



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



PRÁCTICA #1:

Comunicación Cliente – Servidor Vía Sockets Java con Sistema Manejador de Bases de Datos

MATERIA: Desarrollo de Sistemas Distribuidos

PROFESOR: José Asunción Enriquez Zárate

ALUMNOS:

Juan Luis Morales Angelina
Hector Ivan Cabrera Segura
Ulises Amado Sánchez Bello

FECHA: 7/03/17

GRUPO: 4CV4

Indice de Contenido

INTRODUCCION..... 2

DESARROLLO..... 4

 Requerimientos:..... 4

 Screenshots..... 5

 Códigos..... 7

CONCLUSIONES..... 90

BIBLIOGRAFÍA..... 90

INTRODUCCION

SQL es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas. SQL (Lenguaje estructurado de consulta) es un lenguaje específico de dominio que se usa en la programación y está diseñado para manejar los datos que hay en un sistema de base de datos, o para el procesamiento de flujos de datos en un sistema de gestión de flujos.

El lenguaje de definición de datos (en inglés *Data Definition Language*, o *DDL*), es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen 3 operaciones básicas: CREATE, ALTER y DROP.

CREATE

Este comando permite crear objetos de datos, como nuevas bases de datos, tablas, vistas y procedimientos almacenados.

Ejemplo (crear una tabla)

```
CREATE TABLE 'CUSTOMERS';
```

ALTER

Este comando permite modificar la estructura de una tabla u objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

Ejemplo (agregar columna a una tabla)

```
ALTER TABLE 'ALUMNOS' ADD EDAD INT UNSIGNED;
```

DROP

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

Ejemplo

```
DROP TABLE 'ALUMNOS';
```

SELECT

Esta sentencia nos permite consultar los datos almacenados en una tabla de la base de datos.

```
SELECT [ALL | DISTINCT ]
        <nombre_campo> [{,<nombre_campo>}]
FROM <nombre_tabla>|<nombre_vista>
        [{,<nombre_tabla>|<nombre_vista>}]
[WHERE <condición> [{ AND|OR <condición>}]]
[GROUP BY <nombre_campo> [{,<nombre_campo >}]]
[HAVING <condición>[{ AND|OR <condición>}]]
[ORDER BY <nombre_campo>|<indice_campo> [ASC | DESC]
        [{,<nombre_campo>|<indice_campo> [ASC | DESC ]}]]
```

SELECT	Palabra clave que indica que la sentencia de SQL que queremos ejecutar es de selección.
ALL	Indica que queremos seleccionar todos los valores. Es el valor por defecto y no suele especificarse casi nunca.
DISTINCT	Indica que queremos seleccionar sólo los valores distintos.
FROM	Indica la tabla (o tablas) desde la que queremos recuperar los datos. En el caso de que exista más de una tabla se denomina a la consulta "consulta combinada" o "join". En las consultas combinadas es necesario aplicar una condición de combinación a través de una cláusula WHERE .
WHERE	Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Admite los operadores lógicos AND y OR .
GROUP BY	Especifica la agrupación que se da a los datos. Se usa siempre en combinación con funciones agregadas.
HAVING	Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Su funcionamiento es similar al de WHERE pero aplicado al conjunto de resultados devueltos por la consulta. Debe aplicarse siempre junto a GROUP BY y la condición debe estar referida a los campos contenidos en ella.
ORDER BY	Presenta el resultado ordenado por las columnas indicadas. El orden puede expresarse con ASC (orden ascendente) y DESC (orden descendente). El valor predeterminado es ASC .

JDBC API

Java Database Connectivity, más conocida por sus siglas **JDBC**,^{1 2} es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de

datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión; para ello provee el localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tarea con la base de datos a la que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

La API de JDBC se conforma de dos paquetes:

- `java.sql`
- `javax.sql`

DESARROLLO

Requerimientos:

Hardware:

- Laboratorio de cómputo
- Red computacional
- Proyector

Software:

- Sistema operativo
- Sistema Manejador de Bases de Datos (MySQL, SQL Server, ORACLE, DB2, etc.)
- Java Development Kit (JSDK 7.x)

Instrucciones:

1. Cree la siguiente base de datos.
2. Crear una interfaz Gráfica en Java Swing para dar soporte a las diferentes actividades de Administracion en cada una de las entidades del punto Anterior, esta interfaz gráfica sera el cliente, que realizará peticiones al Servidor.
3. Crear un SocketServidor que permita mantener comunicación con el servidor de bases de datos MySql y permita dar soporte a las operaciones CRUD Basicas para cada una de las Entidades mostradas en la imagen del punto 1.

Screenshots

Gestor de Alumnos

Nombre

Apellido Paterno

Apellido Materno

Email

Calle

Numero

Colonia

Delegacion

Entidad

Telefono

Opciones

Nuevo

Buscar

Eliminar

Actualiza

Listar

idAlu...	Nombre	Apelli...	Apelli...	Email	Calle	Numero	Colonia	Deleg...	Entidad	Telefo...
5	Maria	Morelos	Pavon	pavar...	orient...	34	Agric...	Iztaca...	DF	6666...
6	Maria	Miranda	Huerta	huari...	orient...	34	Agric...	Iztaca...	DF	6666...
7	Miguel	Montes	Oca	moca...	12 de...	1215	Olivos	Migue...	DF	5859...

Gestor de Asignaciones

Alumno

Curso

Horario

Tipo de curso

0

Opciones

Nuevo

Buscar

Eliminar

Actualiza

Listar

idCurso	idAlumno	Horario	Tipo de Curso
3	6	Horario	1
4	7	alguno	2

Gestor de Cursos

Nombre

Fecha inicio

Fecha fin

Cuota de recuperacion

Opciones

Nuevo

Buscar

Eliminar

Actualiza

Listar

idCurso	Nombre	Fecha de In...	Fecha Fin	Cuota de Re...
2	Programaci...	2017-2-1	2017-6-29	300.5
3	Liderazgo	2017-2-1	2017-6-4	500.0
4	Cosologia	2017-2-1	2017-6-4	500.0

Códigos

Conexion.java

```
package servicios;
import java.sql.Connection;
import java.sql.DriverManager;

/**
 *
 * @author Lock
 */
public class Conexion {

    //Configuracion de los datos de la BD
    private String usuario = "ziggy";
    private String pass = "Morales=10";
    private String host = "127.0.0.1";
    private String nombre_BD = "practical_dsd";
```

```

private Connection con = null;

public Conexion() {
}
//Metodo que se devuelve la conexion o null si hubo un error
public Connection getConexionMYSQL() {
    try{
        Class.forName("org.mariadb.jdbc.Driver").newInstance( );
        String servidor = "jdbc:mariadb://" + host + "/" + nombre_BD;
        con = DriverManager.getConnection(servidor, usuario, pass);
        return con;
    } catch (Exception e) {
        e.printStackTrace();
        return con;
    }
}
}

```

Crud.java

```

package servicios;

import modelosCurso;
import modelos.Alumno;
import modelos.Asignacion;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
import java.time.*;
import java.util.ArrayList;

/**
 *
 * @author Lock
 */
public class Crud {

    private Conexion conexion = new Conexion();
    private Connection con;
    private Statement st;
    private ResultSet rs;

    public Crud() {
        try {
            if ((con = conexion.getConexionMYSQL()) == null) {
                JOptionPane.showMessageDialog(null, "Error con la conexion a la
BD");
                return;
            }
            st = con.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

//-----Aquí comienza el CRUD para la
clase alumno-----
//Insertar alumno
public boolean insertA(Alumno al) {
    try {
        String query = "INSERT INTO alumno VALUES(NULL,'" + al.getNombre() +
        "','" + al.getApellidoPaterno() + "','" + al.getApellidoMaterno() + "','" +
        al.getMail() + "','" + al.getCalle() + "','" + al.getNumero() + "','" +
        al.getColonia() + "','" + al.getDelegación() + "','" + al.getEntidad() + "','" +
        al.getTelefono() + "')";
        st.executeUpdate(query);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Buscar todos los alumnos
public ResultSet buscaTodosA() {
    try {
        String query = "SELECT * FROM alumno";
        rs = st.executeQuery(query);
        return rs;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

// Buscar por nombre
public Alumno buscaNombreA(int id) {
    try {
        String query = "SELECT * FROM alumno WHERE idAlumno = '" + id + "'";
        rs = st.executeQuery(query);
        rs.next();
        return new Alumno(rs.getInt("idAlumno"), rs.getString("nombre"),
rs.getString("paterno"), rs.getString("materno"), rs.getString("email"),
rs.getString("calle"), rs.getString("numero"), rs.getString("colonia"),
rs.getString("delegacion"), rs.getString("entidad"), rs.getString("telefono"));
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//Modificar Alumno
public boolean updateA(Alumno al) {
    try {
        String query = "UPDATE alumno SET " + "nombre='" + al.getNombre() +
        "','" + "paterno='" + al.getApellidoPaterno() + "','" + "materno='" +
        al.getApellidoMaterno() + "','" + "email='" + al.getMail() + "','" + "calle='" +
        al.getCalle() + "','" + "numero='" + al.getNumero() + "','" + "colonia='" +
        al.getColonia() + "','" + "delegacion='" + al.getDelegación() + "','" + "entidad='" +
        al.getEntidad() + "','" + "telefono='" + al.getTelefono() + "' WHERE idAlumno = " +
        al.getIdAlumno() + "'";
        st.executeUpdate(query);
    }
}

```

```

        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Eliminar Alumno
public boolean deleteA(Alumno al) {
    try {
        String query = "DELETE FROM asignacion WHERE idAlumno = " +
al.getIdAlumno() + "";
        st.executeUpdate(query);
        query = "DELETE FROM alumno WHERE idAlumno = " + al.getIdAlumno() + "";
        st.executeUpdate(query);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Obtener el id de un Alumno especifico

public int getIdE(String nombre, String paterno, String materno) {
    try {
        int id;
        String query = "SELECT * FROM alumno WHERE nombre = '" + nombre + '"
AND paterno = '" + paterno + "'" AND materno = '" + materno + "'";
        rs = st.executeQuery(query);
        rs.next();
        id = rs.getInt("idAlumno");
        return id;
    } catch (Exception e) {
        e.printStackTrace();
        return 0;
    }
}

//-----Aquí empuieza el Crud de
curso-----
//Insertar curso
public boolean insertC(Curso cu) {

    LocalDate fechaInicio, fechaTermino;

    //Convertimos los string a LocalDate en formato "yyyy-mm-dd"
    fechaInicio = LocalDate.parse(cu.getFechaInicio());
    fechaTermino = LocalDate.parse(cu.getFechaTermino());

    //Convertimos los LocalDate a SQL date
    java.sql.Date fi = java.sql.Date.valueOf(fechaInicio);
    java.sql.Date fe = java.sql.Date.valueOf(fechaTermino);

    try {
        String query = "INSERT INTO curso VALUES(NULL,'" + cu.getNombre() +
"', '" + fi + "', '" + fe + "', '" + cu.getCuota() + "');";
        st.executeUpdate(query);
    }
}

```

```

        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Buscar todos los cursos
public ResultSet buscaTodosC() {
    try {
        String query = "SELECT * FROM curso";
        rs = st.executeQuery(query);
        return rs;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//Buscar un curso por nombre
public Curso buscaNombreC(int id) {
    try {
        String query = "SELECT * FROM curso WHERE idCurso = '" + id + "'";
        rs = st.executeQuery(query);
        rs.next();
        return new Curso(rs.getInt("idCurso"), rs.getString("nombreC"),
rs.getString("fechaInicio"), rs.getString("fechaTermino"),
rs.getDouble("cuotaDeRecuperacion"));
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//Modificar curso
public boolean updateC(Curso cu) {

    LocalDate fechaInicio, fechaTermino;

    //Convertimos los string a LocalDate en formato "yyyy-mm-dd"
    fechaInicio = LocalDate.parse(cu.getFechaInicio());
    fechaTermino = LocalDate.parse(cu.getFechaTermino());

    //Convertimos los LocalDate a SQL date
    java.sql.Date fi = java.sql.Date.valueOf(fechaInicio);
    java.sql.Date fe = java.sql.Date.valueOf(fechaTermino);

    try {
        String query = "UPDATE curso SET " + "nombreC='" + cu.getNombre() +
        "','" + "fechaInicio='" + fi + "','" + "fechaTermino='" + fe + "','" +
        "cuotaDeRecuperacion='" + cu.getCuota() + "' WHERE idCurso = " + cu.getIdCurso() +
        "''";

        st.executeUpdate(query);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

```

    }
}

//Eliminar curso
public boolean deleteC(int id) {
    try {

        String query = "DELETE FROM asignacion WHERE idCurso = " + id + "";
        st.executeUpdate(query);
        query = "DELETE FROM curso WHERE idCurso = " + id + "";
        st.executeUpdate(query);

        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Obtener el id de un curso con su nombre
public int getIdC(String nombre) {
    try {
        int id;
        String query = "SELECT * FROM curso WHERE nombreC = '" + nombre + "'";
        rs = st.executeQuery(query);
        rs.next();
        id = rs.getInt("idCurso");
        return id;
    } catch (Exception e) {
        e.printStackTrace();
        return 0;
    }
}

