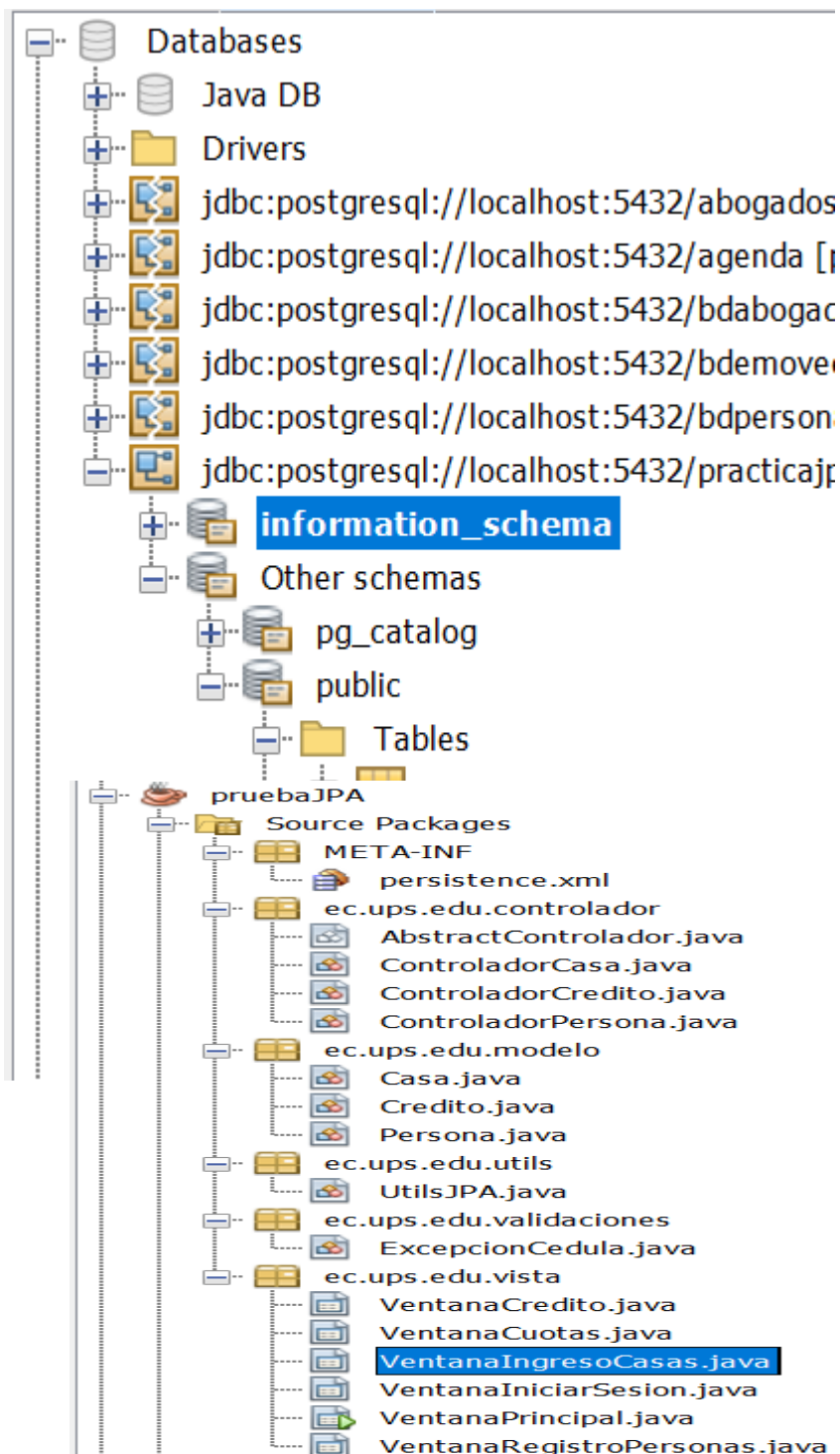
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


CARRERA: Computacion	ASIGNATURA: Programacion Aplicada
----------------------	-----------------------------------

NRO. PRÁCTICA:	TÍTULO PRÁCTICA: PracticaJPA
----------------	------------------------------

### ACTIVIDADES DESARROLLADAS

1.



