UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

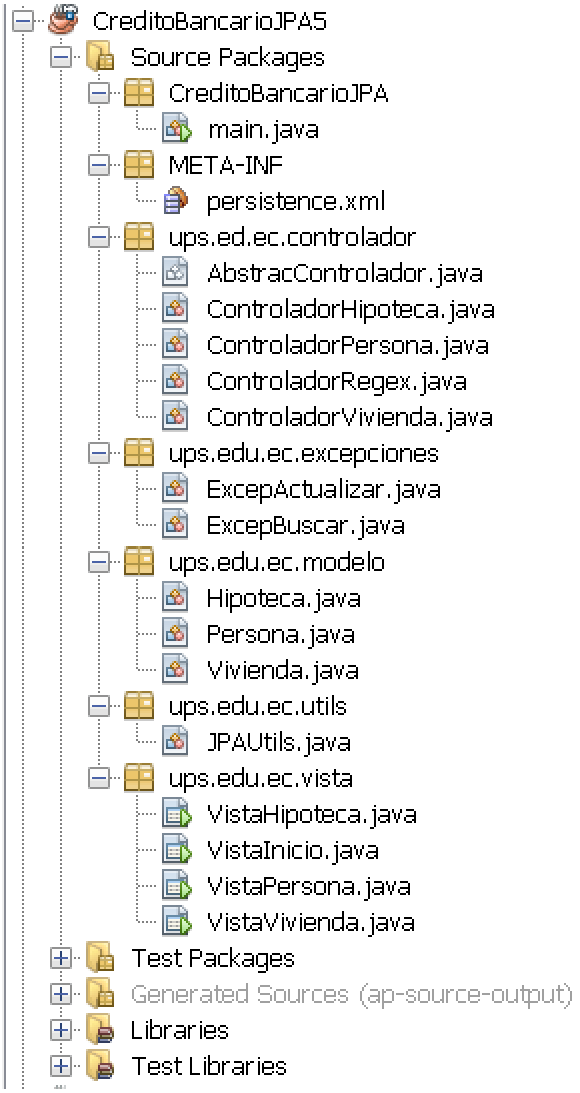
FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES

CARRERA: Computación		**ASIGNATURA**: Programación Aplicada
NRO. PRÁCTICA:		**TÍTULO PRÁCTICA**: CREDITO BANCARIO

OBJETIVO ALCANZADO: Poder realizar las practicas con un éxito al %100 tras haber recibido las calses y temas necesarios para poder realizarlas y a su vez obtener resultados que esten a la perspectiva del docente a cargo.

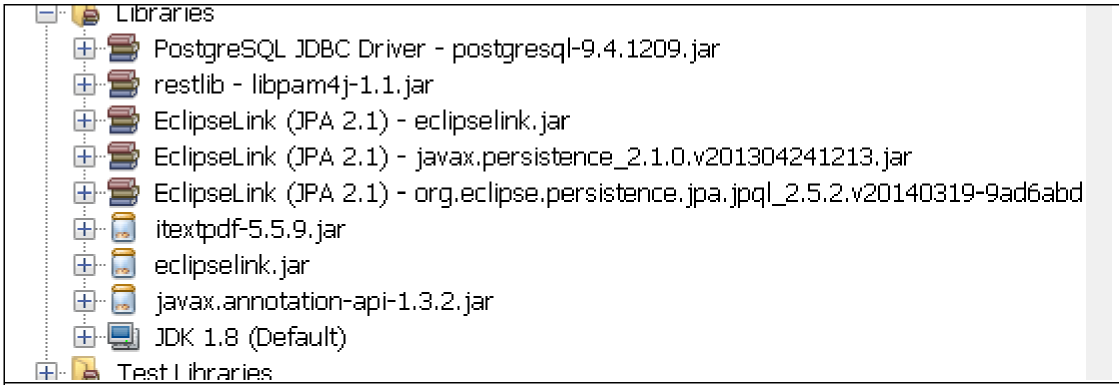
ACTIVIDADES DESARROLLADAS

1. MODELO MVC



```
CreditoBancarioJPA5
  Source Packages
    CreditoBancarioJPA
      main.java
    META-INF
      persistence.xml
    ups.ed.ec.controlador
      AbstracControlador.java
      ControladorHipoteca.java
      ControladorPersona.java
      ControladorRegex.java
      ControladorVivienda.java
    ups.edu.ec.excepciones
      ExcepActualizar.java
      ExcepBuscar.java
    ups.edu.ec.modelo
      Hipoteca.java
      Persona.java
      Vivienda.java
    ups.edu.ec.utils
      JPAUtils.java
    ups.edu.ec.vista
      VistaHipoteca.java
      VistaInicio.java
      VistaPersona.java
      VistaVivienda.java
  Test Packages
  Generated Sources (ap-source-output)
  Libraries
  Test Libraries
```