//Obtener el nombre de un curso con su id
public String getNombreC(int id) {
    try {
        String nombre;
        String query = "SELECT * FROM curso WHERE idCurso = '" + id + "'";
        rs = st.executeQuery(query);
        rs.next();
        nombre = rs.getString("nombreC");
        return nombre;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//-----Aquí comienzan los crud de
asignación-----
//Insertar una asignación
public boolean insertS(Asignacion as) {
    try {
        String query = "INSERT INTO asignacion VALUES('" + as.getIdAlumno() +
        "','" + as.getIdCurso() + "','" + as.getHorario() + "','" + as.getTipo() + "')";
        st.executeUpdate(query);
    }
}

```

```

        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Insertar una asignación
public Asignacion buscaS(Asignacion as) {
    try {
        String query = "SELECT FROM asignacion WHERE idAlumno = " +
as.getIdAlumno() + " AND idCurso = " + as.getIdCurso() + "";
        rs=st.executeQuery(query);
        rs.next();
        as.setHorario(rs.getString("horario"));
        as.setTipo(rs.getInt("tipoCurso"));
        return as;
    } catch (Exception e) {
        e.printStackTrace();
        return as;
    }
}

//Buscar todas las asignaciones
public ResultSet buscaTodosS() {
    try {
        String query = "SELECT nombre, paterno, materno, horario, tipoCurso,
nombreC\n" + "FROM alumno inner join asignacion ON alumno.idAlumno =
asignacion.idAlumno inner join curso ON asignacion.idCurso = curso.idCurso;";
        rs = st.executeQuery(query);
        return rs;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

// Buscar asignación por alumno
public ResultSet buscaAsignacionAlumno(String nombre) {
    try {
        String query = "SELECT nombre, paterno, materno, horario, tipoCurso,
nombreC\n" + "FROM alumno join asignacion ON alumno.idAlumno = asignacion.idAlumno
inner join curso ON asignacion.idCurso = curso.idCurso WHERE nombre = '" + nombre +
"'";
        rs = st.executeQuery(query);
        return rs;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//Buscar asignación por curso

public ResultSet buscaAsignacionCurso(String nombre) {
    try {
        String query = "SELECT nombre, paterno, materno, horario, tipoCurso,
nombreC\n" + "FROM alumno join asignacion ON alumno.idAlumno = asignacion.idAlumno

```

```

inner join curso ON asignacion.idCurso = curso.idCurso where nombreC = '"' + nombre
+ '"';

        rs = st.executeQuery(query);
        return rs;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

//Modificar asignación
public boolean updateS(Asignacion as) {
    try {
        String query = "UPDATE asignacion SET " + "horario='" + as.getHorario()
+ "'," + "tipoCurso=" + as.getTipo() + " WHERE idAlumno = " + as.getIdAlumno() + "
AND idCurso = " + as.getIdCurso() + " ";
        st.executeUpdate(query);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Eliminar asignacion
public boolean deleteS(Asignacion as) {
    try {
        String query = "DELETE FROM asignacion WHERE idAlumno = " +
as.getIdAlumno() + " AND idCurso = " + as.getIdCurso() + " ";
        st.executeUpdate(query);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

//Genera una lista de las asignaciones
public ArrayList<Asignacion> ListaTodosS() {

    ArrayList<Asignacion> asignacion = new ArrayList<Asignacion>();

    try {

        Crud cr = new Crud();
        String query = "select * from asignacion";
        ResultSet res = st.executeQuery(query);

        while (res.next()) {

            asignacion.add(new Asignacion(res.getInt("idAlumno"),
res.getInt("idCurso"), res.getString("horario"), res.getInt("tipoCurso")));

        }

    } catch (Exception e) {
        e.printStackTrace();
    }
}

```



```

        }

        return asignacion;
    }
}

```

EmpacaBusqueda.java

//Esta clase sirve para empaquetar las busquedas en listas para enviarlas al cliente.

```

package servicios;

import modelosCurso;
import modelos.Alumno;
import modelos.Asignacion;
import java.util.*;
import java.sql.*;
import java.time.*;

/**
 *
 * @author Lock
 */
public class EmpacaBusqueda {

    public EmpacaBusqueda() {
    }

    //_____Empaquetados de busqueda
    Alumno_____

    //Genera una lista con todos los alumnos en la tabla

    public ArrayList<Alumno> ListaTodosA() {

        ArrayList<Alumno> alumnos = new ArrayList<Alumno>();

        try{

            Crud cr = new Crud();
            ResultSet res = cr.buscaTodosA();
            if(res!= null){
                while(res.next()) {
                    alumnos.add(new
Alumno(res.getInt("IdAlumno"),res.getString("nombre"), res.getString("paterno"),
res.getString("materno"), res.getString("email"),res.getString("calle"),
res.getString("numero"), res.getString("colonia"), res.getString("delegacion"),
res.getString("entidad"), res.getString("telefono")));
                }

            }

        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

```

        return alumnos;
    }

    //Genera una lista con los alumnos encontrados por nombre

    //      public ArrayList<Alumno> ListaNombreA(String nombre){
    //
    //      ArrayList<Alumno> alumnos = new ArrayList<Alumno>();
    //
    //      try{
    //
    //          Crud cr = new Crud();
    //          ResultSet res = cr.buscaNombreA(nombre);
    //          if(res!= null){
    //              while(res.next()){
    //                  alumnos.add(new
    Alumno(res.getInt("IdAlumno"),res.getString("nombre"), res.getString("paterno"),
    res.getString("materno"), res.getString("email"),res.getString("calle"),
    res.getString("numero"), res.getString("colonia"), res.getString("delegacion"),
    res.getString("entidad"), res.getString("telefono")));
    //              }
    //          }
    //      }catch(Exception e){
    //          e.printStackTrace();
    //      }
    //      return alumnos;
    //  }

    //_____Empaquetados de busqueda de
curso_____

    //Busqueda de todos los cursos

    public ArrayList<Curso> ListaTodosC(){

        ArrayList<Curso> cursos = new ArrayList<Curso>();

        try{

            Crud cr = new Crud();
            ResultSet res = cr.buscaTodosC();
            if(res!= null){
                while(res.next()){
                    LocalDate fechaInicio,fechaTermino;
                    String fi,fe;
                    fechaInicio = res.getDate("fechaInicio").toLocalDate();
                    fechaTermino = res.getDate("fechaTermino").toLocalDate();
                    fi = fechaInicio.getYear()+"-"+fechaInicio.getMonthValue()
+ "-" + fechaInicio.getDayOfMonth();
                    fe = fechaTermino.getYear()+"-"+fechaTermino.getMonthValue()
+ "-" + fechaTermino.getDayOfMonth();

```

```

        cursos.add(new
Curso(res.getInt("IdCurso"),res.getString("nombreC"),
fi,fe,res.getDouble("cuotaDeRecuperacion")));
    }

    }

    }catch(Exception e){
        e.printStackTrace();
    }

    return cursos;
}

// Genera una lista de los cursos buscados por nombre

// public ArrayList<Curso> ListaNombreC(String nombre){
//
//     ArrayList<Curso> cursos = new ArrayList<Curso>();
//
//     try{
//
//         Crud cr = new Crud();
//         ResultSet res = cr.buscaNombreC(nombre);
//         if(res!= null){
//             while(res.next()){
//                 LocalDate fechaInicio,fechaTermino;
//                 String fi,fe;
//                 fechaInicio = res.getDate("fechaInicio").toLocalDate();
//Pasamos de sql a LocalDate
//                 fechaTermino = res.getDate("fechaTermino").toLocalDate();
//                 fi = fechaInicio.getYear()+"-"+fechaInicio.getMonthValue()
+"-"+fechaInicio.getDayOfMonth(); //Construimos el String con los valores del
LocalDate
//                 fe = fechaTermino.getYear()+"-"+fechaTermino.getMonthValue()
+"-"+fechaTermino.getDayOfMonth();
//
//                 cursos.add(new
Curso(res.getInt("IdCurso"),res.getString("nombreC"),
fi,fe,res.getDouble("cuotaDeRecuperacion")));
//
//             }
//
//         }catch(Exception e){
//             e.printStackTrace();
//         }
//
//     return cursos;
// }

//_____Empaquetado de busqueda de
asignacion_____

//Genera una lista de las asignaciones

```

```

//      public ArrayList<Asignacion> ListaTodosS() {
//
//          ArrayList<Asignacion> asignacion = new ArrayList<Asignacion>();
//
//          try{
//
//              Crud cr = new Crud();
//              ResultSet res = cr.buscaTodosS();
//
//              if(res!= null){
//                  while(res.next()){
//
//                      asignacion.add(new
Asignacion(res.getString("nombre"),res.getString("paterno"),res.getString("materno"
),res.getString("nombreC"),res.getString("horario"),res.getInt("tipoCurso")));
//
//                      }
//
//                  }
//
//              }catch(Exception e){
//                  e.printStackTrace();
//              }
//
//          return asignacion;
//      }

//Genera una lista con la busqueda por alumno

//      public ArrayList<Asignacion> ListaNombreAlumnoS(String nombre){
//
//          ArrayList<Asignacion> asignacion = new ArrayList<Asignacion>();
//
//          try{
//
//              Crud cr = new Crud();
//              ResultSet res = cr.buscaAsignacionAlumno(nombre);
//              if(res!= null){
//                  while(res.next()){
//                      asignacion.add(new
Asignacion(res.getString("nombre"),res.getString("paterno"),res.getString("materno"
),res.getString("nombreC"),res.getString("horario"),res.getInt("tipoCurso")));
//
//                      }
//
//                  }
//
//              }catch(Exception e){
//                  e.printStackTrace();
//              }
//
//          return asignacion;
//      }

```

```

////Genera una lista buscando por curso
//
//
//    public ArrayList<Asignacion> ListaNombreCursoS(String nombre){
//
//        ArrayList<Asignacion> asignacion = new ArrayList<Asignacion>();
//
//
//        try{
//
//            Crud cr = new Crud();
//            ResultSet res = cr.buscaAsignacionCurso(nombre);
//            if(res!= null){
//                while(res.next()){
//                    asignacion.add(new
Asignacion(res.getString("nombre"),res.getString("paterno"),res.getString("materno"
),res.getString("nombreC"),res.getString("horario"),res.getInt("tipoCurso")));
//                }
//
//            }
//
//        }catch(Exception e){
//            e.printStackTrace();
//        }
//
//        return asignacion;
//    }
//

//Genera el paquete a insertar en la asignacion

//    public boolean insertaAsignacion( Asignacion as){
//
//
//
//
//        try{
//            Crud cr = new Crud();
//            int idA,idC;
//
//            idA = cr.getIdE(as.getAlumno(), as.getPaterno(),as.getMaterno());
//            idC = cr.getIdC(as.getCurso());
//            cr.insertS(idA, idC, as);
//
//            return true;
//        }catch(Exception e){
//            e.printStackTrace();
//        }return false;
//
//    }

//Genera el paquete a actualizar en la base

//    public boolean actualizaAsignacion( Asignacion as){
//
//
//
//
//        try{

```

```

//          Crud cr = new Crud();
//          int idA,idC;
//
//          idA = cr.getIdE(as.getAlumno(),as.getPaterno(),as.getMaterno());
//          idC = cr.getIdC(as.getCurso());
//          cr.updateS(idA, idC, as);
//
//          return true;
//      }catch(Exception e){
//          e.printStackTrace();
//      }return false;
//
//    }

//Borra la asignacion

//    public boolean borraAsignacion( Asignacion as){
//
//
//
//
//        try{
//            Crud cr = new Crud();
//            int idA,idC;
//
//            idA = cr.getIdE(as.getAlumno(),as.getPaterno(),as.getMaterno());
//            idC = cr.getIdC(as.getCurso());
//            cr.deleteS(idA, idC);
//
//            return true;
//        }catch(Exception e){
//            e.printStackTrace();
//        }return false;
//
//    }

}

```

Alumno.java

```

package modelos;

/**
 *
 * @author Lock
 */
public class Alumno implements java.io.Serializable {
    private String nombre, aPaterno, aMaterno, mail,calle,numero, colonia,
delegación, entidad, telefono;
    private int idAlumno;

    public Alumno()
    {

    }
}

```

```

    public Alumno(int idAlumno,String nombre, String aPaterno, String aMaterno,
String mail, String calle, String numero, String colonia, String delegación, String
entidad, String telefono) {
        this.idAlumno=idAlumno;
        this.nombre = nombre;
        this.aPaterno = aPaterno;
        this.aMaterno = aMaterno;
        this.mail = mail;
        this.calle = calle;
        this.numero = numero;
        this.colonia = colonia;
        this.delegación = delegación;
        this.entidad = entidad;
        this.telefono = telefono;
    }
    public int getIdAlumno() {
        return idAlumno;
    }

    public void setIdAlumno(int idAlumno) {
        this.idAlumno = idAlumno;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getApellidoPaterno() {
        return aPaterno;
    }

    public void setApellidoPaterno(String aPaterno) {
        this.aPaterno = aPaterno;
    }

    public String getApellidoMaterno() {
        return aMaterno;
    }

    public void setApellidoMaterno(String Amaterno) {
        this.aMaterno = Amaterno;
    }

    public String getMail() {
        return mail;
    }

    public void setMail(String mail) {
        this.mail = mail;
    }

    public String getCalle() {

```

```

        return calle;
    }

    public void setCalle(String calle) {
        this.calle = calle;
    }

    public String getNumero() {
        return numero;
    }

    public void setNumero(String numero) {
        this.numero = numero;
    }

    public String getColonia() {
        return colonia;
    }

    public void setColonia(String colonia) {
        this.colonia = colonia;
    }

    public String getDelegación() {
        return delegación;
    }

    public void setDelegación(String delegación) {
        this.delegación = delegación;
    }

    public String getEntidad() {
        return entidad;
    }

    public void setEntidad(String entidad) {
        this.entidad = entidad;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
}

