

### JUSTIFICACIÓN DEL PROYECTO

ESCUELA : TECNOLOGÍAS DE LA INFORMACIÓN  
CARRERA : COMPUTACIÓN E INFORMÁTICA  
CURSO : (1814) ALGORITMOS Y ESTRUCTURA DE DATOS  
SEMESTRE : 2021 – I  
CICLO : SEGUNDO

---

# Instituto de Educación Superior Privado **CIBERTEC**

#### **Integrantes:**

- |                                     |                         |
|-------------------------------------|-------------------------|
| 1. Arturo Contreras Montoya         | 202022525 (Coordinador) |
| 2. Abraham Ramírez Espino           | 202020305               |
| 3. Bryan Alexander Quintana Soriano | 202020399               |
| 4. Santiago Alexander Bonilla Acuña | 202013960               |

#### **Sección:**

- T2BT

#### **Carrera:**

- Escuela de Tecnología

#### **Tema:**

- Proyecto “Registro y matrícula de alumnos”

## 2021

# Contenido

## CIBERTEC

Carátula	1
1. Descripción del Proyecto	3
2. Justificación del Proyecto	3
3. Conclusión de cada uno de los integrantes según su aporte al proyecto	5
4. Código fuente de todos los diseños y pantallas principales	6
5. Un link para descargar el proyecto final empaquetado y un breve audio o video que contenga la exposición de cada uno de los integrantes en función a la parte que le tocó desarrollar	

## 1. DESCRIPCIÓN DEL PROYECTO

Este proyecto tiene como objetivo: Ofrecer una solución al registro y matrícula de alumnos para una institución educativa, en los cuales se podrá modificar, añadir, eliminar, y visualizar diferentes tipos de datos. A través de lo aprendido en el curso, iniciando por su creación a través de algoritmos e implementarlos en el lenguaje Java.

## 2. JUSTIFICACIÓN DEL PROYECTO

El resultado del proyecto culminado son grandes aportes del manejo de objetos y múltiples arreglos para mantener las buenas prácticas en la programación cumpliendo con los estándares EOR (eficiente, optimizado y reducir el trabajo manual).

## 3. CONCLUSIÓN DE CADA UNO DE LOS INTEGRANTES SEGÚN SU APOORTE AL PROYECTO

### Arturo Contreras Montoya:

Clase Alumno
Clase Curso
Arreglo ArregloCurso
Arreglo ArregloMatriculas
Arreglo ArregloRetiro
_02MantenimientoAlumno
_03MantenimientoCurso
_04RegistroMatricula
_05RegistroRetiro
Libreria Fecha
Libreria Libreria

### Abraham Ramírez Espino:

Clase Matricula
Clase Retiro
_06ConsultaAlumnos
_07ConsultaCursos
_08ConsultaMatriculas
_09ConsultaRetiros

### Bryan Alexander Quintana Soriano:

GUI's
Arreglo ArregloAlumno
_01MenuPrincipal
_02MantenimientoAlumno
_10ReporteAlumnosMatriculaPendiente
_11ReporteAlumnosMatriculaVigente
_12ReporteAlumnosMatriculadosCurso

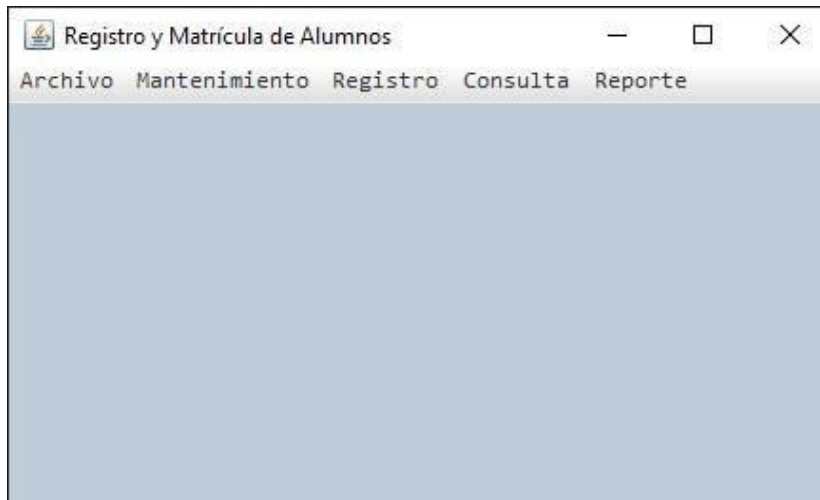
### Santiago Alexander Bonilla Acuña:

GUI's
Justificación del Proyecto

El programa permite que los procesos de las matrículas del alumnado se realicen rápidamente y estos queden almacenados en un archivo de texto. El monitoreo de las matrículas y de los alumnos se pueden revisar de forma eficiente con los reportes que tiene el programa.

#### 4. CÓDIGO FUENTE DE TODOS LOS DISEÑOS Y PANTALLAS PRINCIPALES

// Menú Principal



```
package interfases;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
import java.awt.Font;

public class _01MenuPrincipal extends JFrame implements
ActionListener {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JMenuBar menuProyecto;
    private JMenu mnArchivo;
    private JMenuItem mntmSalir;
    private JMenu mnMantenimiento;
    private JMenuItem mntmCurso;
    private JMenuItem mntmAlumno;
    private JMenu mnRegistro;
    private JMenuItem mntmMatricula;
    private JMenuItem mntmRetiro;
    private JMenuItem mntmAlumnos;
    private JMenuItem mntmCursos;
    private JMenuItem mntmMatriculaPendiente;
    private JMenuItem mntmMatriculaVigente;
    private JMenuItem mntmMatriculadosPorCurso;
```

```

private JMenu mnConsulta;
private JMenu mnReporte;
private JMenuItem mntmMatriculas;
private JMenuItem mntmRetiros;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _01MenuPrincipal frame = new
_01MenuPrincipal();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public _01MenuPrincipal() {
    setTitle("Registro y Matr\u00E9cula de Alumnos");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);

    menuProyecto = new JMenuBar();
    setJMenuBar(menuProyecto);

    mnArchivo = new JMenu("Archivo");
    mnArchivo.setFont(new Font("Consolas", Font.PLAIN, 12));
    menuProyecto.add(mnArchivo);

    mntmSalir = new JMenuItem("Salir");
    mntmSalir.setFont(new Font("Consolas", Font.PLAIN, 12));
    mntmSalir.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            System.exit(0);
        }
    });
    mnArchivo.add(mntmSalir);

    mnMantenimiento = new JMenu("Mantenimiento");
    mnMantenimiento.setFont(new Font("Consolas", Font.PLAIN,
12));
    menuProyecto.add(mnMantenimiento);

```

```

mnmAlumno = new JMenuItem("Alumno");
mnmAlumno.setFont(new Font("Consolas", Font.PLAIN, 12));
mnmAlumno.addActionListener(this);
mnMantenimiento.add(mnmAlumno);

mnmCurso = new JMenuItem("Curso");
mnmCurso.setFont(new Font("Consolas", Font.PLAIN, 12));
mnmCurso.addActionListener(this);
mnMantenimiento.add(mnmCurso);

mnRegistro = new JMenu("Registro");
mnRegistro.setFont(new Font("Consolas", Font.PLAIN, 12));
menuProyecto.add(mnRegistro);

mnmMatricula = new JMenuItem("Matricula");
mnmMatricula.setFont(new Font("Consolas", Font.PLAIN,
12));
mnmMatricula.addActionListener(this);
mnRegistro.add(mnmMatricula);

mnmRetiro = new JMenuItem("Retiro");
mnmRetiro.setFont(new Font("Consolas", Font.PLAIN, 12));
mnmRetiro.addActionListener(this);
mnRegistro.add(mnmRetiro);

mnConsulta = new JMenu("Consulta");
mnConsulta.setFont(new Font("Consolas", Font.PLAIN, 12));
menuProyecto.add(mnConsulta);

mnmAlumnos = new JMenuItem("Alumnos");
mnmAlumnos.setFont(new Font("Consolas", Font.PLAIN,
12));
mnmAlumnos.addActionListener(this);
mnConsulta.add(mnmAlumnos);

mnmCursos = new JMenuItem("Cursos");
mnmCursos.setFont(new Font("Consolas", Font.PLAIN, 12));
mnmCursos.addActionListener(this);
mnConsulta.add(mnmCursos);

mnmMatriculas = new JMenuItem("Matriculas");
mnmMatriculas.addActionListener(this);
mnmMatriculas.setFont(new Font("Consolas", Font.PLAIN,
12));
mnConsulta.add(mnmMatriculas);

mnmRetiros = new JMenuItem("Retiros");
mnmRetiros.addActionListener(this);
mnmRetiros.setFont(new Font("Consolas", Font.PLAIN,
12));
mnConsulta.add(mnmRetiros);

```

```

        mnReporte = new JMenu("Reporte");
        mnReporte.setFont(new Font("Consolas", Font.PLAIN, 12));
        menuProyecto.add(mnReporte);

        mntmMatriculaPendiente = new JMenuItem("Matr\u00EDcula
pendiente");
        mntmMatriculaPendiente.setFont(new Font("Consolas",
Font.PLAIN, 12));
        mntmMatriculaPendiente.addActionListener(this);
        mnReporte.add(mntmMatriculaPendiente);

        mntmMatriculaVigente = new JMenuItem("Matr\u00EDcula
vigente");
        mntmMatriculaVigente.setFont(new Font("Consolas",
Font.PLAIN, 12));
        mntmMatriculaVigente.addActionListener(this);
        mnReporte.add(mntmMatriculaVigente);

        mntmMatriculadosPorCurso = new JMenuItem("Matriculados
por curso");
        mntmMatriculadosPorCurso.setFont(new Font("Consolas",
Font.PLAIN, 12));
        mntmMatriculadosPorCurso.addActionListener(this);
        mnReporte.add(mntmMatriculadosPorCurso);
        contentPane = new JPanel();
        contentPane.setBackground(SystemColor.inactiveCaption);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == mntmMatriculas) {
            actionPerformedMntmMatriculas(e);
        }
        if (e.getSource() == mntmAlumno) {
            actionPerformedMntmAlumno(e);
        }
        if (e.getSource() == mntmCurso) {
            actionPerformedMntmCurso(e);
        }
        if (e.getSource() == mntmMatricula) {
            actionPerformedMntmMatricula(e);
        }
        if (e.getSource() == mntmRetiro) {
            actionPerformedMntmRetiro(e);
        }
        if (e.getSource() == mntmAlumnos) {
            actionPerformedMntmAlumnos(e);
        }
        if (e.getSource() == mntmCursos) {

```

```

        actionPerformedMntmCursos(e);
    }
    if (e.getSource() == mntmMatriculas) {
        actionPerformedMntmMatriculas(e);
    }
    if (e.getSource() == mntmRetiros) {
        actionPerformedMntmRetiros(e);
    }
    if (e.getSource() == mntmMatriculaPendiente) {
        actionPerformedMntmMatriculaPendiente(e);
    }
    if (e.getSource() == mntmMatriculaVigente) {
        actionPerformedMntmMatriculaVigente(e);
    }
    if (e.getSource() == mntmMatriculadosPorCurso) {
        actionPerformedMntmMatriculadoPorCursos(e);
    }
}

/**
 * Enlazando interfaces
 */
protected void actionPerformedMntmAlumno(ActionEvent e) {
    _02MantenimientoAlumno ma = new _02MantenimientoAlumno();
    ma.setLocationRelativeTo(this);
    ma.setVisible(true);
}
protected void actionPerformedMntmCurso(ActionEvent e) {
    _03MantenimientoCurso mc = new _03MantenimientoCurso();
    mc.setLocationRelativeTo(this);
    mc.setVisible(true);
}
protected void actionPerformedMntmMatricula(ActionEvent e) {
    _04RegistroMatricula rm = new _04RegistroMatricula();
    rm.setLocationRelativeTo(this);
    rm.setVisible(true);
}
protected void actionPerformedMntmRetiro(ActionEvent e) {
    _05RegistroRetiro rr = new _05RegistroRetiro();
    rr.setLocationRelativeTo(this);
    rr.setVisible(true);
}
protected void actionPerformedMntmAlumnos(ActionEvent e) {
    _06ConsultaAlumnos ca = new _06ConsultaAlumnos();
    ca.setLocationRelativeTo(this);
    ca.setVisible(true);
}
protected void actionPerformedMntmCursos(ActionEvent e) {
    _07ConsultaCursos ca = new _07ConsultaCursos();
    ca.setLocationRelativeTo(this);
    ca.setVisible(true);
}

```



```

    }
    protected void actionPerformedMntmMatriculas(ActionEvent e) {
        _08ConsultaMatriculas ca = new _08ConsultaMatriculas();
        ca.setLocationRelativeTo(this);
        ca.setVisible(true);
    }
    protected void actionPerformedMntmRetiros(ActionEvent e) {
        _09ConsultaRetiros ca = new _09ConsultaRetiros();
        ca.setLocationRelativeTo(this);
        ca.setVisible(true);
    }
    protected void
actionPerformedMntmMatriculaPendiente(ActionEvent e) {
        _10ReporteAlumnosMatriculaPendiente ramp = new
_10ReporteAlumnosMatriculaPendiente();
        ramp.setLocationRelativeTo(this);
        ramp.setVisible(true);
    }
    protected void actionPerformedMntmMatriculaVigente(ActionEvent
e) {
        _11ReporteAlumnosMatriculaVigente ramv = new
_11ReporteAlumnosMatriculaVigente();
        ramv.setLocationRelativeTo(this);
        ramv.setVisible(true);
    }
    protected void
actionPerformedMntmMatriculadoPorCursos(ActionEvent e) {
        _12ReporteAlumnosMatriculadosCurso ramc = new
_12ReporteAlumnosMatriculadosCurso();
        ramc.setLocationRelativeTo(this);
        ramc.setVisible(true);
    }
}