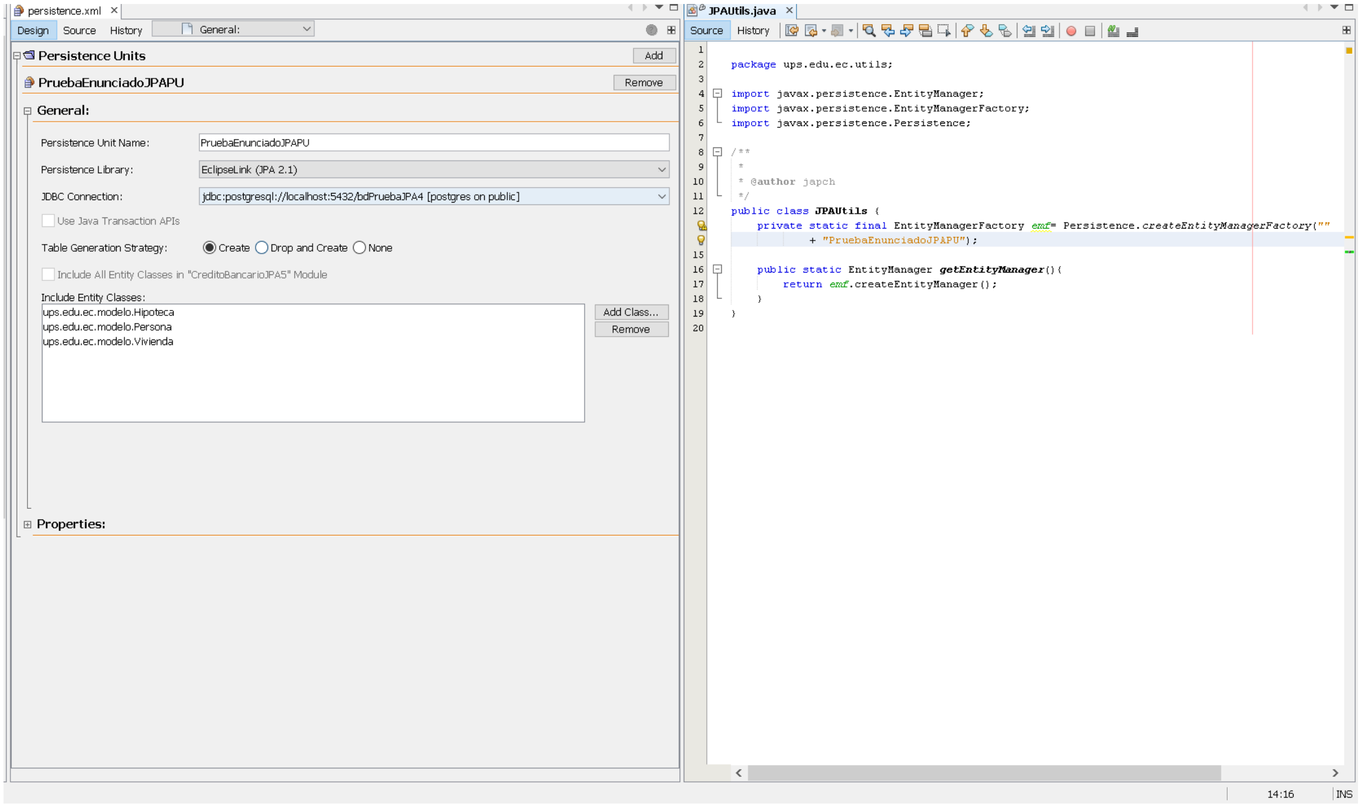
LIBRERIAS UTILIZADAS



Libraries
- PostgreSQL JDBC Driver - postgresql-9.4.1209.jar
- restlib - libpam4j-1.1.jar
- EclipseLink (JPA 2.1) - eclipselink.jar
- EclipseLink (JPA 2.1) - javax.persistence_2.1.0.v201304241213.jar
- EclipseLink (JPA 2.1) - org.eclipse.persistence.jpa.jpql_2.5.2.v20140319-9ad6abd
- itextpdf-5.5.9.jar
- eclipselink.jar
- javax.annotation-api-1.3.2.jar
- JDK 1.8 (Default)

Test Libraries

2.CONEXION BD



Persistence Units

PruebaEnunciadoJPAPU

General:

Persistence Unit Name: PruebaEnunciadoJPAPU

Persistence Library: EclipseLink (JPA 2.1)

JDBC Connection: jdbc:postgresql://localhost:5432/bdPruebaJPA4 [postgres on public]

□ Use Java Transaction APIs

Table Generation Strategy: ● Create ○ Drop and Create ○ None

□ Include All Entity Classes in "CreditoBancarioJPA5" Module

Include Entity Classes:
ups.edu.ec.modelo.Hipoteca
ups.edu.ec.modelo.Persona
ups.edu.ec.modelo.Vivienda

Properties:

```java

package ups.edu.ec.utils;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

/**
 *
 * @author japch
 */
public class JPAUtils {
    private static final EntityManagerFactory emf= Persistence.createEntityManagerFactory(""
            + "PruebaEnunciadoJPAPU");

    public static EntityManager getEntityManager(){
        return emf.createEntityManager();
    }
}

```

3.----------------------------------CONTROLADORES-------------------------

ABSTRACCONTROLADOR

package ups.ed.ec.controlador;

import ups.edu.ec.utils.JPAUtils;

import java.lang.reflect.ParameterizedType;

import java.lang.reflect.Type;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.EntityManager;

import javax.swing.JOptionPane;

```
public abstract class AbstracControlador<E> {


    private Class<E> clase;

    private EntityManager em;


    public AbstracControlador() {

        Type t = getClass().getGenericSuperclass();

        ParameterizedType pt = (ParameterizedType) t;

        clase = (Class) pt.getActualTypeArguments()[0];

        this.em = JPAUtils.getEntityManager();

    }


    public AbstracControlador(EntityManager em) {

        Type t = getClass().getGenericSuperclass();

        ParameterizedType pt = (ParameterizedType) t;

        clase = (Class) pt.getActualTypeArguments()[0];

        this.em = em;

    }


    public void crear(E objeto)  {


            em.getTransaction().begin();

            em.persist(objeto);

            em.getTransaction().commit();


    }


    public void actualizar(E objeto) throws Exception {

        if (validarActualizar(objeto) == true) {

            em.getTransaction().begin();
            em.merge(objeto);

            em.getTransaction().commit();

        }

    }


    public void eliminar(E objeto) throws Exception {

        if (validarActualizar(objeto) == true) {

            em.getTransaction().begin();

            em.remove(em.merge(objeto));

            em.getTransaction().commit();

        }

    }


    public E leer(Object id) throws Exception {

        if (validarExistente(id) == true) {

            return (E) em.find(clase, id);

        } else {

            return null;

        }

    }


    public List<E> buscartodo() throws Exception {

        return em.createQuery("Select t from " + clase.getSimpleName() + " t").getResultList();

    }


    public abstract boolean validarActualizar(E objeto) throws Exception;


    public abstract boolean validarExistente(Object id) throws Exception;


    public Class<E> getClase() {

        return clase;

    }
```

```
    public void setClase(Class<E> clase) {
        this.clase = clase;
    }

    public EntityManager getEm() {
        return em;
    }

    public void setEm(EntityManager em) {
        this.em = em;
    }
}
```

CONTROLADOR HIPOTECA

```
package ups.ed.ec.controlador;

import ups.edu.ec.excepciones.ExcepBuscar;
import ups.edu.ec.excepciones.ExcepActualizar;
import ups.edu.ec.modelo.Hipoteca;
import java.util.List;
import javax.persistence.Query;

/**
 *
 * @author japch
 */
public class ControladorHipoteca extends AbstracControlador<Hipoteca> {



    @Override
    public boolean validarActualizar(Hipoteca objeto) throws ExcepActualizar {
        String queryString = "Select * from hipoteca as h "
```

```
            + " where h.id like ?";
        Query query = getEm().createNativeQuery(queryString, Hipoteca.class);
        List<Hipoteca>hipoteca = query.setParameter(1, objeto.getId()).getResultList();
        if (hipoteca.size()==1) {
            return true;
        }else{
            return false;
        }
    }

    @Override
    public boolean validarExistente(Object id) throws ExcepBuscar  {


        String queryString = "Select * from hipoteca as h "
                + "where h.id like ?";
        Query query = getEm().createNativeQuery(queryString, Hipoteca.class);
        List<Hipoteca> hipoteca = query.setParameter(1, id).getResultList();
        if (hipoteca.size() == 1) {
            return true;
        }else{
            return false;
        }


    }
}
```

CONTROLADOR PERSONA

```
package ups.ed.ec.controlador;


import ups.edu.ec.excepciones.ExcepBuscar;
import ups.edu.ec.excepciones.ExcepActualizar;
import ups.edu.ec.modelo.Persona;
import java.util.List;
import javax.persistence.Query;


/**
 *
 * @author japch
 */
public class ControladorPersona extends AbstracControlador<Persona> {


    Persona persona = new Persona();


    @Override
    public boolean validarActualizar(Persona objeto) throws ExcepActualizar {
        String queryString = "Select * from persona as p "
                + "where p.cedula like ?";
        Query query = getEm().createNativeQuery(queryString, Persona.class);
        List<Persona> persona = query.setParameter(1, objeto.getCedula()).getResultList();
        if (persona.size() == 1) {
            return true;
        } else {
            return false;
        }
    }


    @Override
    public boolean validarExistente(Object id) throws ExcepBuscar {
        String queryString = "Select * from persona as p "
                + "where p.cedula like ?";
        Query query = getEm().createNativeQuery(queryString, Persona.class);
        List<Persona> persona = query.setParameter(1, id).getResultList();
        if (persona.size() == 1) {
            return true;
        }else{
            return false;
        }
    }

    public  boolean consultacedula(String cedula){

        Query consulta =  getEm().createNamedQuery("consultaCedula");
        consulta.setParameter("cedula", cedula);
        return true;
    }

}
```

