TALLER DE DESARROLLO DE APLICACIONES 1

SEMANA: 5

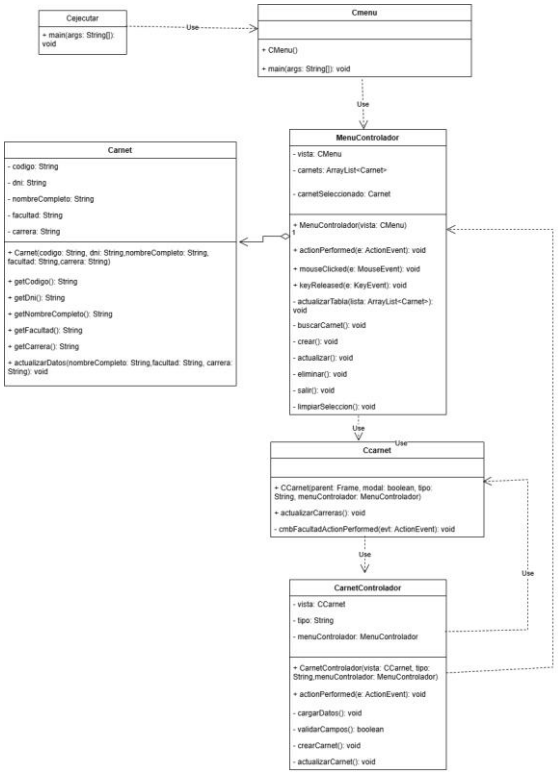
ENTREGADO POR: Mescua Segovia Marino

CÓDIGO: N04058C

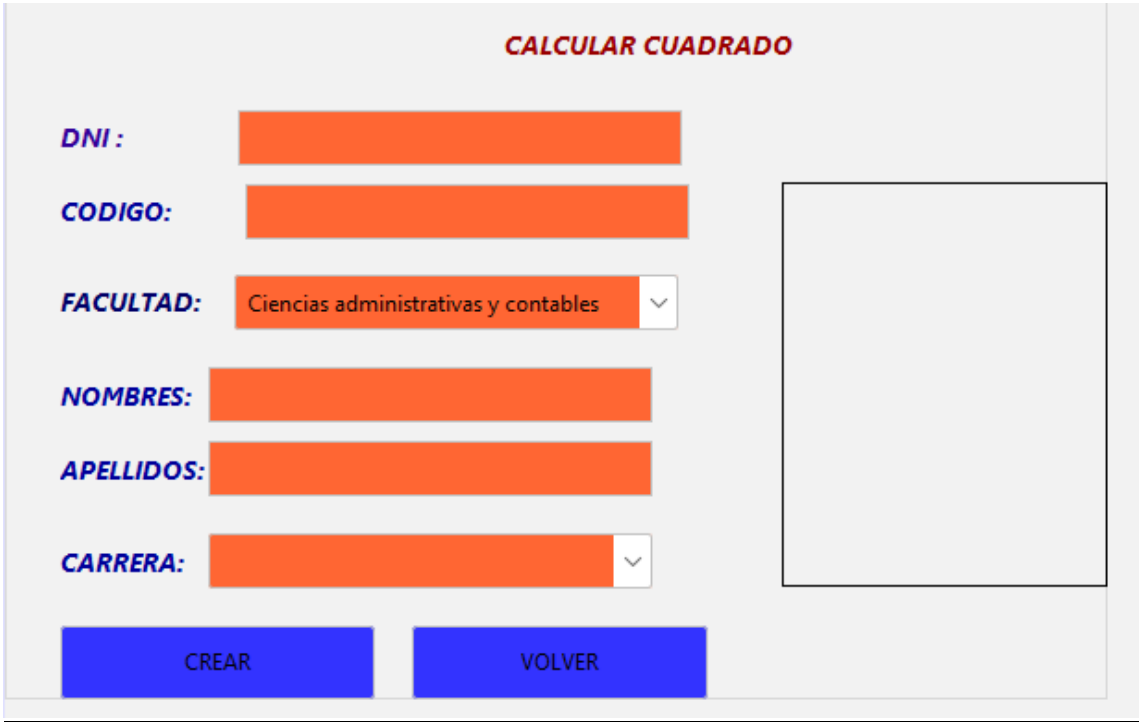
CATEDRÁTICO: Fernández Bejarano Raúl Enrique