```

Curso.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modelos;

```



```

/**
 *
 * @author Lock
 */
public class Curso implements java.io.Serializable {

    private String nombre, fechaInicio, fechaTermino;
    private int idCurso;

    private double cuota;

    public Curso() {

    }

    public Curso(int idCurso,String nombre, String fechaInicio, String
fechaTermino, double cuota) {
        this.idCurso=idCurso;
        this.nombre = nombre;
        this.fechaInicio = fechaInicio;
        this.fechaTermino = fechaTermino;
        this.cuota = cuota;
    }

    public int getIdCurso() {
        return idCurso;
    }

    public void setIdCurso(int curso) {
        this.idCurso = curso;
    }

    public String getNombre() {
        return nombre;
    }

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

    public String getFechaInicio() {
        return fechaInicio;
    }

    public void setFechaInicio(String fechaInicio) {
        this.fechaInicio = fechaInicio;
    }

    public String getFechaTermino() {
        return fechaTermino;
    }

    public void setFechaTermino(String fechaTermino) {
        this.fechaTermino = fechaTermino;
    }
}

```

```

    public double getCuota() {
        return cuota;
    }

    public void setCuota(double cuota) {
        this.cuota = cuota;
    }
}

```

Asignación.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modelos;

/**
 *
 * @author Lock
 */
public class Asignacion implements java.io.Serializable {

    int idAlumno,idCurso;
    String horario;
    int tipo;

    public Asignacion()
    {

    }

    public Asignacion(int idAlumno, int idCurso, String horario, int tipo) {
        this.idAlumno = idAlumno;
        this.idCurso = idCurso;
        this.horario = horario;
        this.tipo = tipo;
    }

    public int getIdAlumno() {
        return idAlumno;
    }

    public void setIdAlumno(int idAlumno) {
        this.idAlumno = idAlumno;
    }

    public int getIdCurso() {
        return idCurso;
    }

    public void setIdCurso(int idCurso) {
        this.idCurso = idCurso;
    }
}

```

```

    public String getHorario() {
        return horario;
    }

    public void setHorario(String horario) {
        this.horario = horario;
    }

    public int getTipo() {
        return tipo;
    }

    public void setTipo(int tipo) {
        this.tipo = tipo;
    }
}

```

Servidor.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package server;

import modelosCurso;
import modelos.Alumno;
import modelos.Asignacion;
import java.util.*;
import java.io.*;
import java.net.*;
import servicios.*;

/**
 *
 * @author Lock
 */
public class Servidor {

    private final static int PORT = 8200;

    public static void startServer() throws IOException {

        ServerSocket servidor = null;
        Socket socketCliente = null;
        ObjectOutputStream respuesta = null;
        ObjectInputStream peticion = null;

        try {

            servidor = new ServerSocket(PORT);
            System.out.println("Socket TCP iniciado. . .");

```

```

        Crud cru = new Crud(); //Objeto de tipo crud
        EmpacaBusqueda em = new EmpacaBusqueda(); //objeto usado para crear las
listas de busquedas y manipulacion de asignaciones
        ArrayList<Alumno> busquedaA; //Lista de estructuras de tipo alumno
        ArrayList<Curso> busquedaC; //Lista de estructuras de tipo curso
        ArrayList<Asignacion> busquedaS; //Lista de estructuras de tipo
asignacion
        Alumno al;
        Curso cu;
        Asignacion as;
        String tabla, accion, nombreA, nombreC;
        socketCliente = servidor.accept();
        respuesta = new ObjectOutputStream(socketCliente.getOutputStream());
        peticion = new ObjectInputStream(socketCliente.getInputStream());

        while ((tabla = (String) peticion.readObject()) != null) {

            if (tabla.equals("alumno")) {

                accion = (String) peticion.readObject();

                switch (accion) {

                    case "insertar":
                        al = (Alumno) peticion.readObject();
                        if (cru.insertA(al)) {
                            respuesta.writeObject(true);
                        } else {
                            respuesta.writeObject(false);
                        }
                        break;

                    case "editar":
                        al = (Alumno) peticion.readObject();
                        if (cru.updateA(al)) {
                            respuesta.writeObject(true);
                        } else {
                            respuesta.writeObject(false);
                        }
                        break;

                    case "borrar":
                        al = (Alumno) peticion.readObject();
                        if (cru.deleteA(al)) {
                            respuesta.writeObject(true);
                        } else {
                            respuesta.writeObject(false);
                        }
                        break;

                    case "buscar":

                        al = (Alumno) peticion.readObject();
                        al = cru.buscaNombreA(al.getIdAlumno());
                        respuesta.writeObject(al);

                        break;

                    case "listar":

```

```

        busquedaA = em.ListaTodosA();
        respuesta.writeObject(busquedaA);
        break;
    }

} else if (tabla.equals("curso")) {

    accion = (String) peticion.readObject();

    switch (accion) {
        case "insertar":
            cu = (Curso) peticion.readObject();
            //System.out.println(" a punto de insertar");
            if (cru.insertC(cu)) {
                respuesta.writeObject(true);
            } else {
                respuesta.writeObject(false);
            }
            break;

        case "editar":
            cu = (Curso) peticion.readObject();
            if (cru.updateC(cu)) {
                respuesta.writeObject(true);
            } else {
                respuesta.writeObject(false);
            }
            break;

        case "borrar":
            cu = (Curso) peticion.readObject();
            if (cru.deleteC(cu.getIdCurso())) {
                respuesta.writeObject(true);
            } else {
                respuesta.writeObject(false);
            }
            break;

        case "buscar":
            cu = (Curso) peticion.readObject();
            cu = cru.buscaNombreC(cu.getIdCurso());
            respuesta.writeObject(cu);
            break;

        case "listar":
            busquedaC = em.ListaTodosC();
            respuesta.writeObject(busquedaC);
            break;
    }

} else if (tabla.equals("asignacion")) {

    accion = (String) peticion.readObject();

    switch (accion) {
        case "insertar":
            as = (Asignacion) peticion.readObject();

```



```

        e.printStackTrace();
        System.out.println("Conexion finalizada");

        respuesta.close();
        peticion.close();
        socketCliente.close();
        servidor.close();

    }

}

public static void main(String[] args) throws IOException {
    int limite = 1000;
    int peticiones = 0;
    do {
        startServer();
    } while (peticiones++ < limite);

}

}

```

AlumnoRequest.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cliente;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import modelos.Alumno;

/**
 *
 * @author Steeltwins
 */
public class AlumnoRequest implements Runnable {
    GUI_Alumno parent;
    private Alumno alumno;
    private int operation;
    private char request;
    private final String IP_ADDRESS = "127.0.0.1";
    private final int PORT = 8200;

    public AlumnoRequest(GUI_Alumno parent, Alumno alumno, int operation, char request) {

```

```

        this.parent = parent;
        this.alumno = alumno;
        this.operation = operation;
        this.request = request;
    }

    public AlumnoRequest(GUI_Alumno parent, Alumno alumno, char request) {
        this.parent = parent;
        this.alumno = alumno;
        this.request = request;
    }

    public boolean sendUpdate() throws UnknownHostException, IOException {
        boolean exitoso = false;
        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            peticion.writeObject("alumno");

            //SE MANDA EL TIPO DE OPERACION
            if (this.operation == 1) {
                peticion.writeObject("insertar");
            } else {
                peticion.writeObject("editar");
            }
            peticion.writeObject(this.alumno); //Se manda el objeto
            System.out.println("Elemento enviado");
            exitoso = (boolean) respuesta.readObject();
            System.out.println("Respuesta devuelta");
            if (exitoso) {
                if (operation == 1) {
                    JOptionPane.showMessageDialog(null, "Insercion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);

                    } else {
                        JOptionPane.showMessageDialog(null, "Actualizacion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);
                    }
                    synchronized (this.parent) {
                        this.parent.getStudents();
                    }
                } else {
                    System.out.println("Respuesta false");
                    JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.ERROR_MESSAGE);
                }
                peticion.close();
                respuesta.close();
                socket.close();
                System.out.println("Conexion cerrada");
            } catch (UnknownHostException e) {
                e.printStackTrace();
                exitoso = false;
            }
        }
    }

```



```

    } catch (IOException e) {
        e.printStackTrace();
        exitoso = false;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        exitoso = false;
    }
    return exitoso;
}

public boolean sendDelete() {
    boolean exitoso = false;

    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("alumno");

        peticion.writeObject("borrar");//Se manda operacion
        peticion.writeObject(this.alumno);//Se manda objeto
        exitoso = (boolean) respuesta.readObject();
        if (exitoso) {
            JOptionPane.showMessageDialog(null, "Alumno Borrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.getStudents();
            }
        } else {
            JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.INFORMATION_MESSAGE);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
    return exitoso;
}

public Alumno sendBuscar() {
    Alumno a = new Alumno();

    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());

```

```

        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("alumno");

        peticion.writeObject("buscar");//Se manda operacion
        peticion.writeObject(this.alumno);//Se manda objeto
        a = (Alumno) respuesta.readObject();

        JOptionPane.showMessageDialog(null, "Alumno encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.setValuesSearched(a);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
    }
    return a;
}

public ArrayList<Alumno> sendListar() {
    ArrayList<Alumno> alumnos = new ArrayList<Alumno>();
    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("alumno");

        peticion.writeObject("listar");//Se manda operacion
        alumnos = (ArrayList<Alumno>) respuesta.readObject();
        //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.listStudents(alumnos);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {

```

```

        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
    return alumnos;
}

@Override
public void run() {
    try {
        if (this.request == 'u') {
            sendUpdate();
        } else if (this.request == 'd') {
            sendDelete();
        } else if (this.request == 'b') {
            sendBuscar();
        } else if (this.request == 'l') {
            sendListar();
        }
    } catch (IOException ex) {
        Logger.getLogger(AlumnoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

CursoRequest.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cliente;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import modelos.Curso;

/**
 *
 * @author Steeltwins
 */
public class CursoRequest implements Runnable{
    GUI_Curso parent;
    private Curso curso;
    private int operation;
    private char request;
    private final String IP_ADDRESS = "127.0.0.1";
    private final int PORT = 8200;

```

```

    public CursoRequest(GUI_Curso parent, Curso curso, int operation, char request)
    {
        this.parent = parent;
        this.curso = curso;
        this.operation = operation;
        this.request = request;
    }

    public CursoRequest(GUI_Curso parent, Curso curso, char request) {
        this.parent = parent;
        this.curso = curso;
        this.request = request;
    }

    public boolean sendUpdate() throws UnknownHostException, IOException {
        boolean exitoso = false;
        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            //System.out.println("Antes de enviar la operacion");
            peticion.writeObject("curso");

            //SE MANDA EL TIPO DE OPERACION
            if (this.operation == 1) {
                peticion.writeObject("insertar");
            } else {
                peticion.writeObject("editar");
            }
            peticion.writeObject(this.curso); //Se manda el objeto
            System.out.println("Elemento enviado");
            exitoso = (boolean) respuesta.readObject();
            System.out.println("Respuesta devuelta");
            if (exitoso) {
                if (operation == 1) {
                    JOptionPane.showMessageDialog(null, "Insercion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);

                    } else {
                        JOptionPane.showMessageDialog(null, "Actualizacion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);
                    }
                    synchronized (this.parent) {
                        this.parent.getCourses();
                    }
                } else {
                    System.out.println("Respuesta false");
                    JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.ERROR_MESSAGE);
                }
                peticion.close();
                respuesta.close();
                socket.close();
            }
        }
    }

```

```

        System.out.println("Conexion cerrada");
    } catch (UnknownHostException e) {
        e.printStackTrace();
        exitoso = false;
    } catch (IOException e) {
        e.printStackTrace();
        exitoso = false;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        exitoso = false;
    }
    return exitoso;
}

public boolean sendDelete() {
    boolean exitoso = false;

    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("curso");

        peticion.writeObject("borrar");//Se manda operacion
        peticion.writeObject(this.curso);//Se manda objeto
        exitoso = (boolean) respuesta.readObject();
        if (exitoso) {
            JOptionPane.showMessageDialog(null, "Alumno Borrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.getCourses();
            }
        } else {
            JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.INFORMATION_MESSAGE);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
    return exitoso;
}

public Curso sendBuscar() {
    Curso curso = new Curso();

```

```

        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            peticion.writeObject("curso");

            peticion.writeObject("buscar");//Se manda operacion
            peticion.writeObject(this.curso);//Se manda objeto
            curso = (Curso) respuesta.readObject();

            JOptionPane.showMessageDialog(null, "Curso encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.setValuesSearched(curso);
            }
            peticion.close();
            respuesta.close();
            socket.close();
            System.out.println("Conexion cerrada");

        } catch (IOException ex) {
            Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return curso;
    }

    public ArrayList<Curso> sendListar() {
        ArrayList<Curso> cursos = new ArrayList<Curso>();
        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            peticion.writeObject("curso");

            peticion.writeObject("listar");//Se manda operacion
            cursos = (ArrayList<Curso>) respuesta.readObject();
            //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.listCourses(cursos);
            }
            peticion.close();
            respuesta.close();
            socket.close();
            System.out.println("Conexion cerrada");

        } catch (IOException ex) {

```

```

        Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
    return cursors;
}

@Override
public void run() {
    try {
        if (this.request == 'u') {
            sendUpdate();
        } else if (this.request == 'd') {
            sendDelete();
        } else if (this.request == 'b') {
            sendBuscar();
        } else if (this.request == 'l') {
            sendListar();
        }
    } catch (IOException ex) {
        Logger.getLogger(CursoRequest.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

GUI_Curso.form

