



SERVICIO NACIONAL DE APRENDIZAJE

**GA7-220501096-AA2-EV01 codificación de módulos del software según
requerimientos del proyecto**

**INSTRUCTORA: SONIA PATRICIA HOSTEN
APRENDIZ: JESUS MORENO
FICHA: 2721532**

MAYO/2024

Introducción

La utilización de CRUD en Java para MySQL se refiere a la capacidad de crear, leer, actualizar y eliminar datos en una base de datos MySQL utilizando el lenguaje de programación Java. Esta combinación es muy común en el desarrollo de aplicaciones empresariales y web, ya que MySQL es un sistema de gestión de bases de datos muy popular y Java es uno de los lenguajes de programación más utilizados.

Para lograr esto, se utiliza la API JDBC (Java Database Connectivity), que proporciona métodos para establecer conexiones con bases de datos, enviar consultas SQL y procesar los resultados. Al utilizar JDBC en conjunto con MySQL, los desarrolladores pueden implementar las operaciones CRUD para interactuar con la base de datos.

Objetivo

proporcionar a los desarrolladores una forma estructurada y eficiente de interactuar con una base de datos. Al implementar las operaciones de creación, lectura, actualización y eliminación, se permite a las aplicaciones Java acceder y manipular datos en una base de datos MySQL de manera coherente y segura. El objetivo de utilizar CRUD en Java para MySQL es facilitar la interacción entre las aplicaciones Java y la base de datos MySQL, permitiendo a los desarrolladores gestionar eficientemente la persistencia de datos en sus aplicaciones.

Evidencia de desempeño: GA7-220501096-AA2-EV01 codificación de módulos del software según requerimientos del proyecto

Teniendo en cuenta las características del software a desarrollar realizar la codificación del módulo del proyecto realizando conexiones con bases de datos por medio de JDBC tomando como referencia lo visto en el componente formativo “Construcción de aplicaciones con JAVA”.

Elementos para tener en cuenta:

- Para la codificación del módulo debe tener en cuenta los artefactos del ciclo del software realizados con anterioridad: diagrama de clases, diagramas de casos de uso, historias de usuario, diseños, prototipos, Informe técnico de plan de trabajo para construcción de software con tecnologías seleccionadas etc.
- Se debe crear el proyecto utilizando herramientas de versionamiento.
- El código debe cumplir con estándares de codificación como:
 - Nombramiento de variables
 - Nombramiento de métodos
 - Nombramiento de clases
 - Nombramiento de paquetes
- Debe tener funcionalidades de inserción, consulta, actualización y eliminación

Realización