CONTROLADOR REGEX

```
package ups.ed.ec.controlador;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author japch
 */
public class ControladorRegex {

    private Pattern patron;
    private Matcher corpus;

    public void ingreseRegex(String regex) {
        patron = Pattern.compile(regex);
    }

    public boolean validar(String texto) {
        corpus = patron.matcher(texto);
        return corpus.find();
    }

    public Pattern getPatron() {
        return patron;
    }

    public void setPatron(Pattern patron) {
        this.patron = patron;
    }

    public Matcher getCorpus() {
        return corpus;
    }

    public void setCorpus(Matcher corpus) {
        this.corpus = corpus;
    }

}
```

CONTROLADOR VIVIENDA

```
package ups.ed.ec.controlador;


import ups.edu.ec.excepciones.ExcepBuscar;
import ups.edu.ec.excepciones.ExcepActualizar;
import ups.edu.ec.modelo.Vivienda;
import java.util.List;
import javax.persistence.Query;


/**
 *
 * @author japch
 */
public class ControladorVivienda extends AbstracControlador<Vivienda> {


    @Override
    public boolean validarActualizar(Vivienda objeto) throws ExcepActualizar {


        String queryString = "Select * from casa as c "
                + "where c.codigoCasa like ?";
        Query query = getEm().createNativeQuery(queryString, Vivienda.class);
        List<Vivienda> casa = query.setParameter(1, objeto.getCodigoCasa()).getResultList();
        if (casa.size() == 1) {
            return true;
        } else {
            return false;
        }

    }

    @Override
    public boolean validarExistente(Object id) throws ExcepBuscar {
        String queryString = "Select * from casa as c "
                + "where c.codigoCasa like ?";
        Query query = getEm().createNativeQuery(queryString, Vivienda.class);
        List<Vivienda> casa = query.setParameter(1, id).getResultList();
        if (casa.size() == 1) {
            return true;
        } else {
            return false;
        }

    }
    public Vivienda verPorCedula(String cedula) {
        String queryString = "SELECT * FROM casa as c"
                + " where c.fk_persona like ?";
        Query query = getEm().createNativeQuery(queryString, Vivienda.class);
        Vivienda casa = (Vivienda) query.setParameter(1, cedula).getSingleResult();

        return casa;
    }

}
```

4.----------------------MODELO-------------------
HIPOITECA

```
package ups.edu.ec.modelo;


import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;



/**
 *
 * @author japch
 */
@Entity
public class Hipoteca implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
private Long id;


@Column
private String CedulaCliente;


@Column
private String CedulaGarante;


@Column
private double interes;
@Column
private double prestamo ;
@Column
private String LapzoDeTiempo;




public String getCedulaCliente() {
    return CedulaCliente;
}


public void setCedulaCliente(String CedulaCliente) {
    this.CedulaCliente = CedulaCliente;
}


public String getCedulaGarante() {
    return CedulaGarante;
}


public void setCedulaGarante(String CedulaGarante) {
    this.CedulaGarante = CedulaGarante;
}
```

```
public double getInteres() {
    return interes;
}

public void setInteres(double interes) {
    this.interes = interes;
}

public double getPrestamo() {
    return prestamo;
}

public void setPrestamo(double prestamo) {
    this.prestamo = prestamo;
}

public String getLapzoDeTiempo() {
    return LapzoDeTiempo;
}

public void setLapzoDeTiempo(String LapzoDeTiempo) {
    this.LapzoDeTiempo = LapzoDeTiempo;
}



public Long getId() {
    return id;
}

public void setId(Long id) {
        this.id = id;
    }



    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id != null ? id.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Hipoteca)) {
            return false;
        }
        Hipoteca other = (Hipoteca) object;
        if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "Hipoteca{" + "id=" + id + ", CedulaCliente=" + CedulaCliente + ", CedulaGarante=" + CedulaGarante + ", interes=" + interes + ", prestamo=" + prestamo + ", LapzoDeTiempo=" + LapzoDeTiempo + '}';
    }




}
```

PERSONA

```
package ups.edu.ec.modelo;


import java.io.Serializable;


import java.util.List;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 *
 * @author japch
 */

@Entity
@NamedQuery(name = "consultaCedula", query ="Select p from Persona p where p.cedula = :cedula")

public class Persona implements Serializable {



    @Id
    @Column
    private String cedula;
```

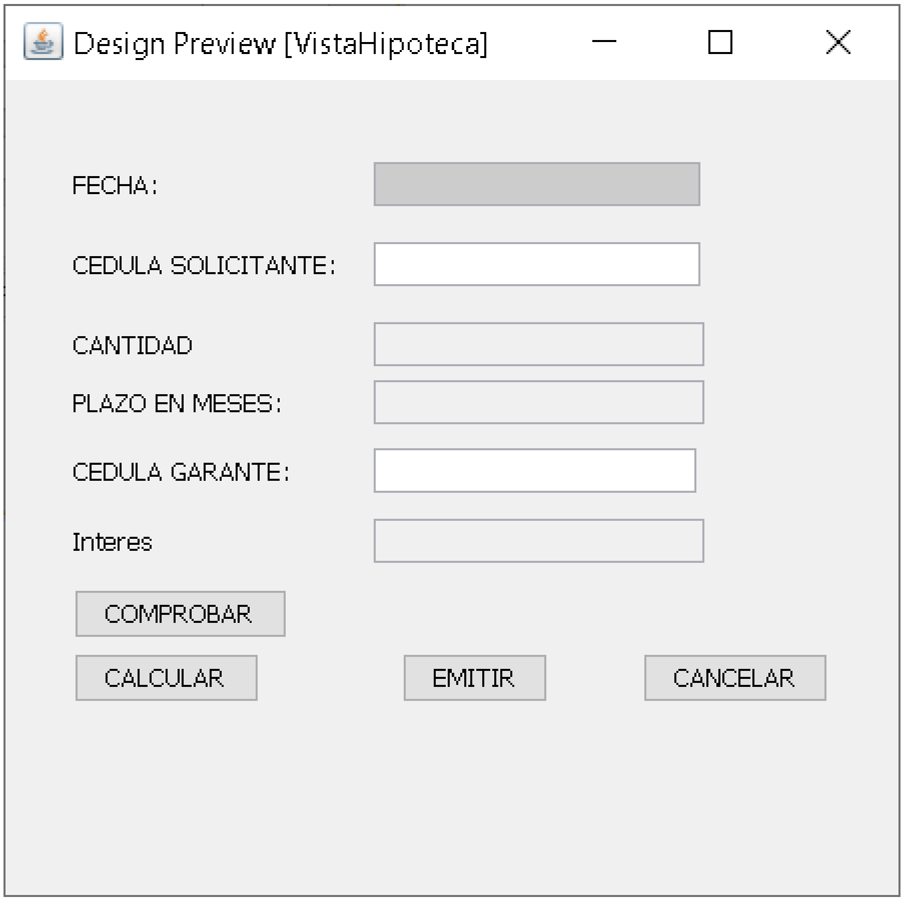
```
@Column
private String nombre;

@Column
private String apellido;

@Column
private String telefono;

@Column
private String fechaDeNacimiento;

@Column
private double salarioMensual;

@OneToMany(mappedBy = "persona", cascade = CascadeType.ALL)
private List<Vivienda> casas;

public List<Vivienda> getCasas() {
    return casas;
}

public void setCasas(List<Vivienda> casas) {
    this.casas = casas;
}


public String getCedula() {
    return cedula;
}

public void setCedula(String cedula) {
    this.cedula = cedula;
}

public String getNombre() {
    return nombre;
}

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

public String getApellido() {
    return apellido;
}

public void setApellido(String apellido) {
    this.apellido = apellido;
}

public String getTelefono() {
    return telefono;
}

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

public String getFechaDeNacimiento() {
    return fechaDeNacimiento;
}

public void setFechaDeNacimiento(String fechaDeNacimiento) {
        this.fechaDeNacimiento = fechaDeNacimiento;
    }



    public double getSalarioMensual() {
        return salarioMensual;
    }

    public void setSalarioMensual(double salarioMensual) {
        this.salarioMensual = salarioMensual;
    }

    @Override
    public String toString() {
        return "Persona{" + "cedula=" + cedula + ", nombre=" + nombre + ", apellido=" + apellido + ", telefono=" + telefono + ", fechaDeNacimiento=" + fechaDeNacimiento + ", salarioMensual=" + salarioMensual + ", casas=" + casas + '}';
    }



}
```