```

## // Mantenimiento Alumno

Mantenimiento | Alumno

Código

202010001

Nombres

Arturo Bruno

Apellidos

Contreras Montoya

DNI

75756454

Edad

25

Celular

964936467

Estado

Registrado

Adicionar

Modificar

Eliminar

Código	Nombres	Apellidos	DNI	Edad	Celular	Estado
202010001	Arturo Bruno	Contreras Montoya	75756454	25	964936467	Registrado
202010002	Bryan Alexander	Quintana Bonilla	70308527	20	940123075	Matriculado
202010003	Santiago Alexander	Bonilla Acuña	73227197	19	970310434	Registrado
202010004	Abraham	Ramirez Espino	75647589	20	915253936	Matriculado
202010005	Magaly Francisca	Zavala Ibañez	75645364	31	968576890	Matriculado
202010006	Alex Jordy	Silva Mogollon	75645675	27	998374675	Retirado
202010007	Anderson Esteban	Cisneros Miñan	88574634	26	993847757	Retirado
202010008	Carlos Daniel	Bonilla Acuña	73226712	21	936636422	Registrado

package interfases;

```
import clases.Alumno;
import arreglos.ArregloAlumno;
import java.awt.EventQueue;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.awt.Color;
import java.awt.SystemColor;
import java.awt.Font;
import javax.swing.JDialog;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumnModel;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JSeparator;

public class _02MantenimientoAlumno extends JDialog implements
ActionListener, MouseListener {

    private static final long serialVersionUID = 1L;
    private JButton btnAdicionar;
```

```

private JButton btnModificar;
private JButton btnEliminar;
private JTextField txtCodigo;
private JTextField txtNombres;
private JTextField txtApellidos;
private JTextField txtDNI;
private JTextField txtEdad;
private JTextField txtCelular;
private JComboBox <String> cboEstado;
private JLabel lblCodigo;
private JLabel lblApellidos;
private JLabel lblNombres;
private JLabel lblDni;
private JLabel lblEdad;
private JScrollPane scrollPane;
private JTable tblAlumno;
private DefaultTableModel modelo;
private JSeparator separator;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _02MantenimientoAlumno dialog = new
_02MantenimientoAlumno();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _02MantenimientoAlumno() {
    setForeground(new Color(0, 0, 0));
    setBackground(Color.WHITE);

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Mantenimiento | Alumno");
    setBounds(100, 100, 650, 450);
    getContentPane().setLayout(null);

```

```

lblCodigo = new JLabel("C\u00F3digo");
lblCodigo.setFont(new Font("Consolas", Font.PLAIN, 10));
lblCodigo.setBounds(10, 10, 46, 14);
getContentPane().add(lblCodigo);

txtCodigo = new JTextField();
txtCodigo.setEditable(false);
txtCodigo.setFont(new Font("Consolas", Font.PLAIN, 10));
txtCodigo.setBounds(73, 8, 130, 20);
getContentPane().add(txtCodigo);
txtCodigo.setColumns(10);

lblNombres = new JLabel("Nombres");
lblNombres.setFont(new Font("Consolas", Font.PLAIN, 10));
lblNombres.setBounds(10, 34, 66, 14);
getContentPane().add(lblNombres);

txtNombres = new JTextField();
txtNombres.setEditable(false);
txtNombres.setFont(new Font("Consolas", Font.PLAIN, 10));
txtNombres.setBounds(73, 32, 130, 20);
getContentPane().add(txtNombres);
txtNombres.setColumns(10);

lblApellidos = new JLabel("Apellidos");
lblApellidos.setFont(new Font("Consolas", Font.PLAIN,
10));
lblApellidos.setBounds(10, 58, 65, 14);
getContentPane().add(lblApellidos);

txtApellidos = new JTextField();
txtApellidos.setEditable(false);
txtApellidos.setFont(new Font("Consolas", Font.PLAIN,
10));
txtApellidos.setBounds(73, 56, 130, 20);
getContentPane().add(txtApellidos);
txtApellidos.setColumns(10);

lblDni = new JLabel("DNI");
lblDni.setFont(new Font("Consolas", Font.PLAIN, 10));
lblDni.setBounds(10, 82, 46, 14);
getContentPane().add(lblDni);

txtDNI = new JTextField();
txtDNI.setEditable(false);
txtDNI.setFont(new Font("Consolas", Font.PLAIN, 10));
txtDNI.setBounds(73, 80, 130, 20);
getContentPane().add(txtDNI);
txtDNI.setColumns(10);

lblEdad = new JLabel("Edad");

```

```

lblEdad.setFont(new Font("Consolas", Font.PLAIN, 10));
lblEdad.setBounds(10, 106, 46, 14);
getContentPane().add(lblEdad);

txtEdad = new JTextField();
txtEdad.setEditable(false);
txtEdad.setFont(new Font("Consolas", Font.PLAIN, 10));
txtEdad.setBounds(73, 104, 130, 20);
getContentPane().add(txtEdad);
txtEdad.setColumns(10);

JLabel lblCelular = new JLabel("Celular");
lblCelular.setFont(new Font("Consolas", Font.PLAIN, 10));
lblCelular.setBounds(10, 130, 46, 14);
getContentPane().add(lblCelular);

txtCelular = new JTextField();
txtCelular.setEditable(false);
txtCelular.setFont(new Font("Consolas", Font.PLAIN, 10));
txtCelular.setBounds(73, 128, 130, 20);
getContentPane().add(txtCelular);
txtCelular.setColumns(10);

JLabel lblEstado = new JLabel("Estado");
lblEstado.setFont(new Font("Consolas", Font.PLAIN, 10));
lblEstado.setBounds(10, 154, 46, 14);
getContentPane().add(lblEstado);

cboEstado = new JComboBox<String>();
cboEstado.setModel(new DefaultComboBoxModel<String>(new
String[] { "Registrado", "Matriculado", "Retirado"}));
cboEstado.setEnabled(false);
cboEstado.setFont(new Font("Consolas", Font.PLAIN, 10));
cboEstado.setBounds(73, 152, 130, 20);
getContentPane().add(cboEstado);

btnAdicionar = new JButton("Adicionar");
btnAdicionar.setForeground(Color.BLACK);
btnAdicionar.setFont(new Font("Consolas", Font.PLAIN,
10));
btnAdicionar.addActionListener(this);
btnAdicionar.setBounds(537, 6, 89, 23);
getContentPane().add(btnAdicionar);

btnModificar = new JButton("Modificar");
btnModificar.setForeground(Color.BLACK);
btnModificar.setFont(new Font("Consolas", Font.PLAIN,
10));
btnModificar.addActionListener(this);
btnModificar.setBounds(537, 30, 89, 23);
getContentPane().add(btnModificar);

```

```

        btnEliminar = new JButton("Eliminar");
        btnEliminar.setForeground(Color.BLACK);
        btnEliminar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnEliminar.addActionListener(this);
        btnEliminar.setBounds(537, 54, 89, 23);
        getContentPane().add(btnEliminar);

        scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 190, 616, 213);
        getContentPane().add(scrollPane);

        tblAlumno = new JTable();
        tblAlumno.addMouseListener(this);
        tblAlumno.getTableHeader().setReorderingAllowed(false);
        tblAlumno.setFillsViewportHeight(true);
        tblAlumno.setFont(new Font("Consolas", Font.PLAIN, 10));
        scrollPane.setViewportView(tblAlumno);

        modelo = new DefaultTableModel();
        modelo.addColumn("Código");
        modelo.addColumn("Nombres");
        modelo.addColumn("Apellidos");
        modelo.addColumn("DNI");
        modelo.addColumn("Edad");
        modelo.addColumn("Celular");
        modelo.addColumn("Estado");
        tblAlumno.setModel(modelo);

        separator = new JSeparator();
        separator.setBounds(10, 178, 616, 2);
        getContentPane().add(separator);

        ajustarAnchoColumnas();
        listar(0);
        obtenerDatos();
        habilitarEntradas(false);
    }

    /**
     * Programación de botones
     */
    ArregloAlumno aa = new ArregloAlumno();

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnAdicionar) {
            actionPerformedBtnAdicionar(arg0);
        }
        if (arg0.getSource() == btnModificar) {
            actionPerformedBtnModificar(arg0);
        }
    }

```

```

    }
    if (arg0.getSource() == btnEliminar) {
        actionPerformedBtnEliminar(arg0);
    }
}

// Adiciona un nuevo alumno verificando que el código no se
repita
protected void actionPerformedBtnAdicionar(ActionEvent arg0) {
    try {
        if (btnAdicionar.getForeground() == Color.BLACK) {
            btnAdicionar.setForeground(Color.BLUE);
            btnModificar.setForeground(Color.BLACK);
            btnEliminar.setForeground(Color.BLACK);
            habilitarEntradas(true);
            limpieza();
            txtNombres.requestFocus();
        } else {
            adicionarModificar();
        }
    } catch (Exception e) {
        mensaje("Datos no validos.");
        limpieza();
    }
}

// Busca un código y si existe modifica los datos del Alumno
protected void actionPerformedBtnModificar(ActionEvent arg0) {
    if (aa.tamaño() > 0) {
        if (btnModificar.getForeground() == Color.BLACK) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLUE);
            btnEliminar.setForeground(Color.BLACK);
            obtenerDatos();
            habilitarEntradas(true);
            txtNombres.requestFocus();
        } else {
            adicionarModificar();
        }
    } else {
        mensaje("No existen alumnos.");
    }
}

// Busca un código y si existe retira el curso del arreglo
protected void actionPerformedBtnEliminar(ActionEvent arg0) {
    if (aa.tamaño() > 0) {
        btnAdicionar.setForeground(Color.BLACK);
        btnModificar.setForeground(Color.BLACK);
        btnEliminar.setForeground(Color.BLUE);
        obtenerDatos();
    }
}

```

```

        habilitarEntradas(false);

        int codigo = leerCodigo();
        Alumno a = aa.buscarAlumno(codigo);

        if (a.getEstado() == 0) {
            String nombre = leerNombre();
            String apellido = leerApellido();
            int confirmacion = confirmar("¿Desea eliminar
al alumno " + nombre + " " + apellido + " ?");

            if (confirmacion == 0) {
                int posFila = tblAlumno.getSelectedRow();

                if (posFila == aa.tamaño() - 1) {
                    posFila--;
                }
                aa.eliminar(a);
                listar(posFila);
                obtenerDatos();
            }
        } else {
            mensaje("Solo se puede eliminar alumnos
registrados");
        }
        btnEliminar.setForeground(Color.BLACK);
    } else {
        mensaje("No existen alumnos.");
    }
}

/**
 * Métodos adicionales
 */
// Métodos sin valor de retorno
void adicionarModificar() {
    int codigo = aa.numeroCorrelativo();
    String nombre = leerNombre();
    String apellido = leerApellido();
    String dni = leerDNI();
    int edad = leerEdad();
    int celular = leerCelular();
    int estado = leerEstado();
    int posFila = tblAlumno.getSelectedRow();

    try {
        if (btnAdicionar.setForeground() == Color.BLUE) {
            if (aa.buscarAlumno(codigo) == null) {
                Alumno a = new Alumno(codigo, nombre,
apellido, dni, edad, celular, estado);
                posFila = aa.tamaño();
            }
        }
    }
}

```



```

        aa.adicionar(a);
        limpieza();
        txtCodigo.setText("" +
aa.numeroCorrelativo());
        txtNombres.requestFocus();
    } else {
        mensaje("El código ya existe.");
        limpieza();
    }
}
if (btnModificar.getForeground() == Color.BLUE) {
    Alumno a = aa.obtener(posFila);
    a.setNombres(nombre);
    a.setApellidos(apellido);
    a.setDNI(dni);
    a.setEdad(edad);
    a.setCelular(celular);
    a.setEstado(estado);
    btnModificar.setForeground(Color.BLACK);
    habilitarEntradas(false);
}
listar(posFila);
} catch (Exception e) {
    mensaje("Datos no validos.");
    limpieza();
}
}

void habilitarEntradas(boolean trueFalse) {
    txtNombres.setEditable(trueFalse);
    txtApellidos.setEditable(trueFalse);

    if (btnModificar.getForeground() == Color.BLUE) {
        txtDNI.setEditable(false);
    } else {
        txtDNI.setEditable(trueFalse);
    }
    txtEdad.setEditable(trueFalse);
    txtCelular.setEditable(trueFalse);

    if (btnAdicionar.getForeground() == Color.BLUE) {
        cboEstado.setEnabled(false);
    } else {
        cboEstado.setEnabled(trueFalse);
    }
}

void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s, "Información", 1);
}

void listar(int posFila) {
    modelo.setRowCount(0);
    for (int i = 0; i < aa.tamaño(); i++) {

```

```

        Object[] fila = {
            aa.obtener(i).getCodAlumno(),
            aa.obtener(i).getNombres(),
            aa.obtener(i).getApellidos(),
            aa.obtener(i).getDNI(),
            aa.obtener(i).getEdad(),
            aa.obtener(i).getCelular(),

leerEstadoString(aa.obtener(i).getEstado())
        };
        modelo.addRow(fila);
    }

    tblAlumno.getSelectionModel().setSelectionInterval(posFila,
posFila);
}
void obtenerDatos() {
    if (aa.tamaño() == 0) {
        limpieza();
    } else {
        Alumno a = aa.obtener(tblAlumno.getSelectedRow());
        txtCodigo.setText("" + a.getCodAlumno());
        txtNombres.setText("" + a.getNombres());
        txtApellidos.setText("" + a.getApellidos());
        txtDNI.setText("" + a.getDNI());
        txtEdad.setText("" + a.getEdad());
        txtCelular.setText("" + a.getCelular());
        cboEstado.setSelectedIndex(a.getEstado());
    }
}
void limpieza() {
    txtCodigo.setText("" + aa.numeroCorrelativo());
    txtNombres.setText("");
    txtApellidos.setText("");
    txtDNI.setText("");
    txtEdad.setText("");
    txtCelular.setText("");
    cboEstado.setSelectedIndex(0);
}
private void ajustarAnchoColumnas() {
    TableColumnModel tcm = tblAlumno.getColumnModel();
    tcm.getColumn(0).setPreferredWidth(anchoColumna(6)); //
Código
    tcm.getColumn(1).setPreferredWidth(anchoColumna(15)); //
Nombre
    tcm.getColumn(2).setPreferredWidth(anchoColumna(15)); //
Apellidos
    tcm.getColumn(3).setPreferredWidth(anchoColumna(10)); //
DNI
    tcm.getColumn(4).setPreferredWidth(anchoColumna(4)); //
Edad