- El objetivo, crear una base de datos llamada: bdescuela
- con una tabla llamada: alumnos, la cual va a tener id, nombres, apellidos
- Agregar al apache netbeans el conector JDBC para interactuar con la base de datos de MSQL
- Realizar la creación de diferentes sentencias, clases e importaciones para la ejecución de una interfaz que permita interactuar con la base de datos y permita actualizar, modificar y eliminar datos de la tabla alumnos

Uso de la herramienta Git y Github:

```
MINGW64/c:/Users/jesus/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01
jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01
$ git init
Initialized empty Git repository in C:/Users/jesus/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01/.git/

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git config --global user.name "jesusjh"

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git config --global user.email "jesushmoreno@gmail.com"

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git add .

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git commit -m "carga de archivos"
[master (root-commit) 6285aee] carga de archivos
5 files changed, 63 insertions(+)
create mode 100644 BACKUP_BDESCUELA(ALUNNOS).sql
create mode 100644 "Caj-220501096-AA2-EV01 codificaci\303\263n de n\303\263dulo
s del software seg\303\272n.docx"
create mode 100644 SCRIPT BASE DE DATOS ALUNNOS.sql
create mode 100644 diagrama_de_clase_alumnos.mmb
create mode 100644 "-57-220501096-AA2-EV01 codificaci\303\263n de n\303\263dulo
s del software seg\303\272n.docx"

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git remote add origin https://github.com/JESUSH313/SENA_codificacion_de_modulos.git

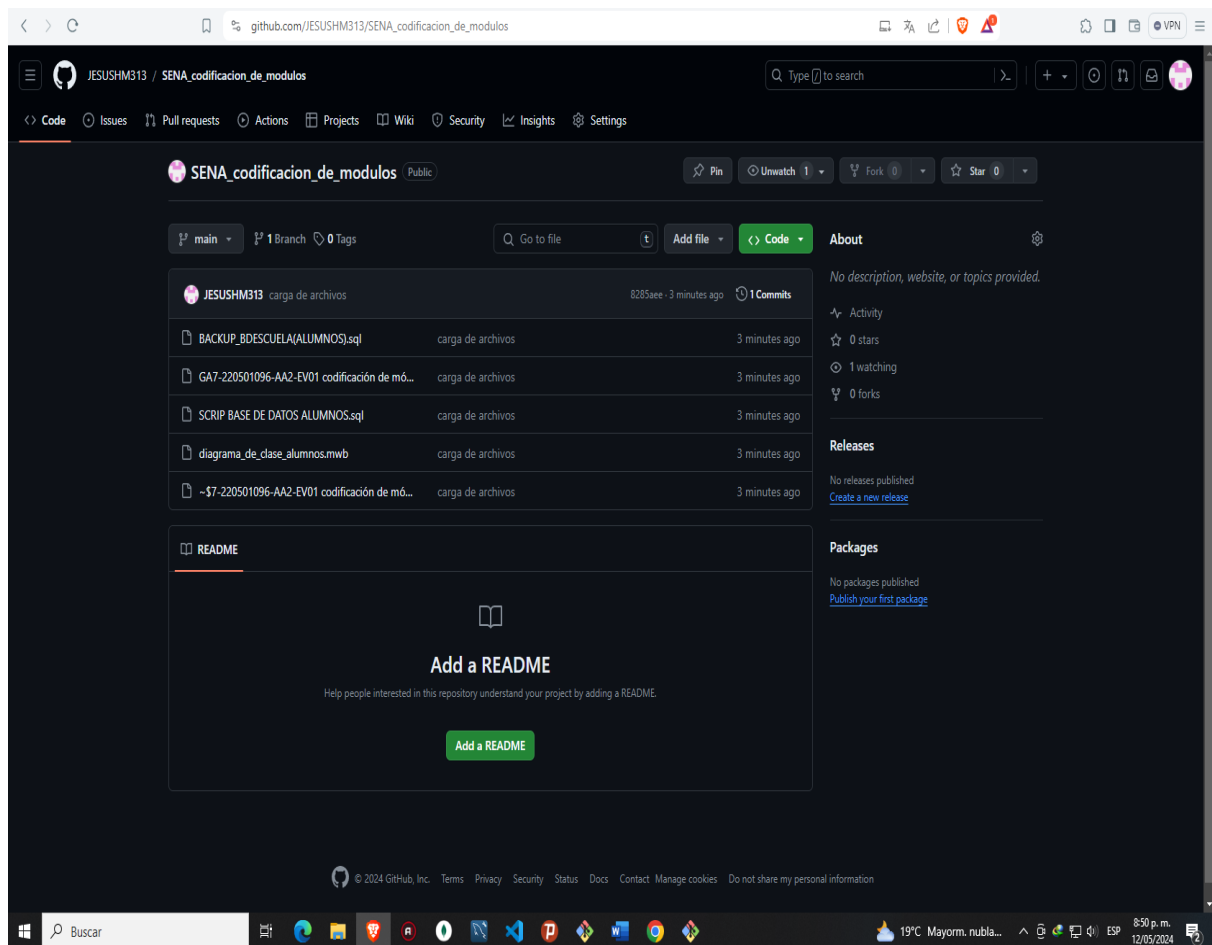
jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git push -u origin main
error: src refspec main does not match any
error: Failed to push some refs to 'https://github.com/JESUSH313/SENA_codificacion_de_modulos.git'

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (master)
$ git branch -m main

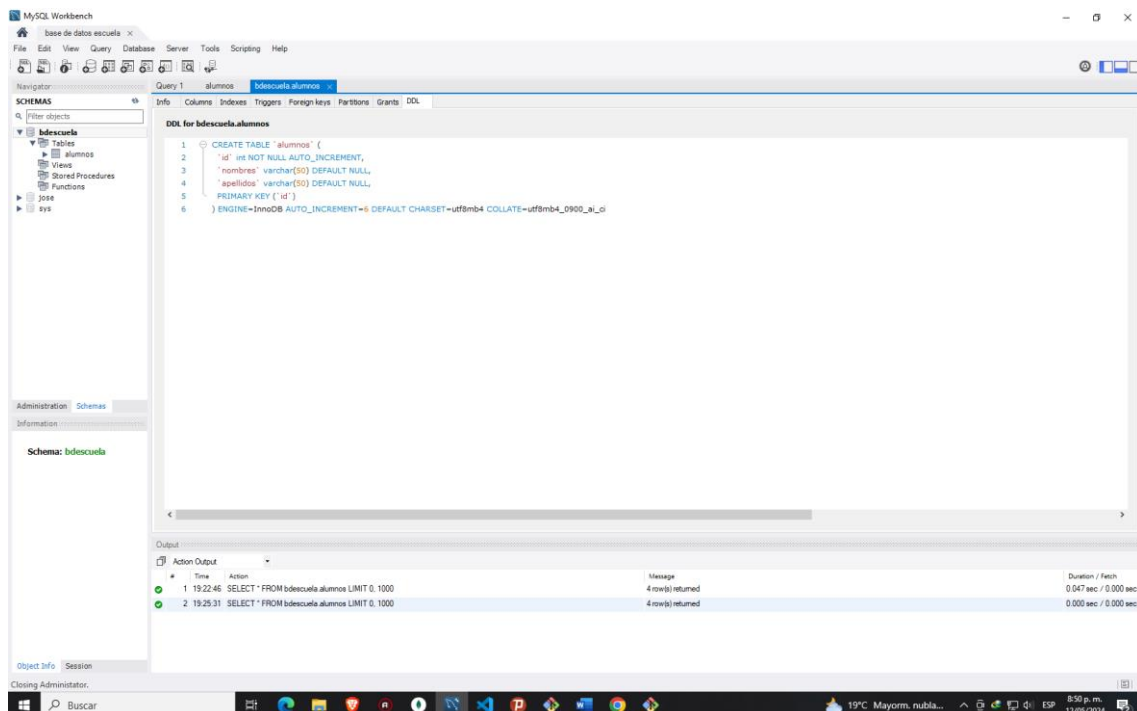
jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (main)
$ git remote add origin https://github.com/JESUSH313/SENA_codificacion_de_modulos.git
error: remote origin already exists.

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 20.06 KiB | 10.03 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JESUSH313/SENA_codificacion_de_modulos.git
 * [new branch] main -> main
branch 'main' set up to track 'origin/main'.

jesus@MLINA-MOLINA MINGW64 ~/Desktop/actividades/JESUSHERNANDEZ_AA2_EV01 (main)
$
```

