

## TEMA: ARQUITECTURA DE SOFTWARE TRES CAPAS – LENGUAJE JAVA

### OBJETIVO:

Conocer y desarrollar aplicaciones con arquitectura tres capas (Presentación – Lógica – Persistencia) en el lenguaje de Programación JAVA

### I. CAPA DE PERSISTENCIA (“Capa de Datos-Base De Datos”)

Utilizando un motor de bases de datos MySQL ,cree una base de datos de nombre “Directorio” y en ella cree una tabla con el nombre “*clientes*” con las siguientes distribución de campos:

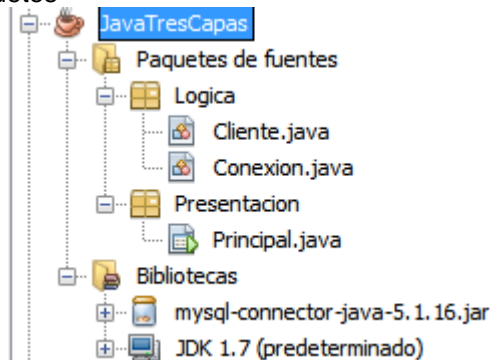
	Nombre campo	Tipo de dato	Tamaño del Campo
Primary Key	Id_Cliente	Autonumerico (Int)	Entero
	Identificacion	Número (BigInt)	Entero largo
	Nombre	Texto (Varchar)	150
	Apellido	Texto (Varchar)	150
	Fijo	Texto (Varchar)	50
	Celular	Texto (Varchar)	50



A continuación se muestran dos script SQL que permiten crear una tabla con las características antes mencionadas en los motores MySQL y SQL Server respectivamente:

Script SQL – Motor: MySQL	Script SQL – Motor: SQL Server
<pre>CREATE TABLE Clientes (   Id_Cliente INT AUTO_INCREMENT,   identificacion BIGINT,   Nombre VARCHAR(150),   Apellido VARCHAR(150),   Fijo VARCHAR(50),   Celular VARCHAR(50),   PRIMARY KEY (Id_Cliente) );</pre>	<pre>CREATE TABLE Clientes (   Id_Cliente INT Identity(0,1),   identificacion numeric(18,0),   Nombre VARCHAR(150),   Apellido VARCHAR(150),   Fijo VARCHAR(50),   Celular VARCHAR(50),   PRIMARY KEY (Id_Cliente) );</pre>

### II. CAPA DE LOGICA (“Capa de Negocio – Lógica de la aplicación”):

1. Cree un Nuevo proyecto “JavaApplication” teniendo en cuenta la siguiente distribución de archivos, clases y paquetes



 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA Nº 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 2 de 12</b>		

**Importante:** Tenga en cuenta que el Archivo “mysql-connector-java-5.1.16.jar” es una librería en formato .jar que corresponde al driver de conexión entre la Plataforma Java y el motor de bases de datos MySQL. Si se usa un Motor de bases de datos diferente se debe agregar el driver de conexión correspondiente.

En la capa de lógica del negocio se ubican los componentes y se programa la secuencia que permite la conexión con el motor de base de datos haciendo uso del correspondiente “Driver de conexión”. En la clase “conexion” se definen los parámetros de conexión, los cuales deben ser coherentes con el motor de base de datos seleccionado, cada motor requiere de sus propios formatos de cadena de conexión:

<b>Motor de Base de Datos</b>	<b>Formato de Cadena de Conexión - Java</b>
SQL Server 2005/2008 (Autenticación Windows)	driver = “com.microsoft.sqlserver.jdbc.SQLServerDriver” connectString = “jdbc:sqlserver://localhost;databaseName=AdventureWorks;integratedSecurity=true;”
SQL Server 2005/2008 (Autenticación SQL Server)	driver = “com.microsoft.sqlserver.jdbc.SQLServerDriver” connectString = “jdbc:sqlserver://localhost:1433;databaseName= nombre_base; user=sa;password= password_del_usuario;”
MySQL	driver = “com.mysql.jdbc.Driver”; connectString = “jdbc:mysql:localhost:3306/nombre_base”