ACTIVIDAD 1:

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:



CÓDIGO DE LA APLICACIÓN:

```java
public class CarnetControlador implements ActionListener {

    private final CCarnet vista;
    private final String tipo;
    private final MenuControlador menuControlador;

    public CarnetControlador(CCarnet vista, String tipo, MenuControlador menuControlador) {
        this.vista = vista;
        this.tipo = tipo;
        this.menuControlador = menuControlador;

        this.vista.btnCrear.addActionListener(this);
        this.vista.btnVolver.addActionListener(this);

        if (tipo.equalsIgnoreCase("actualizar")) {
            cargarDatos();
        }
    }

    private void cargarDatos() {
        Carnet carnet = menuControlador.getCarnetSeleccionado();
        if (carnet != null) {
            vista.txtCodigo.setText(carnet.getCodigo());
            vista.txtDni.setText(carnet.getDni());
            vista.txtNombre.setText(carnet.getNombreCompleto());
            vista.cmbFacultad.setSelectedItem(carnet.getFacultad());

            vista.actualizarCarreras(); // 🔹 Ahora llamas a la vista para actualizar carreras.

            vista.cmbCarrera.setSelectedItem(carnet.getCarrera());

            vista.txtCodigo.setEditable(false);
            vista.txtDni.setEditable(false);
        }
    }

    private boolean validarCampos() {
        if (vista.txtCodigo.getText().trim().isEmpty()
                || vista.txtDni.getText().trim().isEmpty()
                || vista.txtNombre.getText().trim().isEmpty()
                || vista.cmbFacultad.getSelectedIndex() == -1
                || vista.cmbCarrera.getSelectedIndex() == -1) {
            JOptionPane.showMessageDialog(vista, "Todos los campos deben estar completos.", "Error", JOptionPane.ERROR_MESSAGE);
            return false;
        }
        return true;
    }

```

```
    private void crearCarnet() {
        Carnet nuevo = new Carnet(
                vista.txtCodigo.getText(),
                vista.txtDni.getText(),
                vista.txtNombre.getText(),
                (String) vista.cmbFacultad.getSelectedItem(),
                (String) vista.cmbCarrera.getSelectedItem()
        );
        menuControlador.agregarCarnet(nuevo);
        JOptionPane.showMessageDialog(vista, "Carnet creado exitosamente.");
        vista.dispose();
    }

    private void actualizarCarnet() {
        Carnet carnet = menuControlador.getCarnetSeleccionado();
        if (carnet != null) {
            carnet.actualizarDatos(
                    vista.txtNombre.getText(),
                    (String) vista.cmbFacultad.getSelectedItem(),
                    (String) vista.cmbCarrera.getSelectedItem()
            );
            JOptionPane.showMessageDialog(vista, "Carnet actualizado exitosamente.");
            vista.dispose();
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();

        if (source == vista.btnVolver) {
            vista.dispose();
        } else if (source == vista.btnCrear) {
            if (validarCampos()) {
                if (tipo.equalsIgnoreCase("crear")) {
                    crearCarnet();
                } else if (tipo.equalsIgnoreCase("actualizar")) {
                    actualizarCarnet();
                }
            }
        }
    }
}
```

```java
public class MenuControlador implements ActionListener, MouseListener, KeyListener {

    private final CMenu vista;
    private static final ArrayList<Carnet> carnets = new ArrayList<>();
    private Carnet carnetSeleccionado = null;

    public MenuControlador(CMenu vista) {
        this.vista = vista;

        this.vista.btnCrear.addActionListener(this);
        this.vista.btnActualizar.addActionListener(this);
        this.vista.btnEliminar.addActionListener(this);
        this.vista.btnSalir.addActionListener(this);

        this.vista.tblDatos.addMouseListener(this);
        this.vista.txtBuscar.addKeyListener(this);

        this.vista.btnActualizar.setEnabled(false);
        this.vista.btnEliminar.setEnabled(false);

        actualizarTabla(carnets);
    }

    private void actualizarTabla(ArrayList<Carnet> lista) {
        DefaultTableModel modelo = new DefaultTableModel();
        modelo.addColumn("Código");
        modelo.addColumn("DNI");
        modelo.addColumn("Nombre Completo");
        modelo.addColumn("Facultad");
        modelo.addColumn("Carrera");

        for (Carnet c : lista) {
            modelo.addRow(new Object[]{
                c.getCodigo(),
                c.getDni(),
                c.getNombreCompleto(),
                c.getFacultad(),
                c.getCarrera()
            });
        }

        vista.tblDatos.setModel(modelo);
    }