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

## 2.

### Clase casa

```

/*
 * 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 ec.ups.edu.modelo;


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

/**
 *
 * @author santi
 */
@Entity
public class Casa implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Column
    private String ubicacion;
    @Column
    private String callePrimaria;
    @Column
    private String calleSecundaria;
    @Column
    private String numeroPisos;
    @Column
    private double valorCasa;

    public Long getId() {
        return id;
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

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

public String getCallePrimaria() {
    return callePrimaria;
}

public void setCallePrimaria(String callePrimaria) {
    this.callePrimaria = callePrimaria;
}

public String getCalleSecundaria() {
    return calleSecundaria;
}

public void setCalleSecundaria(String calleSecundaria) {
    this.calleSecundaria = calleSecundaria;
}

public String getNumeroPisos() {
    return numeroPisos;
}

public void setNumeroPisos(String numeroPisos) {
    this.numeroPisos = numeroPisos;
}

public double getValorCasa() {
    return valorCasa;
}

public void setValorCasa(double valorCasa) {
    this.valorCasa = valorCasa;
}

public String getUbicacion() {
    return ubicacion;
}

public void setUbicacion(String ubicacion) {
    this.ubicacion = ubicacion;
}

@Override
public int hashCode() {
    int hash = 5;
    hash = 47 * hash + Objects.hashCode(this.id);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {

```

```
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Casa other = (Casa) obj;
    if (!Objects.equals(this.id, other.id)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Casa{" + "id=" + id + ", ubicacion=" + ubicacion + ", callePrimaria="
+ callePrimaria + ", calleSecundaria=" + calleSecundaria + ", numeroPisos=" +
numeroPisos + ", valorCasa=" + valorCasa + '}';
}
}
```


## Clase Credito

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

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

/**
 *
 * @author santi
 */
@Entity
public class Credito implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Column
    private Date fechaPrestamo;
    @Column
    private double monto;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

@Column
private float tasaInteres;
@Column
private String sistemaAmortizacion;
@Column
private int plazo;
@Column
private Date fechaInicio;
@Column
private Date fechaFin;

public Long getId() {
    return id;
}

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

public Date getFechaPrestamo() {
    return fechaPrestamo;
}

public void setFechaPrestamo(Date fechaPrestamo) {
    this.fechaPrestamo = fechaPrestamo;
}

public double getMonto() {
    return monto;
}

public void setMonto(double monto) {
    this.monto = monto;
}

public float getTasaInteres() {
    return tasaInteres;
}


public void setTasaInteres(float tasaInteres) {
    this.tasaInteres = tasaInteres;
}

public String getSistemaAmortizacion() {
    return sistemaAmortizacion;
}

public void setSistemaAmortizacion(String sistemaAmortizacion) {
    this.sistemaAmortizacion = sistemaAmortizacion;
}

public int getPlazo() {
    return plazo;
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public void setPlazo(int plazo) {
    this.plazo = plazo;
}

public Date getFechaInicio() {
    return fechaInicio;
}

public void setFechaInicio(Date fechaInicio) {
    this.fechaInicio = fechaInicio;
}


public Date getFechaFin() {
    return fechaFin;
}

public void setFechaFin(Date fechaFin) {
    this.fechaFin = fechaFin;
}

@Override
public int hashCode() {
    int hash = 5;
    hash = 59 * hash + Objects.hashCode(this.id);
    hash = 59 * hash + Objects.hashCode(this.fechaPrestamo);
    hash = 59 * hash + Objects.hashCode(this.fechaInicio);
    hash = 59 * hash + Objects.hashCode(this.fechaFin);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Credito other = (Credito) obj;
    if (!Objects.equals(this.id, other.id)) {
        return false;
    }
    if (!Objects.equals(this.fechaPrestamo, other.fechaPrestamo)) {
        return false;
    }
    if (!Objects.equals(this.fechaInicio, other.fechaInicio)) {
        return false;
    }
    if (!Objects.equals(this.fechaFin, other.fechaFin)) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Credito{" + "id=" + id + ", fechaPrestamo=" + fechaPrestamo + ",
monto=" + monto + ", tasaInteres=" + tasaInteres + ", sistemaAmortizacion=" +
sistemaAmortizacion + ", plazo=" + plazo + ", fechaInicio=" + fechaInicio + ",
fechaFin=" + fechaFin + '}';
}