- Dentro del Paquete Lógica en el archivo llamado “Conexion.java” ubique cada bloque de código para escribir las instrucciones según se muestra a continuación: (para MySQL)

```
package Logica;
import java.sql.*;
```

```
public class Conexion {
```

```
    public Connection getConexion(){
        String driver = "com.mysql.jdbc.Driver";
        String connectionString = "jdbc:mysql://localhost:3306/Directorio";
        String user = "usuario MySql";
        String password = "password MySQL";
```



```
    try
    {
```

```
        Class.forName(driver);
        Connection con = DriverManager.getConnection(connectionString, user , password);
        return con;
```

```
    }
    catch ( Exception e )
```

```
    {
        System.out.println(e.getMessage());
        return null;
```

```
    }
}
}
```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 3 de 12</b>		

3. Dentro del Paquete Lógica en el archivo llamado “Cliente.java” ubique cada bloque de código para escribir las instrucciones según se muestra a continuación:

```
package Logica;
import java.sql.*;
import javax.swing.table.DefaultTableModel;

public class Cliente {

    static Connection con;
    static Conexion cx;

    private int identificacion;
    private String nombre;
    private String apellido;
    private String fijo;
    private String celular;

    public int getIdentificacion() {
        return identificacion;
    }

    public void setIdentificacion(int identificacion) {
        this.identificacion = identificacion;
    }

    public String getNombre() {
        return nombre;
    }

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



    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public String getFijo() {
        return fijo;
    }

    public void setFijo(String fijo) {
        this.fijo = fijo;
    }

    public String getCelular() {
        return celular;
    }
}
```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 4 de 12</b>		

```
public void setCelular(String celular) {
    this.celular = celular;
}
```

```
public boolean GuardarCliente(){
```

```
    try
    {
        cx = new Conexion();
        con = cx.getConnection();
        PreparedStatement stmt = con.prepareStatement("INSERT INTO Clientes (identificacion,
        Nombre,Apellido, Fijo, Celular) VALUES (?, ?, ?, ?, ?)");

        stmt.setString(1,"" + this.identificacion);
        stmt.setString(2,this.nombre);
        stmt.setString(3,this.apellido);
        stmt.setString(4,this.fijo);
        stmt.setString(5,this.celular);

        stmt.executeUpdate();
        stmt.close();
        con.close()

        return true;
    }
    catch ( Exception e )
    {
        System.out.println(e.getMessage());
        return false;
    }
}
```



```
public boolean ActualizarCliente(){
```

```
    try
    {
        cx = new Conexion();
        con = cx.getConnection();

        PreparedStatement stmt = con.prepareStatement("UPDATE Clientes SET Nombre = " +
        this.nombre + ", Apellido = " + this.apellido + ", Fijo=" + this.fijo + ", Celular=" + this.celular +
        " WHERE (identificacion = " + this.identificacion + ")");

        stmt.executeUpdate();
        stmt.close();
        con.close();

        return true;
    }
}
```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 5 de 12</b>		

```

catch ( Exception e )
{
    System.out.println(e.getMessage());
    return false;
}
}

```

```

public boolean ConsultarCliente(int IdentificacionABuscar){

    try
    {
        boolean consultaOK=false;
        cx = new Conexion();
        con = cx.getConexion();
        Statement stmt = con.createStatement();

        ResultSet rs = stmt.executeQuery ("SELECT * FROM Clientes where identificacion = " +
        IdentificacionABuscar);



        if (rs.next()==true) {
            rs.first();
            this.identificacion= Integer.parseInt(rs.getString("Identificacion"));
            this.nombre=rs.getString("Nombre");
            this.apellido=rs.getString("Apellido");
            this.fijo= rs.getString("Fijo");
            this.celular=rs.getString("Celular");
            consultaOK= true;
        }
        else
        {
            consultaOK=false;
        }

        stmt.close();
        con.close();

        return consultaOK;

    catch ( Exception e )
    {
        System.out.println(e.getMessage());
        return false;
    }
}

```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA Nº 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 6 de 12</b>		

```

public DefaultTableModel ConsultarTodosClientes(){

    DefaultTableModel modelo = new DefaultTableModel();

    try
    {
        cx = new Conexion();
        con = cx.getConexion();
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM Clientes");

        modelo.addColumn("id");
        modelo.addColumn("identificacion");
        modelo.addColumn("nombre");
        modelo.addColumn("Apellido");
        modelo.addColumn("Fijo");
        modelo.addColumn("Celular");
        int NumColumnas = 6;

        while (rs.next())
        {

            // Se crea un array que será una de las filas de la tabla.
            Object[] fila= new Object[NumColumnas];

            // Se rellena cada posición del array con una de las columnas de la tabla en base de datos.
            for (int i=0;i<NumColumnas;i++)
            {
                fila[i] = rs.getObject(i+1); // El primer indice en rs es 1, no el cero, por eso se suma 1.
            }
            modelo.addRow(fila); // se añade la fila al modelo tipo DefaultTableModel
        }

        stmt.close();
        con.close();

        return modelo;
    }

    catch ( Exception e )
    {
        System.out.println(e.getMessage());
        return modelo;
    }
}