```

```java
    private void buscarCarnet() {
        String texto = vista.txtBuscar.getText().toLowerCase();
        ArrayList<Carnet> resultados = new ArrayList<>();

        for (Carnet c : carnets) {
            if (c.getDni().toLowerCase().contains(texto)
                    || c.getCodigo().toLowerCase().contains(texto)
                    || c.getNombreCompleto().toLowerCase().contains(texto)) {
                resultados.add(c);
            }
        }

        actualizarTabla(resultados);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();

        if (source == vista.btnCrear) {
            crear();
        } else if (source == vista.btnActualizar) {
            actualizar();
        } else if (source == vista.btnEliminar) {
            eliminar();
        } else if (source == vista.btnSalir) {
            salir();
        }
    }

    private void crear() {
        CCarnet dialogo = new CCarnet(vista, true, "crear", this);
        dialogo.setVisible(true);
        actualizarTabla(carnets);
        limpiarSeleccion();
    }

    private void actualizar() {
        if (carnetSeleccionado != null) {
            CCarnet dialogo = new CCarnet(vista, true, "actualizar", this);
            dialogo.setVisible(true);
            actualizarTabla(carnets);
            limpiarSeleccion();
        } else {
            JOptionPane.showMessageDialog(vista, "Seleccione un carnet primero.");
        }
    }
```

```
    private void eliminar() {
        if (carnetSeleccionado != null) {
            int confirm = JOptionPane.showConfirmDialog(vista, "¿Seguro de eliminar este carnet?", "Confirmar", JOptionPane.YES_NO_OPTION);
            if (confirm == JOptionPane.YES_OPTION) {
                carnets.remove(carnetSeleccionado);
                actualizarTabla(carnets);
                carnetSeleccionado = null;
                limpiarSeleccion();
            }
        } else {
            JOptionPane.showMessageDialog(vista, "Seleccione un carnet primero.");
        }
    }

    private void salir() {
        int respuesta = JOptionPane.showOptionDialog(vista, "¿Estás seguro de salir?", "Salir", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MES
        if (respuesta == 0) {
            System.exit(0);
        }
        vista.txtBuscar.requestFocus();
    }

    private void limpiarSeleccion() {
        carnetSeleccionado = null;
        vista.btnActualizar.setEnabled(false);
        vista.btnEliminar.setEnabled(false);
        vista.tblDatos.clearSelection();
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        int fila = vista.tblDatos.getSelectedRow();
        if (fila >= 0) {
            String dniSeleccionado = (String) vista.tblDatos.getValueAt(fila, 1); // columna 1 = DNI
            for (Carnet c : carnets) {
                if (c.getDni().equals(dniSeleccionado)) {
                    carnetSeleccionado = c;
                    break;
                }
            }
            vista.btnActualizar.setEnabled(true);
            vista.btnEliminar.setEnabled(true);
        }
    }
        }
    }

    // Métodos vacíos que se deben implementar
    @Override
    public void mousePressed(MouseEvent e) {
    }

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) {
    }

    @Override
    public void mouseExited(MouseEvent e) {
    }

    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    // Métodos para acceder desde el CarnetControlador
    public void agregarCarnet(Carnet c) {
        carnets.add(c);
    }

    public Carnet getCarnetSeleccionado() {
        return carnetSeleccionado;
    }
}
```

```java

public class CEjecutar {

    public static void main(String[] args) {
        CMenu menu = new CMenu();
        menu.setVisible(true);
    }
}

