MySqlConnectionar().getConectar();
        //2
        String sql="delete from tb_libro where cod_lib=?";
        //3
        pstmt=cn.prepareStatement(sql);
        //4
        pstmt.setInt(1, cod);
        //5
        valor=pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    finally{
        try {
            if(pstmt!=null)pstmt.close();
            if(cn!=null)cn.close();
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
    return valor;
}

//método que retorna un arreglo de objetos de la clase LibroDTO
public ArrayList<LibroDTO> listarLibro(){
    ArrayList<LibroDTO> data=new ArrayList<LibroDTO>();
    Connection cn=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;
    try {
        //1
        cn=new MySqlConnectionar().getConectar();
        //2
        String sql="select *from tb_libro";
        //3
        pstmt=cn.prepareStatement(sql);
        //4 para.
        //5 ejecutar
        rs=pstmt.executeQuery();
        //6bucle
        while (rs.next()) {
            LibroDTO lib=new LibroDTO(rs.getInt(1),
                                     rs.getString(2),
                                     rs.getDouble(3),
                                     rs.getInt(4));

            //agregar el objeto al arreglo
            data.add(lib);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }finally{
        try {
            if(rs!=null) rs.close();
            if(pstmt!=null) pstmt.close();
            if(cn!=null) cn.close();
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
    return data;
}
}

```

4. Programar en el formulario **frmLibro**:

```
public class frmLibro extends JInternalFrame implements ActionListener {
    //crear un modelo para el objeto tblibro
    DefaultTableModel modelo=new DefaultTableModel();

    //crear un objeto de la clase MySQLLibroDAO
    MySQLLibroDAO obj=new MySQLLibroDAO();
    //crear una variable de tipo boolean,
    //esta variable sirve para indicar si registra o actualiza
    //estado=true---->registrarLibro
    //estado=false---->actualizarLibro
    boolean estado=false;

    private JTextField txtCodigo;
    private JTextField txtTitulo;
    private JTextField txtPrecio;
```

Crear los métodos en el formulario **frmLibro**:

```
void mensaje(String m){
    JOptionPane.showMessageDialog(null, m);
}
//METODO PARA MOSTRAR LOS REGISTROS QUE RETORNA EL METODO LISTARLIBRO QUE SE ENCUENTRA EN LA CLASE MYSQLLIBRODAO
void Listar(){
    //limpiar el objeto "modelo"
    modelo.setRowCount(0);
    //crear un objeto de la clase arraylist
    ArrayList<LibroDTO> info=obj.listarLibro();
    for(LibroDTO x:info){
        //crear un object de tipo arreglo
        Object fila[]={x.getCodigo(),x.getDescripcion(),x.getPrecio(),x.getCantidad()};
        //enviar el objeto "fila" al objeto "modelo"
        modelo.addRow(fila);
    }
}
void Mostrar(){
    //variable para almacenar la fila seleccionada del control tblibro
    int fila=tblibro.getSelectedRow();
    //mostrar el codigo seleccionado de la fila en la caja txtcodigo
    //getValueAt es un metodo del objeto tblibro
    //retorna el valor que se encuentra en la interseccion de una fila y columna
    txtCodigo.setText(""+tblibro.getValueAt(fila, 0));
    txtTitulo.setText(""+tblibro.getValueAt(fila, 1));
    txtPrecio.setText(""+tblibro.getValueAt(fila, 2));
    txtCantidad.setText(""+tblibro.getValueAt(fila, 3));
}
}
```

Antes de cerrar el constructor del formulario **frmLibro**:

```
btnNuevo = new JButton("Nuevo");
btnNuevo.addActionListener(this);
btnNuevo.setBounds(85, 465, 138, 31);
getContentPane().add(btnNuevo);

//crear las columnas al objeto "modelo"
modelo.addColumn("CODIGO");
modelo.addColumn("DESCRIPCION");
modelo.addColumn("PRECIO");
modelo.addColumn("STOCK");
//enviar el objeto "modelo" al control tblibro
tblibro.setModel(modelo);
//invocar el metodo listar
Listar();
```

5. Código botones:

```

protected void btnNuevoActionPerformed(ActionEvent arg0) {
    txtCodigo.setText("");txtTitulo.setText("");
    txtPrecio.setText("");txtCantidad.setText("");
    txtTitulo.requestFocus();
    estado=true;
}

protected void btnGrabarActionPerformed(ActionEvent arg0) {
    if(estado==true){
        //crear un objeto de la clase LibroDTO
        LibroDTO l=new LibroDTO(0, txtTitulo.getText(),Double.parseDouble(txtPrecio.getText()),
                                Integer.parseInt(txtCantidad.getText()));
        //variable para almacenar el valor que retorna el método registrarLibro
        int valor=obj.registrarLibro(l);
        if(valor==1){
            mensaje("Registro Ok");
            estado=false;
            Listar();
        }
        else
            mensaje("Error");
    }
    else{
        //crear un objeto de la clase LibroDTO
        LibroDTO l=new LibroDTO(Integer.parseInt(txtCodigo.getText()),
                                txtTitulo.getText(),Double.parseDouble(txtPrecio.getText()),
                                Integer.parseInt(txtCantidad.getText()));
        //variable para almacenar el valor que retorna el método actualizarLibro
        int valor=obj.actualizarLibro(l);
        if(valor==1){
            mensaje("ACTUALIZO");
            Listar();
        }
        else
            mensaje("Error");
    }
}

protected void btnEliminarActionPerformed(ActionEvent arg0) {
    int valor=obj.eliminarLibro(
        Integer.parseInt(txtCodigo.getText()));
    if(valor==1)
        mensaje("Registro eliminado");
    else
        mensaje("Error");
}

```

Trabajo:

frmLector(JInternalframe)

Lector

MANTENIMIENTO DE LECTOR

Código:

Nombre: Apellido:

DNI: Sexo:

Código	Nombre	Apellido	DNI	Sexo
--------	--------	----------	-----	------

Nota:

Realizar el mantenimiento de la tabla Lector.

Suerte A.B.P.