```

```

public boolean EliminarCliente(int IdentificacionABuscar){

    try
    {

        cx = new Conexion();
        con = cx.getConexion();

        PreparedStatement stmt = con.prepareStatement("DELETE FROM Clientes WHERE
        (identificacion = " + IdentificacionABuscar + ")");

        stmt.executeUpdate();
        stmt.close();
        con.close();
        return true;
    }

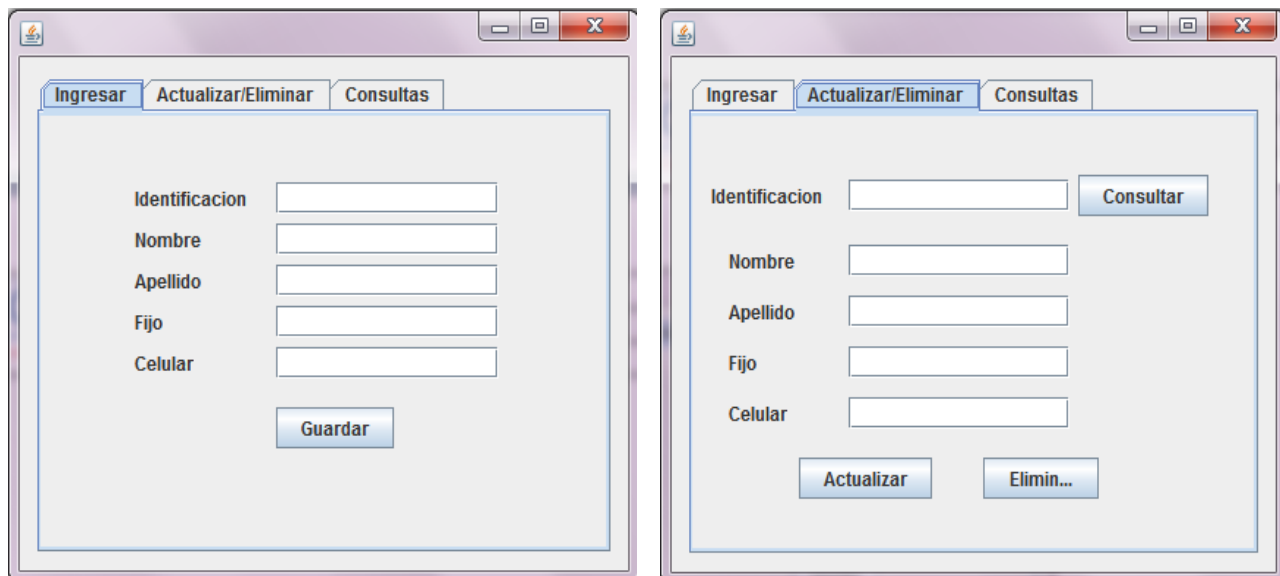
    catch (Exception e)
    {
        System.out.println(e.getMessage());
        return false;
    }
}

```

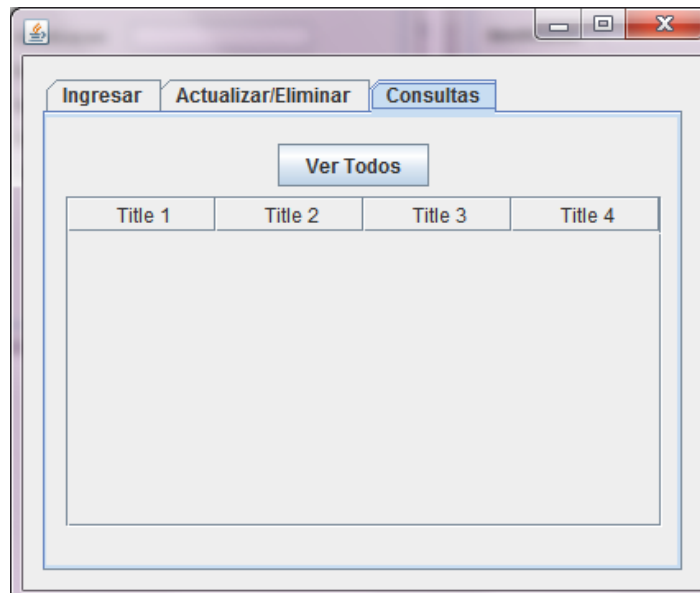
#### IV. CAPA DE PRESENTACIÓN (“I.G.U / Interfaz Gráfica de Usuario”)