```

```java
public class Carnet {

    private String codigo;
    private String dni;
    private String nombreCompleto;
    private String facultad;
    private String carrera;

    public Carnet(String codigo, String dni, String nombreCompleto, String facultad, String carrera) {
        this.codigo = codigo;
        this.dni = dni;
        this.nombreCompleto = nombreCompleto;
        this.facultad = facultad;
        this.carrera = carrera;
    }

    public String getCodigo() {
        return codigo;
    }

    public String getDni() {
        return dni;
    }

    public String getNombreCompleto() {
        return nombreCompleto;
    }

    public String getFacultad() {
        return facultad;
    }

    public String getCarrera() {
        return carrera;
    }

    public void actualizarDatos(String nombreCompleto, String facultad, String carrera) {
        this.nombreCompleto = nombreCompleto;
        this.facultad = facultad;
        this.carrera = carrera;
    }
}

```

```java
public class CCarnet extends javax.swing.JDialog {

    /**
     * Creates new form CCarnet
     */
    public CCarnet(java.awt.Frame parent, boolean modal, String tipo, MenuControlador menuControlador) {
        super(parent, modal);
        initComponents();
        new CarnetControlador(this, tipo, menuControlador);
    }

    public void actualizarCarreras() {
        String facultadSeleccionada = (String) cmbFacultad.getSelectedItem();
        DefaultComboBoxModel<String> modeloCarreras = new DefaultComboBoxModel<>();

        if (facultadSeleccionada == null) {
            return;
        }

        if (facultadSeleccionada.equalsIgnoreCase("Ciencias administrativas y contables")) {
            modeloCarreras.addElement("Administración");
            modeloCarreras.addElement("Administración e Inteligencia de Negocios");
            modeloCarreras.addElement("Administración y Negocios Globales");
            modeloCarreras.addElement("Administración y Gestión del Talento Humano");
            modeloCarreras.addElement("Contabilidad y Finanzas");
        } else if (facultadSeleccionada.equalsIgnoreCase("Derecho y ciencias políticas")) {
            modeloCarreras.addElement("Derecho");
            modeloCarreras.addElement("Educación Inicial");
            modeloCarreras.addElement("Educación Primaria");
        } else if (facultadSeleccionada.equalsIgnoreCase("Ingeniería")) {
            modeloCarreras.addElement("Arquitectura");
            modeloCarreras.addElement("Ingeniería Civil");
            modeloCarreras.addElement("Ingeniería Industrial");
            modeloCarreras.addElement("Ingeniería del Medio Ambiente y Desarrollo");
            modeloCarreras.addElement("Ingeniería de Sistemas y Computación");
        } else if (facultadSeleccionada.equalsIgnoreCase("Ciencias de la Salud")) {
            modeloCarreras.addElement("Enfermería");
            modeloCarreras.addElement("Farmacia y Bioquímica");
            modeloCarreras.addElement("Medicina Veterinaria y Zootecnia");
            modeloCarreras.addElement("Nutrición Humana");
            modeloCarreras.addElement("Obstetricia");
            modeloCarreras.addElement("Odontología");
            modeloCarreras.addElement("Psicología");
            modeloCarreras.addElement("Tecnología Médica");
        } else if (facultadSeleccionada.equalsIgnoreCase("Medicina Humana")) {
            modeloCarreras.addElement("Medicina Humana");
        }
```

```java
public class CMenu extends javax.swing.JFrame {

    /**
     * Creates new form CMenu
     */
    public CMenu() {
        initComponents();
        new MenuControlador(this);
    }

    /**
     * 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

    /**
     * @param args the command line arguments
```