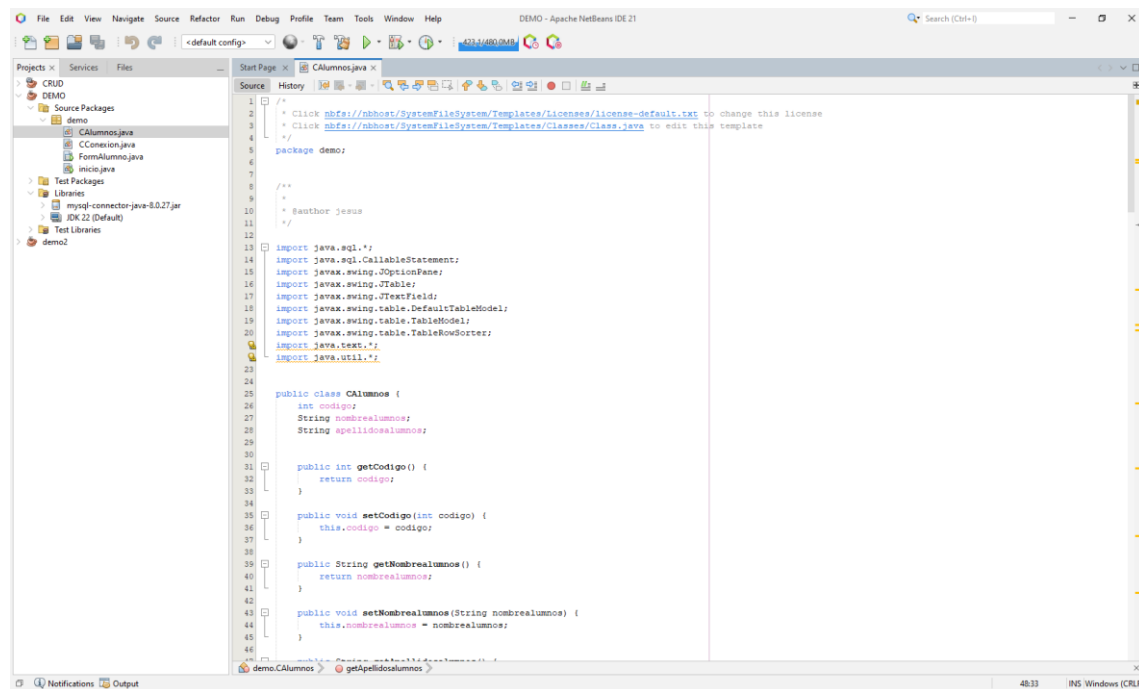


MSQL WORKBENCH

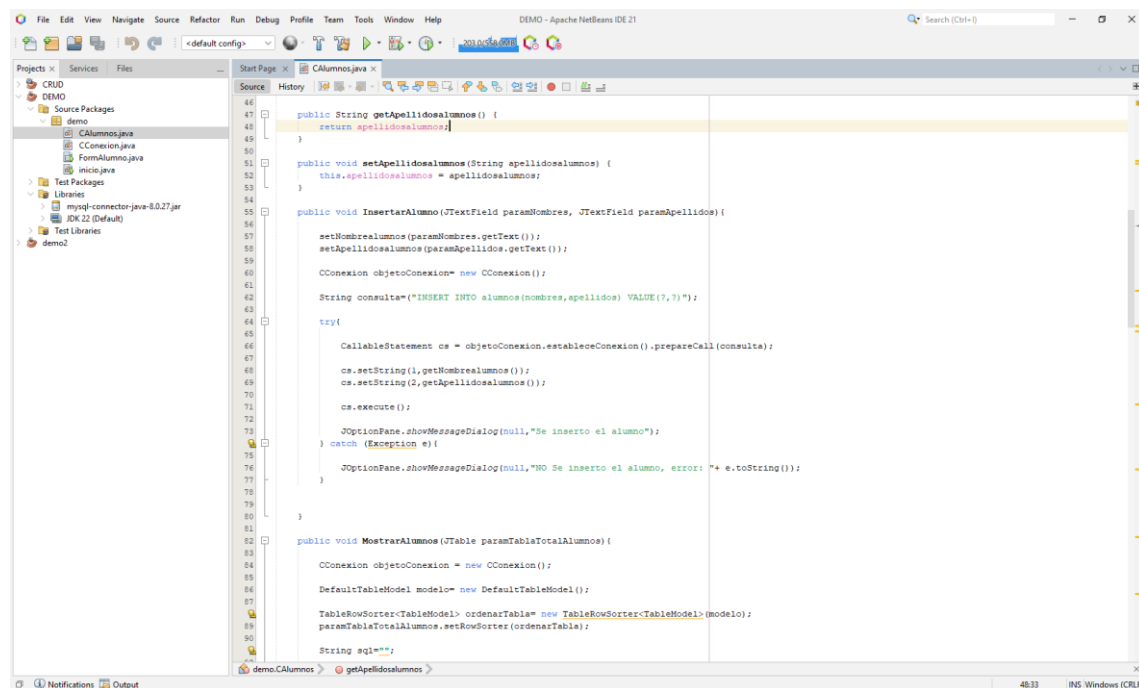


Apache net beans JAVA

CAlumnos.java:



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package demo;
6
7
8  /**
9   *
10   * @author Jesus
11   */
12
13  import java.sql.*;
14  import java.sql.CallableStatement;
15  import javax.swing.JOptionPane;
16  import javax.swing.JTable;
17  import javax.swing.JTextField;
18  import javax.swing.table.DefaultTableModel;
19  import javax.swing.table.TableModel;
20  import javax.swing.table.TableRowSorter;
21  import java.text.*;
22  import java.util.*;
23
24
25  public class CAalumnos {
26      int codigo;
27      String nombrealumnos;
28      String apellidosalumnos;
29
30
31      public int getCodigo() {
32          return codigo;
33      }
34
35      public void setCodigo(int codigo) {
36          this.codigo = codigo;
37      }
38
39      public String getNombrealumnos() {
40          return nombrealumnos;
41      }
42
43      public void setNombrealumnos(String nombrealumnos) {
44          this.nombrealumnos = nombrealumnos;
45      }
46  }
```



```
46
47      public String getApellidosalumnos() {
48          return apellidosalumnos;
49      }
50
51      public void setApellidosalumnos(String apellidosalumnos) {
52          this.apellidosalumnos = apellidosalumnos;
53      }
54
55      public void InsertarAlumno(JTextField paramNombres, JTextField paramApellidos) {
56          setNombrealumnos(paramNombres.getText());
57          setApellidosalumnos(paramApellidos.getText());
58
59          CConexion objetoConexion= new CConexion();
60
61          String consulta=("INSERT INTO alumnos(nombres,apellidos) VALUES(?,?)");
62
63          try{
64
65              CallableStatement cs = objetoConexion.estableceConexion().prepareCall(consulta);
66
67              cs.setString(1,getNombrealumnos());
68              cs.setString(2,getApellidosalumnos());
69
70              cs.execute();
71
72              JOptionPane.showMessageDialog(null,"Se inserto el alumno");
73          } catch (Exception e) {
74              JOptionPane.showMessageDialog(null,"NO Se inserto el alumno, error: "+ e.toString());
75          }
76
77      }
78
79
80
81      public void MostrarAlumnos(JTable paramTablaTotalAlumnos){
82
83          CConexion objetoConexion = new CConexion();
84
85          DefaultTableModel modelo= new DefaultTableModel();
86
87          TableRowSorter<TableModel> ordenarTabla= new TableRowSorter<TableModel>(modelo);
88          paramTablaTotalAlumnos.setRowSorter(ordenarTabla);
89
90          String sql="";
91      }
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help DEMO - Apache NetBeans IDE 21
Source History
CAlumnos.java
TableRowSorter<TableModel> ordenarTabla= new TableRowSorter<TableModel>(modelo);
paramTablaTotalAlumnos.setRowSorter(ordenarTabla);

String sql="";
modelo.addColumn("id");
modelo.addColumn("Nombres");
modelo.addColumn("Apellidos");
paramTablaTotalAlumnos.setModel(modelo);

sql="select * from alumnos";
String [] datos = new String[3];
Statement st;

try{
    st= objetoConexion.estableceConexion().createStatement();
    ResultSet rs= st.executeQuery(sql);

    while(rs.next()){
        datos[0]=rs.getString(1);
        datos[1]=rs.getString(2);
        datos[2]=rs.getString(3);
        modelo.addRow(datos);
    }

    paramTablaTotalAlumnos.setModel(modelo);
}catch(Exception e){
    JOptionPane.showMessageDialog(null,"NO Se pudo mostrar los registros: "+ e.toString());
}

public void SeleccionarAlumno(JTable paramTablaAlumnos, JTextField paramId, JTextField paramNombres, JTextField paramApellidos){
    try{
        int fila=paramTablaAlumnos.getSelectedRow();

        if(fila >=0){
            // ... (code continues in the next screenshot)
        }
    }
}
```

```
int fila=paramTablaAlumnos.getSelectedRow();

if(fila >=0){
    paramId.setText((paramTablaAlumnos.getValueAt(fila,0).toString()));
    paramNombres.setText((paramTablaAlumnos.getValueAt(fila,1).toString()));
    paramApellidos.setText((paramTablaAlumnos.getValueAt(fila,2).toString()));
}

else{
    JOptionPane.showMessageDialog(null, "fila no seleccionada");
}

}catch (Exception e){
    JOptionPane.showMessageDialog(null, "error de seleccion, error: "+ e.toString());
}

public void ModificarAlumnos(JTextField paramCodigo, JTextField paramNombres, JTextField paramApellidos){
    setCodigo(Integer.parseInt(paramCodigo.getText()));
    setNombres(paramNombres.getText());
    setApellidos(paramApellidos.getText());

    CConexion objetoConexion = new CConexion();

