

```
public class Iniciosesion extends javax.swing.JFrame {
    Admin admin ;
    Administrador administrador;

    public Iniciosesion() {

        this.admin = new Admin(usuario: "admin", password: "admin");
        this.administrador=new Administrador();
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void BtninicioActionPerformed(java.awt.event.ActionEvent evt) {
        String codigoQuemadoAdmin = "admin";
        String contraseñaQuemadaAdmin = "admin";

        if(this.admin.getUsuario().equals(anObject: codigoQuemadoAdmin) && this.admin.getPassword().equals(anObject: contraseñaQuemadaAdmin)){
            this.administrador.setVisible(b: true);
            this.setVisible(b: false);
        }else{
            JOptionPane.showMessageDialog(parentComponent: null, message: "Las credenciales son incorrectas!", title: "Alert", messageType: JOptionPane.INFORMATION_MESSAGE);
        }
    }
}
```

```
package proyectooool.vistas;
import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.pdf.PdfPTable;
import com.lowagie.text.pdf.PdfWriter;
import java.awt.BorderLayout;
import java.awt.Container;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import javax.swing.JFileChooser;
import proyectooool.clases.Profesor;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import proyectooool.clases.Admin;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PiePlot;
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.data.general.PieDataset;
import proyectooool.clases.Cursor;
```

```

public class Administrador extends javax.swing.JFrame {

    public Administrador() {
        initComponents();
        this.recuperarProfesores();
        this.mostrarProfesores();
        this.recuperarCursos();
        this.mostrarCursos();

        PieDataset dataset = createDatasetFromProfessors();
        JFreeChart chart = createPieChart(dataset);
        ChartPanel chartPanel = new ChartPanel(chart);
        JPanel panelGrafica = new JPanel();
        panelGrafica.add(comp: chartPanel);
        panelGrafica.setPreferredSize(new Dimension(jPanelp.getSize().width, jPanelp.getSize().height));
        Container panelProfesores = getContentPane();
        panelProfesores.add(comp: panelGrafica, constraints: BorderLayout.CENTER);
        jPanelp.add(comp: panelGrafica);

    }

```

```

public void recuperarProfesores() {
    try {
        FileInputStream fis = new FileInputStream(name: "C:\\Users\\herme\\OneDrive\\Documentos\\NetBeansProjects\\proyecto0001\\profesores.dat");
        ObjectInputStream ois = new ObjectInputStream(in: fis);
        ArrayList<Profesor> profesores = (ArrayList<Profesor>) ois.readObject();
        Admin.setprofesor(profesores);
        fis.close();
        ois.close();
        System.out.println("Profesores recuperados: " + profesores.size());
    } catch (Exception e) {
        System.out.println("Error al recuperar profesores: " + e.getMessage());
    }
}

private void mostrarProfesores() {
    String[] columnNames = {"CODIGO", "NOMBRE", "APELLIDO", "CORREO", "genero"};

    DefaultTableModel model = new DefaultTableModel(columnNames, rowCount: 0);

    for (Profesor profesor : Admin.getprofesores()) {
        Object[] rowData = {profesor.getCodigo(), profesor.getNombre(), profesor.getApellido(),
            profesor.getCorreo(), profesor.getGenero()};
        model.addRow(rowData);
    }
    tablaprofesores.setModel(dataModel: model);
}

```

```

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

    Crearprofesor crearp = new Crearprofesor();
    crearp.setResizable(resizable: false);
    crearp.setVisible(b: true);
    crearp.pack();
    crearp.setLocationRelativeTo(c: null);
    crearp.setDefaultCloseOperation(operation: JFrame.DISPOSE_ON_CLOSE);

    this.dispose();
}

```

```

int Codigop = (Integer.parseInt(s: Txtcodigoprof.getText()));
String Nombrep = Txtnombrepref.getText();
String Apellidop = Txtapellidoprof.getText();
String Correop = Txtcorreoprof.getText();
String generop = CBGeneroprof.getSelectedItem().toString();

try{
Profesor profe = new Profesor(codigo:Codigop, nombre:Nombrep, apellido: Apellidop, correo:Correop, genero:generop);
Admin.getprofesores().add(e: profe);
JOptionPane.showMessageDialog(parentComponent: null, message: "Ingresó de profesor exitosamente");

FileOutputStream fos = new FileOutputStream(name: "C:\\Users\\herme\\OneDrive\\Documentos\\NetBeansProjects\\proyectoool\\profesores.dat");
System.out.println(s: "que pasa brou ");
ObjectOutputStream oos = new ObjectOutputStream(out: fos);
oos.writeObject(obj: Admin.getprofesores());
fos.close();
oos.close();

} catch (Exception e) {
JOptionPane.showMessageDialog(parentComponent: null, message: e.getMessage());
}

Administrador admin = new Administrador();
admin.setVisible(b: true);
this.dispose();