```

```

        tcm.getColumnModel(5).setPreferredWidth(anchoColumna(5)); //
Celular
        tcm.getColumnModel(6).setPreferredWidth(anchoColumna(10)); //
Estado
    }

    // Métodos con valor de retorno
    int leerCodigo() {
        return Integer.parseInt(txtCodigo.getText().trim());
    }
    String leerNombre() {
        return txtNombres.getText().trim();
    }
    String leerApellido() {
        return txtApellidos.getText().trim();
    }
    String leerDNI() {
        return txtDNI.getText().trim();
    }
    int leerEdad() {
        return Integer.parseInt(txtEdad.getText().trim());
    }
    int leerCelular() {
        return Integer.parseInt(txtCelular.getText().trim());
    }
    int leerEstado() {
        return cboEstado.getSelectedIndex();
    }
    String leerEstadoString(int i) {
        return cboEstado.getItemAt(i);
    }
    int confirmar(String s) {
        return JOptionPane.showConfirmDialog(this, s,
"Confirmación", 0, 1, null);
    }
    int anchoColumna(int porcentaje) {
        return porcentaje * scrollPane.getWidth() / 100;
    }

    // Acciones del mouse
    public void mouseClicked(MouseEvent arg0) {
        if (arg0.getSource() == tblAlumno) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLACK);
            habilitarEntradas(false);
            obtenerDatos();
        }
    }
    public void mouseEntered(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }

```

```
    public void mouseExited(MouseEvent arg0) {  
        // TODO Auto-generated method stub  
    }  
    public void mousePressed(MouseEvent arg0) {  
        // TODO Auto-generated method stub  
    }  
    public void mouseReleased(MouseEvent arg0) {  
        // TODO Auto-generated method stub  
    }  
}
```

## // Mantenimiento Curso



Código	Asignatura	Ciclo	Créditos	Horas
1814	Algoritmos y Estructura de Datos	Segundo	4	50
1760	Habilidades Comunicativas III	Tercero	3	30
2243	Desarrollo Personal II	Segundo	1	10
2349	Base de Datos	Segundo	4	60
2389	Modelado de Procesos de Negocio	Segundo	3	25
2394	Gestión de Datos Dinámicos	Tercero	3	25

```
package interfases;
```

```
import java.awt.EventQueue;
import javax.swing.JDialog;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumnModel;
import arreglos.ArregloCurso;
import arreglos.ArregloMatricula;
import clases.Curso;
import clases.Matricula;
import java.awt.SystemColor;
import java.awt.Font;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import java.awt.Color;
import javax.swing.JSeparator;

public class _03MantenimientoCurso extends JDialog implements
ActionListener, MouseListener {

    private static final long serialVersionUID = 1L;
    private JButton btnAdicionar;
```

```

private JButton btnModificar;
private JButton btnEliminar;
private JTextField txtCodigo;
private JComboBox <String> cboCiclo;
private JTextField txtCreditos;
private JTextField txtHoras;
private JTextField txtAsignatura;
private JLabel lblCodigo;
private JLabel lblCiclo;
private JLabel lblCreditos;
private JLabel lblHoras;
private JLabel lblAsignatura;
private JScrollPane scrollPane;
private DefaultTableModel modelo;
private JTable tblCurso;
private JSeparator separator;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _03MantenimientoCurso dialog = new
_03MantenimientoCurso();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _03MantenimientoCurso() {

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setAutoRequestFocus(false);
    setTitle("Mantenimiento | Curso");
    setBounds(100, 100, 650, 450);
    getContentPane().setLayout(null);

    btnAdicionar = new JButton("Adicionar");
    btnAdicionar.setForeground(Color.BLACK);
    btnAdicionar.setFont(new Font("Consolas", Font.PLAIN,
10));

```

```

        btnAdicionar.addActionListener(this);
        btnAdicionar.setBounds(530, 8, 96, 21);
        getContentPane().add(btnAdicionar);

        btnModificar = new JButton("Modificar");
        btnModificar.setForeground(Color.BLACK);
        btnModificar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnModificar.addActionListener(this);
        btnModificar.setBounds(530, 32, 96, 21);
        getContentPane().add(btnModificar);

        btnEliminar = new JButton("Eliminar");
        btnEliminar.setForeground(Color.BLACK);
        btnEliminar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnEliminar.addActionListener(this);
        btnEliminar.setBounds(530, 55, 96, 21);
        getContentPane().add(btnEliminar);

        txtCodigo = new JTextField();
        txtCodigo.setEditable(false);
        txtCodigo.setFont(new Font("Consolas", Font.PLAIN, 10));
        txtCodigo.setBounds(73, 9, 130, 20);
        getContentPane().add(txtCodigo);
        txtCodigo.setColumns(10);

        cboCiclo = new JComboBox<String>();
        cboCiclo.setModel(new DefaultComboBoxModel<String>(new
String[] {"Primero", "Segundo", "Tercero", "Cuarto", "Quinto",
"Sexto"}));
        cboCiclo.setEnabled(false);
        cboCiclo.setFont(new Font("Consolas", Font.PLAIN, 10));
        cboCiclo.setBounds(73, 56, 130, 20);
        getContentPane().add(cboCiclo);

        txtCreditos = new JTextField();
        txtCreditos.setEditable(false);
        txtCreditos.setFont(new Font("Consolas", Font.PLAIN,
10));

        txtCreditos.setBounds(73, 80, 130, 20);
        getContentPane().add(txtCreditos);
        txtCreditos.setColumns(10);

        txtHoras = new JTextField();
        txtHoras.setEditable(false);
        txtHoras.setFont(new Font("Consolas", Font.PLAIN, 10));
        txtHoras.setBounds(73, 104, 130, 19);
        getContentPane().add(txtHoras);
        txtHoras.setColumns(10);

```

```

txtAsignatura = new JTextField();
txtAsignatura.setEditable(false);
txtAsignatura.setFont(new Font("Consolas", Font.PLAIN,
10));

txtAsignatura.setBounds(73, 33, 130, 20);
getContentPane().add(txtAsignatura);
txtAsignatura.setColumns(10);

lblCodigo = new JLabel("C\u00F3digo");
lblCodigo.setFont(new Font("Consolas", Font.PLAIN, 10));
lblCodigo.setBounds(10, 10, 96, 16);
getContentPane().add(lblCodigo);

lblCiclo = new JLabel("Ciclo");
lblCiclo.setFont(new Font("Consolas", Font.PLAIN, 10));
lblCiclo.setBounds(10, 57, 96, 16);
getContentPane().add(lblCiclo);

lblCreditos = new JLabel("Cr\u00E9ditos");
lblCreditos.setFont(new Font("Consolas", Font.PLAIN,
10));

lblCreditos.setBounds(10, 81, 85, 16);
getContentPane().add(lblCreditos);

lblHoras = new JLabel("Horas");
lblHoras.setFont(new Font("Consolas", Font.PLAIN, 10));
lblHoras.setBounds(10, 105, 96, 16);
getContentPane().add(lblHoras);

lblAsignatura = new JLabel("Asignatura");
lblAsignatura.setFont(new Font("Consolas", Font.PLAIN,
10));

lblAsignatura.setBounds(10, 34, 96, 16);
getContentPane().add(lblAsignatura);

scrollPane = new JScrollPane();
scrollPane.setBounds(10, 145, 616, 258);
getContentPane().add(scrollPane);

tblCurso = new JTable();
tblCurso.addMouseListener(this);
tblCurso.getTableHeader().setReorderingAllowed(false);
tblCurso.setFillViewportHeight(true);
tblCurso.setFont(new Font("Consolas", Font.PLAIN, 10));
scrollPane.setViewportView(tblCurso);

separator = new JSeparator();
separator.setBounds(10, 133, 616, 2);
getContentPane().add(separator);

modelo = new DefaultTableModel();

```



```

        modelo.addColumn("Código");
        modelo.addColumn("Asignatura");
        modelo.addColumn("Ciclo");
        modelo.addColumn("Créditos");
        modelo.addColumn("Horas");
        tblCurso.setModel(modelo);

        ajustarAnchoColumnas();
        listar(0);
        obtenerDatos();
        habilitarEntradas(false);
    }

    /**
     * Programación de botones
     */
    ArregloCurso ac = new ArregloCurso();
    ArregloMatricula am = new ArregloMatricula();

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnAdicionar) {
            actionPerformedBtnAdicionar(arg0);
        }
        if (arg0.getSource() == btnModificar) {
            actionPerformedBtnModificar(arg0);
        }
        if (arg0.getSource() == btnEliminar) {
            actionPerformedBtnEliminar(arg0);
        }
    }

    // Adiciona un nuevo curso verificando que el código no se
    repita
    protected void actionPerformedBtnAdicionar(ActionEvent arg0) {
        try {
            if (btnAdicionar.setForeground() == Color.BLACK) {
                btnAdicionar.setForeground(Color.BLUE);
                btnModificar.setForeground(Color.BLACK);
                btnEliminar.setForeground(Color.BLACK);
                habilitarEntradas(true);
                limpieza();
                txtCodigo.requestFocus();
            } else {
                adicionarModificar();
            }
        } catch (Exception e) {
            mensaje("Datos no válidos.");
            limpieza();
        }
    }
}

```

```

// Busca un código y si existe modifica los datos del curso
protected void actionPerformedBtnModificar(ActionEvent arg0) {
    if (ac.tamaño() > 0) {
        if (btnModificar.getForeground() == Color.BLACK) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLUE);
            btnEliminar.setForeground(Color.BLACK);
            obtenerDatos();
            habilitarEntradas(true);
            txtAsignatura.requestFocus();
        } else {
            adicionarModificar();
        }
    } else {
        mensaje("No existen cursos.");
    }
}

// Busca un código y si existe retira el curso del arreglo
protected void actionPerformedBtnEliminar(ActionEvent arg0) {
    if (ac.tamaño() > 0) {
        btnAdicionar.setForeground(Color.BLACK);
        btnModificar.setForeground(Color.BLACK);
        btnEliminar.setForeground(Color.BLUE);
        obtenerDatos();
        habilitarEntradas(false);

        int codigo = leerCodigo();
        Curso c = ac.buscarCurso(codigo);
        Matricula m = am.buscarCurso(codigo);

        if (m == null) {
            String asignatura = leerAsignatura();
            int confirmacion = confirmar("Desea eliminar la
asignatura " + asignatura + " ?");

            if (confirmacion == 0) {
                int posFila = tblCurso.getSelectedRow();

                if (posFila == ac.tamaño() - 1) {
                    posFila--;
                }
                ac.eliminar(c);
                listar(posFila);
                obtenerDatos();
            }
        } else {
            mensaje("Hay alumnos matriculados en este
curso.");
        }
    }
}

```

```

        btnEliminar.setForeground(Color.BLACK);
    } else {
        mensaje("No existen asignaturas.");
    }
}

/**
 * Métodos adicionales
 */
// Métodos sin valor de retorno
void adicionarModificar() {
    int codigo = leerCodigo();
    String asignatura = leerAsignatura();
    int ciclo = leerCiclo();
    int credits = leerCredits();
    int horas = leerHoras();
    int posFila = tblCurso.getSelectedRow();

    try {
        if (btnAdicionar.setForeground() == Color.BLUE) {
            if (ac.buscarCurso(codigo) == null) {
                Curso c = new Curso(codigo, asignatura,
ciclo, credits, horas);

                posFila = ac.tamaño();
                ac.adicionar(c);
                limpieza();
                txtCodigo.requestFocus();
            } else {
                mensaje("El código ya existe.");
                limpieza();
            }
        }
        if (btnModificar.setForeground() == Color.BLUE) {
            Curso c = ac.obtener(posFila);
            c.setCodCurso(codigo);
            c.setAsignatura(asignatura);
            c.setCiclo(ciclo);
            c.setCredits(credits);
            c.setHoras(horas);
            btnModificar.setForeground(Color.BLACK);
            habilitarEntradas(false);
        }
        listar(posFila);
    } catch (Exception e) {
        mensaje("Datos no válidos.");
        limpieza();
    }
}

void habilitarEntradas(boolean trueFalse) {
    if (btnModificar.setForeground() == Color.BLUE) {
        txtCodigo.setEditable(false);
    }
}

```

```

        } else {
            txtCodigo.setEditable(trueFalse);
        }
        txtAsignatura.setEditable(trueFalse);
        cboCiclo.setEnabled(trueFalse);
        txtCreditos.setEditable(trueFalse);
        txtHoras.setEditable(trueFalse);
    }
    void mensaje(String s) {
        JOptionPane.showMessageDialog(this, s);
    }
    void listar(int posFila) {
        modelo.setRowCount(0);
        for (int i = 0; i < ac.tamaño(); i++) {
            Object[] fila = {
                ac.obtener(i).getCodCurso(),
                ac.obtener(i).getAsignatura(),

                leerCicloString(ac.obtener(i).getCiclo()),
                ac.obtener(i).getCreditos(),
                ac.obtener(i).getHoras()
            };
            modelo.addRow(fila);
        }

        tblCurso.getSelectionModel().setSelectionInterval(posFila,
posFila);
    }
    void obtenerDatos() {
        if (ac.tamaño() == 0) {
            limpieza();
        } else {
            Curso c = ac.obtener(tblCurso.getSelectedRow());
            txtCodigo.setText("" + c.getCodCurso());
            txtAsignatura.setText("" + c.getAsignatura());
            cboCiclo.setSelectedIndex(c.getCiclo());
            txtCreditos.setText("" + c.getCreditos());
            txtHoras.setText("" + c.getHoras());
        }
    }
    void limpieza() {
        txtCodigo.setText("");
        txtAsignatura.setText("");
        cboCiclo.setSelectedIndex(0);
        txtCreditos.setText("");
        txtHoras.setText("");
    }
    private void ajustarAnchoColumnas() {
        TableColumnModel tcm = tblCurso.getColumnModel();
        tcm.getColumn(0).setPreferredWidth(anchoColumna(10)); //

```

Código

```

        tcm.getColumnModel(1).setPreferredWidth(anchoColumna(30)); //
Asignatura        tcm.getColumnModel(2).setPreferredWidth(anchoColumna(10)); //
Ciclo            tcm.getColumnModel(3).setPreferredWidth(anchoColumna(10)); //
Créditos        tcm.getColumnModel(4).setPreferredWidth(anchoColumna(10)); //
Horas
    }

    // Métodos con valor de retorno
    int leerCodigo() {
        return Integer.parseInt(txtCodigo.getText().trim());
    }
    String leerAsignatura() {
        return txtAsignatura.getText().trim();
    }
    int leerCiclo() {
        return cboCiclo.getSelectedIndex();
    }
    String leerCicloString(int i) {
        return cboCiclo.getItemAt(i);
    }
    int leerCreditos() {
        return Integer.parseInt(txtCreditos.getText().trim());
    }
    int leerHoras() {
        return Integer.parseInt(txtHoras.getText().trim());
    }
    int confirmar(String s) {
        return JOptionPane.showConfirmDialog(this, s,
"Confirmación", 0, 1, null);
    }
    int anchoColumna(int porcentaje) {
        return porcentaje * scrollPane.getWidth() / 100;
    }

    // Acciones del mouse
    public void mouseClicked(MouseEvent arg0) {
        if (arg0.getSource() == tblCurso) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLACK);
            habilitarEntradas(false);
            obtenerDatos();
        }
    }
    public void mouseEntered(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }
    public void mouseExited(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }

```

```
}  
public void mousePressed(MouseEvent arg0) {  
    // TODO Auto-generated method stub  
}  
public void mouseReleased(MouseEvent arg0) {  
    // TODO Auto-generated method stub  
}  
}
```

## // Registro de Matrícula

Registro | Matricula

Número de Matrícula

100001

Adicionar

Código Alumno

202010002

Modificar

Código Curso

1814

Eliminar

Alumno

Bryan Alexander Quintana Bonilla

Curso

Algoritmos y Estructura de Datos

Número	Cod. Alumno	Cod. Curso	Fecha	Hora	Estado
100001	202010002	1814	24/06/2021	23:54:50	Matriculado
100002	202010007	1760	24/06/2021	23:55:43	Retirado
100003	202010006	2394	24/06/2021	23:55:59	Retirado
100004	202010005	1814	24/06/2021	23:56:49	Matriculado
100005	202010004	2243	24/06/2021	23:57:15	Matriculado

package interfases;

```
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import java.awt.SystemColor;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import arreglos.ArregloAlumno;
import arreglos.ArregloCurso;
import arreglos.ArregloMatricula;
import clases.Alumno;
import clases.Curso;
import clases.Matricula;
import libreria.Fecha;
import libreria.Libreria;
import javax.swing.JSeparator;
```

```

public class _04RegistroMatricula extends JDialog implements
ActionListener, MouseListener {

    private static final long serialVersionUID = 1L;
    private final JPanel contentPanel = new JPanel();
    private JScrollPane scp;
    private JLabel lblNumeroMatricula;
    private JLabel lblCodigoAlumno;
    private JLabel lblCodigoCurso;
    private JTextField txtNumeroMatricula;
    private JTextField txtCodigoAlumno;
    private JTextField txtCodigoCurso;
    private JButton btnAdicionar;
    private JButton btnModificar;
    private JButton btnEliminar;
    private JTable tblMatricula;
    private DefaultTableModel modelo;
    private JLabel lblAlumno;
    private JLabel lblCurso;
    private JTextField txtAlumno;
    private JTextField txtCurso;
    private JSeparator separator;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        try {
            _04RegistroMatricula dialog = new
_04RegistroMatricula();

            dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
            dialog.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * Create the dialog.
     */
    public _04RegistroMatricula() {
        setTitle("Registro | Matricula");
        setBounds(100, 100, 650, 450);
        getContentPane().setLayout(new BorderLayout());
        contentPanel.setBackground(SystemColor.inactiveCaption);
        contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
        getContentPane().add(contentPanel, BorderLayout.CENTER);
        contentPanel.setLayout(null);

        scp = new JScrollPane();

```



```

        scp.setBounds(10, 143, 616, 260);
        contentPanel.add(scp);

        lblNumeroMatricula = new JLabel("N\u00F3mero de
Matr\u00EDcula");
        lblNumeroMatricula.setFont(new Font("Consolas",
Font.PLAIN, 10));
        lblNumeroMatricula.setBounds(10, 10, 127, 16);
        contentPanel.add(lblNumeroMatricula);

        lblCodigoAlumno = new JLabel("C\u00F3digo Alumno");
        lblCodigoAlumno.setFont(new Font("Consolas", Font.PLAIN,
10));
        lblCodigoAlumno.setBounds(10, 34, 127, 16);
        contentPanel.add(lblCodigoAlumno);

        lblCodigoCurso = new JLabel("C\u00F3digo Curso");
        lblCodigoCurso.setFont(new Font("Consolas", Font.PLAIN,
10));
        lblCodigoCurso.setBounds(10, 57, 127, 16);
        contentPanel.add(lblCodigoCurso);

        txtNumeroMatricula = new JTextField();
        txtNumeroMatricula.setFont(new Font("Consolas",
Font.PLAIN, 10));
        txtNumeroMatricula.setEditable(false);
        txtNumeroMatricula.setColumns(10);
        txtNumeroMatricula.setBounds(147, 9, 130, 20);
        contentPanel.add(txtNumeroMatricula);

        txtCodigoAlumno = new JTextField();
        txtCodigoAlumno.setFont(new Font("Consolas", Font.PLAIN,
10));
        txtCodigoAlumno.setEditable(false);
        txtCodigoAlumno.setColumns(10);
        txtCodigoAlumno.setBounds(147, 33, 130, 20);
        contentPanel.add(txtCodigoAlumno);

        txtCodigoCurso = new JTextField();
        txtCodigoCurso.setFont(new Font("Consolas", Font.PLAIN,
10));
        txtCodigoCurso.setEditable(false);
        txtCodigoCurso.setColumns(10);
        txtCodigoCurso.setBounds(147, 56, 130, 20);
        contentPanel.add(txtCodigoCurso);

        btnAdicionar = new JButton("Adicionar");
        btnAdicionar.addActionListener(this);
        btnAdicionar.setForeground(Color.BLACK);
        btnAdicionar.setFont(new Font("Consolas", Font.PLAIN,
10));

```

```

        btnAdicionar.setBounds(530, 8, 96, 21);
        contentPanel.add(btnAdicionar);

        btnModificar = new JButton("Modificar");
        btnModificar.addActionListener(this);
        btnModificar.setForeground(Color.BLACK);
        btnModificar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnModificar.setBounds(530, 32, 96, 21);
        contentPanel.add(btnModificar);

        btnEliminar = new JButton("Eliminar");
        btnEliminar.addActionListener(this);
        btnEliminar.setForeground(Color.BLACK);
        btnEliminar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnEliminar.setBounds(530, 55, 96, 21);
        contentPanel.add(btnEliminar);

        lblAlumno = new JLabel("Alumno");
        lblAlumno.setFont(new Font("Consolas", Font.PLAIN, 10));
        lblAlumno.setBounds(10, 81, 127, 16);
        contentPanel.add(lblAlumno);

        lblCurso = new JLabel("Curso");
        lblCurso.setFont(new Font("Consolas", Font.PLAIN, 10));
        lblCurso.setBounds(10, 105, 127, 16);
        contentPanel.add(lblCurso);

        txtAlumno = new JTextField();
        txtAlumno.setFont(new Font("Consolas", Font.PLAIN, 10));
        txtAlumno.setEditable(false);
        txtAlumno.setColumns(10);
        txtAlumno.setBounds(147, 80, 235, 20);
        contentPanel.add(txtAlumno);

        txtCurso = new JTextField();
        txtCurso.setFont(new Font("Consolas", Font.PLAIN, 10));
        txtCurso.setEditable(false);
        txtCurso.setColumns(10);
        txtCurso.setBounds(147, 104, 235, 20);
        contentPanel.add(txtCurso);

        tblMatricula = new JTable();
        tblMatricula.addMouseListener(this);

        tblMatricula.getTableHeader().setReorderingAllowed(false);
        tblMatricula.setFillViewportHeight(true);
        tblMatricula.setFont(new Font("Consolas", Font.PLAIN,
10));

        scp.setViewportView(tblMatricula);

```

```

        modelo = new DefaultTableModel();
        modelo.addColumn("Número");
        modelo.addColumn("Cod. Alumno");
        modelo.addColumn("Cod. Curso");
        modelo.addColumn("Fecha");
        modelo.addColumn("Hora");
        modelo.addColumn("Estado");
        tblMatricula.setModel(modelo);

        separator = new JSeparator();
        separator.setBounds(10, 131, 616, 2);
        contentPanel.add(separator);

        listar(0);
        obtenerDatos();
        habilitarEntradas(false);
    }

    /**
     * Programación de botones
     */
    ArregloMatricula am = new ArregloMatricula();
    ArregloAlumno aa = new ArregloAlumno();
    ArregloCurso ac = new ArregloCurso();

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnAdicionar) {
            actionPerformedBtnAdicionar(arg0);
        }
        if (arg0.getSource() == btnModificar) {
            actionPerformedBtnModificar(arg0);
        }
        if (arg0.getSource() == btnEliminar) {
            actionPerformedBtnEliminar(arg0);
        }
    }

    // Adiciona una nueva matrícula
    protected void actionPerformedBtnAdicionar(ActionEvent arg0) {
        try {
            if (btnAdicionar.getForeground() == Color.BLACK) {
                btnAdicionar.setForeground(Color.BLUE);
                btnModificar.setForeground(Color.BLACK);
                btnEliminar.setForeground(Color.BLACK);
                habilitarEntradas(true);
                limpieza();
                txtCodigoAlumno.requestFocus();
            } else {
                adicionarModificar();
            }
        }
    }

```

```

    } catch (Exception e) {
        mensaje("Datos no válidos.");
        limpieza();
    }
}

// Busca un código y si existe modifica los datos de la
matrícula
protected void actionPerformedBtnModificar(ActionEvent e) {
    if (am.tamaño() > 0) {
        if (btnModificar.getForeground() == Color.BLACK) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLUE);
            btnEliminar.setForeground(Color.BLACK);
            obtenerDatos();
            habilitarEntradas(true);
            txtCodigoCurso.requestFocus();
        } else {
            adicionarModificar();
        }
    } else {
        mensaje("No existen matrículas.");
    }
}

// Busca un código y si existe retira la matrícula del arreglo
protected void actionPerformedBtnEliminar(ActionEvent e) {
    if (am.tamaño() > 0) {
        btnAdicionar.setForeground(Color.BLACK);
        btnModificar.setForeground(Color.BLACK);
        btnEliminar.setForeground(Color.BLUE);

        obtenerDatos();
        habilitarEntradas(false);

        int numeroMatricula = leerNumeroMatricula();
        Matricula m = am.buscarMatricula(numeroMatricula);
        Alumno a = aa.buscarAlumno(m.getCodAlumno());

        if (a.getEstado() == 2) {
            mensaje("El alumno se encuentra retirado.");
        } else {
            int confirmacion = confirmar("¿Desea eliminar
el registro?");

            if (confirmacion == 0) {
                int posFila =
tblMatricula.getSelectedRow();

                if (posFila == am.tamaño() - 1) {
                    posFila--;

```

```

        }
        a.setEstado(0);
        aa.actualizarArchivo();
        am.eliminar(m);
        listar(posFila);
        obtenerDatos();
    }
}
btnEliminar.setForeground(Color.BLACK);
} else {
    mensaje("No existen matrículas.");
}
}

/**
 * Métodos adicionales
 */
// Métodos sin valor de retorno
void adicionarModificar() {
    int numMatricula = leerNumeroMatricula();
    int codAlumno = leerCodigoAlumno();
    int codCurso = leerCodigoCurso();
    int posFila = tblMatricula.getSelectedRow();

    Alumno a = aa.buscarAlumno(codAlumno);
    Curso c = ac.buscarCurso(codCurso);
    Matricula m = am.buscarAlumno(codAlumno);

    try {
        if (btnAdicionar.setForeground() == Color.BLUE) {

            if (a != null && c != null) {
                if (m == null) {
                    Matricula nuevaMatricula = new
Matricula(numMatricula, codAlumno, codCurso, Fecha.fechaActual(),
Fecha.horaActual(), 1);

                    posFila = am.tamaño();
                    am.adicionar(nuevaMatricula);
                    a.setEstado(1);
                    aa.actualizarArchivo();
                    limpieza();
                    txtCodigoAlumno.requestFocus();
                } else {
                    mensaje("El alumno ya ha sido
matriculado.");

                    limpieza();
                }
            } else {
                mensaje("El alumno o el curso no se
encuentran registrados.");
                limpieza();
            }
        }
    }
}

```

```

        }
    }
    if (btnModificar.getForeground() == Color.BLUE) {

        if (c != null) {
            m.setNumMatricula(numMatricula);
            m.setCodAlumno(codAlumno);
            m.setCodCurso(codCurso);
            btnModificar.setForeground(Color.BLACK);
            habilitarEntradas(false);
        } else {
            mensaje("El código de curso ingresado no
existe.");
        }
    }
    listar(posFila);
} catch (Exception e) {
    mensaje("Datos no validos.");
    limpieza();
}
}

void habilitarEntradas(boolean trueFalse) {
    if (btnModificar.getForeground() == Color.BLUE) {
        txtCodigoAlumno.setEditable(false);
    } else {
        txtCodigoAlumno.setEditable(trueFalse);
    }
    txtCodigoCurso.setEditable(trueFalse);
}

void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s, "Información", 1);
}

void listar(int posFila) {
    modelo.setRowCount(0);
    for (int i = 0; i < am.tamaño(); i++) {
        Object[] fila = {
            am.obtener(i).getNumMatricula(),
            am.obtener(i).getCodAlumno(),
            am.obtener(i).getCodCurso(),
            am.obtener(i).getFecha(),
            am.obtener(i).getHora(),

Libreria.estadoAlumno[am.obtener(i).getEstado()]
        };
        modelo.addRow(fila);
    }

    tblMatricula.getSelectionModel().setSelectionInterval(posFila,
posFila);
}

void obtenerDatos() {