```

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.3" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
    <Properties>
        <Property name="defaultCloseOperation" type="int" value="3"/>
        <Property name="title" type="java.lang.String" value="Gestor de cursos"/>
    </Properties>
    <SyntheticProperties>
        <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
        <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
    </SyntheticProperties>
    <AuxValues>
        <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>
        <AuxValue name="FormSettings_autoSetComponentName" type="java.lang.Boolean"
value="false"/>
        <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>
        <AuxValue name="FormSettings_generateMnemonicsCode" type="java.lang.Boolean"
value="false"/>
        <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
        <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
    </AuxValues>
</Form>

```

```

    <AuxValue name="FormSettings_listenerGenerationStyle" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
  </AuxValues>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" attributes="0">
          <EmptySpace min="21" pref="21" max="-2" attributes="0"/>
          <Group type="103" groupAlignment="0" attributes="0">
            <Group type="102" attributes="0">
              <Group type="103" groupAlignment="1" attributes="0">
                <Group type="102" alignment="1" attributes="0">
                  <Component id="jLabel14" min="-2" max="-2"
attributes="0"/>
                  <EmptySpace type="separate" max="-2" attributes="0"/>
                  <Component id="txtCuotaRecuperacion" min="-2"
pref="104" max="-2" attributes="0"/>
                </Group>
                <Group type="102" alignment="1" attributes="0">
                  <Component id="jLabel13" min="-2" max="-2"
attributes="0"/>
                  <EmptySpace type="separate" max="-2" attributes="0"/>
                  <Component id="txtFechaFin" min="-2" pref="104"
max="-2" attributes="0"/>
                </Group>
                <Group type="102" alignment="1" attributes="0">
                  <Component id="jLabel2" min="-2" max="-2"
attributes="0"/>
                  <EmptySpace type="separate" max="-2" attributes="0"/>
                  <Component id="txtFechaInicio" min="-2" pref="104"
max="-2" attributes="0"/>
                </Group>
                <Group type="102" alignment="1" attributes="0">
                  <Component id="jLabel11" min="-2" max="-2"
attributes="0"/>
                  <EmptySpace type="separate" max="-2" attributes="0"/>
                  <Component id="txtNombre" min="-2" pref="104" max="-
2" attributes="0"/>
                </Group>
              </Group>
            <EmptySpace pref="269" max="32767" attributes="0"/>
            <Component id="jPanel4" min="-2" max="-2" attributes="0"/>
            <EmptySpace min="-2" pref="40" max="-2" attributes="0"/>
          </Group>
          <Group type="102" alignment="0" attributes="0">
            <Component id="jScrollPane1" min="-2" pref="482" max="-2"
attributes="0"/>
            <EmptySpace max="32767" attributes="0"/>
          </Group>
        </Group>
      </Group>
    </Layout>
  </Group>
</Group>

```



```

</DimensionLayout>
<DimensionLayout dim="1">
  <Group type="103" groupAlignment="0" attributes="0">
    <Group type="102" attributes="0">
      <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="0" attributes="0">
          <EmptySpace min="-2" pref="34" max="-2" attributes="0"/>
          <Group type="103" groupAlignment="3" attributes="0">
            <Component id="txtNombre" alignment="3" min="-2" max="-2"
attributes="0"/>
            <Component id="jLabel1" alignment="3" min="-2" max="-2"
attributes="0"/>
          </Group>
          <EmptySpace type="unrelated" max="-2" attributes="0"/>
          <Group type="103" groupAlignment="3" attributes="0">
            <Component id="txtFechaInicio" alignment="3" min="-2"
max="-2" attributes="0"/>
            <Component id="jLabel2" alignment="3" min="-2" max="-2"
attributes="0"/>
          </Group>
          <EmptySpace type="unrelated" max="-2" attributes="0"/>
          <Group type="103" groupAlignment="3" attributes="0">
            <Component id="txtFechaFin" alignment="3" min="-2" max="-2"
attributes="0"/>
            <Component id="jLabel3" alignment="3" min="-2" max="-2"
attributes="0"/>
          </Group>
          <EmptySpace type="unrelated" max="-2" attributes="0"/>
          <Group type="103" groupAlignment="3" attributes="0">
            <Component id="txtCuotaRecuperacion" alignment="3" min="-2"
max="-2" attributes="0"/>
            <Component id="jLabel4" alignment="3" min="-2" max="-2"
attributes="0"/>
          </Group>
          <EmptySpace min="-2" pref="52" max="-2" attributes="0"/>
          <Component id="jScrollPane1" min="-2" pref="248" max="-2"
attributes="0"/>
        </Group>
        <Group type="102" alignment="0" attributes="0">
          <EmptySpace min="-2" pref="46" max="-2" attributes="0"/>
          <Component id="jPanel4" min="-2" max="-2" attributes="0"/>
        </Group>
      </Group>
      <EmptySpace pref="45" max="32767" attributes="0"/>
    </Group>
  </DimensionLayout>
</Layout>
<SubComponents>
  <Component class="javax.swing.JLabel" name="jLabel1">
    <Properties>
      <Property name="text" type="java.lang.String" value="Nombre"/>
    </Properties>
  </Component>
  <Component class="javax.swing.JTextField" name="txtNombre">
  </Component>
  <Component class="javax.swing.JLabel" name="jLabel2">

```

```

    <Properties>
      <Property name="text" type="java.lang.String" value="Fecha inicio"/>
    </Properties>
  </Component>
  <Component class="javax.swing.JTextField" name="txtFechaInicio">
  </Component>
  <Component class="javax.swing.JLabel" name="jLabel3">
    <Properties>
      <Property name="text" type="java.lang.String" value="Fecha fin"/>
    </Properties>
  </Component>
  <Component class="javax.swing.JTextField" name="txtFechaFin">
  </Component>
  <Component class="javax.swing.JLabel" name="jLabel4">
    <Properties>
      <Property name="text" type="java.lang.String" value="Cuota de
recuperacion"/>
    </Properties>
  </Component>
  <Component class="javax.swing.JTextField" name="txtCuotaRecuperacion">
  </Component>
  <Container class="javax.swing.JPanel" name="jPanel4">
    <Properties>
      <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
      <Border info="org.netbeans.modules.form.compat2.border.TitledBorderInfo">
        <TitledBorder title="Opciones"/>
      </Border>
    </Property>
  </Properties>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="0" attributes="0">
          <EmptySpace max="-2" attributes="0"/>
          <Group type="103" groupAlignment="0" attributes="0">
            <Component id="btnActualizaCurso" alignment="0" min="-2"
max="-2" attributes="0"/>
            <Component id="btnEliminarCurso" alignment="0" min="-2"
max="-2" attributes="0"/>
            <Component id="btnBuscarCurso" alignment="0" min="-2" max="-
2" attributes="0"/>
            <Component id="btnNuevoCurso" alignment="0" min="-2" max="-2"
attributes="0"/>
            <Group type="102" attributes="0">
              <EmptySpace min="10" pref="10" max="-2" attributes="0"/>
              <Component id="btnListarCursos" min="-2" max="-2"
attributes="0"/>
            </Group>
          </Group>
        </Group>
      <EmptySpace pref="11" max="32767" attributes="0"/>
    </Group>
  </DimensionLayout>
  <DimensionLayout dim="1">
    <Group type="103" groupAlignment="0" attributes="0">

```

```

        <Group type="102" alignment="0" attributes="0">
            <EmptySpace min="-2" pref="21" max="-2" attributes="0"/>
            <Component id="btnNuevoCurso" min="-2" max="-2" attributes="0"/>
            <EmptySpace type="unrelated" max="-2" attributes="0"/>
            <Component id="btnBuscarCurso" min="-2" max="-2" attributes="0"/>
            <EmptySpace max="-2" attributes="0"/>
            <Component id="btnEliminarCurso" min="-2" max="-2"
attributes="0"/>
            <EmptySpace type="unrelated" max="-2" attributes="0"/>
            <Component id="btnActualizaCurso" min="-2" max="-2"
attributes="0"/>
            <EmptySpace type="unrelated" max="-2" attributes="0"/>
            <Component id="btnListarCursos" min="-2" max="-2"
attributes="0"/>
            <EmptySpace pref="35" max="32767" attributes="0"/>
        </Group>
    </Group>
</DimensionLayout>
</Layout>
<SubComponents>
    <Component class="javax.swing.JButton" name="btnNuevoCurso">
        <Properties>
            <Property name="text" type="java.lang.String" value="Nuevo"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnNuevoCursoActionPerformed"/>
        </Events>
    </Component>
    <Component class="javax.swing.JButton" name="btnBuscarCurso">
        <Properties>
            <Property name="text" type="java.lang.String" value="Buscar"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnBuscarCursoActionPerformed"/>
        </Events>
    </Component>
    <Component class="javax.swing.JButton" name="btnEliminarCurso">
        <Properties>
            <Property name="text" type="java.lang.String" value="Eliminar"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnEliminarCursoActionPerformed"/>
        </Events>
    </Component>
    <Component class="javax.swing.JButton" name="btnActualizaCurso">
        <Properties>
            <Property name="text" type="java.lang.String" value="Actualiza"/>
        </Properties>
        <Events>

```

```

        <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnActualizarCursoActionPerformed"/>
    </Events>
</Component>
<Component class="javax.swing.JButton" name="btnListarCursos">
    <Properties>
        <Property name="text" type="java.lang.String" value="Listar"/>
    </Properties>
    <Events>
        <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnListarCursosActionPerformed"/>
    </Events>
</Component>
</SubComponents>
</Container>
<Container class="javax.swing.JScrollPane" name="jScrollPane1">
    <AuxValues>
        <AuxValue name="autoScrollPane" type="java.lang.Boolean" value="true"/>
    </AuxValues>

    <Layout
class="org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout"/
>
    <SubComponents>
        <Component class="javax.swing.JTable" name="tblShowTable">
            <Properties>
                <Property name="model" type="javax.swing.table.TableModel"
editor="org.netbeans.modules.form.editors2.TableModelEditor">
                    <Table columnCount="5" rowCount="4">
                        <Column editable="true" title="idCurso" type="java.lang.Object"/>
                        <Column editable="true" title="Nombre" type="java.lang.Object"/>
                        <Column editable="true" title="Fecha Inicio"
type="java.lang.Object"/>
                        <Column editable="true" title="Fecha Fin" type="java.lang.Object"/>
                        <Column editable="true" title="Cuota de recuperacion"
type="java.lang.Object"/>
                    </Table>
                </Property>
                <Property name="columnModel" type="javax.swing.table.TableColumnModel"
editor="org.netbeans.modules.form.editors2.TableColumnModelEditor">
                    <TableColumnModel selectionModel="0">
                        <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                            <Title/>
                            <Editor/>
                            <Renderer/>
                        </Column>
                        <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                            <Title/>
                            <Editor/>
                            <Renderer/>
                        </Column>
                        <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">

```

```

                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
        </TableColumnModel>
    </Property>
    <Property name="tableHeader" type="javax.swing.table.JTableHeader"
editor="org.netbeans.modules.form.editors2.JTableHeaderEditor">
        <TableHeader reorderingAllowed="true" resizingAllowed="true"/>
    </Property>
</Properties>
</Component>
</SubComponents>
</Container>
</SubComponents>
</Form>

```

GUI_Curso.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cliente;

import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import modelos.Curso;

/**
 *
 * @author Steeltwins
 */
public class GUI_Curso extends javax.swing.JFrame {

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

    public void getCourses() {

```

```

        CursoRequest cliente;
        Curso curso = new Curso();
        cliente = new CursoRequest(this, curso, 'l');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }

    public void listCourses(ArrayList<Curso> cursos) {
        Object[][] courses = new Object[cursos.size()][5];
        String[] columns = {"idCurso", "Nombre", "Fecha de Inicio", "Fecha Fin",
"Cuota de Recuperacion"};

        for (int i = 0, j = 0; i < cursos.size(); i++) {
            courses[i][j] = "" + cursos.get(i).getIdCurso();
            courses[i][j+1] = "" + cursos.get(i).getNombre();
            courses[i][j + 2] = "" + cursos.get(i).getFechaInicio();
            courses[i][j + 3] = "" + cursos.get(i).getFechaTermino();
            courses[i][j + 4] = "" + cursos.get(i).getCuota();
        }
        tblShowTable.setModel(new DefaultTableModel(courses, columns));
    }

    public void setValuesSearched(Curso curso) {
        txtNombre.setText("" + curso.getNombre());
        txtFechaInicio.setText(curso.getFechaInicio());
        txtFechaFin.setText(curso.getFechaTermino());
        txtCuotaRecuperacion.setText(""+curso.getCuota());
    }

    public Curso buscarCurso() {
        CursoRequest cliente;
        Curso curso = new Curso();
        int idCurso = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingrese
el id del Curso ", "Buscar curso", JOptionPane.QUESTION_MESSAGE));
        curso.setIdCurso(idCurso);
        cliente = new CursoRequest(this, curso, 'b');
        Thread peticion = new Thread(cliente);
        peticion.start();
        return curso;
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        txtNombre = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        txtFechaInicio = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();