La capa de presentación de este ejercicio consta de un formulario con tres pestañas distribuidas llamadas “Ingresar, Actualizar/Eliminar y Consultas” respectivamente

1. Dentro del paquete llamado “Presentacion” en el archivo llamado “Presentacion.java” realice el diseño de la Interfaz Gráfica De usuario del ejercicio propuesto de forma similar al que se muestra a continuación:



The image shows two screenshots of a Java Swing application window. The window has a title bar with standard Windows controls (minimize, maximize, close). Inside the window, there are three tabs: 'Ingresar', 'Actualizar/Eliminar', and 'Consultas'. The 'Ingresar' tab is selected in both screenshots. The form contains five text input fields labeled 'Identificacion', 'Nombre', 'Apellido', 'Fijo', and 'Celular'. In the first screenshot, there is a 'Guardar' button at the bottom. In the second screenshot, there is a 'Consultar' button next to the 'Identificacion' field, and 'Actualizar' and 'Elimin...' buttons at the bottom.



**Nota:** Las imágenes corresponde a un mismo formulario (JFrame) donde los controles son distribuidos por medio de pestañas (JTabbedPane). En la tercera pestaña “Consultas” el control a agregar es una tabla de swing (JTable)

2. Dentro del paquete llamado “Presentacion” en el archivo llamado “Presentacion.java” ubique cada bloque de código para escribir las instrucciones según se muestra a continuación:

```
package Presentacion;
import Logica.*;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Principal extends javax.swing.JFrame {



    /** Creates new form Principal */
    public Principal() {
        initComponents();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try
        {

            Cliente ObjCliente = new Cliente();
            ObjCliente.setIdentificacion(Integer.parseInt(jTextField1.getText()));
            ObjCliente.setNombre(jTextField2.getText());
            ObjCliente.setApellido(jTextField3.getText());
            ObjCliente.setFijo(jTextField4.getText());
            ObjCliente.setCelular(jTextField5.getText());

            boolean GuardarOK = ObjCliente.GuardarCliente();
```



 <b>UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS</b>	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 <b>UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS</b>
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 9 de 12</b>		

```



if (GuardarOK)
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    JOptionPane.showMessageDialog(rootPane, "Este cliente ha sido Guardado
    Corrcitamente");
}
else
{
    JOptionPane.showMessageDialog(rootPane, "Se ha presentado un Error y No se ha
    podido guardar Este cliente en el sistema. Verifique e intente nuevamente");
}
}

catch ( Exception e )
{
    JOptionPane.showMessageDialog(rootPane,"Se ha presentado un Error. " +
    e.getMessage());
}
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
try
{
    Cliente ObjCliente = new Cliente();
    boolean ConsultaOK=ObjCliente.ConsultarCliente(Integer.parseInt(jTextField6.getText()));

    if (ConsultaOK)
    {
        jTextField6.setEnabled(false);
        jTextField7.setText(ObjCliente.getNombre());
        jTextField8.setText(ObjCliente.getApellido());
        jTextField9.setText(ObjCliente.getFijo());
        jTextField10.setText(ObjCliente.getCelular());
        JOptionPane.showMessageDialog(rootPane, "Consulta Realizada Con Exito");
    }
    else
    {
        JOptionPane.showMessageDialog(rootPane, "No se ha encontrado ningun Cliente con esta
        identificación, Verifique e intente nuevamente ");
    }
}
}
catch ( Exception e )
{
    JOptionPane.showMessageDialog(rootPane,"Se ha presentado un Error. " +
    e.getMessage());
}
}

```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 10 de 12</b>		

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    try
    {

        Cliente ObjCliente = new Cliente();
        ObjCliente.setIdentificacion(Integer.parseInt(jTextField6.getText()));
        ObjCliente.setNombre(jTextField7.getText());
        ObjCliente.setApellido(jTextField8.getText());
        ObjCliente.setFijo(jTextField9.getText());
        ObjCliente.setCelular(jTextField10.getText());

        boolean ActualizarOK = ObjCliente.ActualizarCliente();

        if (ActualizarOK)
        {
            jTextField6.setText("");
            jTextField6.setEnabled(true);
            jTextField7.setText("");
            jTextField8.setText("");
            jTextField9.setText("");
            jTextField10.setText("");