```

```

        if (am.tamaño() == 0) {
            limpieza();
        } else {
            Matricula m =
am.obtener(tblMatricula.getSelectedRow());
            txtNumeroMatricula.setText("" +
m.getNumMatricula());
            txtCodigoAlumno.setText("" + m.getCodAlumno());
            txtCodigoCurso.setText("" + m.getCodCurso());
            txtAlumno.setText(nombreAlumno());
            txtCurso.setText(nombreCurso());
        }
    }
    void limpieza() {
        txtNumeroMatricula.setText("" + am.numeroCorrelativo());
        txtCodigoAlumno.setText("");
        txtCodigoCurso.setText("");
        txtAlumno.setText("");
        txtCurso.setText("");
    }

    // Métodos con valor de retorno
    int leerNumeroMatricula() {
        return
Integer.parseInt(txtNumeroMatricula.getText().trim());
    }
    int leerCodigoAlumno() {
        return
Integer.parseInt(txtCodigoAlumno.getText().trim());
    }
    int leerCodigoCurso() {
        return Integer.parseInt(txtCodigoCurso.getText().trim());
    }
    String nombreAlumno() {
        Alumno a = aa.buscarAlumno(leerCodigoAlumno());
        return a.getNombres() + " " + a.getApellidos();
    }
    String nombreCurso() {
        Curso c = ac.buscarCurso(leerCodigoCurso());
        return c.getAsignatura();
    }
    int confirmar(String s) {
        return JOptionPane.showConfirmDialog(this, s, "Alerta",
0, 1, null);
    }

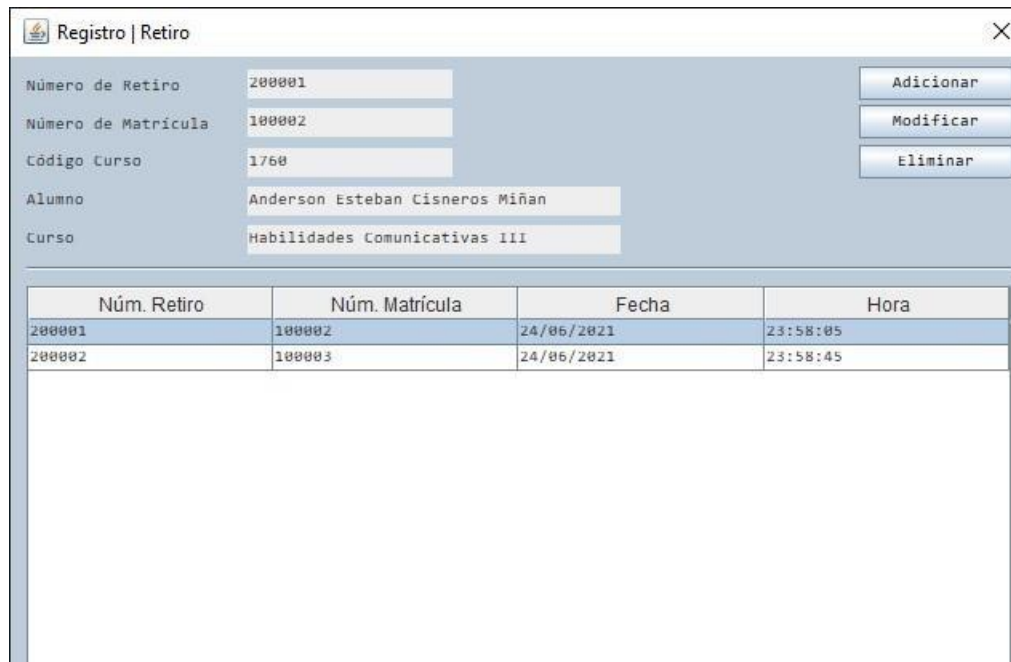
    // Acciones del mouse
    public void mouseClicked(MouseEvent arg0) {
        if (arg0.getSource() == tblMatricula) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLACK);
        }
    }

```

```
        habilitarEntradas(false);
        obtenerDatos();
    }
}
public void mouseEntered(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
public void mouseExited(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
public void mousePressed(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
public void mouseReleased(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
}
```



## // Registro Retiro



Núm. Retiro	Núm. Matrícula	Fecha	Hora
200001	100002	24/06/2021	23:58:05
200002	100003	24/06/2021	23:58:45

```
package interfases;
```

```
import java.awt.EventQueue;
import javax.swing.JDialog;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.ActionEvent;
import javax.swing.JTextField;
import java.awt.SystemColor;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JSeparator;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import arreglos.ArregloAlumno;
import arreglos.ArregloCurso;
import arreglos.ArregloMatricula;
import arreglos.ArregloRetiro;
import clases.Alumno;
import clases.Curso;
import clases.Matricula;
import clases.Retiro;
import libreria.Fecha;
```

```
public class _05RegistroRetiro extends JDialog implements
ActionListener, MouseListener {
```

```

private static final long serialVersionUID = 1L;
private JScrollPane scp;
private JLabel lblNumeroRetiro;
private JLabel lblNumeroMatricula;
private JLabel lblAlumno;
private JTextField txtNumeroRetiro;
private JTextField txtNumeroMatricula;
private JTextField txtAlumno;
private JButton btnAdicionar;
private JButton btnModificar;
private JButton btnEliminar;
private JSeparator separator;
private JTable tblRetiro;
private DefaultTableModel modelo;
private JLabel lblCurso;
private JTextField txtCurso;
private JLabel lblCodigoCurso;
private JTextField txtCodigoCurso;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _05RegistroRetiro dialog = new
_05RegistroRetiro();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _05RegistroRetiro() {

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Registro | Retiro");
    setBounds(100, 100, 650, 450);
    getContentPane().setLayout(null);

    scp = new JScrollPane();

```

```

        scp.setBounds(10, 143, 616, 260);
        getContentPane().add(scp);

        lblNumeroRetiro = new JLabel("N\u00F3mero de Retiro");
        lblNumeroRetiro.setFont(new Font("Consolas", Font.PLAIN,
10));

        lblNumeroRetiro.setBounds(10, 10, 127, 16);
        getContentPane().add(lblNumeroRetiro);

        lblNumeroMatricula = new JLabel("N\u00F3mero de
Matr\u00E9cula");
        lblNumeroMatricula.setFont(new Font("Consolas",
Font.PLAIN, 10));
        lblNumeroMatricula.setBounds(10, 34, 127, 16);
        getContentPane().add(lblNumeroMatricula);

        lblAlumno = new JLabel("Alumno");
        lblAlumno.setFont(new Font("Consolas", Font.PLAIN, 10));
        lblAlumno.setBounds(10, 81, 127, 16);
        getContentPane().add(lblAlumno);

        txtNumeroRetiro = new JTextField();
        txtNumeroRetiro.setFont(new Font("Consolas", Font.PLAIN,
10));

        txtNumeroRetiro.setEditable(false);
        txtNumeroRetiro.setColumns(10);
        txtNumeroRetiro.setBounds(147, 7, 130, 20);
        getContentPane().add(txtNumeroRetiro);

        txtNumeroMatricula = new JTextField();
        txtNumeroMatricula.setFont(new Font("Consolas",
Font.PLAIN, 10));
        txtNumeroMatricula.setEditable(false);
        txtNumeroMatricula.setColumns(10);
        txtNumeroMatricula.setBounds(147, 31, 130, 20);
        getContentPane().add(txtNumeroMatricula);

        txtAlumno = new JTextField();
        txtAlumno.setFont(new Font("Consolas", Font.PLAIN, 10));
        txtAlumno.setEditable(false);
        txtAlumno.setColumns(10);
        txtAlumno.setBounds(147, 80, 235, 20);
        getContentPane().add(txtAlumno);

        btnAdicionar = new JButton("Adicionar");
        btnAdicionar.addActionListener(this);
        btnAdicionar.setForeground(Color.BLACK);
        btnAdicionar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnAdicionar.setBounds(530, 6, 96, 21);
        getContentPane().add(btnAdicionar);

```

```

        btnModificar = new JButton("Modificar");
        btnModificar.addActionListener(this);
        btnModificar.setForeground(Color.BLACK);
        btnModificar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnModificar.setBounds(530, 30, 96, 21);
        getContentPane().add(btnModificar);

        btnEliminar = new JButton("Eliminar");
        btnEliminar.addActionListener(this);
        btnEliminar.setForeground(Color.BLACK);
        btnEliminar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnEliminar.setBounds(530, 55, 96, 21);
        getContentPane().add(btnEliminar);

        separator = new JSeparator();
        separator.setBounds(10, 131, 616, 2);
        getContentPane().add(separator);

        lblCurso = new JLabel("Curso");
        lblCurso.setFont(new Font("Consolas", Font.PLAIN, 10));
        lblCurso.setBounds(10, 105, 127, 16);
        getContentPane().add(lblCurso);

        txtCurso = new JTextField();
        txtCurso.setFont(new Font("Consolas", Font.PLAIN, 10));
        txtCurso.setEditable(false);
        txtCurso.setColumns(10);
        txtCurso.setBounds(147, 104, 235, 20);
        getContentPane().add(txtCurso);

        lblCodigoCurso = new JLabel("Codigo Curso");
        lblCodigoCurso.setFont(new Font("Consolas", Font.PLAIN,
10));

        lblCodigoCurso.setBounds(10, 57, 127, 16);
        getContentPane().add(lblCodigoCurso);

        txtCodigoCurso = new JTextField();
        txtCodigoCurso.setFont(new Font("Consolas", Font.PLAIN,
10));

        txtCodigoCurso.setEditable(false);
        txtCodigoCurso.setColumns(10);
        txtCodigoCurso.setBounds(147, 56, 130, 20);
        getContentPane().add(txtCodigoCurso);

        tblRetiro = new JTable();
        tblRetiro.addMouseListener(this);
        tblRetiro.getTableHeader().setReorderingAllowed(false);
        tblRetiro.setFillViewportHeight(true);

```

```

tblRetiro.setFont(new Font("Consolas", Font.PLAIN, 10));
scp.setViewportView(tblRetiro);

modelo = new DefaultTableModel();
modelo.addColumn("Núm. Retiro");
modelo.addColumn("Núm. Matrícula");
modelo.addColumn("Fecha");
modelo.addColumn("Hora");
tblRetiro.setModel(modelo);

listar(0);
obtenerDatos();
habilitarEntradas(false);
}

/**
 * Programación de botones
 */
ArregloRetiro ar = new ArregloRetiro();
ArregloMatricula am = new ArregloMatricula();
ArregloAlumno aa = new ArregloAlumno();
ArregloCurso ac = new ArregloCurso();

public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnAdicionar) {
        actionPerformedBtnAdicionar(arg0);
    }
    if (arg0.getSource() == btnModificar) {
        actionPerformedBtnModificar(arg0);
    }
    if (arg0.getSource() == btnEliminar) {
        actionPerformedBtnEliminar(arg0);
    }
}

// Adiciona un nuevo retiro
protected void actionPerformedBtnAdicionar(ActionEvent arg0) {
    try {
        if (btnAdicionar.getForeground() == Color.BLACK) {
            btnAdicionar.setForeground(Color.BLUE);
            btnModificar.setForeground(Color.BLACK);
            btnEliminar.setForeground(Color.BLACK);
            habilitarEntradas(true);
            limpieza();
            txtNumeroMatricula.requestFocus();
        } else {
            adicionarModificar();
        }
    } catch (Exception e) {

```

```

        mensaje("Datos no validos.");
        limpieza();
    }
}

// Modifica un retiro existente
protected void actionPerformedBtnModificar(ActionEvent arg0) {
    if (ar.tamaño() > 0) {
        if (btnModificar.getForeground() == Color.BLACK) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLUE);
            btnEliminar.setForeground(Color.BLACK);
            obtenerDatos();
            habilitarEntradas(true);
            txtCodigoCurso.requestFocus();
        } else {
            adicionarModificar();
        }
    } else {
        mensaje("No existen registros.");
    }
}

// Elimina un registro de retiro
protected void actionPerformedBtnEliminar(ActionEvent arg0) {
    if (ar.tamaño() > 0) {
        btnAdicionar.setForeground(Color.BLACK);
        btnModificar.setForeground(Color.BLACK);
        btnEliminar.setForeground(Color.BLUE);
        obtenerDatos();
        habilitarEntradas(false);

        int numeroRetiro = leerNumeroRetiro();
        Retiro r = ar.buscarRetiro(numeroRetiro);
        Matricula m =
am.buscarMatricula(r.getNumMatricula());
        Alumno a = aa.buscarAlumno(m.getCodAlumno());

        if (a.getEstado() == 2) {
            int confirmacion = confirmar("¿Desea eliminar
el registro?");

            if (confirmacion == 0) {
                int posFila = tblRetiro.getSelectedRow();

                if (posFila == ar.tamaño() - 1) {
                    posFila--;
                }
                ar.eliminar(r);
                listar(posFila);
            }
        }
    }
}

```

```

        obtenerDatos();
    }
} else {
    mensaje("El alumno no está retirado.");
}
} else {
    mensaje("No existen retiros.");
}
}

/**
 * Métodos adicionales
 */
// Métodos sin valor de retorno
void adicionarModificar() {
    int numRetiro = leerNumeroRetiro();
    int numMatricula = leerNumeroMatricula();
    int posFila = tblRetiro.getSelectedRow();

    Matricula m = am.buscarMatricula(numMatricula);

    try {
        if (btnAdicionar.getForeground() == Color.BLUE) {
            if (m != null) {
                Curso c =
ac.buscarCurso(m.getCodCurso());
                int confirmacion = confirmar("¿Deseas
retirar al alumno " + nombreAlumno() + " del curso " +
c.getAsignatura() + "?");

                if (confirmacion == 0) {
                    Retiro nuevoRetiro = new
Retiro(numRetiro, numMatricula, Fecha.fechaActual(),
Fecha.horaActual());

                    posFila = ar.tamaño();

                    ar.adicionar(nuevoRetiro);
                    Alumno a =
aa.buscarAlumno(m.getCodAlumno());

                    a.setEstado(2);
                    aa.actualizarArchivo();
                    m.setEstado(2);
                    am.actualizarArchivo();

                    limpieza();
                    txtNumeroMatricula.requestFocus();
                }
            } else {
                mensaje("El código de matrícula no
existe.");
            }
        }
    }
}

```

```

        limpieza();
        txtNumeroMatricula.requestFocus();
    }
}
if (btnModificar.getForeground() == Color.BLUE) {

    int codCurso = leerCodigoCurso();
    Curso c = ac.buscarCurso(codCurso);

    if (c != null) {
        Retiro r =
ar.buscarMatricula(numMatricula);
        r.setNumRetiro(numRetiro);
        r.setNumMatricula(numMatricula);
        btnModificar.setForeground(Color.BLACK);
        habilitarEntradas(false);
    } else {
        mensaje("El código de curso ingresado no
existe.");
    }
}
listar(posFila);
} catch (Exception e) {
    mensaje("dfdgh");
    limpieza();
}
}
void habilitarEntradas(boolean trueFalse) {
    if (btnModificar.getForeground() == Color.BLUE) {
        txtNumeroMatricula.setEditable(false);
        txtCodigoCurso.setEditable(true);
    } else {
        txtNumeroMatricula.setEditable(trueFalse);
        txtCodigoCurso.setEditable(false);
    }
}
void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s);
}
void listar(int posFila) {
    modelo.setRowCount(0);
    for (int i = 0; i < ar.tamaño(); i++) {
        Object[] fila = {
            ar.obtener(i).getNumRetiro(),
            ar.obtener(i).getNumMatricula(),
            ar.obtener(i).getFecha(),
            ar.obtener(i).getHora()
        };
        modelo.addRow(fila);
    }
}