```

```

txtFechaFin = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
txtCuotaRecuperacion = new javax.swing.JTextField();
jPanel4 = new javax.swing.JPanel();
btnNuevoCurso = new javax.swing.JButton();
btnBuscarCurso = new javax.swing.JButton();
btnEliminarCurso = new javax.swing.JButton();
btnActualizaCurso = new javax.swing.JButton();
btnListarCursos = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
tblShowTable = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Gestor de cursos");

jLabel1.setText("Nombre");

jLabel2.setText("Fecha inicio");

jLabel3.setText("Fecha fin");

jLabel4.setText("Cuota de recuperacion");

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder("Opciones"));

btnNuevoCurso.setText("Nuevo");
btnNuevoCurso.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnNuevoCursoActionPerformed(evt);
    }
});

btnBuscarCurso.setText("Buscar");
btnBuscarCurso.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarCursoActionPerformed(evt);
    }
});

btnEliminarCurso.setText("Eliminar");
btnEliminarCurso.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEliminarCursoActionPerformed(evt);
    }
});

btnActualizaCurso.setText("Actualiza");
btnActualizaCurso.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizaCursoActionPerformed(evt);
    }
});

btnListarCursos.setText("Listar");
btnListarCursos.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```



```

jScrollPane1.setViewportView(tblShowTable);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(21, 21, 21)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel4)
        .addGap(18, 18, 18)
        .addComponent(txtCuotaRecuperacion,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel3)
            .addGap(18, 18, 18)
            .addComponent(txtFechaFin,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(18, 18, 18)
                .addComponent(txtFechaInicio,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addGap(18, 18, 18)
                    .addComponent(txtNombre,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 269,
Short.MAX_VALUE)
        .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(40, 40, 40)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 482,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())
        .addGap(34, 34, 34)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtNombre,
            javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel1))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtFechaInicio,
            javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtFechaFin,
            javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtCuotaRecuperacion,
            javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel4))
        .addGap(52, 52, 52)
        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.PREFERRED_SIZE, 248,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addGap(46, 46, 46)
        .addComponent(jPanel4,
            javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(45, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void btnNuevoCursoActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btnNuevoCursoActionPerformed
    CursoRequest cliente;
    Curso curso = new Curso();
    curso.setNombre(txtNombre.getText());
    curso.setFechaInicio(txtFechaInicio.getText());
    curso.setFechaTermino(txtFechaFin.getText());
    curso.setCuota(Double.parseDouble(txtCuotaRecuperacion.getText()));
    int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
insertar el curso?", "Insertar nuevo curso", JOptionPane.YES_NO_OPTION);

```

```

        if (option == 0) {
            cliente = new CursoRequest(this, curso, 1, 'u');
            Thread peticion=new Thread(cliente);
            peticion.start();
        }
    }
}

//GEN-LAST:event_btnNuevoCursoActionPerformed

private void btnBuscarCursoActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btnBuscarCursoActionPerformed
        Curso alumno = new Curso();
        alumno = buscarCurso();
    }
    //GEN-LAST:event_btnBuscarCursoActionPerformed

private void btnEliminarCursoActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btnEliminarCursoActionPerformed
        CursoRequest cliente;
        Curso curso = new Curso();
        curso = buscarCurso();
        int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
eliminar al curso?", "Eliminar curso", JOptionPane.YES_NO_OPTION);
        if (option == 0) {
            cliente = new CursoRequest(this, curso, 'd');
            Thread peticion = new Thread(cliente);
            peticion.start();
        }
    }
    //GEN-LAST:event_btnEliminarCursoActionPerformed

private void btnActualizaCursoActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btnActualizaCursoActionPerformed
        Curso curso = new Curso();
        CursoRequest cliente;
        curso = buscarCurso();
        curso.setNombre(txtNombre.getText());
        curso.setFechaInicio(txtFechaInicio.getText());
        curso.setFechaTermino(txtFechaFin.getText());
        curso.setCuota(Double.parseDouble(txtCuotaRecuperacion.getText()));

        int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
actualizar el curso?", "Actualizar curso", JOptionPane.YES_NO_OPTION);
        if (option == 0) {

            cliente = new CursoRequest(this, curso, 2, 'u');//Primer
parametro=datos,Segundo parametro: 1=insert 2=update,Tercer
parametro:'u'=update&insert 'd'=delete 'b'=buscar 'l'=listar
            Thread peticion = new Thread(cliente);
            peticion.start();//Peticion asincrona

        }
    }
    //GEN-LAST:event_btnActualizaCursoActionPerformed

private void btnListarCursosActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btnListarCursosActionPerformed
        getCourses();
    }
    //GEN-LAST:event_btnListarCursosActionPerformed

/**
 * @param args the command line arguments

```

```

    */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(GUI_Curso.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(GUI_Curso.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(GUI_Curso.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(GUI_Curso.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
        }
    } //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new GUI_Curso().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnActualizarCurso;
private javax.swing.JButton btnBuscarCurso;
private javax.swing.JButton btnEliminarCurso;
private javax.swing.JButton btnListarCursos;
private javax.swing.JButton btnNuevoCurso;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel4;
private javax.swing.JScrollPane jScrollPane1;

```

```

private javax.swing.JTable tblShowTable;
private javax.swing.JTextField txtCuotaRecuperacion;
private javax.swing.JTextField txtFechaFin;
private javax.swing.JTextField txtFechaInicio;
private javax.swing.JTextField txtNombre;
// End of variables declaration//GEN-END:variables
}

```

GUI_Alumno.form

```

<?xml version="1.0" encoding="UTF-8" ?>

<Form version="1.5" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <Properties>
    <Property name="defaultCloseOperation" type="int" value="3"/>
    <Property name="title" type="java.lang.String" value="Gestor de alumnos"/>
  </Properties>
  <SyntheticProperties>
    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
    <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
  </SyntheticProperties>
  <AuxValues>
    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_autoSetComponentName" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>
    <AuxValue name="FormSettings_generateMnemonicsCode" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
  </AuxValues>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="0" attributes="0">
          <EmptySpace min="-2" pref="27" max="-2" attributes="0"/>
          <Group type="103" groupAlignment="1" attributes="0">
            <Group type="102" alignment="1" attributes="0">
              <Component id="jLabel6" min="-2" max="-2" attributes="0"/>
              <EmptySpace type="separate" max="-2" attributes="0"/>
              <Component id="txtNumero" min="-2" pref="101" max="-2"
attributes="0"/>
            </Group>
            <Group type="102" alignment="1" attributes="0">
              <Component id="jLabel5" min="-2" max="-2" attributes="0"/>

```

```

        <EmptySpace type="separate" max="-2" attributes="0"/>
        <Component id="txtCalle" min="-2" pref="101" max="-2"
attributes="0"/>
    </Group>
    <Group type="102" alignment="1" attributes="0">
        <Component id="jLabel4" min="-2" max="-2" attributes="0"/>
        <EmptySpace type="separate" max="-2" attributes="0"/>
        <Component id="txtEmail" min="-2" pref="101" max="-2"
attributes="0"/>
    </Group>
    <Group type="102" alignment="1" attributes="0">
        <Component id="jLabel3" min="-2" max="-2" attributes="0"/>
        <EmptySpace type="separate" max="-2" attributes="0"/>
        <Component id="txtApellidoM" min="-2" pref="101" max="-2"
attributes="0"/>
    </Group>
    <Group type="102" alignment="1" attributes="0">
        <Component id="jLabel2" min="-2" max="-2" attributes="0"/>
        <EmptySpace type="separate" max="-2" attributes="0"/>
        <Component id="txtApellidoP" min="-2" pref="101" max="-2"
attributes="0"/>
    </Group>
    <Group type="102" alignment="1" attributes="0">
        <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
        <EmptySpace type="separate" max="-2" attributes="0"/>
        <Component id="txtNombreAlumno" min="-2" pref="101" max="-2"
attributes="0"/>
    </Group>
    <Group type="102" alignment="1" attributes="0">
        <Group type="103" groupAlignment="1" attributes="0">
            <Component id="jLabel10" min="-2" max="-2"
attributes="0"/>
            <Component id="jLabel8" min="-2" max="-2"
attributes="0"/>
            <Component id="jLabel7" min="-2" max="-2"
attributes="0"/>
            <Component id="jLabel9" alignment="1" min="-2" max="-2"
attributes="0"/>
        </Group>
        <EmptySpace type="separate" max="-2" attributes="0"/>
        <Group type="103" groupAlignment="0" attributes="0">
            <Group type="103" alignment="1" groupAlignment="0" max="-
2" attributes="0">
                <Component id="txtEntidad" alignment="0" max="32767"
attributes="0"/>
                <Component id="txtTelefono" alignment="0" min="-2"
pref="101" max="-2" attributes="0"/>
            </Group>
            <Group type="103" alignment="0" groupAlignment="0" max="-
2" attributes="0">
                <Component id="txtColonia" max="32767"
attributes="0"/>
                <Component id="txtDelegacion" min="-2" pref="101"
max="-2" attributes="0"/>
            </Group>
        </Group>
    </Group>

```

```

        </Group>
        <EmptySpace max="32767" attributes="0"/>
        <Component id="jPanel2" min="-2" max="-2" attributes="0"/>
        <EmptySpace min="-2" pref="51" max="-2" attributes="0"/>
    </Group>
    <Group type="102" alignment="0" attributes="0">
        <EmptySpace max="-2" attributes="0"/>
        <Component id="jScrollPane1" pref="646" max="32767" attributes="0"/>
        <EmptySpace max="-2" attributes="0"/>
    </Group>
</Group>
</DimensionLayout>
<DimensionLayout dim="1">
    <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="0" attributes="0">
            <EmptySpace min="-2" pref="26" max="-2" attributes="0"/>
            <Group type="103" groupAlignment="1" attributes="0">
                <Group type="102" attributes="0">
                    <Group type="103" groupAlignment="0" attributes="0">
                        <Component id="txtNombreAlumno" alignment="0" min="-2"
max="-2" attributes="0"/>
                        <Component id="jLabel1" alignment="1" min="-2" max="-2"
attributes="0"/>
                    </Group>
                    <EmptySpace type="unrelated" max="-2" attributes="0"/>
                    <Group type="103" groupAlignment="3" attributes="0">
                        <Component id="txtApellidoP" alignment="3" min="-2"
max="-2" attributes="0"/>
                        <Component id="jLabel2" alignment="3" min="-2" max="-2"
attributes="0"/>
                    </Group>
                    <EmptySpace max="-2" attributes="0"/>
                    <Group type="103" groupAlignment="3" attributes="0">
                        <Component id="txtApellidoM" alignment="3" min="-2"
max="-2" attributes="0"/>
                        <Component id="jLabel3" alignment="3" min="-2" max="-2"
attributes="0"/>
                    </Group>
                    <EmptySpace max="-2" attributes="0"/>
                    <Group type="103" groupAlignment="3" attributes="0">
                        <Component id="txtEmail" alignment="3" min="-2" max="-2"
attributes="0"/>
                        <Component id="jLabel4" alignment="3" min="-2" max="-2"
attributes="0"/>
                    </Group>
                    <EmptySpace max="-2" attributes="0"/>
                    <Group type="103" groupAlignment="3" attributes="0">
                        <Component id="txtCalle" alignment="3" min="-2" max="-2"
attributes="0"/>
                        <Component id="jLabel5" alignment="3" min="-2" max="-2"
attributes="0"/>
                    </Group>
                    <EmptySpace max="-2" attributes="0"/>
                    <Group type="103" groupAlignment="3" attributes="0">
                        <Component id="txtNumero" alignment="3" min="-2" max="-2"
attributes="0"/>

```

```

        <Component id="jLabel6" alignment="3" min="-2" max="-2"
attributes="0"/>
    </Group>
    <Group type="103" groupAlignment="0" attributes="0">
        <Group type="102" alignment="0" attributes="0">
            <EmptySpace max="-2" attributes="0"/>
            <Component id="txtColonia" min="-2" max="-2"
attributes="0"/>
            <EmptySpace type="unrelated" max="-2"
attributes="0"/>
            <Component id="txtDelegacion" min="-2" max="-2"
attributes="0"/>
            <EmptySpace max="-2" attributes="0"/>
            <Group type="103" groupAlignment="3" attributes="0">
                <Component id="txtEntidad" alignment="3" min="-2"
max="-2" attributes="0"/>
                <Component id="jLabel9" alignment="3" min="-2"
max="-2" attributes="0"/>
            </Group>
        </Group>
    </Group>
    <Group type="102" alignment="0" attributes="0">
        <EmptySpace min="-2" pref="14" max="-2"
attributes="0"/>
        <Component id="jLabel7" min="-2" max="-2"
attributes="0"/>
        <EmptySpace type="unrelated" max="-2"
attributes="0"/>
        <Component id="jLabel8" min="-2" max="-2"
attributes="0"/>
    </Group>
</Group>
<Component id="jPanel2" min="-2" max="-2" attributes="0"/>
</Group>
<EmptySpace type="unrelated" max="-2" attributes="0"/>
<Group type="103" groupAlignment="3" attributes="0">
    <Component id="txtTelefono" alignment="3" min="-2" max="-2"
attributes="0"/>
    <Component id="jLabel10" alignment="3" min="-2" max="-2"
attributes="0"/>
</Group>
<EmptySpace min="-2" pref="42" max="-2" attributes="0"/>
<Component id="jScrollPane1" pref="164" max="32767" attributes="0"/>
<EmptySpace min="-2" pref="68" max="-2" attributes="0"/>
</Group>
</Group>
</DimensionLayout>
</Layout>
<SubComponents>
    <Container class="javax.swing.JPanel" name="jPanel2">
        <Properties>
            <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
                <Border info="org.netbeans.modules.form.compat2.border.TitledBorderInfo">
                    <TitledBorder title="Opciones"/>
                </Border>
            </Property>

```