VIVIENDA

```
package ups.edu.ec.modelo;


import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
```

```
/**
 *
 * @author japch
 */
@Entity
public class Vivienda implements Serializable {

    @Id
    @Column
    private int codigoCasa;

    @Column
    private String direccionCalle1;
    @Column
    private String direccionCalle2;
    @Column
    private double valorDeCasa;

    @ManyToOne
    @JoinColumn(name="fk_persona")
    private Persona persona;



    public int getCodigoCasa() {
        return codigoCasa;
    }

    public void setCodigoCasa(int codigoCasa) {
        this.codigoCasa = codigoCasa;
    }

    public String getDireccionCalle1() {
        return direccionCalle1;
    }

    public void setDireccionCalle1(String direccionCalle1) {
        this.direccionCalle1 = direccionCalle1;
    }

    public String getDireccionCalle2() {
        return direccionCalle2;
    }

    public void setDireccionCalle2(String direccionCalle2) {
        this.direccionCalle2 = direccionCalle2;
    }

    public double getValorDeCasa() {
        return valorDeCasa;
    }

    public void setValorDeCasa(double valorDeCasa) {
        this.valorDeCasa = valorDeCasa;
    }

    public Persona getPersona() {
        return persona;
    }

    public void setPersona(Persona persona) {
        this.persona = persona;
    }



}
```