```



```

        tblRetiro.getSelectionModel().setSelectionInterval(posFila,
posFila);
    }
    void obtenerDatos() {
        if (ar.tamaño() == 0) {
            limpieza();
        } else {
            Retiro r = ar.obtener(tblRetiro.getSelectedRow());
            Matricula m =
am.buscarMatricula(r.getNumMatricula());
            Curso c = ac.buscarCurso(m.getCodCurso());
            Alumno a = aa.buscarAlumno(m.getCodAlumno());

            txtNumeroRetiro.setText("" + r.getNumRetiro());
            txtNumeroMatricula.setText("" +
r.getNumMatricula());
            txtCodigoCurso.setText("" + m.getCodCurso());
            txtAlumno.setText(a.getNombres() + " " +
a.getApellidos());
            txtCurso.setText(c.getAsignatura());
        }
    }
    void limpieza() {
        txtNumeroRetiro.setText("" + ar.numeroCorrelativo());
        txtNumeroMatricula.setText("");
        txtCodigoCurso.setText("");
        txtAlumno.setText("");
        txtCurso.setText("");
    }

    // Métodos con valor de retorno
    int leerNumeroRetiro() {
        return
Integer.parseInt(txtNumeroRetiro.getText().trim());
    }
    int leerNumeroMatricula() {
        return
Integer.parseInt(txtNumeroMatricula.getText().trim());
    }
    int leerCodigoCurso(){
        return Integer.parseInt(txtCodigoCurso.getText().trim());
    }
    String nombreAlumno() {
        Matricula m = am.buscarMatricula(leerNumeroMatricula());
        Alumno a = aa.buscarAlumno(m.getCodAlumno());
        return "" + a.getNombres() + " " + a.getApellidos();
    }
    int confirmar(String s) {
        return JOptionPane.showConfirmDialog(this, s, "Alerta",
0, 1, null);
    }

```

```

    }

    // Acciones del mouse
    public void mouseClicked(MouseEvent arg0) {
        if (arg0.getSource() == tblRetiro) {
            btnAdicionar.setForeground(Color.BLACK);
            btnModificar.setForeground(Color.BLACK);
            habilitarEntradas(false);
            obtenerDatos();
        }
    }
    public void mouseEntered(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }
    public void mouseExited(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }
    public void mousePressed(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }
    public void mouseReleased(MouseEvent arg0) {
        // TODO Auto-generated method stub
    }
}

```

// Consulta Alumnos



Consulta | Alumnos

Código de Alumno: 202010002 Consultar

Alumno	: 202010002
Nombres	: Bryan Alexander
Apellidos	: Quintana Bonilla
DNI	: 70308527
Edad	: 20
Celular	: 940123075
Estado	: Matriculado
Asignatura	: Algoritmos y Estructura de Datos
Ciclo	: Segundo
Cantidad de créditos	: 4
Cantidad de horas	: 50

```
package interfases;
```

```
import java.awt.EventQueue;
import javax.swing.JDialog;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import arreglos.ArregloAlumno;
import arreglos.ArregloCurso;
import arreglos.ArregloMatricula;
import clases.Alumno;
import clases.Curso;
import clases.Matricula;
import libreria.Libreria;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
import javax.swing.JSeparator;
import java.awt.Font;

public class _06ConsultaAlumnos extends JDialog implements
ActionListener {

    private static final long serialVersionUID = 1L;
    private JTextField txtCodigoAlumno;
    private JScrollPane scp;
    private JTextArea txtS;
    private JLabel lblCodigoAlumno;
    private JButton btnConsultar;
```

```

private JSeparator separator;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _06ConsultaAlumnos dialog = new
_06ConsultaAlumnos();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _06ConsultaAlumnos() {
    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Consulta | Alumnos");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    txtCodigoAlumno = new JTextField();
    txtCodigoAlumno.setFont(new Font("Consolas", Font.PLAIN,
10));

    txtCodigoAlumno.setBounds(129, 9, 125, 20);
    getContentPane().add(txtCodigoAlumno);
    txtCodigoAlumno.setColumns(10);

    scp = new JScrollPane();
    scp.setBounds(10, 48, 414, 202);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setEditable(false);
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    scp.setViewportViewView(txtS);

    lblCodigoAlumno = new JLabel("C\u00F3digo de Alumno");
    lblCodigoAlumno.setFont(new Font("Consolas", Font.PLAIN,
10));

    lblCodigoAlumno.setBounds(10, 10, 109, 16);

```

```

        getContentPane().add(lblCodigoAlumno);

        btnConsultar = new JButton("Consultar");
        btnConsultar.setFont(new Font("Consolas", Font.PLAIN,
10));
        btnConsultar.addActionListener(this);
        btnConsultar.setBounds(321, 8, 103, 21);
        getContentPane().add(btnConsultar);

        separator = new JSeparator();
        separator.setBounds(10, 36, 416, 2);
        getContentPane().add(separator);

        soloNumeros(txtCodigoAlumno);
    }

    /**
     * Programación de botones
     */
    ArregloAlumno aa = new ArregloAlumno();
    ArregloCurso ac = new ArregloCurso();
    ArregloMatricula am = new ArregloMatricula();

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnConsultar) {
            actionPerformedBtnConsultar(arg0);
        }
    }

    // Botón consultar
    protected void actionPerformedBtnConsultar(ActionEvent arg0) {
        try {
            int codigoAlumno = leerCodigoAlumno();
            Alumno a = aa.buscarAlumno(codigoAlumno);

            if (a != null) {
                txtS.setText("");
                listar();
            } else {
                mensaje("El código ingresado no existe.");
            }
        } catch (Exception e) {
            // TODO: handle exception
        }
    }

    // Métodos sin valor de retorno
    void listar(){
        Alumno a = aa.buscarAlumno(leerCodigoAlumno());
    }

```

```

        imprimir("Alumno           : " + a.getCodAlumno());
        imprimir("Nombres         : " + a.getNombres());
        imprimir("Apellidos        : " + a.getApellidos());
        imprimir("DNI              : " + a.getDNI());
        imprimir("Edad             : " + a.getEdad());
        imprimir("Celular          : " + a.getCelular());
        imprimir("Estado           : " +
Libreria.estadoAlumno[a.getEstado()]);

        Matricula m = am.buscarAlumno(LeerCodigoAlumno());
        Curso c = ac.buscarCurso(m.getCodCurso());

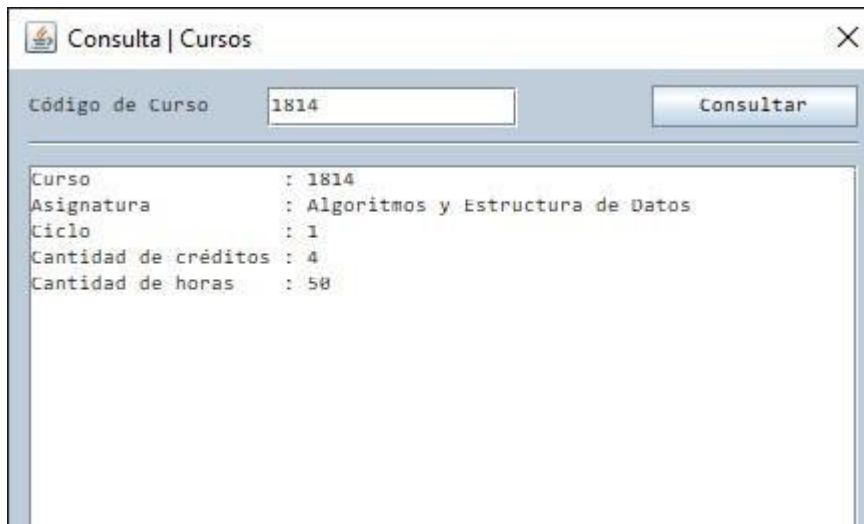
        imprimir("Asignatura         : " + c.getAsignatura());
        imprimir("Ciclo             : " +
Libreria.ciclocursos[c.getCiclo()]);
        imprimir("Cantidad de créditos : " + c.getCreditos());
        imprimir("Cantidad de horas   : " + c.getHoras());
    }

    // Métodos con valor de retorno
    void imprimir(String s) {
        txtS.append(s + "\n");
    }
    void mensaje(String s) {
        JOptionPane.showMessageDialog(this, s, "Alerta", 1);
    }
    int leerCodigoAlumno(){
        return
Integer.parseInt(txtCodigoAlumno.getText().trim());
    }

    // Validación de caracteres
    public void soloNumeros(JTextField a) {
        a.addKeyListener(new KeyAdapter() {
            public void keyTyped(KeyEvent e) {
                char c = e.getKeyChar();
                if(!Character.isDigit(c)) {
                    e.consume();
                }
            }
        });
    }
}

```

// Consulta Cursos



Consulta | Cursos

Código de Curso: 1814 Consultar

Curso : 1814  
Asignatura : Algoritmos y Estructura de Datos  
Ciclo : 1  
Cantidad de créditos : 4  
Cantidad de horas : 50

```
package interfases;
```

```
import java.awt.EventQueue;  
import javax.swing.JDialog;  
import javax.swing.JButton;  
import javax.swing.JScrollPane;  
import javax.swing.JTextArea;  
import javax.swing.JTextField;  
import arreglos.ArregloCurso;  
import clases.Curso;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import java.awt.event.ActionListener;  
import java.awt.event.KeyAdapter;  
import java.awt.event.KeyEvent;  
import java.awt.event.ActionEvent;  
import java.awt.Font;  
import java.awt.SystemColor;  
import javax.swing.JSeparator;  
import javax.swing.UIManager;
```

```
public class _07ConsultaCursos extends JDialog implements  
ActionListener {
```

```
    private static final long serialVersionUID = 1L;  
    private JTextField txtCodigoCurso;  
    private JScrollPane scp;  
    private JTextArea txtS;  
    private JButton btnConsultar;  
    private JLabel lblCodigoCurso;  
    private JSeparator separator;
```

```
    /**  
     * Launch the application.
```

```

    */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    _07ConsultaCursos dialog = new
_07ConsultaCursos();

                    dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                    dialog.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the dialog.
     */
    public _07ConsultaCursos() {

        getContentPane().setBackground(SystemColor.inactiveCaption);

        getContentPane().setForeground(SystemColor.inactiveCaption);
        setTitle("Consulta | Cursos");
        setBounds(100, 100, 450, 300);
        getContentPane().setLayout(null);

        txtCodigoCurso = new JTextField();
        txtCodigoCurso.setFont(new Font("Consolas", Font.PLAIN,
10));

        txtCodigoCurso.setBounds(129, 9, 125, 20);
        getContentPane().add(txtCodigoCurso);
        txtCodigoCurso.setColumns(10);

        scp = new JScrollPane();
        scp.setBounds(10, 48, 414, 205);
        getContentPane().add(scp);

        txtS = new JTextArea();
        txtS.setEditable(false);
        txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
        scp.setViewportView(txtS);

        btnConsultar = new JButton("Consultar");
        btnConsultar.setFont(new Font("Consolas", Font.PLAIN,
10));

        btnConsultar.setBackground(UIManager.getColor("Button.backgroun
d"));
    }

```



```

        btnConsultar.addActionListener(this);
        btnConsultar.setForeground(SystemColor.desktop);
        btnConsultar.setBounds(321, 8, 103, 21);
        getContentPane().add(btnConsultar);

        lblCodigoCurso = new JLabel("C\u00F3digo de Curso");
        lblCodigoCurso.setFont(new Font("Consolas", Font.PLAIN,
10));

        lblCodigoCurso.setBounds(10, 10, 109, 16);
        getContentPane().add(lblCodigoCurso);

        separator = new JSeparator();
        separator.setBounds(10, 36, 414, 2);
        getContentPane().add(separator);

        soloNumeros(txtCodigoCurso);
    }

    /**
     * Programación de botones
     */
    ArregloCurso ac = new ArregloCurso();

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnConsultar) {
            actionPerformedBtnConsultar(arg0);
        }
    }

    // Botón consultar
    protected void actionPerformedBtnConsultar(ActionEvent arg0) {

        try {
            int codigoCurso = leerCodigoCurso();
            Curso c = ac.buscarCurso(codigoCurso);

            if (c != null) {
                txtS.setText("");
                listar();
            } else {
                mensaje("El código ingresado no existe.");
            }
        } catch (Exception e){
            // TODO: handle exception
        }
    }

    // Métodos sin valor de retorno
    void listar() {
        Curso c = ac.buscarCurso(leerCodigoCurso());
    }

```

```

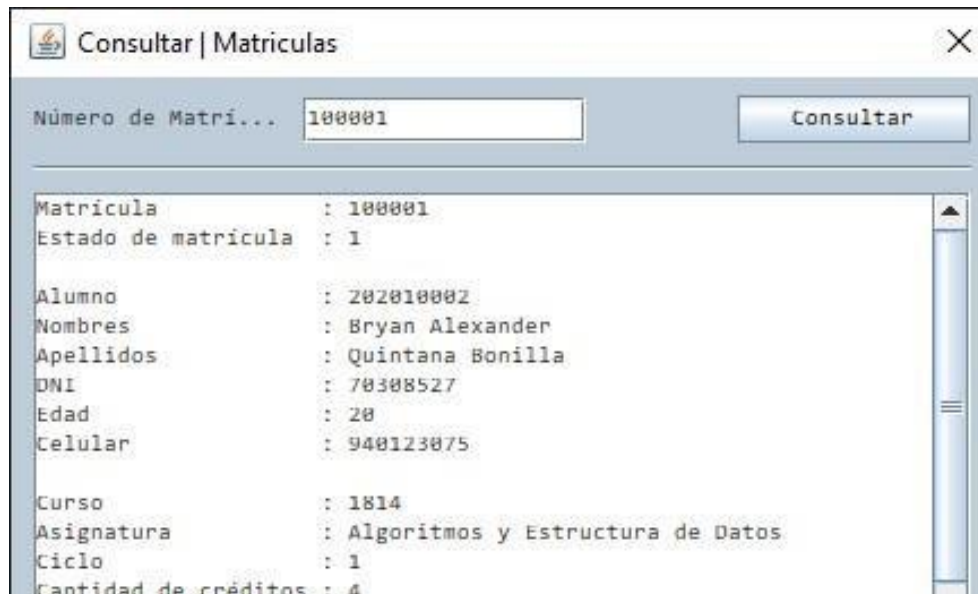
        imprimir("Curso                : " + c.getCodCurso());
        imprimir("Asignatura           : " + c.getAsignatura());
        imprimir("Ciclo                 : " + c.getCiclo());
        imprimir("Cantidad de créditos : " + c.getCreditos());
        imprimir("Cantidad de horas    : " + c.getHoras());
    }
    void imprimir(String s) {
        txtS.append(s + "\n");
    }
    void mensaje(String s) {
        JOptionPane.showMessageDialog(this, s, "Alerta", 1);
    }

    // Métodos con valor de retorno
    int leerCodigoCurso() {
        return Integer.parseInt(txtCodigoCurso.getText().trim());
    }

    // Validación de caracteres
    public void soloNumeros(JTextField a) {
        a.addKeyListener(new KeyAdapter() {
            public void keyTyped(KeyEvent e) {
                char c = e.getKeyChar();
                if(!Character.isDigit(c)) {
                    e.consume();
                }
            }
        });
    }
}