```

</Properties>

<Layout>
  <DimensionLayout dim="0">
    <Group type="103" groupAlignment="0" attributes="0">
      <Group type="102" alignment="0" attributes="0">
        <EmptySpace max="-2" attributes="0"/>
        <Group type="103" groupAlignment="0" attributes="0">
          <Component id="btnListarAlumnos" alignment="0" min="-2"
max="-2" attributes="0"/>
          <Component id="btnActualizaAlumno" alignment="0" min="-2"
max="-2" attributes="0"/>
          <Component id="btnEliminarAlumno" alignment="0" min="-2"
max="-2" attributes="0"/>
          <Component id="btnBuscarAlumno" alignment="0" min="-2" max="-
2" attributes="0"/>
          <Component id="btnNuevoAlumno" alignment="0" min="-2" max="-
2" attributes="0"/>
        </Group>
        <EmptySpace pref="11" max="32767" attributes="0"/>
      </Group>
    </Group>
  </DimensionLayout>
  <DimensionLayout dim="1">
    <Group type="103" groupAlignment="0" attributes="0">
      <Group type="102" alignment="0" attributes="0">
        <EmptySpace min="-2" pref="21" max="-2" attributes="0"/>
        <Component id="btnNuevoAlumno" min="-2" max="-2" attributes="0"/>
        <EmptySpace type="unrelated" max="-2" attributes="0"/>
        <Component id="btnBuscarAlumno" min="-2" max="-2"
attributes="0"/>
        <EmptySpace max="-2" attributes="0"/>
        <Component id="btnEliminarAlumno" min="-2" max="-2"
attributes="0"/>
        <EmptySpace type="unrelated" max="-2" attributes="0"/>
        <Component id="btnActualizaAlumno" min="-2" max="-2"
attributes="0"/>
        <EmptySpace type="unrelated" max="-2" attributes="0"/>
        <Component id="btnListarAlumnos" min="-2" max="-2"
attributes="0"/>
        <EmptySpace pref="35" max="32767" attributes="0"/>
      </Group>
    </Group>
  </DimensionLayout>
</Layout>
<SubComponents>
  <Component class="javax.swing.JButton" name="btnNuevoAlumno">
    <Properties>
      <Property name="text" type="java.lang.String" value="Nuevo"/>
    </Properties>
    <Events>
      <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnNuevoAlumnoActionPerformed"/>
    </Events>
  </Component>
  <Component class="javax.swing.JButton" name="btnBuscarAlumno">

```

```

        <Properties>
            <Property name="text" type="java.lang.String" value="Buscar"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnBuscarAlumnoActionPerformed"/>
        </Events>
    </Component>
    <Component class="javax.swing.JButton" name="btnEliminarAlumno">
        <Properties>
            <Property name="text" type="java.lang.String" value="Eliminar"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnEliminarAlumnoActionPerformed"/>
        </Events>
    </Component>
    <Component class="javax.swing.JButton" name="btnActualizaAlumno">
        <Properties>
            <Property name="text" type="java.lang.String" value="Actualiza"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnActualizaAlumnoActionPerformed"/>
        </Events>
    </Component>
    <Component class="javax.swing.JButton" name="btnListarAlumnos">
        <Properties>
            <Property name="text" type="java.lang.String" value="Listar"/>
        </Properties>
        <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="btnListarAlumnosActionPerformed"/>
        </Events>
    </Component>
</SubComponents>
</Container>
<Component class="javax.swing.JLabel" name="jLabel5">
    <Properties>
        <Property name="text" type="java.lang.String" value="Calle"/>
    </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtCalle">
</Component>
<Component class="javax.swing.JLabel" name="jLabel6">
    <Properties>
        <Property name="text" type="java.lang.String" value="Numero"/>
    </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtApellidoP">
</Component>
<Component class="javax.swing.JTextField" name="txtNumero">
</Component>

```

```

<Component class="javax.swing.JLabel" name="jLabel1">
  <Properties>
    <Property name="text" type="java.lang.String" value="Nombre"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="jLabel7">
  <Properties>
    <Property name="text" type="java.lang.String" value="Colonia"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtNombreAlumno">
</Component>
<Component class="javax.swing.JTextField" name="txtColonia">
</Component>
<Component class="javax.swing.JLabel" name="jLabel2">
  <Properties>
    <Property name="text" type="java.lang.String" value="Apellido Paterno"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="jLabel8">
  <Properties>
    <Property name="text" type="java.lang.String" value="Delegacion"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtApellidoM">
</Component>
<Component class="javax.swing.JTextField" name="txtDelegacion">
</Component>
<Component class="javax.swing.JLabel" name="jLabel3">
  <Properties>
    <Property name="text" type="java.lang.String" value="Apellido Materno"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtEmail">
</Component>
<Component class="javax.swing.JLabel" name="jLabel4">
  <Properties>
    <Property name="text" type="java.lang.String" value="Email"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="jLabel9">
  <Properties>
    <Property name="text" type="java.lang.String" value="Entidad"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtEntidad">
</Component>
<Component class="javax.swing.JLabel" name="jLabel10">
  <Properties>
    <Property name="text" type="java.lang.String" value="Telefono"/>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="txtTelefono">
</Component>
<Container class="javax.swing.JScrollPane" name="jScrollPane1">
  <AuxValues>
    <AuxValue name="autoScrollPane" type="java.lang.Boolean" value="true"/>
  </AuxValues>

```



```

                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            <Column maxWidth="-1" minWidth="-1" prefWidth="-1"
resizable="true">
                <Title/>
                <Editor/>
                <Renderer/>
            </Column>
            </TableColumnModel>
        </Property>
        <Property name="tableHeader" type="javax.swing.table.JTableHeader"
editor="org.netbeans.modules.form.editors2.JTableHeaderEditor">
            <TableHeader reorderingAllowed="true" resizingAllowed="true"/>
        </Property>
    </Properties>
</Component>
</SubComponents>
</Container>
</SubComponents>
</Form>

```

GUI_Alumno.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates

```

```

    * and open the template in the editor.
    */
package cliente;

import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import modelos.Alumno;

/**
 *
 * @author Steeltwins
 */
public class GUI_Alumno extends javax.swing.JFrame {

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

    public void getStudents() {
        AlumnoRequest cliente;
        Alumno alumno = new Alumno();
        cliente = new AlumnoRequest(this, alumno, '1');
        Thread petition = new Thread(cliente);
        petition.start();
    }

    public void listStudents(ArrayList<Alumno> alumnos) {
        Object[][] students = new Object[alumnos.size()][11];
        String[] columns = {"idAlumno", "Nombre", "Apellido Paterno", "Apellido Materno", "Email", "Calle", "Numero", "Colonia", "Delegacion", "Entidad", "Telefono"};

        for (int i = 0, j = 0; i < alumnos.size(); i++) {
            students[i][j] = "" + alumnos.get(i).getIdAlumno();
            students[i][j + 1] = "" + alumnos.get(i).getNombre();
            students[i][j + 2] = "" + alumnos.get(i).getApellidoPaterno();
            students[i][j + 3] = "" + alumnos.get(i).getApellidoMaterno();
            students[i][j + 4] = "" + alumnos.get(i).getMail();
            students[i][j + 5] = "" + alumnos.get(i).getCalle();
            students[i][j + 6] = "" + alumnos.get(i).getNumero();
            students[i][j + 7] = "" + alumnos.get(i).getColonia();
            students[i][j + 8] = "" + alumnos.get(i).getDelegación();
            students[i][j + 9] = "" + alumnos.get(i).getEntidad();
            students[i][j + 10] = "" + alumnos.get(i).getTelefono();
        }
        tblShowTable.setModel(new DefaultTableModel(students, columns));
    }

    public void setValuesSearched(Alumno alumno) {
        txtNombreAlumno.setText("" + alumno.getNombre());
        txtApellidoP.setText(alumno.getApellidoPaterno());
        txtApellidoM.setText(alumno.getApellidoMaterno());
    }
}

```

```

        txtEmail.setText(alumno.getMail());
        txtCalle.setText(alumno.getCalle());
        txtNumero.setText(alumno.getNumero());
        txtColonia.setText(alumno.getColonia());
        txtDelegacion.setText(alumno.getDelegación());
        txtEntidad.setText(alumno.getEntidad());
        txtTelefono.setText(alumno.getTelefono());
    }

    public Alumno buscarAlumno() {
        AlumnoRequest cliente;
        Alumno alumno = new Alumno();
        int idAlumno = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingrese
el nombre del Alumno ", "Buscar alumno", JOptionPane.QUESTION_MESSAGE));
        alumno.setIdAlumno(idAlumno);
        cliente = new AlumnoRequest(this, alumno, 'b');
        Thread peticion = new Thread(cliente);
        peticion.start();
        return alumno;
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        btnNuevoAlumno = new javax.swing.JButton();
        btnBuscarAlumno = new javax.swing.JButton();
        btnEliminarAlumno = new javax.swing.JButton();
        btnActualizaAlumno = new javax.swing.JButton();
        btnListarAlumnos = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
        txtCalle = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        txtApellidoP = new javax.swing.JTextField();
        txtNumero = new javax.swing.JTextField();
        jLabel11 = new javax.swing.JLabel();
        jLabel17 = new javax.swing.JLabel();
        txtNombreAlumno = new javax.swing.JTextField();
        txtColonia = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        txtApellidoM = new javax.swing.JTextField();
        txtDelegacion = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        txtEmail = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        jLabel19 = new javax.swing.JLabel();
        txtEntidad = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();

```



```

        .addComponent(btnActualizarAlumno)
        .addComponent(btnEliminarAlumno)
        .addComponent(btnBuscarAlumno)
        .addComponent(btnNuevoAlumno)
        .addContainerGap(11, Short.MAX_VALUE))
    );
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(21, 21, 21)
        .addComponent(btnNuevoAlumno)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnBuscarAlumno)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnEliminarAlumno)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnActualizarAlumno)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnListarAlumnos)
        .addContainerGap(35, Short.MAX_VALUE))
    );

    jLabel5.setText("Calle");
    jLabel6.setText("Numero");
    jLabel11.setText("Nombre");
    jLabel7.setText("Colonia");
    jLabel12.setText("Apellido Paterno");
    jLabel8.setText("Delegacion");
    jLabel3.setText("Apellido Materno");
    jLabel4.setText("Email");
    jLabel19.setText("Entidad");
    jLabel10.setText("Telefono");

tblShowTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null, null, null, null, null}
    },
    new String [] {
        "IdAlumno", "Nombre", "Apellido Paterno", "Apellido Materno",
        "Email", "Calle", "Numero", "Colonia", "Delegacion", "Entidad", "Telefono"
    }

```

```

    }
    ));
    jScrollPane1.setViewportView(tblShowTable);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel6)
                .addGap(18, 18, 18)
                .addComponent(txtNumero,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel5)
                    .addGap(18, 18, 18)
                    .addComponent(txtCalle,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel4)
                    .addGap(18, 18, 18)
                    .addComponent(txtEmail,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel3)
                    .addGap(18, 18, 18)
                    .addComponent(txtApellidoM,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addGap(18, 18, 18)
                    .addComponent(txtApellidoP,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel11)
                    .addGap(18, 18, 18)
                    .addComponent(txtNombreAlumno,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel10)
                    .addComponent(jLabel8)
                    .addComponent(jLabel7)
                    .addComponent(jLabel9))
                .addGap(18, 18, 18)
            )
        )
    );

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(txtEntidad)
                .addComponent(txtTelefono,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
            .addComponent(txtColonia)
            .addComponent(txtDelegacion,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(51, 51, 51))
        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
646, Short.MAX_VALUE)
            .addContainerGap()
        );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup())
                .addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(txtNombreAlumno,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.TRAILING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtApellidoP,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtApellidoM,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel3))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtEmail,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel14))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtCalle,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel15))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtNumero,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel16))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txtColonia,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(txtDelegacion,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtEntidad,
        javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel19)))
    .addGroup(layout.createSequentialGroup()
        .addGap(14, 14, 14)
        .addComponent(jLabel17))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabel18)))
    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(txtTelefono,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel10))
        .addGap(42, 42, 42)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
164, Short.MAX_VALUE)
        .addGap(68, 68, 68))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void btnNuevoAlumnoActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btnNuevoAlumnoActionPerformed
    AlumnoRequest cliente;
    Alumno alumno = new Alumno();
    alumno.setNombre(txtNombreAlumno.getText());
    alumno.setApellidoPaterno(txtApellidoP.getText());
    alumno.setApellidoMaterno(txtApellidoM.getText());
    alumno.setMail(txtEmail.getText());
    alumno.setCalle(txtCalle.getText());
    alumno.setNumero(txtNumero.getText());
    alumno.setColonia(txtColonia.getText());
    alumno.setDelegación(txtDelegacion.getText());
    alumno.setEntidad(txtEntidad.getText());
    alumno.setTelefono(txtTelefono.getText());
    int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
insertar al alumno?", "Insertar nuevo alumno", JOptionPane.YES_NO_OPTION);
    if (option == 0) {
        cliente = new AlumnoRequest(this, alumno, 1, 'u');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }
} // GEN-LAST:event_btnNuevoAlumnoActionPerformed

private void btnBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btnBuscarAlumnoActionPerformed
    Alumno alumno = new Alumno();
    alumno = buscarAlumno();

} // GEN-LAST:event_btnBuscarAlumnoActionPerformed

private void btnEliminarAlumnoActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_btnEliminarAlumnoActionPerformed
    AlumnoRequest cliente;
    Alumno alumno = new Alumno();
    alumno = buscarAlumno();
    int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
eliminar al alumno?", "Eliminar alumno", JOptionPane.YES_NO_OPTION);
    if (option == 0) {
        cliente = new AlumnoRequest(this, alumno, 'd');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }
} // GEN-LAST:event_btnEliminarAlumnoActionPerformed