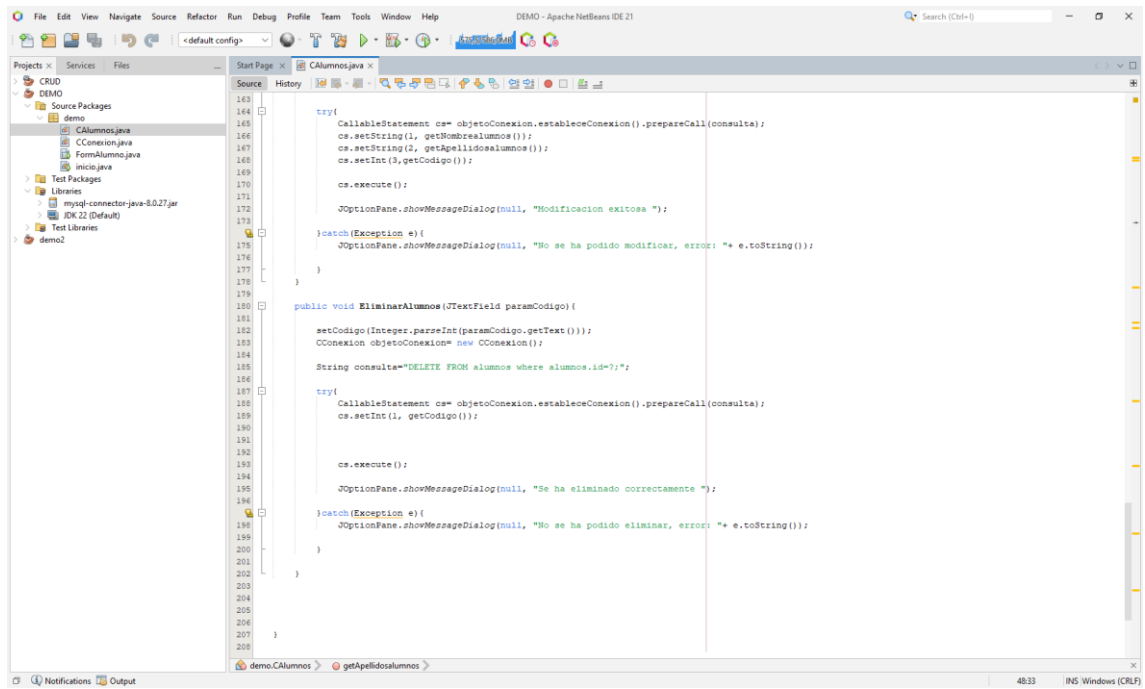
    String consulta="UPDATE alumnos set alumnos.nombres=?, alumnos.apellidos=? WHERE alumnos.id=?";

    try{
        CallableStatement cs= objetoConexion.estableceConexion().prepareCall(consulta);
        cs.setString(1, getNombresAlumnos());
        cs.setString(2, getApellidosAlumnos());
        cs.setInt(3, getCodigo());

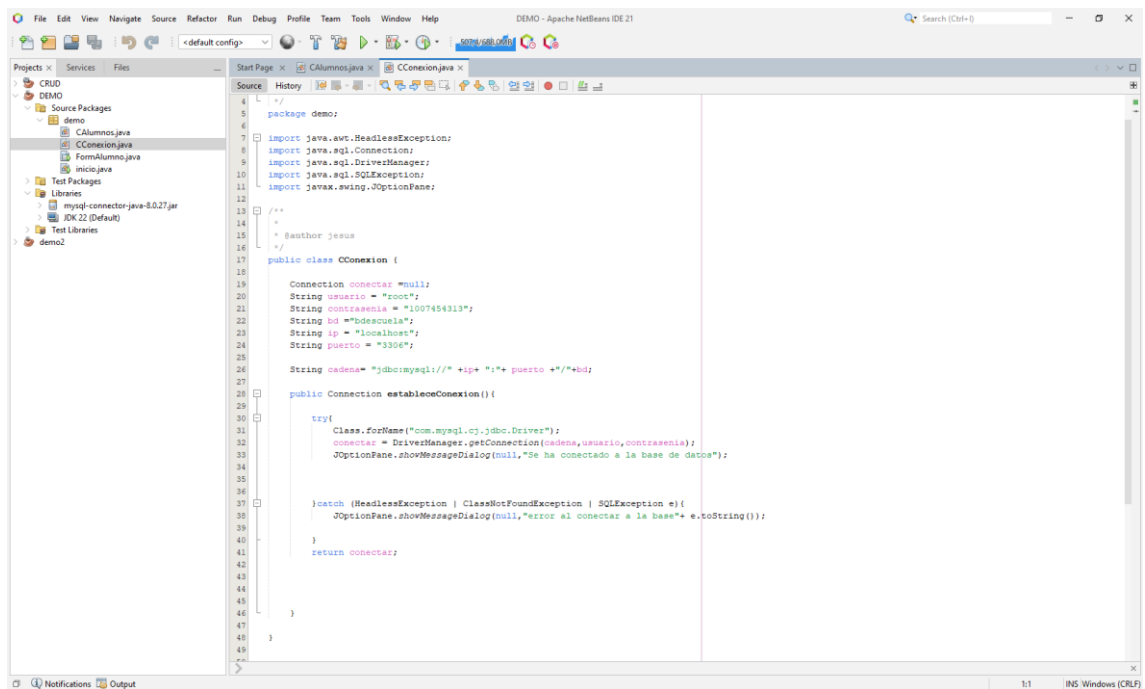
        cs.execute();