5.-----------------------------VISTA---------------------------



```
package ups.edu.ec.vista;


import com.itextpdf.text.Document;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.pdf.PdfPCell;

import com.itextpdf.text.pdf.PdfPTable;

import com.itextpdf.text.pdf.PdfWriter;

import ups.ed.ec.controlador.ControladorHipoteca;

import ups.ed.ec.controlador.ControladorPersona;

import ups.ed.ec.controlador.ControladorRegex;

import ups.ed.ec.controlador.ControladorVivienda;

import ups.edu.ec.modelo.Hipoteca;

import ups.edu.ec.modelo.Persona;

import ups.edu.ec.modelo.Vivienda;

import java.io.FileOutputStream;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author japch
 */
public class VistaHipoteca extends javax.swing.JFrame {

    private ControladorVivienda controladorC = new ControladorVivienda();
    private ControladorHipoteca controladorH= new ControladorHipoteca();
    private ControladorPersona controladorP = new ControladorPersona();
    private ControladorRegex controladorR = new ControladorRegex();
    /**
     * Creates new form VistaHipoteca
     */
    public VistaHipoteca() {
        initComponents();
   //     habiliatrCajones();
    btnCalcular.setEnabled(false);
    btnEmitir.setEnabled(false);

    }



     public void limpiar() {
        txtFecha.setText("");
        txtCedula.setText("");
        txtCedulaGarante.setText("");
        txtcantidad.setText("");
        txtPlazo.setText("");
        txtinteres.setText("");
    }


    public void habiliatrCajones() {
        txtcantidad.setEditable(true);
        txtPlazo.setEditable(true);
        txtCedulaGarante.setEditable(true);
        txtinteres.setEditable(true);


    }


    public void deshabiliatrCajones() {
        txtcantidad.setEditable(false);
        txtPlazo.setEditable(false);
        txtCedulaGarante.setEditable(false);
    }
    public void comprobarcedulas(){


        try {
            if (controladorP.leer(txtCedulaGarante.getText()) != null) {
                if (controladorP.leer(txtCedula.getText()) != null) {
                    JOptionPane.showMessageDialog(null, "Las personas estan registradas ");
                    habiliatrCajones();
                    btncomprobar.setEnabled(false);
                     btnCalcular.setEnabled(true);
                     btnEmitir.setEnabled(true);


                }else{
                    JOptionPane.showMessageDialog(null, "cedula del solicitante no existe ");
                }


            }else{
                JOptionPane.showMessageDialog(null, "cedula del garante no existe ");
            }
        } catch (Exception ex) {
            Logger.getLogger(VistaHipoteca.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        txtFecha = new javax.swing.JTextField();
        txtCedulaGarante = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        lblCedulaGarante = new javax.swing.JLabel();
        txtCedula = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        txtcantidad = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        txtPlazo = new javax.swing.JTextField();
        btnCalcular = new javax.swing.JToggleButton();
        btnEmitir = new javax.swing.JToggleButton();
        btnCancelar = new javax.swing.JToggleButton();
        jLabel1 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        txtinteres = new javax.swing.JTextField();
        btncomprobar = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


txtFecha.setEditable(false);
txtFecha.setBackground(new java.awt.Color(204, 204, 204));


txtCedulaGarante.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCedulaGaranteKeyTyped(evt);
    }
});


jLabel2.setText("CEDULA SOLICITANTE:");


lblCedulaGarante.setText("CEDULA GARANTE:");


txtCedula.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        txtCedulaKeyTyped(evt);
    }
});


jLabel3.setText("CANTIDAD");


txtcantidad.setEditable(false);


jLabel4.setText("PLAZO EN MESES:");


txtPlazo.setEditable(false);


btnCalcular.setText("CALCULAR");
btnCalcular.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCalcularActionPerformed(evt);
    }
});

btnEmitir.setText("EMITIR");
btnEmitir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEmitirActionPerformed(evt);
    }
});

btnCancelar.setText("CANCELAR");
btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

jLabel1.setText("FECHA:");

jLabel5.setText("Interes ");

txtinteres.setEditable(false);

btncomprobar.setText("COMPROBAR");
btncomprobar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btncomprobarActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

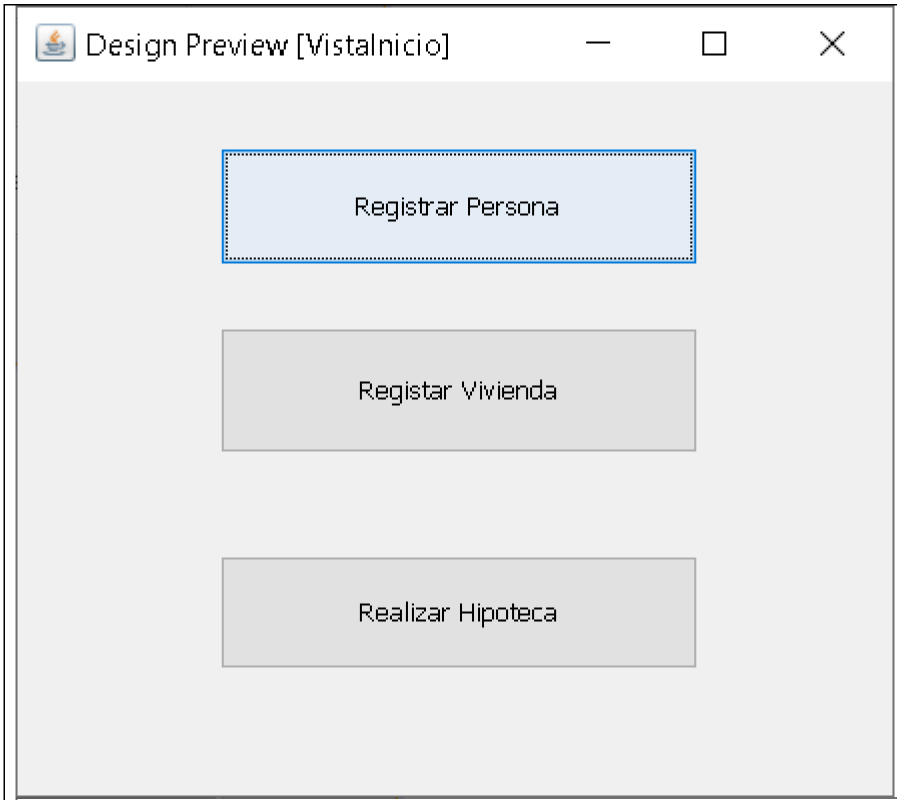
```
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(34, 34, 34)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(btncomprobar)
                .addComponent(jLabel5)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(btnCalcular)
                    .addGap(71, 71, 71)
                    .addComponent(btnEmitir)
                    .addGap(47, 47, 47)
                    .addComponent(btnCancelar))
                .addComponent(jLabel1)
                .addComponent(jLabel3)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addGap(18, 18, 18)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(txtcantidad, javax.swing.GroupLayout.PREFERRED_SIZE, 165, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                            .addComponent(txtFecha, javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(txtCedula, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 163, Short.MAX_VALUE))))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(txtinteres, javax.swing.GroupLayout.PREFERRED_SIZE, 165, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel4)
                            .addComponent(lblCedulaGarante))
```

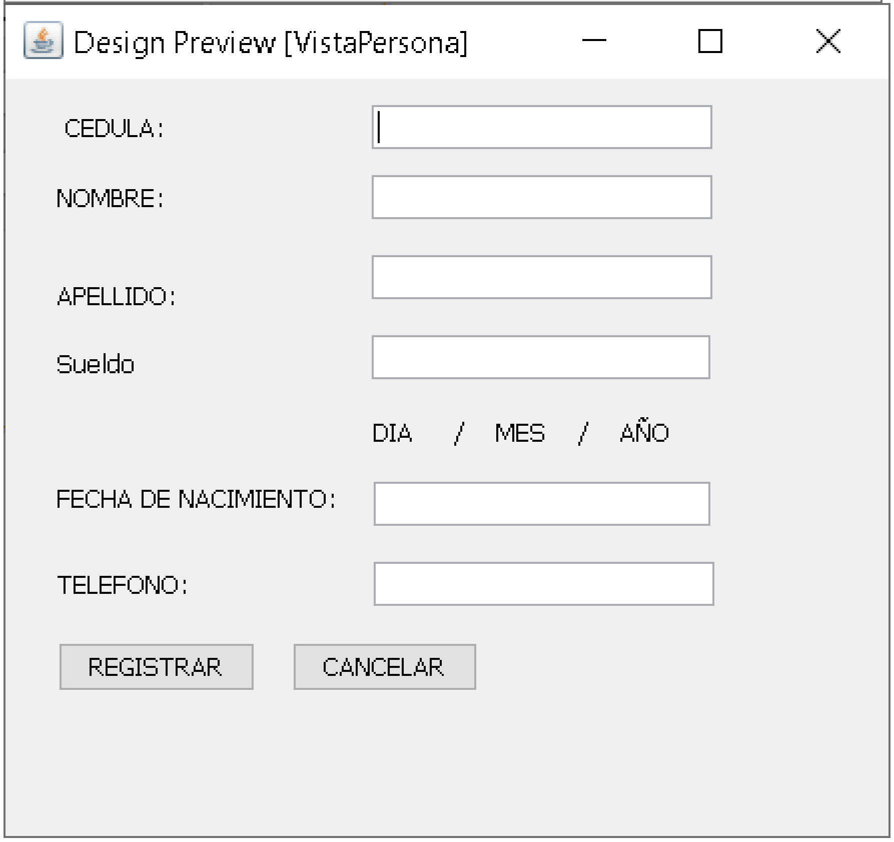
```
              .addGap(41, 41, 41)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txtCedulaGarante, javax.swing.GroupLayout.PREFERRED_SIZE, 161,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtPlazo, javax.swing.GroupLayout.PREFERRED_SIZE, 165,
javax.swing.GroupLayout.PREFERRED_SIZE)))))
        .addContainerGap(35, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(41, 41, 41)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel1)
          .addComponent(txtFecha, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel2)
          .addComponent(txtCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel3)
          .addComponent(txtcantidad, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel4)
          .addComponent(txtPlazo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(12, 12, 12)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(lblCedulaGarante)
```

```
                .addComponent(txtCedulaGarante, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(13, 13, 13)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel5)
                .addComponent(txtinteres, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(btncomprobar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(btnCalcular)
                .addComponent(btnEmitir)
                .addComponent(btnCancelar))
            .addContainerGap(96, Short.MAX_VALUE))
    );

    pack();
}// </editor-fold>

private void txtCedulaGaranteKeyTyped(java.awt.event.KeyEvent evt) {
}
private void txtCedulaKeyTyped(java.awt.event.KeyEvent evt) {


}
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {


    emitirregistro();


}


private void btnEmitirActionPerformed(java.awt.event.ActionEvent evt) {
        if (!txtcantidad.getText().isEmpty() && !txtCedula.getText().isEmpty() && !txtPlazo.getText().isEmpty())
{


            Hipoteca h= new Hipoteca();
            try {


                h.setCedulaCliente(txtCedula.getText());
                h.setCedulaGarante(txtCedulaGarante.getText());
                Double interes = Double.parseDouble(txtinteres.getText());
                h.setInteres(interes);
                h.setLapzoDeTiempo(txtPlazo.getText());
                h.setPrestamo(Double.parseDouble(txtcantidad.getText()));
                controladorH.crear(h);
                limpiar();
                deshabiliatrCajones();
                txtCedula.setEditable(true);
                JOptionPane.showMessageDialog(null, "HIPOTECA EMITIDO CON EXITO");
            } catch (Exception ex) {
                ex.printStackTrace();
            }