```

```

package proyectoool.clases;

import java.io.Serializable;

public abstract class Persona implements Serializable{

private int codigo;
private String nombre;
private String apellido;
private String correo;
private String genero;
private String usuario;
private String password;

public Persona(int codigo, String nombre, String apellido, String correo, String genero, String usuario, String password) {
this.codigo = codigo;
this.nombre = nombre;
this.apellido = apellido;
this.correo = correo;
this.genero = genero;
this.usuario = usuario;
this.password = password;
}

public Persona(int codigo, String nombre, String apellido, String correo, String genero) {
this.codigo = codigo;
this.nombre = nombre;
this.apellido = apellido;
this.correo = correo;
this.genero = genero;
}

public Persona(String usuario, String password) {
this.usuario = usuario;
this.password = password;
}
}

```

```

package proyectoool.clases;

import javax.swing.ImageIcon;

public class Profesor extends Persona {

public Profesor(int codigo, String nombre, String apellido, String correo, String genero, String usuario, String password, ImageIcon foto) {
super(codigo, nombre, apellido, correo, genero, usuario, password);
}

public Profesor(int codigo, String nombre, String apellido, String correo, String genero) {
super(codigo, nombre, apellido, correo, genero);
}

}

```

```

private static ArrayList<Profesor> profesores = new ArrayList<>();
private static ArrayList<Alumno> alumnos = new ArrayList<>();
private static ArrayList<Curso> cursos = new ArrayList<>();

public Admin(String usuario, String password) {
    super(usuario, password);
}

public static ArrayList<Profesor> getprofesores() {
    return profesores;
}

public static void setprofesor(ArrayList<Profesor> profesores) {
    Admin.profesores = profesores;
}

```

```

private void BtbactualizarprofActionPerformed(java.awt.event.ActionEvent evt) {
    int filaSeleccionada = tablaprofesores.getSelectedRow();

    if (filaSeleccionada >= 0) {
        // Obtén el profesor seleccionado en la fila
        Profesor profesorAAActualizar = Admin.getprofesores().get(index: filaSeleccionada);

        // Abre una nueva ventana de actualización de datos y pásale el profesor
        Actualizarprofesor actualizarVentana = new Actualizarprofesor();
        actualizarVentana.setProfesorAAActualizar(profesor: profesorAAActualizar); // Pasa el profesor seleccionado
        actualizarVentana.setVisible(b: true);
        this.dispose();
    } else {
        JOptionPane.showMessageDialog(parentComponent: this, message: "Por favor, selecciona un profesor para actualizar.");
    }
}

```

```

package proyectooool.vistas;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import proyectooool.clases.Admin;
import proyectooool.clases.Cursor;
import proyectooool.clases.Profesor;

public class Actualizarcurso extends javax.swing.JFrame {

    private Cursor cursoActualizar;

    public Actualizarcurso() {
        initComponents();
        ArrayList<Profesor> profesores = Admin.getprofesores();

        // Crear un nuevo arraylist con los nombres y apellidos de los profesores
        ArrayList<String> nombresApellidos = new ArrayList<>();
        for (Profesor profesor : profesores) {
            nombresApellidos.add(profesor.getNombre() + " " + profesor.getApellido());
        }
    }

    public void setCursoActualizar(Curso curso) {
        this.cursoActualizar = curso;
        // Configura los campos de texto con los datos del curso
        Txtcodigocurso.setText(t: String.valueOf(i: curso.getCodigoCurso()));
        Txtnombrecurso.setText(t: curso.getNombreCurso());
        Txtcreditocurso.setText(t: String.valueOf(i: curso.getCreditosCurso()));
        // Carga los datos del profesor en el combobox
        ArrayList<Profesor> profesores = Admin.getprofesores();
        ArrayList<String> nombresApellidos = new ArrayList<>();
        for (Profesor profesor : profesores) {
            nombresApellidos.add(profesor.getNombre() + " " + profesor.getApellido());
        }
        jComboBox2.setModel(new DefaultComboBoxModel(items: nombresApellidos.toArray()));
        jComboBox2.setSelectedItem(anObject: curso.getProfesorCurso());
    }
}

```

```

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

    try {
        // Obtén los valores editados de los campos de texto
        int Codigoc = (Integer.parseInt(s: Txtcodigocurso.getText()));
        String Nombrec = Txtnombrecurso.getText();
        int Creditosc = (Integer.parseInt(s: Txtcreditocurso.getText()));
        String profesoresc = jComboBox2.getSelectedItem().toString();

        // Actualiza los datos del profesor
        cursoActualizar.setCodigoCurso(codigoCurso: Codigoc);
        cursoActualizar.setNombreCurso(nombreCurso: Nombrec);
        cursoActualizar.setCreditosCurso(creditosCurso: Creditosc);
        cursoActualizar.setProfesorCurso(profesorCurso: profesoresc);

        // Actualiza la lista de profesores en la clase Admin
        ArrayList<Curso> cursos = Admin.getCursos();
        // Encuentra y reemplaza el profesor actualizado en la lista
        for (int i = 0; i < cursos.size(); i++) {
            if (cursos.get(index: i).getCodigoCurso() == cursoActualizar.getCodigoCurso()) {
                cursos.set(index: i, element: cursoActualizar);
                break; // Sal del bucle una vez que se haya encontrado y actualizado el profesor
            }
        }
    }
}

```