        JOptionPane.showMessageDialog(null, "Modificacion exitosa ");
    }

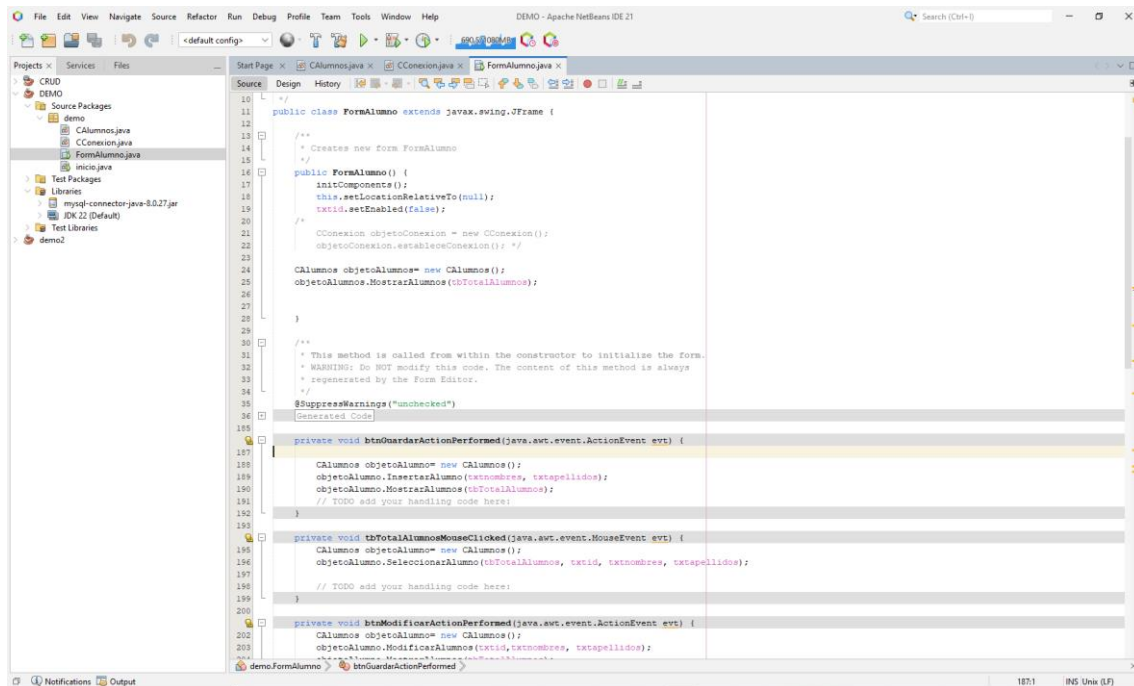
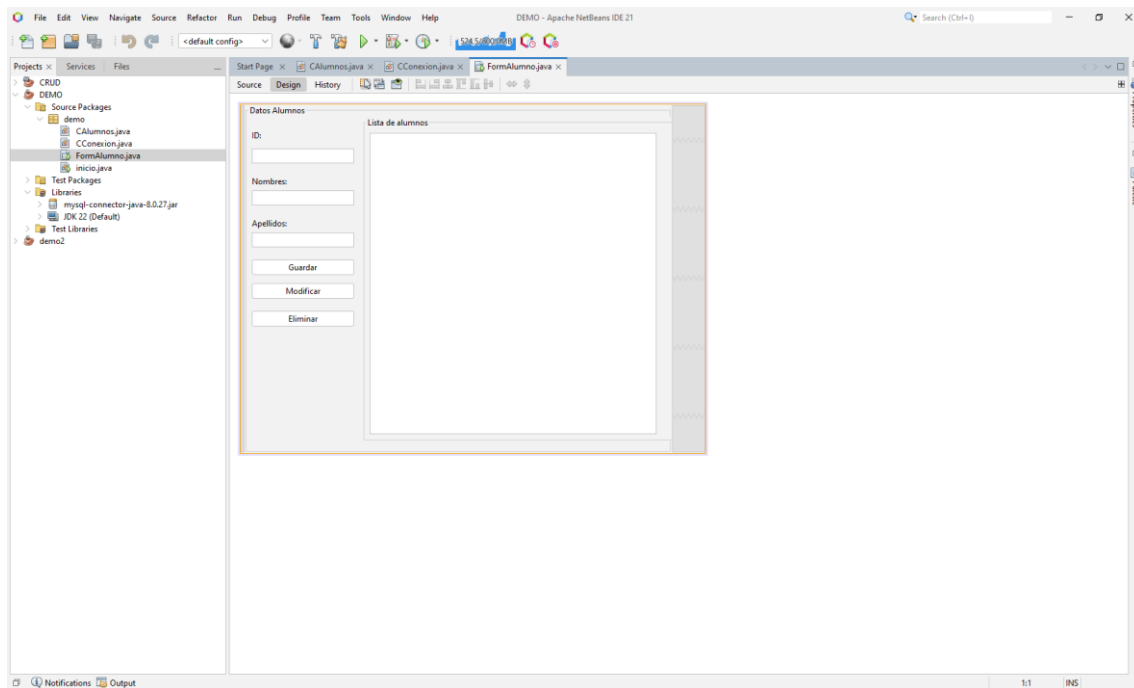
    catch(Exception e){
        JOptionPane.showMessageDialog(null, "No se ha podido modificar, error: "+ e.toString());
    }
}