```

## // Consulta Matrículas



**package** interfases;

```
import java.awt.EventQueue;  
import javax.swing.JDialog;  
import javax.swing.JButton;  
import javax.swing.JScrollPane;  
import javax.swing.JTextArea;  
import javax.swing.JTextField;  
import arreglos.ArregloAlumno;  
import arreglos.ArregloCurso;  
import arreglos.ArregloMatricula;  
import clases.Alumno;  
import clases.Curso;  
import clases.Matricula;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import java.awt.Color;  
import java.awt.event.ActionListener;  
import java.awt.event.KeyAdapter;  
import java.awt.event.KeyEvent;  
import java.awt.event.ActionEvent;  
import javax.swing.JSeparator;  
import java.awt.Font;  
import java.awt.SystemColor;  
import javax.swing.UIManager;
```

```
public class _08ConsultaMatriculas extends JDialog implements  
ActionListener {
```

```
    private static final long serialVersionUID = 1L;  
    private JTextField txtMatricula;  
    private JScrollPane scp;  
    private JTextArea txtS;
```

```

private JButton btnConsultar;
private JLabel lblCodigoDeAlumno;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _08ConsultaMatriculas dialog = new
_08ConsultaMatriculas();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _08ConsultaMatriculas() {
    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Consultar | Matriculas");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    txtMatricula = new JTextField();
    txtMatricula.setFont(new Font("Consolas", Font.PLAIN,
10));

    txtMatricula.setBounds(129, 9, 125, 20);
    getContentPane().add(txtMatricula);
    txtMatricula.setColumns(10);

    scp = new JScrollPane();
    scp.setBounds(10, 51, 414, 199);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setEditable(false);
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    scp.setViewportView(txtS);

    btnConsultar = new JButton("Consultar");
    btnConsultar.setFont(new Font("Consolas", Font.PLAIN,
10));

```

```

        btnConsultar.setBackground(UIManager.getColor("Button.background
d"));
        btnConsultar.addActionListener(this);
        btnConsultar.setForeground(Color.BLACK);
        btnConsultar.setBounds(321, 8, 103, 21);
        getContentPane().add(btnConsultar);

        lblCodigoDeAlumno = new JLabel("N\u00F3FAmerno de
Matr\u000EDDcula");
        lblCodigoDeAlumno.setFont(new Font("Consolas",
Font.PLAIN, 10));
        lblCodigoDeAlumno.setBounds(10, 10, 109, 16);
        getContentPane().add(lblCodigoDeAlumno);

        soloNumeros(txtMatricula);

        separator = new JSeparator();
        separator.setBounds(10, 39, 414, 2);
        getContentPane().add(separator);
    }

    /**
     * Programación de botones
     */
    ArregloMatricula am = new ArregloMatricula();
    ArregloAlumno aa = new ArregloAlumno();
    ArregloCurso ac = new ArregloCurso();
    private JSeparator separator;

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnConsultar) {
            actionPerformedBtnConsultar(arg0);
        }
    }

    // Botón consultar
    protected void actionPerformedBtnConsultar(ActionEvent arg0) {
        try {
            int numeroMatricula = leerNumeroMatricula();
            Matricula m = am.buscarMatricula(numeroMatricula);

            if (m != null) {
                txtS.setText("");
                listar();
            } else {
                mensaje("El código ingresado no existe.");
            }
        } catch (Exception e) {
            // TODO: handle exception

```

```

    }
}

// Métodos sin valor de retorno
void listar(){
    Matricula m = am.buscarMatricula(leerNumeroMatricula());
    Alumno a = aa.buscarAlumno(m.getCodAlumno());
    Curso c = ac.buscarCurso(m.getCodCurso());

    imprimir("Matrícula           : " +
m.getNumMatricula());
    imprimir("Estado de matrícula : " + m.getEstado());
    imprimir();
    imprimir("Alumno           : " + a.getCodAlumno());
    imprimir("Nombres          : " + a.getNombres());
    imprimir("Apellidos         : " + a.getApellidos());
    imprimir("DNI              : " + a.getDNI());
    imprimir("Edad             : " + a.getEdad());
    imprimir("Celular           : " + a.getCelular());
    imprimir();
    imprimir("Curso             : " + c.getCodCurso());
    imprimir("Asignatura          : " + c.getAsignatura());
    imprimir("Ciclo             : " + c.getCiclo());
    imprimir("Cantidad de créditos : " + c.getCreditos());
    imprimir("Cantidad de horas    : " + c.getHoras());
}
void imprimir(){
    imprimir("");
}
void imprimir(String s) {
    txtS.append(s + "\n");
}
void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s, "Alerta", 1);
}

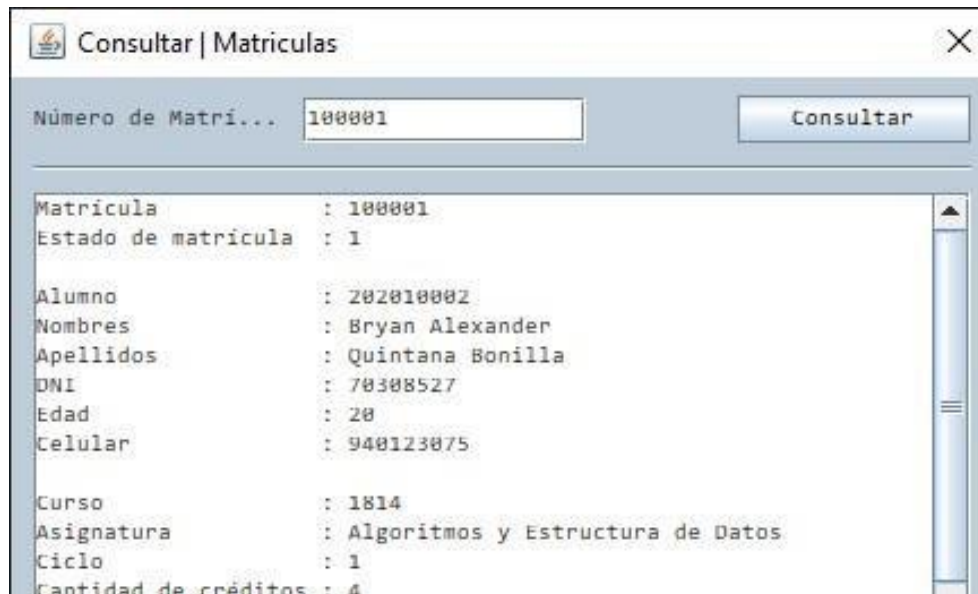
// Métodos con valor de retorno
int leerNumeroMatricula() {
    return Integer.parseInt(txtMatricula.getText().trim());
}

// Validación de caracteres
public void soloNumeros(JTextField a) {
    a.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent e) {
            char c = e.getKeyChar();
            if(!Character.isDigit(c)) {
                e.consume();
            }
        }
    });
}

```

} }

## // Consulta Matrículas



**package** interfases;

```
import java.awt.EventQueue;
import javax.swing.JDialog;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import arreglos.ArregloAlumno;
import arreglos.ArregloCurso;
import arreglos.ArregloMatricula;
import clases.Alumno;
import clases.Curso;
import clases.Matricula;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionEvent;
import javax.swing.JSeparator;
import java.awt.Font;
import java.awt.SystemColor;
import javax.swing.UIManager;
```

```
public class _08ConsultaMatriculas extends JDialog implements
ActionListener {
```

```
    private static final long serialVersionUID = 1L;
    private JTextField txtMatricula;
    private JScrollPane scp;
    private JTextArea txtS;
```



```

private JButton btnConsultar;
private JLabel lblCodigoDeAlumno;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _08ConsultaMatriculas dialog = new
_08ConsultaMatriculas();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _08ConsultaMatriculas() {
    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Consultar | Matriculas");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    txtMatricula = new JTextField();
    txtMatricula.setFont(new Font("Consolas", Font.PLAIN,
10));

    txtMatricula.setBounds(129, 9, 125, 20);
    getContentPane().add(txtMatricula);
    txtMatricula.setColumns(10);

    scp = new JScrollPane();
    scp.setBounds(10, 51, 414, 199);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setEditable(false);
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    scp.setViewportView(txtS);

    btnConsultar = new JButton("Consultar");
    btnConsultar.setFont(new Font("Consolas", Font.PLAIN,
10));

```

```

        btnConsultar.setBackground(UIManager.getColor("Button.background
d"));
        btnConsultar.addActionListener(this);
        btnConsultar.setForeground(Color.BLACK);
        btnConsultar.setBounds(321, 8, 103, 21);
        getContentPane().add(btnConsultar);

        lblCodigoDeAlumno = new JLabel("N\u00F3FAmerno de
Matr\u00E9Dcula");
        lblCodigoDeAlumno.setFont(new Font("Consolas",
Font.PLAIN, 10));
        lblCodigoDeAlumno.setBounds(10, 10, 109, 16);
        getContentPane().add(lblCodigoDeAlumno);

        soloNumeros(txtMatricula);

        separator = new JSeparator();
        separator.setBounds(10, 39, 414, 2);
        getContentPane().add(separator);
    }

    /**
     * Programación de botones
     */
    ArregloMatricula am = new ArregloMatricula();
    ArregloAlumno aa = new ArregloAlumno();
    ArregloCurso ac = new ArregloCurso();
    private JSeparator separator;

    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnConsultar) {
            actionPerformedBtnConsultar(arg0);
        }
    }

    // Botón consultar
    protected void actionPerformedBtnConsultar(ActionEvent arg0) {
        try {
            int numeroMatricula = leerNumeroMatricula();
            Matricula m = am.buscarMatricula(numeroMatricula);

            if (m != null) {
                txtS.setText("");
                listar();
            } else {
                mensaje("El código ingresado no existe.");
            }
        } catch (Exception e) {
            // TODO: handle exception

```

```

    }
}

// Métodos sin valor de retorno
void listar(){
    Matricula m = am.buscarMatricula(leerNumeroMatricula());
    Alumno a = aa.buscarAlumno(m.getCodAlumno());
    Curso c = ac.buscarCurso(m.getCodCurso());

    imprimir("Matrícula          : " +
m.getNumMatricula());
    imprimir("Estado de matrícula : " + m.getEstado());
    imprimir();
    imprimir("Alumno          : " + a.getCodAlumno());
    imprimir("Nombres         : " + a.getNombres());
    imprimir("Apellidos        : " + a.getApellidos());
    imprimir("DNI            : " + a.getDNI());
    imprimir("Edad           : " + a.getEdad());
    imprimir("Celular         : " + a.getCelular());
    imprimir();
    imprimir("Curso           : " + c.getCodCurso());
    imprimir("Asignatura        : " + c.getAsignatura());
    imprimir("Ciclo           : " + c.getCiclo());
    imprimir("Cantidad de créditos : " + c.getCreditos());
    imprimir("Cantidad de horas   : " + c.getHoras());
}
void imprimir(){
    imprimir("");
}
void imprimir(String s) {
    txtS.append(s + "\n");
}
void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s, "Alerta", 1);
}

// Métodos con valor de retorno
int leerNumeroMatricula() {
    return Integer.parseInt(txtMatricula.getText().trim());
}

// Validación de caracteres
public void soloNumeros(JTextField a) {
    a.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent e) {
            char c = e.getKeyChar();
            if(!Character.isDigit(c)) {
                e.consume();
            }
        }
    });
}

```

}

}

## // Consulta Retiros



```
package interfases;
```

```
import java.awt.EventQueue;
import javax.swing.JDialog;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import arreglos.ArregloAlumno;
import arreglos.ArregloCurso;
import arreglos.ArregloMatricula;
import arreglos.ArregloRetiro;
import clases.Alumno;
import clases.Curso;
import clases.Matricula;
import clases.Retiro;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionEvent;
import javax.swing.JSeparator;
import java.awt.Font;
import java.awt.SystemColor;

public class _09ConsultaRetiros extends JDialog implements
ActionListener {

    private static final long serialVersionUID = 1L;
    private JTextField txtRetiro;
```

```

private JScrollPane scp;
private JTextArea txtS;
private JButton btnConsultar;
private JLabel lblCodigoDeAlumno;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                _09ConsultaRetiros dialog = new
_09ConsultaRetiros();

                dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
                dialog.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the dialog.
 */
public _09ConsultaRetiros() {

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Consulta | Retiros");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    txtRetiro = new JTextField();
    txtRetiro.setFont(new Font("Consolas", Font.PLAIN, 10));
    txtRetiro.setBounds(129, 9, 125, 20);
    getContentPane().add(txtRetiro);
    txtRetiro.setColumns(10);

    scp = new JScrollPane();
    scp.setBounds(10, 51, 414, 199);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setEditable(false);
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    scp.setViewportView(txtS);

    btnConsultar = new JButton("Consultar");