        }else{
            System.out.println("EXISTEN CAMPOS VACIOS");
        }
    }

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

        limpiar();
        txtCedula.setEditable(true);
        deshabiliatrCajones();
    }
    private void btncomprobarActionPerformed(java.awt.event.ActionEvent evt) {


    comprobarcedulas();
        // TODO add your handling code here:
    }

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

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

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

    public void emitirregistro(){

         try {
            Persona solicitante = controladorP.leer(txtCedula.getText());
            Vivienda v = controladorC.verPorCedula(solicitante.getCedula());
            Persona garante= new Persona();
            if (!txtCedulaGarante.getText().isEmpty()) {
                garante = controladorP.leer(txtCedulaGarante.getText());
            }
             double prestamo = Double.parseDouble(txtcantidad.getText());
            int tiempo = Integer.parseInt(txtPlazo.getText());
            double capital = prestamo / tiempo;



            Document document = new Document();
            PdfWriter.getInstance(document, new FileOutputStream("Montos.pdf"));
          //  D:\Metodos Numericos\CreditoBancarioJPA5
            document.open();
            Paragraph into = new Paragraph("REGISTRO DE PAGOS DE HIPOTECA ");
            Paragraph space = new Paragraph(" ");


//solicitante
            PdfPTable tablasolicitante = new PdfPTable(4);


            PdfPCell csolicitante = new PdfPCell(new Paragraph("CEDULA SOLICITANTE"));
            PdfPCell nsolicitante = new PdfPCell(new Paragraph("NOMBRE SOLICITANTE"));
            PdfPCell asolicitante = new PdfPCell(new Paragraph("APELLIDO SOLICITANTE"));
            PdfPCell tsolicitante = new PdfPCell(new Paragraph("TELEFONO SOLICITANTE"));


                tablasolicitante.addCell(csolicitante);
                tablasolicitante.addCell(nsolicitante);
                tablasolicitante.addCell(asolicitante);
                tablasolicitante.addCell(tsolicitante);


            csolicitante = new PdfPCell(new Paragraph(solicitante.getCedula()));
            nsolicitante = new PdfPCell(new Paragraph(solicitante.getNombre()));
            asolicitante = new PdfPCell(new Paragraph(solicitante.getApellido()));
            tsolicitante = new PdfPCell(new Paragraph(solicitante.getTelefono()));
            tablasolicitante.addCell(csolicitante);
            tablasolicitante.addCell(nsolicitante);
            tablasolicitante.addCell(asolicitante);
            tablasolicitante.addCell(tsolicitante);


//garante
            PdfPTable tablagarante = new PdfPTable(4);


            PdfPCell cgarante = new PdfPCell(new Paragraph("CEDULA GARANTE"));
            PdfPCell ngarante = new PdfPCell(new Paragraph("NOMBRE GARANTE"));
            PdfPCell agaranate = new PdfPCell(new Paragraph("APELLIDO GARANTE"));
            PdfPCell tgarante = new PdfPCell(new Paragraph("TELEFONO GARANTE"));

                tablagarante.addCell(cgarante);
                tablagarante.addCell(ngarante);
                tablagarante.addCell(agaranate);
                tablagarante.addCell(tgarante);

            cgarante = new PdfPCell(new Paragraph(garante.getCedula()));
            ngarante = new PdfPCell(new Paragraph(garante.getNombre()));
            agaranate = new PdfPCell(new Paragraph(garante.getApellido()));
            tgarante = new PdfPCell(new Paragraph(garante.getTelefono()));
            tablagarante.addCell(cgarante);
            tablagarante.addCell(ngarante);
            tablagarante.addCell(agaranate);
            tablagarante.addCell(tgarante);
//tablaregistro de pagos
            PdfPTable tablapagos = new PdfPTable(5);
            PdfPCell cmes = new PdfPCell(new Paragraph("MES"));
            PdfPCell ccapital = new PdfPCell(new Paragraph("CAPITAL"));
            PdfPCell csaldo = new PdfPCell(new Paragraph("SALDO"));
            PdfPCell cinteres = new PdfPCell(new Paragraph("INTERES"));
            PdfPCell ctotal = new PdfPCell(new Paragraph("TOTAL"));

                tablapagos.addCell(cmes);
                tablapagos.addCell(ccapital);
                tablapagos.addCell(csaldo);
                tablapagos.addCell(cinteres);
                tablapagos.addCell(ctotal);

            for (int i = 0; i < tiempo; i++) {
                double saldo = prestamo - (i * capital);
                double interes = saldo * (.17 / 12);
            cmes = new PdfPCell(new Paragraph((i + 1) + ""));
            ccapital = new PdfPCell(new Paragraph(String.format("%,.2f", capital)));
            csaldo = new PdfPCell(new Paragraph(String.format("%,.2f", saldo)));
            cinteres = new PdfPCell(new Paragraph(String.format("%,.2f", interes)));
            ctotal = new PdfPCell(new Paragraph(String.format("%,.2f", (capital + interes))));

            tablapagos.addCell(cmes);
            tablapagos.addCell(ccapital);
            tablapagos.addCell(csaldo);
            tablapagos.addCell(cinteres);
            tablapagos.addCell(ctotal);

        }

          document.add(into);
          document.add(space);
          document.add(tablasolicitante);
          document.add(space);
          document.add(tablagarante);
          document.add(space);
          document.add(tablapagos);
          document.close();

        JOptionPane.showMessageDialog(null, "Registro creado exitosamente");

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


}
```

```
    // Variables declaration - do not modify
    private javax.swing.JToggleButton btnCalcular;
    private javax.swing.JToggleButton btnCancelar;
    private javax.swing.JToggleButton btnEmitir;
    private javax.swing.JButton btncomprobar;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel lblCedulaGarante;
    private javax.swing.JTextField txtCedula;
    private javax.swing.JTextField txtCedulaGarante;
    private javax.swing.JTextField txtFecha;
    private javax.swing.JTextField txtPlazo;
    private javax.swing.JTextField txtcantidad;
    private javax.swing.JTextField txtinteres;
    // End of variables declaration
}
```





/*

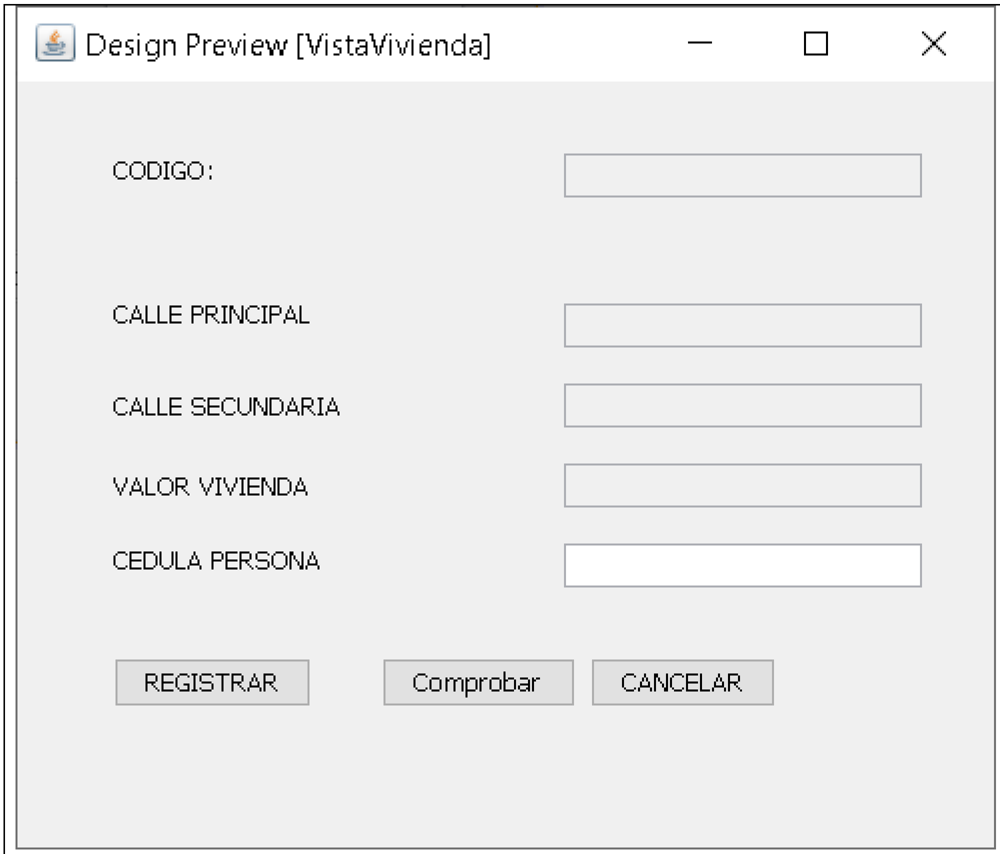
*** To change this license header, choose License Headers in Project Properties.**

*** To change this template file, choose Tools | Templates**