```

```

        private void btnActualizaAlumnoActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btnActualizaAlumnoActionPerformed
        Alumno alumno = new Alumno();
        AlumnoRequest cliente;
        alumno = buscarAlumno();
        alumno.setNombre(txtNombreAlumno.getText());
        alumno.setApellidoPaterno(txtApellidoP.getText());
        alumno.setApellidoMaterno(txtApellidoM.getText());
        alumno.setMail(txtEmail.getText());
        alumno.setCalle(txtCalle.getText());
        alumno.setNumero(txtNumero.getText());
        alumno.setColonia(txtColonia.getText());
        alumno.setDelegación(txtDelegacion.getText());
        alumno.setEntidad(txtEntidad.getText());
        alumno.setTelefono(txtTelefono.getText());
        int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
actualizar al alumno?", "Actualizar alumno", JOptionPane.YES_NO_OPTION);
        if (option == 0) {

            cliente = new AlumnoRequest(this, alumno, 2, 'u');//Primer
parametro=datos,Segundo parametro: 1=insert 2=update,Tercer
parametro:'u'=update&insert 'd'=delete 'b'=buscar 'l'=listar
            Thread peticion = new Thread(cliente);
            peticion.start();//Peticion asincrona

        }
    }
    //GEN-LAST:event_btnActualizaAlumnoActionPerformed

    private void btnListarAlumnosActionPerformed(java.awt.event.ActionEvent evt)
{
    //GEN-FIRST:event_btnListarAlumnosActionPerformed
        getStudents();
    }
    //GEN-LAST:event_btnListarAlumnosActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(GUI_Alumno.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        }
    }
}

```

```

        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(GUI_Alumno.class.getName()).log(java.util.logging
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(GUI_Alumno.class.getName()).log(java.util.logging
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(GUI_Alumno.class.getName()).log(java.util.logging
g.Level.SEVERE, null, ex);
        }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new GUI_Alumno().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnActualizarAlumno;
private javax.swing.JButton btnBuscarAlumno;
private javax.swing.JButton btnEliminarAlumno;
private javax.swing.JButton btnListarAlumnos;
private javax.swing.JButton btnNuevoAlumno;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblShowTable;
private javax.swing.JTextField txtApellidoM;
private javax.swing.JTextField txtApellidoP;
private javax.swing.JTextField txtCalle;
private javax.swing.JTextField txtColonia;
private javax.swing.JTextField txtDelegacion;
private javax.swing.JTextField txtEmail;
private javax.swing.JTextField txtEntidad;
private javax.swing.JTextField txtNombreAlumno;
private javax.swing.JTextField txtNumero;
private javax.swing.JTextField txtTelefono;
// End of variables declaration//GEN-END:variables
}

```

AsignacionRequest.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cliente;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import modelos.Alumno;
import modelos.Asignacion;
import modelos.Cursoro;

/**
 *
 * @author Steeltwins
 */
public class AsignacionRequest implements Runnable {
    GUI_Asignacion parent;
    private Asignacion asignacion;
    private int operation;
    private char request;
    private final String IP_ADDRESS = "127.0.0.1";
    private final int PORT = 8200;

    public AsignacionRequest(GUI_Asignacion parent, Asignacion asignacion, int
operation, char request) {
        this.parent = parent;
        this.asignacion = asignacion;
        this.operation = operation;
        this.request = request;
    }

    public AsignacionRequest(GUI_Asignacion parent, Asignacion asignacion, char
request) {
        this.parent = parent;
        this.asignacion = asignacion;
        this.request = request;
    }

    public AsignacionRequest(GUI_Asignacion parent, char request) {
        this.parent = parent;
        this.request = request;
    }

    public boolean sendUpdate() throws UnknownHostException, IOException {
        boolean exitoso = false;
        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);

```



```

        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        //System.out.println("Antes de enviar la operacion");
        peticion.writeObject("asignacion");

        //SE MANDA EL TIPO DE OPERACION
        if (this.operation == 1) {
            peticion.writeObject("insertar");
        } else {
            peticion.writeObject("editar");
        }
        peticion.writeObject(this.asignacion); //Se manda el objeto
        System.out.println("Elemento enviado");
        exitoso = (boolean) respuesta.readObject();
        System.out.println("Respuesta devuelta");
        if (exitoso) {
            if (operation == 1) {
                JOptionPane.showMessageDialog(null, "Insercion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);

                } else {
                    JOptionPane.showMessageDialog(null, "Actualizacion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);
                }
                synchronized (this.parent) {
                    this.parent.getAssignations();
                }
            } else {
                System.out.println("Respuesta false");
                JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.ERROR_MESSAGE);
            }
            peticion.close();
            respuesta.close();
            socket.close();
            System.out.println("Conexion cerrada");
        } catch (UnknownHostException e) {
            e.printStackTrace();
            exitoso = false;
        } catch (IOException e) {
            e.printStackTrace();
            exitoso = false;
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            exitoso = false;
        }
        return exitoso;
    }

    public boolean sendDelete() {
        boolean exitoso = false;

        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);

```

```

        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("asignacion");

        peticion.writeObject("borrar");//Se manda operacion
        peticion.writeObject(this.asignacion);//Se manda objeto
        exitoso = (boolean) respuesta.readObject();
        if (exitoso) {
            JOptionPane.showMessageDialog(null, "Alumno Borrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.getAssignations();
            }
        } else {
            JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.INFORMATION_MESSAGE);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    }
    return exitoso;
}

public Asignacion sendBuscar() {
    Asignacion asignacion = new Asignacion();

    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("asignacion");

        peticion.writeObject("buscar");//Se manda operacion
        peticion.writeObject(this.asignacion);//Se manda objeto
        asignacion = (Asignacion) respuesta.readObject();

        JOptionPane.showMessageDialog(null, "Curso encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.setValuesSearched(asignacion);
        }
        peticion.close();
    }
}

```

```

        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return asignacion;
}

public ArrayList<Asignacion> sendListar() {
    ArrayList<Asignacion> asignacion = new ArrayList<Asignacion>();
    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("asignacion");
        System.out.println("asignacion enviada");
        peticion.writeObject("listar");//Se manda operacion
        asignacion = (ArrayList<Asignacion>) respuesta.readObject();
        //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.listAssignations(asignacion);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return asignacion;
}

public ArrayList<Curso> sendListarCursos() {
    ArrayList<Curso> cursos = new ArrayList<Curso>();
    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("curso");
    }

```

```

        petition.writeObject("listar");//Se manda operacion
        cursos = (ArrayList<Curso>) respuesta.readObject();
        //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.listCourses(cursos);
        }
        petition.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    }
    return cursos;
}

public ArrayList<Alumno> sendListarAlumnos() {
    ArrayList<Alumno> alumnos = new ArrayList<Alumno>();
    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream petition = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        petition.writeObject("alumno");

        petition.writeObject("listar");//Se manda operacion
        alumnos = (ArrayList<Alumno>) respuesta.readObject();
        //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.listStudents(alumnos);
        }
        petition.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    }
    return alumnos;
}

```

```

@Override
public void run() {
    try {
        if (this.request == 'u') {
            sendUpdate();
        } else if (this.request == 'd') {
            sendDelete();
        } else if (this.request == 'b') {
            sendBuscar();
        } else if (this.request == 'l') {
            sendListar();
        } else if (this.request == 'c') {
            sendListarCursos();
        } else if (this.request == 's') {
            sendListarAlumnos();
        }

        } catch (IOException ex) {
            Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}

```

GUI_Asignacion.form

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cliente;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import modelos.Alumno;
import modelos.Asignacion;
import modelos.Cursoro;

/**
 *
 * @author Steeltwins
 */
public class AsignacionRequest implements Runnable {
    GUI_Asignacion parent;
    private Asignacion asignacion;
    private int operation;
    private char request;
    private final String IP_ADDRESS = "127.0.0.1";
    private final int PORT = 8200;

```

```

    public AsignacionRequest(GUI_Asignacion parent, Asignacion asignacion, int
operation, char request) {
        this.parent = parent;
        this.asignacion = asignacion;
        this.operation = operation;
        this.request = request;
    }

    public AsignacionRequest(GUI_Asignacion parent, Asignacion asignacion, char
request) {
        this.parent = parent;
        this.asignacion = asignacion;
        this.request = request;
    }

    public AsignacionRequest(GUI_Asignacion parent, char request) {
        this.parent = parent;
        this.request = request;
    }

    public boolean sendUpdate() throws UnknownHostException, IOException {
        boolean exitoso = false;
        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            //System.out.println("Antes de enviar la operacion");
            peticion.writeObject("asignacion");

            //SE MANDA EL TIPO DE OPERACION
            if (this.operation == 1) {
                peticion.writeObject("insertar");
            } else {
                peticion.writeObject("editar");
            }
            peticion.writeObject(this.asignacion); //Se manda el objeto
            System.out.println("Elemento enviado");
            exitoso = (boolean) respuesta.readObject();
            System.out.println("Respuesta devuelta");
            if (exitoso) {
                if (operation == 1) {
                    JOptionPane.showMessageDialog(null, "Insercion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);

                } else {
                    JOptionPane.showMessageDialog(null, "Actualizacion exitosa",
"Exito", JOptionPane.INFORMATION_MESSAGE);
                }
                synchronized (this.parent) {
                    this.parent.getAssignations();
                }
            } else {
                System.out.println("Respuesta false");
            }
        }
    }

```

```

        JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.ERROR_MESSAGE);
    }
    peticion.close();
    respuesta.close();
    socket.close();
    System.out.println("Conexion cerrada");
} catch (UnknownHostException e) {
    e.printStackTrace();
    exitoso = false;
} catch (IOException e) {
    e.printStackTrace();
    exitoso = false;
} catch (ClassNotFoundException e) {
    e.printStackTrace();
    exitoso = false;
}
return exitoso;
}

public boolean sendDelete() {
    boolean exitoso = false;

    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream peticion = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        peticion.writeObject("asignacion");

        peticion.writeObject("borrar");//Se manda operacion
        peticion.writeObject(this.asignacion);//Se manda objeto
        exitoso = (boolean) respuesta.readObject();
        if (exitoso) {
            JOptionPane.showMessageDialog(null, "Alumno Borrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.getAssignations();
            }
        } else {
            JOptionPane.showMessageDialog(null, "Se produjo un error", "Error",
JOptionPane.INFORMATION_MESSAGE);
        }
        peticion.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

```

```

        return exitoso;
    }

    public Asignacion sendBuscar() {
        Asignacion asignacion = new Asignacion();

        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream petition = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            petition.writeObject("asignacion");

            petition.writeObject("buscar");//Se manda operacion
            petition.writeObject(this.asignacion);//Se manda objeto
            asignacion = (Asignacion) respuesta.readObject();

            JOptionPane.showMessageDialog(null, "Curso encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.setValuesSearched(asignacion);
            }
            petition.close();
            respuesta.close();
            socket.close();
            System.out.println("Conexion cerrada");

        } catch (IOException ex) {
            Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return asignacion;
    }

    public ArrayList<Asignacion> sendListar() {
        ArrayList<Asignacion> asignacion = new ArrayList<Asignacion>();
        try {
            Socket socket = new Socket(IP_ADDRESS, PORT);
            ObjectOutputStream petition = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
            //SE MANDA CADENA DE LA TABLA ESPECÍFICA
            petition.writeObject("asignacion");
            System.out.println("asignacion enviada");
            petition.writeObject("listar");//Se manda operacion
            asignacion = (ArrayList<Asignacion>) respuesta.readObject();
            //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
            synchronized (this.parent) {
                this.parent.listAssignations(asignacion);
            }
        }
    }

```



```

        petition.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return asignacion;
}

public ArrayList<Curso> sendListarCursos() {
    ArrayList<Curso> cursos = new ArrayList<Curso>();
    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream petition = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA
        petition.writeObject("curso");

        petition.writeObject("listar");//Se manda operacion
        cursos = (ArrayList<Curso>) respuesta.readObject();
        //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.listCourses(cursos);
        }
        petition.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

    } catch (IOException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return cursos;
}

public ArrayList<Alumno> sendListarAlumnos() {
    ArrayList<Alumno> alumnos = new ArrayList<Alumno>();
    try {
        Socket socket = new Socket(IP_ADDRESS, PORT);
        ObjectOutputStream petition = new
ObjectOutputStream(socket.getOutputStream());
        ObjectInputStream respuesta = new
ObjectInputStream(socket.getInputStream());
        //SE MANDA CADENA DE LA TABLA ESPECÍFICA