```

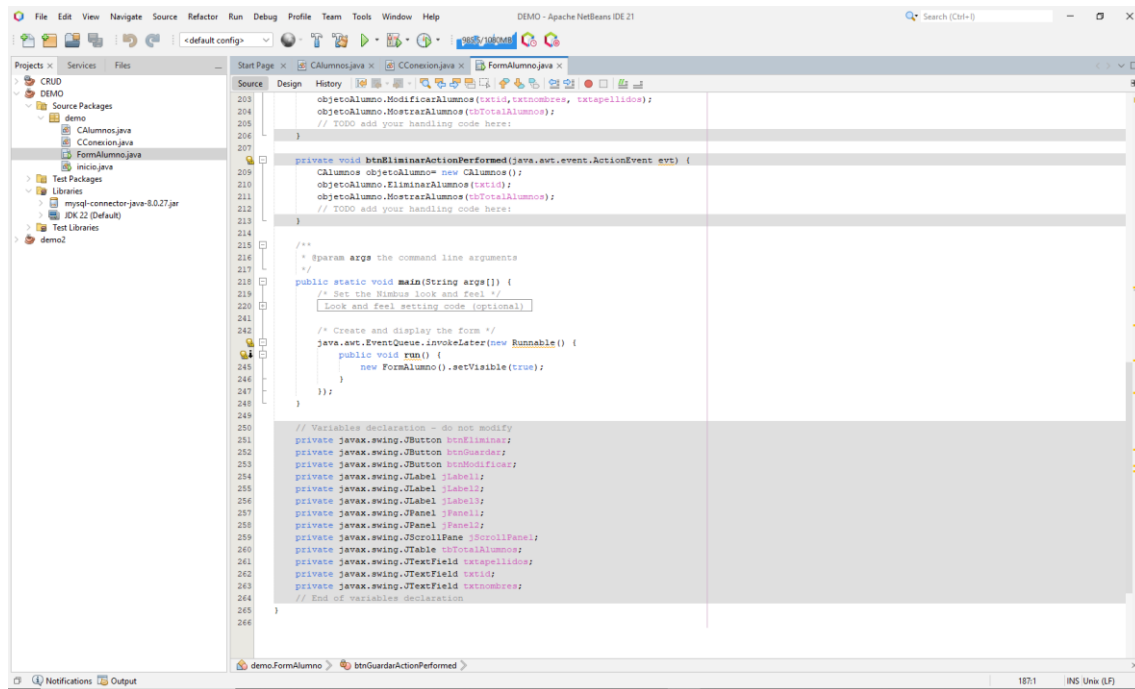



CCONEXION.JAVA:

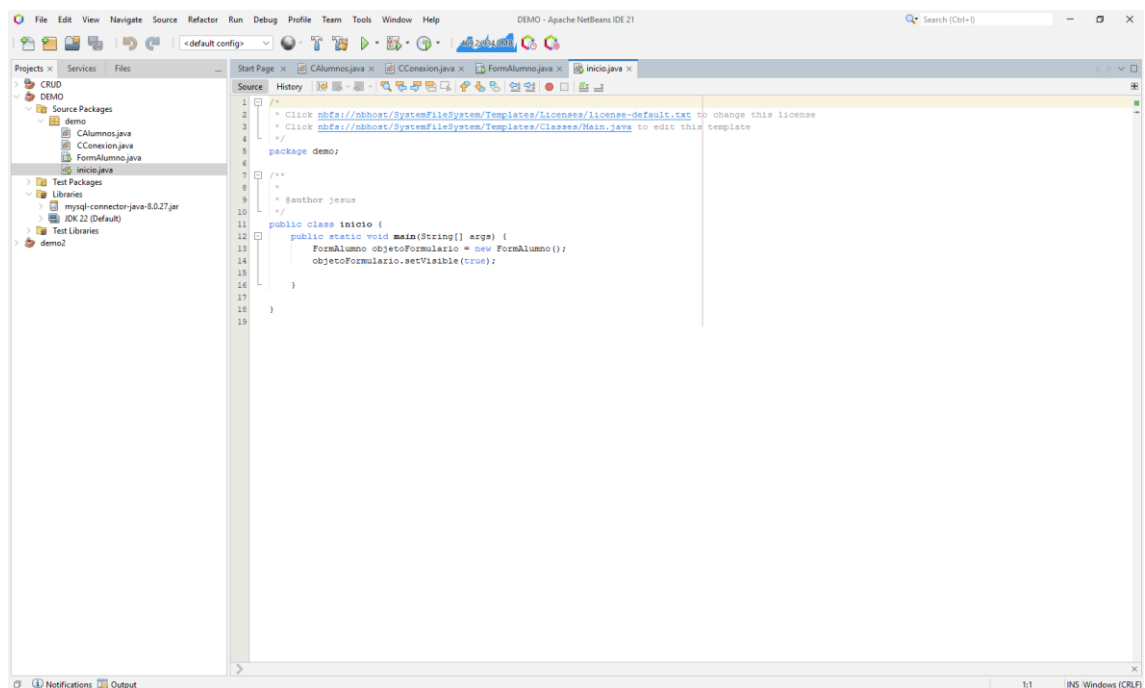


FORMALUMNO.JAVA:

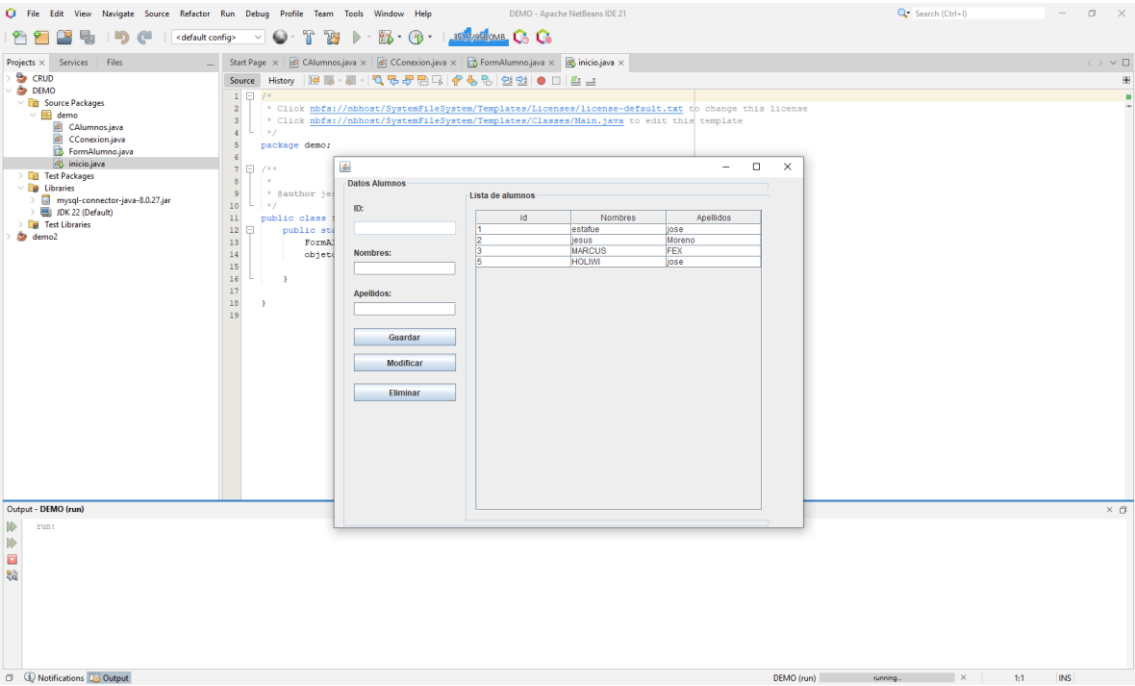
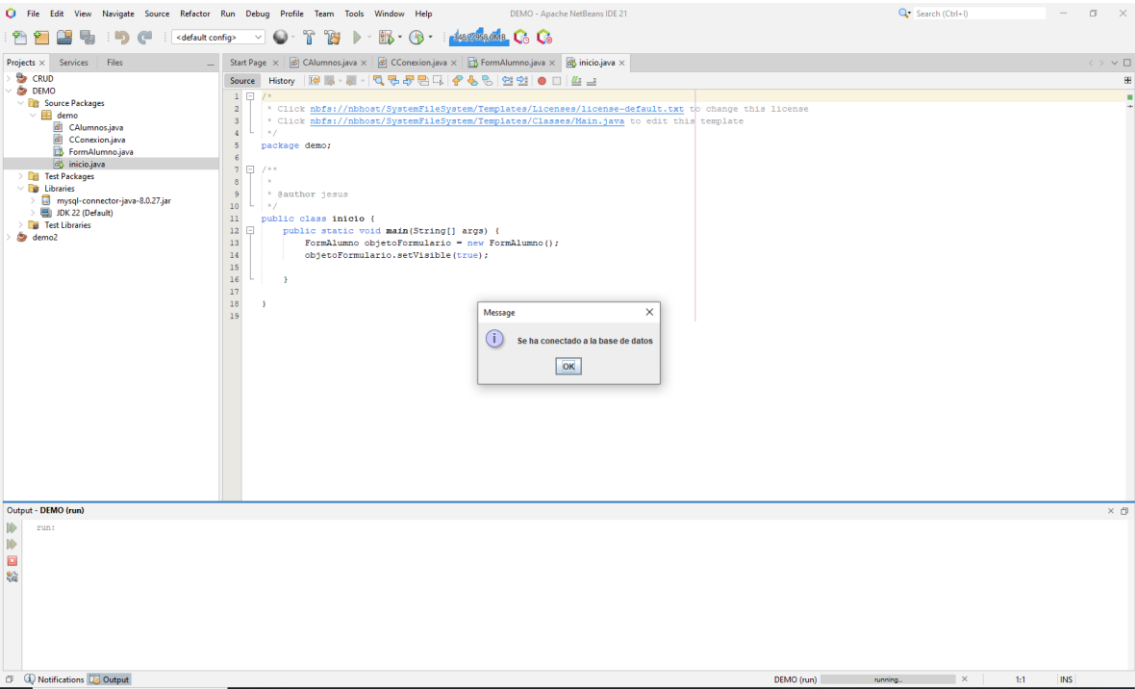




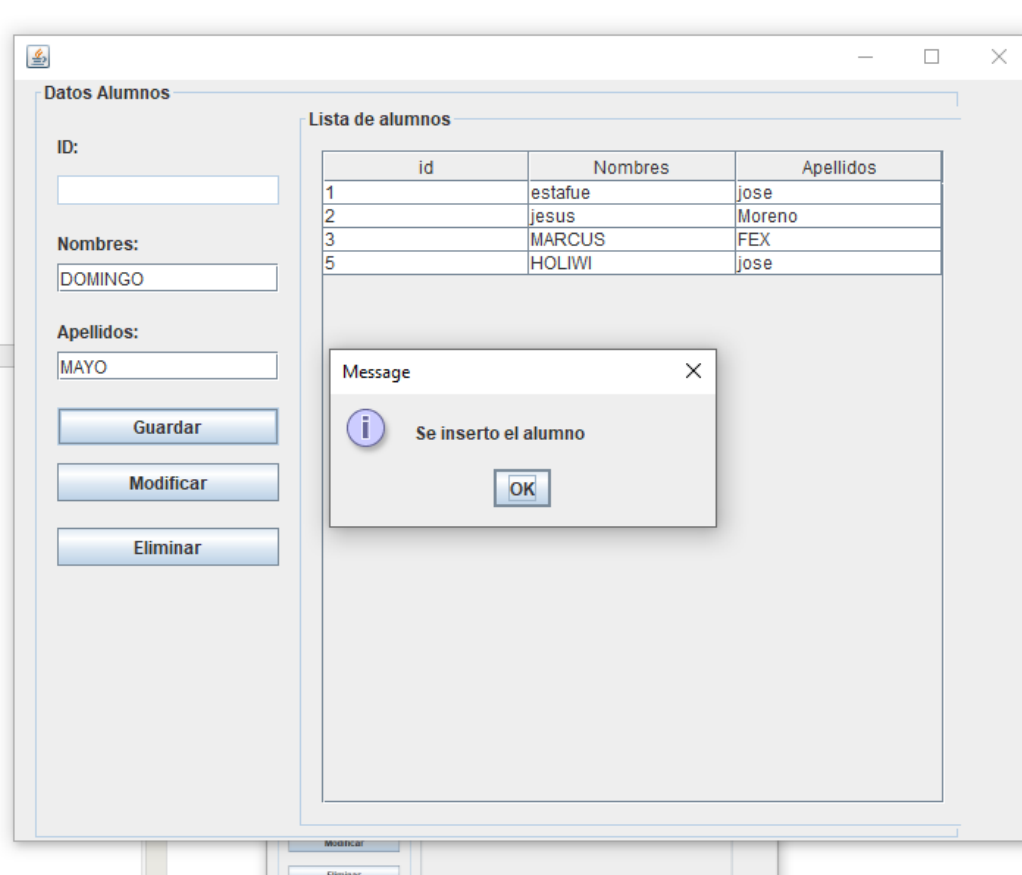
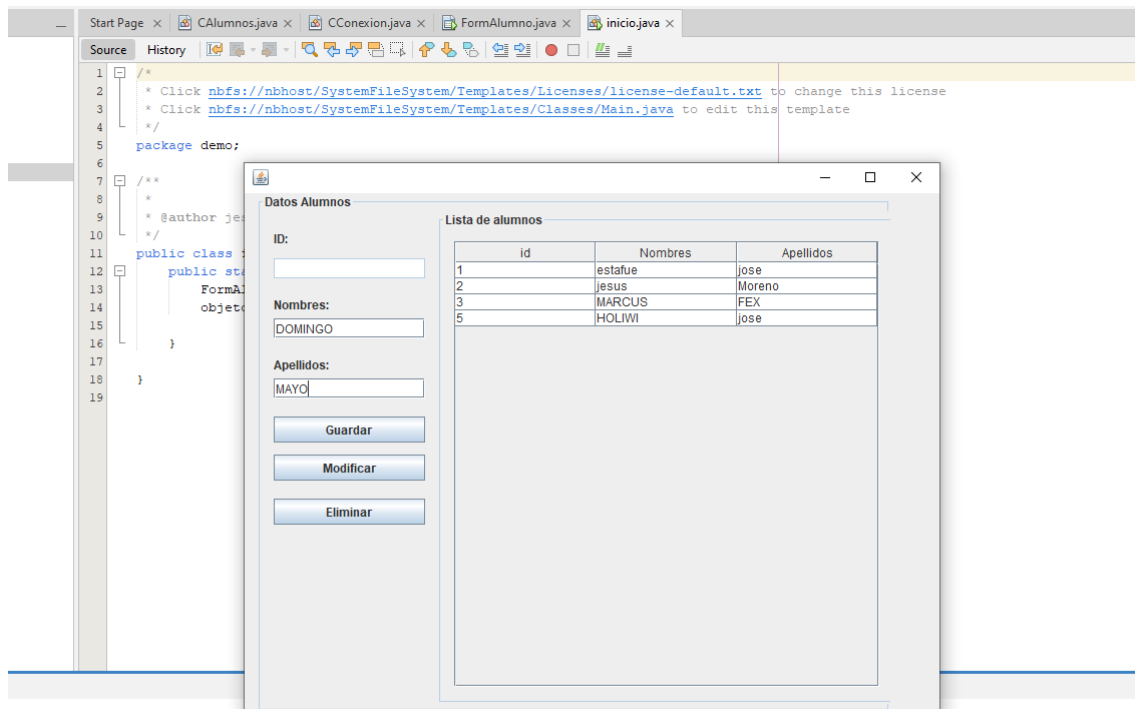