```
 * and open the template in the editor.
 */
package ups.edu.ec.vista;

import ups.ed.ec.controlador.ControladorHipoteca;
import ups.ed.ec.controlador.ControladorPersona;
import ups.ed.ec.controlador.ControladorRegex;
import ups.ed.ec.controlador.ControladorVivienda;
import ups.edu.ec.modelo.Vivienda;
import ups.edu.ec.modelo.Persona;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;

/**
 *
 * @author japch
 */
public class VistaPersona extends javax.swing.JFrame {
    private ControladorVivienda cv = new ControladorVivienda();
    private ControladorHipoteca ch = new ControladorHipoteca();
    private ControladorPersona cp = new ControladorPersona();
    private ControladorRegex cr = new ControladorRegex();
    /**
     * Creates new form VistaPersonas
     */
    public VistaPersona() {
        initComponents();
    }

        public void crearpersona(){
         if (!txtcedula.getText().isEmpty() && !txtnombre.getText().isEmpty() && !txtapellido.getText().isEmpty()
        &&
!txttelefono.getText().isEmpty()&&!txtsueldo.getText().isEmpty()&&!txtfechanac.getText().isEmpty()) {


            cr.ingreseRegex("^\\d{9}-?\\d{1}$");
            boolean validarcedula = cr.validar(txtcedula.getText());
            if (validarcedula) {
                try {
                    Persona p = new Persona();
                    List<Vivienda> casas = new ArrayList<>();
                    p.setCedula(txtcedula.getText());
                    p.setNombre(txtnombre.getText());
                    p.setApellido(txtapellido.getText());
                    p.setTelefono(txttelefono.getText());
                    p.setFechaDeNacimiento(txtfechanac.getText());
                    p.setSalarioMensual(Double.parseDouble(txtsueldo.getText()));
                    p.setCasas(casas);
                    cp.crear(p);
                    JOptionPane.showMessageDialog(null, "Persona registrada");


                    limpiar();
                } catch (Exception ex) {
                    ex.printStackTrace();
                }


            } else {
                JOptionPane.showMessageDialog(null, "Cedula invalida");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Llenar todos los campos");
        }
    }
      public void limpiar() {
        txtcedula.setText("");
        txtnombre.setText("");

        txtapellido.setText("");

        txttelefono.setText("");

        txtsueldo.setText("");

        txtfechanac.setText("");

    }

    /**

     * This method is called from within the constructor to initialize the form.

     * WARNING: Do NOT modify this code. The content of this method is always

     * regenerated by the Form Editor.

     */

    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed" desc="Generated Code">

    private void initComponents() {


        btnCancelar = new javax.swing.JToggleButton();

        jLabel4 = new javax.swing.JLabel();

        jLabel1 = new javax.swing.JLabel();

        txtsueldo = new javax.swing.JTextField();

        txtcedula = new javax.swing.JTextField();

        jLabel6 = new javax.swing.JLabel();

        jLabel2 = new javax.swing.JLabel();

        txtnombre = new javax.swing.JTextField();

        jLabel7 = new javax.swing.JLabel();

        jLabel3 = new javax.swing.JLabel();

        txtapellido = new javax.swing.JTextField();

        jLabel5 = new javax.swing.JLabel();

        txttelefono = new javax.swing.JTextField();

        btnRegistrar = new javax.swing.JButton();

        txtfechanac = new javax.swing.JTextField();


        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
btnCancelar.setText("CANCELAR");
btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

jLabel4.setText("Sueldo");

jLabel1.setText("CEDULA:");

jLabel6.setText("FECHA DE NACIMIENTO:");

jLabel2.setText("NOMBRE:");

jLabel7.setText("DIA     /     MES     /     AÑO");

jLabel3.setText("APELLIDO:");

jLabel5.setText("TELEFONO:");

btnRegistrar.setText("REGISTRAR");
btnRegistrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegistrarActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
          .addGap(26, 26, 26)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
              .addComponent(btnRegistrar)
              .addGap(18, 18, 18)
              .addComponent(btnCancelar))
            .addGroup(layout.createSequentialGroup()
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.TRAILING))
              .addGap(103, 103, 103)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txtcedula, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtapellido, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtnombre, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtsueldo, javax.swing.GroupLayout.PREFERRED_SIZE, 169,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addComponent(jLabel3)
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createSequentialGroup()
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel6)
                .addComponent(jLabel5))
              .addGap(18, 18, 18)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txttelefono, javax.swing.GroupLayout.PREFERRED_SIZE, 170,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
```

```
                .addComponent(txtfechanac, javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel7, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 168, Short.MAX_VALUE)))))
        .addContainerGap(87, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel1)
          .addComponent(txtcedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel2)
          .addComponent(txtnombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addGroup(layout.createSequentialGroup()
            .addComponent(txtapellido, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(txtsueldo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
          .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel3)
            .addGap(18, 18, 18)
            .addComponent(jLabel4)))
        .addGap(18, 18, 18)
        .addComponent(jLabel7)
        .addGap(17, 17, 17)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel6)
          .addComponent(txtfechanac, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel5)
          .addComponent(txttelefono, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(btnRegistrar)
          .addComponent(btnCancelar))
        .addContainerGap(72, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    limpiar();
    this.setVisible(false);
  }

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

    crearpersona();