}

```

## Clase Persona

```


/*
 * 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 ec.ups.edu.modelo;

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

/**
 *
 * @author santi
 */
@Entity
public class Persona implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @Column
    private String cedula;
    @Column
    private String nombre;
    @Column
    private String apellido;
    @Column
    private String direccion;
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

@Column
private double sueldo;
@Column
private Date fechaNacimiento;
@Column
private String correo;
@Column
private String contrasena;

public Long getId() {
    return id;
}

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

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 getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public double getSueldo() {
    return sueldo;
}

public void setSueldo(double sueldo) {

```



```
        this.sueldo = sueldo;
    }

    public Date getFechaNacimiento() {
        return fechaNacimiento;
    }

    public void setFechaNacimiento(Date fechaNacimiento) {
        this.fechaNacimiento = fechaNacimiento;
    }

    public String getCorreo() {
        return correo;
    }


    public void setCorreo(String correo) {
        this.correo = correo;
    }

    public String getContrasena() {
        return contrasena;
    }

    public void setContrasena(String contrasena) {
        this.contrasena = contrasena;
    }

    @Override
    public int hashCode() {
        int hash = 5;
        hash = 67 * hash + Objects.hashCode(this.id);
        hash = 67 * hash + Objects.hashCode(this.cedula);
        hash = 67 * hash + Objects.hashCode(this.correo);
        hash = 67 * hash + Objects.hashCode(this.contrasena);
        return hash;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final Persona other = (Persona) obj;
        if (!Objects.equals(this.cedula, other.cedula)) {
            return false;
        }
        if (!Objects.equals(this.correo, other.correo)) {
            return false;
        }
    }
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }
    if (!Objects.equals(this.contrasena, other.contrasena)) {
        return false;
    }
    if (!Objects.equals(this.id, other.id)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Persona{" + "id=" + id + ", cedula=" + cedula + ", nombre=" + nombre
+ ", apellido=" + apellido + ", direccion=" + direccion + ", sueldo=" + sueldo + ",
fechaNacimiento=" + fechaNacimiento + ", correo=" + correo + ", contrasena=" +
contrasena + '}';
}
}

```

### Clase AbstractControlador

```

/*
 * 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 ec.ups.edu.vista;

import java.awt.Panel;
import javax.swing.JLabel;

/**
 *
 * @author santi
 */
public class VentanaCuotas extends javax.swing.JInternalFrame {

    /**
     * Creates new form VentanaCuotas
     */
    double prestamo;
    double interes;
    int plazo;

    public VentanaCuotas(double prestamo, double interes, int plazo) {
        initComponents();

        this.prestamo = prestamo;
        this.interes = interes;
        this.plazo = plazo;
    }
}


```

```
public void mostrar() {

    jPanel1.removeAll();
    jPanel1.updateUI();
    double cuota = prestamo / plazo;
    JLabel label;
    double[][] matriz = new double[plazo][5];
    matriz[0][0] = 0;
    matriz[0][1] = 0;
    matriz[0][2] = 0;
    matriz[0][3] = 0;
    matriz[0][4] = prestamo;
    for (int i = 1; i < plazo; i++) {
        for (int j = 0; j < 5; j++) {
            if (j == 0) {
                matriz[i][j] = i;
            }
            if (j == 1) {
                matriz[i][j] = cuota;
            }
            if (j == 2) {
                matriz[i][j] = matriz[i - 1][4] * (interes / 100);
            }
            if (j == 3) {
                matriz[i][j] = matriz[i][2] + cuota;
            }
            if (j == 4) {
                matriz[i][j] = matriz[i - 1][4] - cuota;
            }
        }
    }
    String[] c = {"Mes", "Cuota", "Interes", "Amortizacion", "Capital"};
    for (int i = 0; i < 5; i++) {
        label = new JLabel();
        label.setSize(110, 30);
        label.setText(c[i]);
        jPanel1.add(label);
    }
    jPanel1.updateUI();
    for (int i = 0; i < plazo; i++) {
        for (int j = 0; j < 5; j++) {
            label = new JLabel();
            label.setSize(110, 30);
            label.setText(matriz[i][j] + "");
            jPanel1.add(label);
        }
    }
    jPanel1.updateUI();