            JOptionPane.showMessageDialog(rootPane, "Este Cliente Fue Actualizado
            Correctamente");
        }
        else
        {
            JOptionPane.showMessageDialog(rootPane, "Se ha presentado un Error y No se ha
            podido Actualizar este Cliente. Verifique e intente nuevamente");
        }
    }
    catch ( Exception e )
    {
        JOptionPane.showMessageDialog(rootPane,"Se ha presentado un Error. " +
        e.getMessage());
    }

}



private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    try
    {

        Cliente ObjCliente = new Cliente();

        boolean EliminarOK = ObjCliente.EliminarCliente(Integer.parseInt(jTextField6.getText()));

```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 11 de 12</b>		

```

if (EliminarOK==true)
{
    jTextField6.setText("");
    jTextField6.setEnabled(true);
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");

    JOptionPane.showMessageDialog(rootPane, "Este Cliente Fue Eliminado Correctamente");

}
else
{
    JOptionPane.showMessageDialog(rootPane, "Se ha presentado un Error y No se ha
    podido Eliminar este Cliente. Verifique e intente nuevamente");
}

}

catch ( Exception e )
{
    JOptionPane.showMessageDialog(rootPane,"Se ha presentado un Error. " +
    e.getMessage());
}

}



private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    Cliente ObjCliente = new Cliente();

    DefaultTableModel modelo = ObjCliente.ConsultarTodosClientes();
    jTable1.setVisible(true);
    jTable1.setModel(modelo);

}

```

 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS	<b>GUÍA N° 5 - PROGRAMACIÓN APLICADA</b>		 UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
	<b>Ingeniería Electrónica</b>	<b>Ing. Nestor Raul Suarez P.</b>	
	<b>Página 12 de 12</b>		

### **TALLER**

Desarrollar una aplicación con arquitectura tres capas que permita administrar el préstamo de libros de la Biblioteca en una institución educativa. Esta aplicación debe satisfacer los siguientes requerimientos:

1. Permitir ingresar los datos correspondientes a cada estudiante que solicita un libro, los cuales deben ser: Identificación, Nombres, Apellidos, Género, Teléfono, Celular, Fecha de nacimiento, Correo electrónico, Curso o Programa, Ciudad De Residencia, Dirección, Barrio y Estrato.
2. Permitir registrar libros con los siguientes datos (Código, Nombre, Autor, Descripción, Fecha de publicación, Número de ejemplar, Estado, Fecha De Préstamo, Identificación De Estudiante)
3. Permitir modificar/eliminar la información almacenada de cualquier Estudiante o Libro.
4. Debe tener un módulo de consultas donde se tengan las siguientes opciones de búsqueda
  - ✓ **Estudiantes:** Búsqueda Por Identificación, Por Nombre, Por Apellido, Por Curso o Programa, Por Ciudad de residencia, Por (Ciudad y Estrato ->Simultáneos)
  - ✓ **Libros:** Búsqueda Por Código, Por Nombre, Por Autor, Por Fecha De Préstamo, Por (Nombre y Autor -> Simultáneos)
  - ✓ Cuando un libro se encuentre prestado, en la misma pantalla se deben mostrar los datos del libro y del estudiante al cual se le prestó (Tanto al consultar un estudiante como al consultar un libro, siempre y cuando la consulta arroje un único registro como respuesta)
5. Cuando un libro sea prestado, en la tabla que almacena los datos de cada libro debe reflejarse en su campo "Estado" que este está prestado, se debe guardar el dato de la fecha de préstamo y se debe almacenar en el campo "*IdentificacionEstudiante*" la identificación del estudiante al que se le presta el libro. El sistema debe permitir que al devolver un libro este quede nuevamente disponible en el campo "Estado".

#### ***Recomendaciones Importantes:***

- ✓ Para este ejercicio asuma que la institución educativa no requiere almacenar el histórico de todos los prestamos realizados.
- ✓ Se debe crear una base de datos con mínimo dos tablas que deben estar relacionadas (Estudiantes - Libros)
- ✓ Se debe emplear los conceptos y pilares del paradigma de la programación orientada a objetos en el desarrollo del código correspondiente.