Inicio.java:



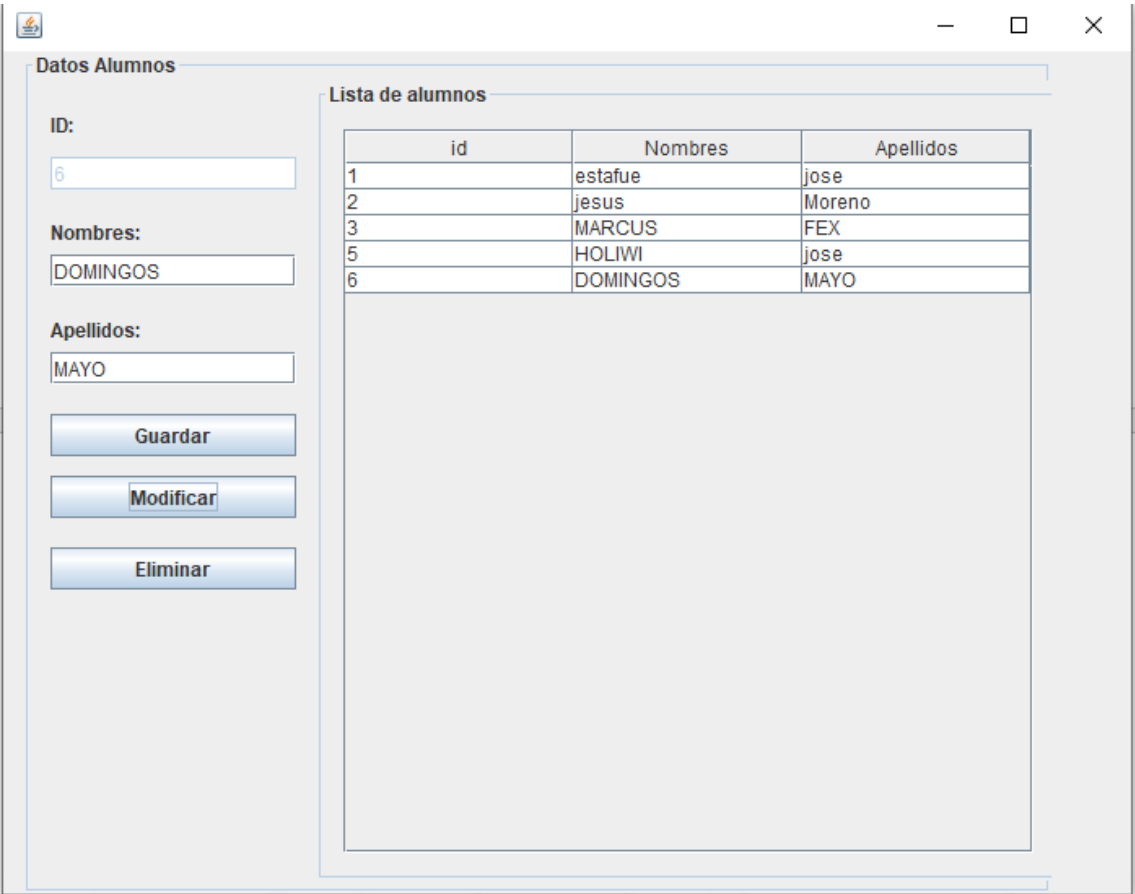
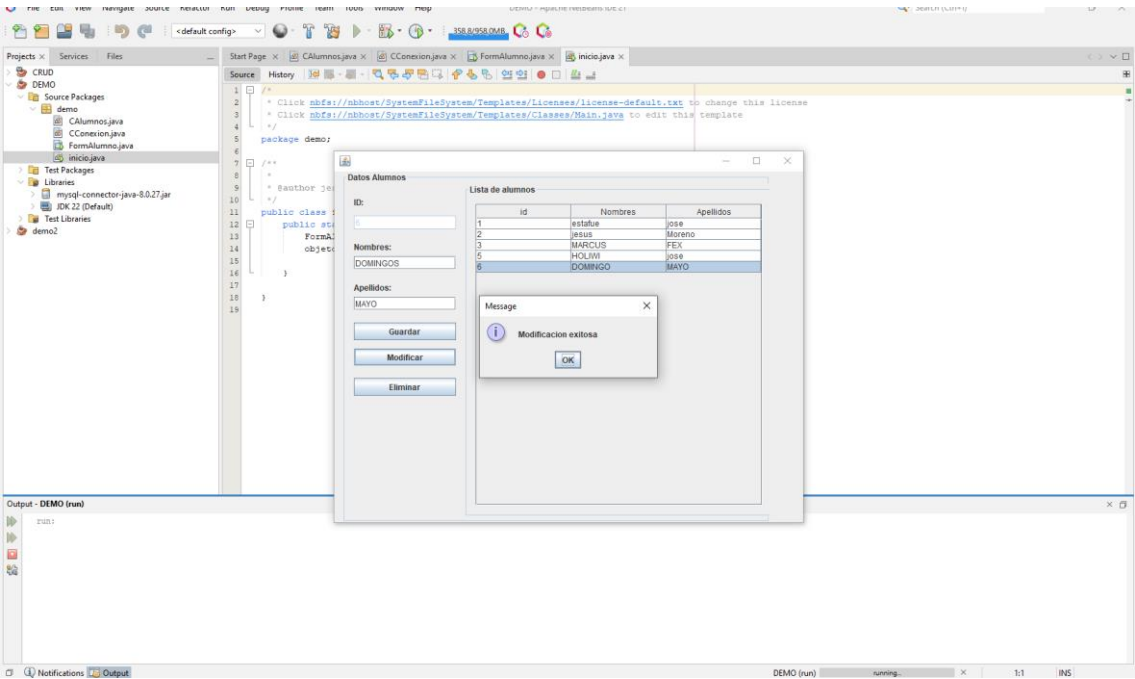
EJECUCION:



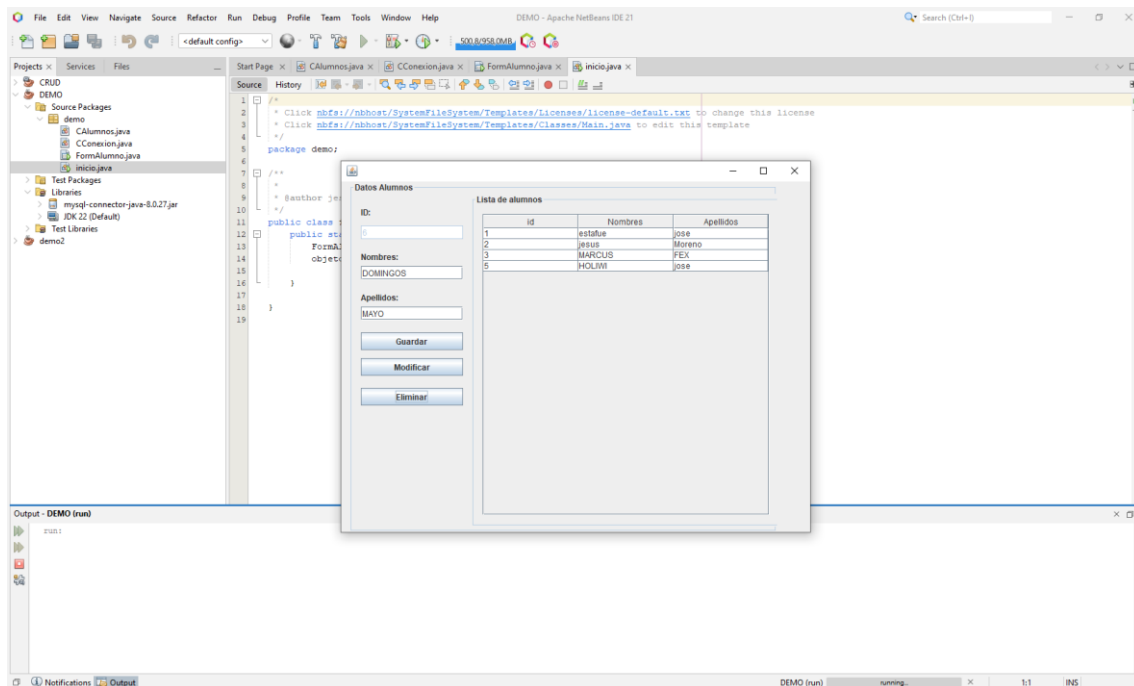
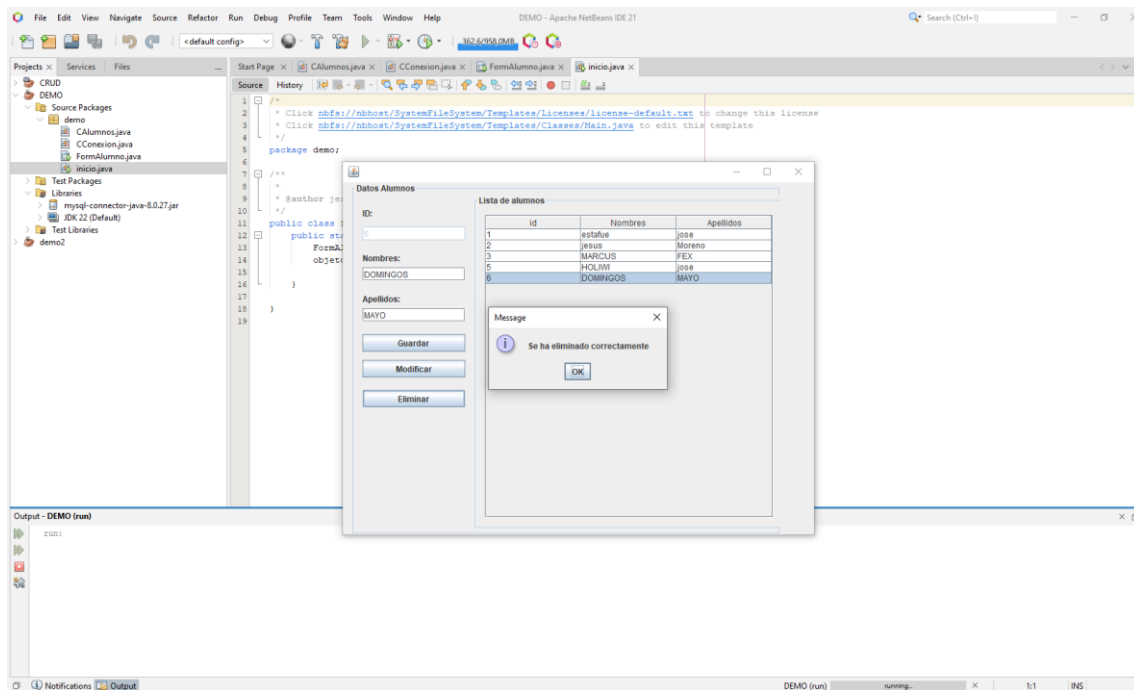
GUARDAR:



MODIFICACION:



ELIMINAR:



Enlace GitHub:

https://github.com/JESUSHM313/SENA_codificacion_de_modulos

CONCLUSION

La utilización de CRUD en Java para MySQL es fundamental para el desarrollo de aplicaciones que requieren interacción con bases de datos, ya que proporciona las herramientas necesarias para gestionar la información de manera eficiente y segura.