```

```

        petition.writeObject("alumno");

        petition.writeObject("listar");//Se manda operacion
        alumnos = (ArrayList<Alumno>) respuesta.readObject();
        //JOptionPane.showMessageDialog(null, "Producto encontrado", "Exito",
JOptionPane.INFORMATION_MESSAGE);
        synchronized (this.parent) {
            this.parent.listStudents(alumnos);
        }
        petition.close();
        respuesta.close();
        socket.close();
        System.out.println("Conexion cerrada");

        } catch (IOException ex) {
            Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return alumnos;
    }

    @Override
    public void run() {
        try {
            if (this.request == 'u') {
                sendUpdate();
            } else if (this.request == 'd') {
                sendDelete();
            } else if (this.request == 'b') {
                sendBuscar();
            } else if (this.request == 'l') {
                sendListar();
            } else if (this.request == 'c') {
                sendListarCursos();
            } else if (this.request == 's') {
                sendListarAlumnos();
            }

            } catch (IOException ex) {
                Logger.getLogger(AsignacionRequest.class.getName()).log(Level.SEVERE,
null, ex);
            }
        }
    }
}

```

GUI_Asigancion.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package cliente;

```

```

import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import modelos.Alumno;
import modelos.Asignacion;
import modelos.Cursoro;

/**
 *
 * @author Steeltwins
 */
public class GUI_Asignacion extends javax.swing.JFrame {

    /**
     * Creates new form GUI_Asignacion
     */
    public GUI_Asignacion() {
        initComponents();
        cboAlumno.removeAllItems();
        cboCurso.removeAllItems();
        // getStudents();
        //getCourses();
    }

    public void getCourses() {
        AsignacionRequest cliente;
        cliente = new AsignacionRequest(this, 'c');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }

    public void listCourses(ArrayList<Curso> cursos) {
        for (int i = 0, j = 0; i < cursos.size(); i++) {
            cboCurso.addItem(cursos.get(i).getNombre() + "-" +
cursos.get(i).getIdCurso());
        }
    }

    public void getStudents() {
        AsignacionRequest cliente;
        cliente = new AsignacionRequest(this, 's');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }

    public void listStudents(ArrayList<Alumno> alumnos) {
        for (int i = 0, j = 0; i < alumnos.size(); i++) {
            cboAlumno.addItem(alumnos.get(i).getNombre() + "-" +
alumnos.get(i).getIdAlumno());
        }
    }

    public void getAssignations() {
        AsignacionRequest cliente;
        cliente = new AsignacionRequest(this, 'l');
        Thread peticion = new Thread(cliente);
    }

```

```

        peticion.start();
    }

    public void listAssignations(ArrayList<Asignacion> asignacion) {
        Object[][] assigned = new Object[asignacion.size()][4];
        String[] columns = {"idCurso", "idAlumno", "Horario", "Tipo de Curso"};

        for (int i = 0, j = 0; i < asignacion.size(); i++) {
            assigned[i][j] = "" + asignacion.get(i).getIdCurso();
            assigned[i][j + 1] = "" + asignacion.get(i).getIdAlumno();
            assigned[i][j + 2] = "" + asignacion.get(i).getHorario();
            assigned[i][j + 3] = "" + asignacion.get(i).getTipo();
        }
        tblShowTable.setModel(new DefaultTableModel(assigned, columns));
    }

    public void setValuesSearched(Asignacion asignacion) {
        // cboAlumno.setSelectedItem(asignacion.getIdAlumno());
        // cboCurso.setSelectedItem(asignacion.getIdCurso());
        // txtHorario.setText(asignacion.getHorario());
        // spTipoDeCurso.setValue(asignacion.getTipo());
        Object[][] assigned = new Object[1][4];
        String[] columns = {"idCurso", "idAlumno", "Horario", "Tipo de Curso"};
        assigned[0][0] = "" + asignacion.getIdCurso();
        assigned[0][1] = "" + asignacion.getIdAlumno();
        assigned[0][2] = "" + asignacion.getHorario();
        assigned[0][3] = "" + asignacion.getTipo();
        tblShowTable.setModel(new DefaultTableModel(assigned, columns));
    }

    public Asignacion buscarAsignacion() {
        AsignacionRequest cliente;
        Asignacion asignacion = new Asignacion();
        int idCurso = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingrese
el id del Curso ", "Buscar asignacion", JOptionPane.QUESTION_MESSAGE));
        asignacion.setIdCurso(idCurso);
        int idAlumno = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingrese
el id del Alumno ", "Buscar asignacion", JOptionPane.QUESTION_MESSAGE));
        asignacion.setIdAlumno(idAlumno);
        cliente = new AsignacionRequest(this, asignacion, 'b');
        Thread peticion = new Thread(cliente);
        peticion.start();
        return asignacion;
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-
BEGIN: initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();

```

```

cboAlumno = new javax.swing.JComboBox<>();
jLabel2 = new javax.swing.JLabel();
cboCurso = new javax.swing.JComboBox<>();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
txtHorario = new javax.swing.JTextField();
jPanel4 = new javax.swing.JPanel();
btnNuevoAsignacion = new javax.swing.JButton();
btnBuscarAsignacion = new javax.swing.JButton();
btnEliminarAsignacion = new javax.swing.JButton();
btnActualizaAsignacion = new javax.swing.JButton();
btnListarAsignaciones = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
tblShowTable = new javax.swing.JTable();
spTipoDeCurso = new javax.swing.JSpinner();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Gestor de Asignaciones");

jLabel1.setText("Alumno");

cboAlumno.setModel(new javax.swing.DefaultComboBoxModel<>(new String[]
{ "Item 1", "Item 2", "Item 3", "Item 4" }));
cboAlumno.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cboAlumnoMouseClicked(evt);
    }
});

jLabel2.setText("Curso");

cboCurso.setModel(new javax.swing.DefaultComboBoxModel<>(new String[]
{ "Item 1", "Item 2", "Item 3", "Item 4" }));
cboCurso.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cboCursoMouseClicked(evt);
    }
});

jLabel3.setText("Horario");

jLabel4.setText("Tipo de curso");

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder("Opciones"));

btnNuevoAsignacion.setText("Nuevo");
btnNuevoAsignacion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnNuevoAsignacionActionPerformed(evt);
    }
});

btnBuscarAsignacion.setText("Buscar");
btnBuscarAsignacion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarAsignacionActionPerformed(evt);
    }
});

```

```

    }
});

btnEliminarAsignacion.setText("Eliminar");
btnEliminarAsignacion.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEliminarAsignacionActionPerformed(evt);
    }
});

btnActualizarAsignacion.setText("Actualizar");
btnActualizarAsignacion.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarAsignacionActionPerformed(evt);
    }
});

btnListarAsignaciones.setText("Listar");
btnListarAsignaciones.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnListarAsignacionesActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addComponent(btnActualizarAsignacion)
        .addComponent(btnEliminarAsignacion)
        .addComponent(btnBuscarAsignacion)
        .addComponent(btnNuevoAsignacion)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(btnListarAsignaciones))
        .addGap(11, Short.MAX_VALUE))
    );
jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(21, 21, 21)
        .addComponent(btnNuevoAsignacion)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnBuscarAsignacion)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(btnEliminarAsignacion)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(btnActualizaAsignacion)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(btnListarAsignaciones)
    .addContainerGap(35, Short.MAX_VALUE))
);

tblShowTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "idCurso", "idAlumno", "Horario", "Tipo de curso"
    }
));
jScrollPane.setViewportView(tblShowTable);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(42, 42, 42)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel4)
                    .addGap(18, 18, 18)
                    .addComponent(spTipoDeCurso))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addGap(47, 47, 47)
                    .addComponent(cboAlumno,
javax.swing.GroupLayout.PREFERRED_SIZE, 116,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)))
            .addGap(35, 35, 35)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addComponent(btnEliminarAsignacion)
                .addComponent(btnActualizaAsignacion)
                .addComponent(btnListarAsignaciones))
        );

```



```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 38,
Short.MAX_VALUE)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
198, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(29, 29, 29))
    );

    pack();
} // </editor-fold> // GEN-END: initComponents

private void btnNuevoAsignacionActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_btnNuevoAsignacionActionPerformed
    AsignacionRequest cliente;
    Asignacion asignacion = new Asignacion();
    System.out.println("idAlumno: " +
cboAlumno.getSelectedItem().toString().substring(cboAlumno.getSelectedItem().toStri
ng().indexOf("-")));
    System.out.println("idAlumno: " +
cboCurso.getSelectedItem().toString().substring(cboCurso.getSelectedItem().toStri
ng().indexOf("-") + 1));

    asignacion.setIdAlumno(Integer.parseInt(cboAlumno.getSelectedItem().toString().subs
tring(cboAlumno.getSelectedItem().toString().indexOf("-") + 1)));

    asignacion.setIdCurso(Integer.parseInt(cboCurso.getSelectedItem().toString().substr
ing(cboCurso.getSelectedItem().toString().indexOf("-") + 1)));
    asignacion.setHorario(txtHorario.getText());
    asignacion.setTipo((int) spTipoDeCurso.getValue());
    int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea crear
la asignacion?", "Crear asignacion", JOptionPane.YES_NO_OPTION);
    if (option == 0) {
        cliente = new AsignacionRequest(this, asignacion, 1, 'u');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }
} // GEN-LAST: event_btnNuevoAsignacionActionPerformed

private void btnBuscarAsignacionActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_btnBuscarAsignacionActionPerformed
    Asignacion asignacion = new Asignacion();
    asignacion = buscarAsignacion();
} // GEN-LAST: event_btnBuscarAsignacionActionPerformed

private void btnEliminarAsignacionActionPerformed(java.awt.event.ActionEvent
evt) { // GEN-FIRST: event_btnEliminarAsignacionActionPerformed
    AsignacionRequest cliente;
    Asignacion asignacion = new Asignacion();
    asignacion = buscarAsignacion();
    int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
eliminar la asignacion?", "Eliminar asignacion", JOptionPane.YES_NO_OPTION);
    if (option == 0) {
        cliente = new AsignacionRequest(this, asignacion, 'd');
        Thread peticion = new Thread(cliente);
        peticion.start();
    }
} // GEN-LAST: event_btnEliminarAsignacionActionPerformed

```

```

        private void btnActualizaAsignacionActionPerformed(java.awt.event.ActionEvent
evt) {
//GEN-FIRST:event_btnActualizaAsignacionActionPerformed
        Asignacion asignacion = new Asignacion();
        AsignacionRequest cliente;
        asignacion = buscarAsignacion();
        asignacion.setHorario(txtHorario.getText());
        asignacion.setTipo((int) spTipoDeCurso.getValue());

        int option = JOptionPane.showConfirmDialog(null, "¿Seguro que desea
actualizar la asignacion?", "Actualizar asignacion", JOptionPane.YES_NO_OPTION);
        if (option == 0) {

            cliente = new AsignacionRequest(this, asignacion, 2, 'u');//Primer
parametro=datos, Segundo parametro: 1=insert 2=update, Tercer
parametro: 'u'=update&insert 'd'=delete 'b'=buscar 'l'=listar
            Thread peticion = new Thread(cliente);
            peticion.start();//Peticion asincrona

        }
    }
//GEN-LAST:event_btnActualizaAsignacionActionPerformed

    private void btnListarAsignacionesActionPerformed(java.awt.event.ActionEvent
evt) {
//GEN-FIRST:event_btnListarAsignacionesActionPerformed
        getAssignations();
    }
//GEN-LAST:event_btnListarAsignacionesActionPerformed

    private void cboCursoMouseClicked(java.awt.event.MouseEvent evt) {
//GEN-FIRST:event_cboCursoMouseClicked
        cboCurso.removeAllItems();
        getCourses();
    }
//GEN-LAST:event_cboCursoMouseClicked

    private void cboAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
//GEN-FIRST:event_cboAlumnoMouseClicked
        cboAlumno.removeAllItems();
        getStudents();
    }
//GEN-LAST:event_cboAlumnoMouseClicked

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Main.class).log(java.util.logging.
Level.SEVERE, "The LookAndFeel class named {0} could not be found. This
probably means the LookAndFeel is not installed or was not loaded correctly.",
ex.getMessage());
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Main.class).log(java.util.logging.
Level.SEVERE, "Failed to instantiate the LookAndFeel class named {0}. This
usually means you cannot run this JAR as a standalone application.",
ex.getMessage());
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Main.class).log(java.util.logging.
Level.SEVERE, "Unable to initialize the LookAndFeel class named {0}. This
usually means you cannot run this JAR as a standalone application.",
ex.getMessage());
        }
    }
}

```

```

        }
    }
    catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(GUI_Asignacion.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
    catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(GUI_Asignacion.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
    catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(GUI_Asignacion.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
    catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(GUI_Asignacion.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new GUI_Asignacion().setVisible(true);
    }
});
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton btnActualizarAsignacion;
private javax.swing.JButton btnBuscarAsignacion;
private javax.swing.JButton btnEliminarAsignacion;
private javax.swing.JButton btnListarAsignaciones;
private javax.swing.JButton btnNuevoAsignacion;
private javax.swing.JComboBox<String> cboAlumno;
private javax.swing.JComboBox<String> cboCurso;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSpinner spTipoDeCurso;
private javax.swing.JTable tblShowTable;
private javax.swing.JTextField txtHorario;
// End of variables declaration//GEN-END:variables
}

```

CONCLUSIONES

La arquitectura cliente servidor es una de las mas antiguas en el área de redes y su eficacia se ha probado por muchos años. La ventaja de implementar este tipo de arquitectura en java es que el lenguaje ya permite mucha facilidad a la hora de escribir el código debido a que ya contiene muchas clases que hacen mas fácil la implementación.

Hacer esta implementación en lenguajes diferentes al estilo C tomaría mucho tiempo y haría que la productividad disminuyera, pues en C se requiere hacer todo a mano, así como estar revisando los mensajes enviados y los mensajes que se reciben.

BIBLIOGRAFÍA

<https://es.wikipedia.org/wiki/SQL>

<https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/>