```

```

10));

    btnConsultar.setFont(new Font("Consolas", Font.PLAIN,

    btnConsultar.addActionListener(this);
    btnConsultar.setForeground(Color.BLACK);
    btnConsultar.setBounds(321, 8, 103, 21);
    getContentPane().add(btnConsultar);

    lblCodigoDeAlumno = new JLabel("Numero de Retiro");
    lblCodigoDeAlumno.setFont(new Font("Consolas",
Font.PLAIN, 10));
    lblCodigoDeAlumno.setBounds(10, 10, 109, 16);
    getContentPane().add(lblCodigoDeAlumno);

    soloNumeros(txtRetiro);

    separator = new JSeparator();
    separator.setBounds(10, 39, 414, 2);
    getContentPane().add(separator);
}

/**
 * Programación de botones
 */
ArregloRetiro ar = new ArregloRetiro();
ArregloMatricula am = new ArregloMatricula();
ArregloAlumno aa = new ArregloAlumno();
ArregloCurso ac = new ArregloCurso();
private JSeparator separator;

public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnConsultar) {
        actionPerformedBtnConsultar(arg0);
    }
}

// Botón consultar
protected void actionPerformedBtnConsultar(ActionEvent arg0) {
    try {
        int numeroRetiro = leerNumeroRetiro();
        Retiro r = ar.buscarRetiro(numeroRetiro);

        if(r != null) {
            txtS.setText("");
            listar();
        } else {
            mensaje("El código ingresado no existe.");
        }
    } catch (Exception e){
        // TODO: handle exception
    }
}

```

```

// Métodos sin valor de retorno
void listar() {
    Retiro r = ar.buscarRetiro(LeerNumeroRetiro());
    Matricula m = am.buscarMatricula(r.getNumMatricula());
    Alumno a = aa.buscarAlumno(m.getCodAlumno());
    Curso c = ac.buscarCurso(m.getCodCurso());

    imprimir("Número de retiro      : " + r.getNumRetiro());
    imprimir("Número de matrícula   : " +
r.getNumMatricula());
    imprimir();
    imprimir("Alumno                  : " + a.getCodAlumno());
    imprimir("Nombres                 : " + a.getNombres());
    imprimir("Apellidos              : " + a.getApellidos());
    imprimir("DNI                   : " + a.getDNI());
    imprimir("Edad                  : " + a.getEdad());
    imprimir("Celular                : " + a.getCelular());
    imprimir();
    imprimir("Curso                   : " + c.getCodCurso());
    imprimir("Asignatura              : " + c.getAsignatura());
    imprimir("Ciclo                 : " + c.getCiclo());
    imprimir("Cantidad de créditos   : " + c.getCreditos());
    imprimir("Cantidad de horas      : " + c.getHoras());
}
void imprimir(){
    imprimir("");
}
void imprimir(String s) {
    txtS.append(s + "\n");
}
void mensaje(String s) {
    JOptionPane.showMessageDialog(this, s, "Alerta", 1);
}

// Métodos con valor de retorno
int leerNumeroRetiro() {
    return Integer.parseInt(txtRetiro.getText().trim());
}

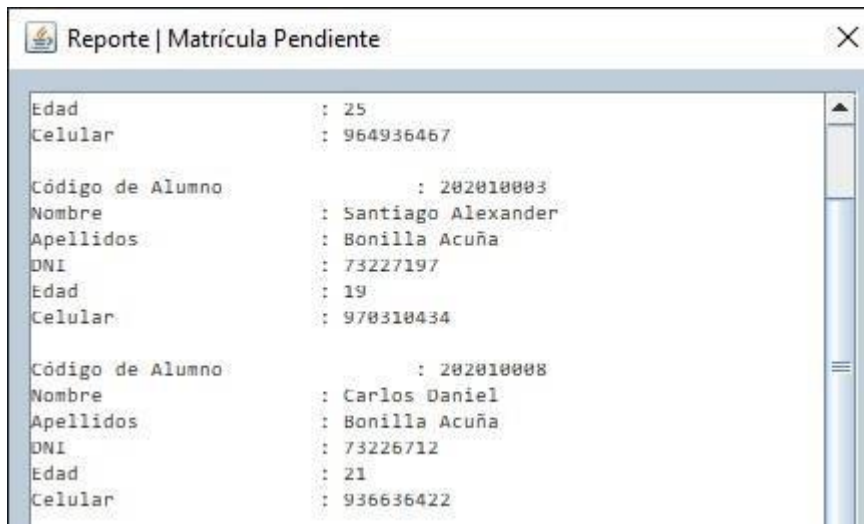
// Validación de caracteres
public void soloNumeros(JTextField a) {
    a.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent e) {
            char c = e.getKeyChar();
            if(!Character.isDigit(c)) {
                e.consume();
            }
        }
    });
}
}

```



}

## // Reporte de Alumnos por Matrícula Pendiente



```
package interfases;
```

```
import java.awt.EventQueue;  
import javax.swing.JDialog;  
import arreglos.ArregloAlumno;  
import clases.Alumno;  
import javax.swing.JScrollPane;  
import javax.swing.JTextArea;  
import java.awt.SystemColor;  
import java.awt.Font;
```

```
public class _10ReporteAlumnosMatriculaPendiente extends JDialog {  
  
    private static final long serialVersionUID = 1L;  
    private JScrollPane scp;  
    private JTextArea txtS;  
  
    /**  
     * Launch the application.  
     */  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    _10ReporteAlumnosMatriculaPendiente  
dialog = new _10ReporteAlumnosMatriculaPendiente();  
  
                    dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);  
                    dialog.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }  
}
```

```

}

/**
 * Create the dialog.
 */
public _10ReporteAlumnosMatriculaPendiente() {

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Reporte | Matr\u00EDcula Pendiente");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    scp = new JScrollPane();
    scp.setBounds(10, 11, 414, 239);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    txtS.setEditable(false);
    scp.setViewportView(txtS);
    listar();
}

// Métodos sin valor de retorno
void listar() {
    ArregloAlumno aa = new ArregloAlumno();
    Alumno a;

    txtS.setText("");

    for (int i = 0; i < aa.tamaño(); i++) {
        a = aa.obtener(i);

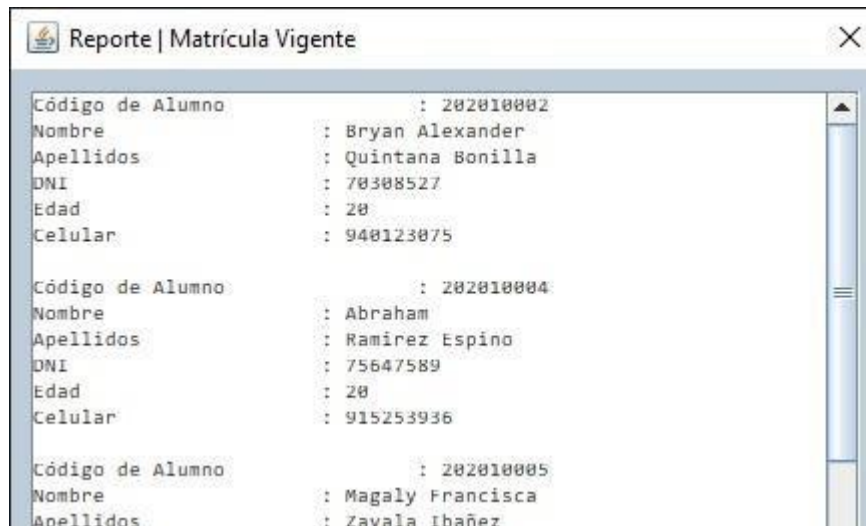
        if (a.getEstado() == 0) {
            a = aa.buscarAlumno(a.getCodAlumno());
            imprimir("Código de Alumno      : " +
a.getCodAlumno());
            imprimir("Nombre              : " +
a.getNombres());
            imprimir("Apellidos           : " +
a.getApellidos());
            imprimir("DNI                : " + a.getDNI());
            imprimir("Edad              : " + a.getEdad());
            imprimir("Celular          : " +
a.getCelular());
            imprimir();
        }
    }
}

void imprimir() {
    imprimir("");
}

```

```
    }  
    void imprimir(String s) {  
        txtS.append(s + "\n");  
    }  
}
```

## // Reporte de Alumnos con Matrícula Vigente



```
package interfases;
```

```
import java.awt.EventQueue;  
import javax.swing.JDialog;  
import javax.swing.JScrollPane;  
import arreglos.ArregloAlumno;  
import clases.Alumno;  
import javax.swing.JTextArea;  
import java.awt.Font;  
import java.awt.SystemColor;
```

```
public class _11ReporteAlumnosMatriculaVigente extends JDialog {  
  
    private static final long serialVersionUID = 1L;  
    private JTextArea txtS;  
    private JScrollPane scp;  
  
    /**  
     * Launch the application.  
     */  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    _11ReporteAlumnosMatriculaVigente dialog  
= new _11ReporteAlumnosMatriculaVigente();  
  
                    dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);  
                    dialog.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }  
}
```

```

}

/**
 * Create the dialog.
 */
public _11ReporteAlumnosMatriculaVigente() {

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Reporte | Matr\u00E9cula Vigente");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    scp = new JScrollPane();
    scp.setBounds(10, 11, 414, 239);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    txtS.setEditable(false);
    scp.setViewportView(txtS);
    listar();
}

// Métodos sin valor de retorno
void listar() {
    ArregloAlumno aa = new ArregloAlumno();
    Alumno a;

    txtS.setText("");

    for (int i = 0; i < aa.tamaño(); i++) {
        a = aa.obtener(i);

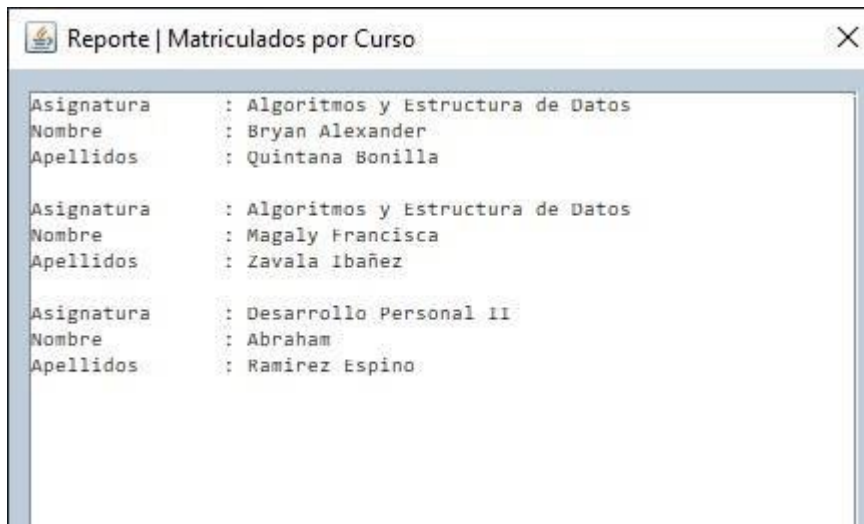
        if (a.getEstado() == 1) {
            a = aa.buscarAlumno(a.getCodAlumno());
            imprimir("Código de Alumno      : " +
a.getCodAlumno());
            imprimir("Nombre              : " +
a.getNombres());
            imprimir("Apellidos           : " +
a.getApellidos());
            imprimir("DNI                : " + a.getDNI());
            imprimir("Edad              : " + a.getEdad());
            imprimir("Celular          : "
+a.getCelular());
            imprimir();
        }
    }
}

void imprimir() {
    imprimir("");
}

```

```
    }  
    void imprimir(String s) {  
        txtS.append(s + "\n");  
    }  
}
```

## // Reporte de Alumnos Matriculados por Curso



```
package interfases;
```

```
import java.awt.EventQueue;  
import javax.swing.JDialog;  
import javax.swing.JScrollPane;  
import javax.swing.JTextArea;  
import arreglos.ArregloAlumno;  
import arreglos.ArregloCurso;  
import arreglos.ArregloMatricula;  
import clases.Alumno;  
import clases.Curso;  
import clases.Matricula;  
import java.awt.SystemColor;  
import java.awt.Font;
```

```
public class _12ReporteAlumnosMatriculadosCurso extends JDialog {  
  
    private static final long serialVersionUID = 1L;  
    private JScrollPane scp;  
    private JTextArea txtS;  
  
    /**  
     * Launch the application.  
     */  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    _12ReporteAlumnosMatriculadosCurso dialog  
= new _12ReporteAlumnosMatriculadosCurso();  
  
                    dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);  
                    dialog.setVisible(true);  
                } catch (Exception e) {
```



```

        e.printStackTrace();
    }
}

});
}

/**
 * Create the dialog.
 */
public _12ReporteAlumnosMatriculadosCurso() {

    getContentPane().setBackground(SystemColor.inactiveCaption);
    setTitle("Reporte | Matriculados por Curso");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(null);

    scp = new JScrollPane();
    scp.setBounds(10, 11, 414, 239);
    getContentPane().add(scp);

    txtS = new JTextArea();
    txtS.setFont(new Font("Consolas", Font.PLAIN, 10));
    txtS.setEditable(false);
    scp.setViewportView(txtS);
    listar();
}

// Métodos sin valor de retorno
void listar() {
    ArregloAlumno aa = new ArregloAlumno();
    ArregloCurso ac = new ArregloCurso();
    ArregloMatricula am = new ArregloMatricula();

    Alumno a;
    Curso c;
    Matricula m;
    txtS.setText("");

    for (int i = 0; i < am.tamaño(); i++) {
        m = am.obtener(i);

        if (m.getEstado() == 1) {
            c = ac.buscarCurso(m.getCodCurso());
            a = aa.buscarAlumno(m.getCodAlumno());
            imprimir("Asignatura : " + c.getAsignatura());

            imprimir("Nombre : " + a.getNombres());
            imprimir("Apellidos : " + a.getApellidos());
            imprimir();
        }
    }
}

```

```
    }  
    void imprimir() {  
        imprimir("");  
    }  
    void imprimir(String s) {  
        txtS.append(s + "\n");  
    }  
}
```

**5. UN LINK PARA DESCARGAR EL PROYECTO FINAL EMPAQUETADO Y UN BREVE AUDIO O VIDEO QUE CONTENGA LA EXPOSICIÓN DE CADA UNO DE LOS INTEGRANTES EN FUNCIÓN A LA PARTE QUE LE TOCÓ DESARROLLAR**

Link del repositorio del proyecto:

[arturobrunocm/ProyectoDeCicloAED: Proyecto de ciclo de Algoritmia y Estructura de Datos \(github.com\)](https://github.com/arturobrunocm/ProyectoDeCicloAED)

Exposición del proyecto:

[Exposición Final -Proyecto Algoritmia - YouTube](#)