  }

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

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

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

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

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

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

    // Variables declaration - do not modify
    private javax.swing.JToggleButton btnCancelar;
    private javax.swing.JButton btnRegistrar;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JTextField txtapellido;
    private javax.swing.JTextField txtcedula;
    private javax.swing.JTextField txtfechanac;
    private javax.swing.JTextField txtnombre;
    private javax.swing.JTextField txtsueldo;
    private javax.swing.JTextField txttelefono;
    // End of variables declaration
}
```



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


import ups.ed.ec.controlador.ControladorHipoteca;
import ups.ed.ec.controlador.ControladorPersona;
import ups.ed.ec.controlador.ControladorRegex;
import ups.ed.ec.controlador.ControladorVivienda;
import ups.edu.ec.modelo.Vivienda;
import ups.edu.ec.modelo.Persona;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
```

```
/**
 *
 * @author japch
 */
public class VistaVivienda extends javax.swing.JFrame {
    private ControladorVivienda cv = new ControladorVivienda();
    private ControladorHipoteca ch = new ControladorHipoteca();
    private ControladorPersona cp = new ControladorPersona();
    private ControladorRegex cr = new ControladorRegex();
    /**
     * Creates new form VistaVivienda
     */
    public VistaVivienda() {
        initComponents();
        deshabiliatCajones();

    }
   public void limpiar() {
        txtcedula.setText("");
        txtcallep.setText("");
        txtcalles.setText("");
        txtcodigo.setText("");
        txtvalorvivienda.setText("");
    }

    public void habilitarCajones() {
        txtcallep.setEditable(true);
        txtcalles.setEditable(true);
        txtcodigo.setEditable(true);
        txtvalorvivienda.setEditable(true);
        btnRegistrar.setEnabled(true);
    }
```

```
public void deshabiliatCajones() {
    txtcallep.setEditable(false);
    txtcalles.setEditable(false);
    txtcodigo.setEditable(false);
    txtvalorvivienda.setEditable(false);
    btnRegistrar.setEnabled(false);
}

public void validarcedula(){
    String cedula = txtcedula.getText();
    if(cp.consultacedula(cedula)){
        JOptionPane.showMessageDialog(null, "La Persona si esta registrada");
        habilitarCajones();
      }else {
        JOptionPane.showMessageDialog(null, "no existe la cedula");
        txtcedula.setText("");
    }




}
public void crearvivienda(){
    if (!txtcedula.getText().isEmpty() && !txtcallep.getText().isEmpty() && !txtcalles.getText().isEmpty()
        && !txtcodigo.getText().isEmpty() && !txtvalorvivienda.getText().isEmpty()) {
        cr.ingreseRegex("^\\d{1,}$");
        boolean validarC = cr.validar(txtcodigo.getText());
        if (validarC) {

            List<Vivienda> casas = new ArrayList<>();
            try {
                    Persona persona = cp.leer(txtcedula.getText());
                    Vivienda c = new Vivienda();
                    c.setCodigoCasa(Integer.parseInt(txtcodigo.getText()));
                    c.setDireccionCalle1(txtcallep.getText());
                    c.setDireccionCalle2(txtcalles.getText());
                    c.setValorDeCasa(Double.parseDouble(txtvalorvivienda.getText()));
                    c.setPersona(persona);
                    casas.add(c);
                    persona.setCasas(casas);
                    cp.actualizar(persona);
                    JOptionPane.showMessageDialog(null, persona.getNombre().toUpperCase()+" SE HA REGISTRADO SU CASA EXITOSAMENTE");
                    txtcedula.setEditable(true);
                    limpiar();
                    deshabiliatCajones();
                    this.setVisible(false);
                } catch (Exception ex) {
                    ex.printStackTrace();
                }

            } else {
                JOptionPane.showMessageDialog(null, "DEBE INGRESAR UN DIGITO EN EL CMAPO DE CODIGO");
            }

        } else {
            JOptionPane.showMessageDialog(null, "EXISTEN CAMPOS VACIOS");
        }
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
```

```
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel6 = new javax.swing.JLabel();
    txtcodigo = new javax.swing.JTextField();
    txtcedula = new javax.swing.JTextField();
    txtcallep = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    txtcalles = new javax.swing.JTextField();
    jLabel4 = new javax.swing.JLabel();
    txtvalorvivienda = new javax.swing.JTextField();
    jLabel5 = new javax.swing.JLabel();
    btnRegistrar = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel6.setText("CEDULA PERSONA");

    txtcodigo.setEditable(false);

    txtcedula.setToolTipText("1. PRIMERO INGRESE CEDULA");
    txtcedula.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtcedulaKeyTyped(evt);
        }
    });
```

```
txtcallep.setEditable(false);

jLabel3.setText("CALLE SECUNDARIA");

txtcalles.setEditable(false);

jLabel4.setText("VALOR VIVIENDA");

txtvalorvivienda.setEditable(false);

jLabel5.setText("CALLE PRINCIPAL");

btnRegistrar.setText("REGISTRAR");
btnRegistrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegistrarActionPerformed(evt);
    }
});

btnCancelar.setText("CANCELAR");
btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

jLabel1.setText("CODIGO:");

jButton1.setText("Comprobar");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
```

```
      }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(48, 48, 48)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jLabel5)
              .addComponent(jLabel4)
              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addComponent(jLabel3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
              .addComponent(jLabel6)
              .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(txtcodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                .addComponent(txtcallep)
                .addComponent(txtvalorvivienda, javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtcalles, javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(txtcedula, javax.swing.GroupLayout.PREFERRED_SIZE, 179,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(36, 36, 36))
          .addGroup(layout.createSequentialGroup()
            .addComponent(btnRegistrar)
            .addGap(35, 35, 35)
            .addComponent(jButton1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(btnCancelar)
            .addGap(109, 109, 109))))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(36, 36, 36)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addComponent(txtcodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(53, 53, 53)
            .addComponent(txtcallep, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(txtcalles, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel3))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(txtvalorvivienda, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel4)))
          .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addGap(56, 56, 56)
            .addComponent(jLabel5)))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel6)
          .addComponent(txtcedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(35, 35, 35)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(btnRegistrar)
          .addComponent(btnCancelar)
          .addComponent(jButton1))
        .addContainerGap(70, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void txtcedulaKeyTyped(java.awt.event.KeyEvent evt) {


  }

  private void btnRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
    crearvivienda();
  }

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

    limpiar();
    txtcedula.setEditable(true);
    deshabiliatCajones();
    this.setVisible(false);
  }
```

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        validarcedula();
    }

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

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

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

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

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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


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


    // Variables declaration - do not modify
    private javax.swing.JButton btnCancelar;
    private javax.swing.JButton btnRegistrar;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JTextField txtcallep;
    private javax.swing.JTextField txtcalles;
    private javax.swing.JTextField txtcedula;
    private javax.swing.JTextField txtcodigo;
    private javax.swing.JTextField txtvalorvivienda;
    // End of variables declaration
}
```

6.

N.

RESULTADO(S) OBTENIDO(S):
Realización del programa con un éxito satisfactorio de mi persona al saber que se pudo resolver los puntos dados para el trabajo, tratando de realizar con los ejemplos dados en clases además de obtener algunas informaciones de sitios educativos en la web,

CONCLUSIONES: Los temas dados para la realización del trabajo fueron dados con una buena explicación para poder realizarlos además de que abarca los temas suficientes para poder hacerlos, además de la incentivar buscar alternativas para mejorar el programa que pide, ya sea buscar el uso de librerías apartes. Aparte de cierto tipo de programación que nos ayuda a reducir código.

RECOMENDACIONES: La practica constante en el ámbito de programación es lo necesario, no necesariamente se necesita saber el paso a paso memorizado para realizar un programa ya que muchos piden otro objetivo.

***Nombre de estudiante*: ___________Christian Japon__________________**

***Firma de estudiante*: ____________________________________**