}

/**
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

* 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() {

    btnMostrar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    jPanel1 = new javax.swing.JPanel();

    btnMostrar.setText("Mostrar");
    btnMostrar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnMostrarActionPerformed(evt);
        }
    });

    jPanel1.setLayout(new java.awt.GridLayout(0, 5));
    jScrollPane1.setViewportView(jPanel1);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                layout.createSequentialGroup()
                    .addContainerGap(495, Short.MAX_VALUE)
                    .addComponent(btnMostrar)
                    .addGap(19, 19, 19))
            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
355, Short.MAX_VALUE)
                    .addGap(18, 18, 18)
                    .addComponent(btnMostrar)
                    .addContainerGap())
            );


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

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

    mostrar();

}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
// Variables declaration - do not modify
private javax.swing.JButton btnMostrar;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}
```

## Clase controlador Casa

```
package ec.ups.edu.controlador;

import ec.ups.edu.modelo.Casa;
import ec.ups.edu.modelo.Persona;
import static ec.ups.edu.utils.UtilsJPA.getEntityManager;
import javax.persistence.EntityManager;
import javax.persistence.NoResultException;

/**
 *
 * @author santi
 */
public class ControladorCasa extends AbstractControlador<Casa> {

    //Singleton
    public static ControladorCasa instancia;

    public static ControladorCasa getControladorCasa() {

        if (instancia == null) {

            instancia = new ControladorCasa();

        }

        return instancia;

    }

    //Acaba Singleton
    @Override
    public boolean validacion(Casa objeto) throws Exception {

        return true;

    }

    public Casa buscar(double valorcasa) {

        EntityManager em = getEntityManager();

        try {

            String jpql = "Select c from Casa c where c.valorcasa='" + valorcasa +
""";
```

```
Casa c = (Casa) em.createQuery(jpql).getSingleResult();
```

```
return c;
```

```
} catch (NoResultException e) {
```

```
    System.out.println("Error " + e);
```

```
}
```

```
return null;
```

```
}
```

```
}
```

### Clase controlador credito

```
package ec.ups.edu.controlador;
```

```
import ec.ups.edu.modelo.Credito;
```

```
/**
```

```
 *
```

```
 * @author santi
```

```
 */
```

```
public class ControladorCredito extends AbstractControlador<Credito> {
```

```
    public static ControladorCredito instancia;
```

```
    public static ControladorCredito getControladorCasa() {
```

```
        if (instancia == null) {
```

```
            instancia = new ControladorCredito();
```

```
        }
```

```
        return instancia;
```

```
    }
```

```
@Override
```

```
public boolean validacion(Credito objeto) throws Exception {
```

```
    return true;
```

```
}
```

```
}
```

### Clase controlador Persona

```
package ec.ups.edu.controlador;
```

```
import ec.ups.edu.modelo.Persona;
import ec.ups.edu.utils.UtillsJPA;
import static ec.ups.edu.utils.UtillsJPA.getEntityManager;
import ec.ups.edu.validaciones.ExcepcionCedula;
import javax.persistence.EntityManager;
import javax.persistence.NoResultException;
import javax.swing.JOptionPane;

/**
 *
 * @author santi
 */
public class ControladorPersona extends AbstractControlador<Persona> {

    Persona persona;
    UtillsJPA utilJPA = new UtillsJPA();

    //Empieza Singleton
    private static ControladorPersona instancia = new ControladorPersona();

    public static ControladorPersona getControladorPersona() {

        if (instancia == null) {

            instancia = new ControladorPersona();

        }

        return instancia;

    }

    //Fin Singleton.
    @Override
    public boolean validacion(Persona objeto) throws ExcepcionCedula {

        int total = 0;

        String cedula[] = objeto.getCedula().split("");

        if (cedula.length == 10) {

            for (int i = 0; i < cedula.length; i++) {

                if (i % 2 == 0) {

                    total += Integer.parseInt(cedula[i]) * 2;

                    if (Integer.parseInt(cedula[i]) * 2 > 9) {

                        total -= 9;

                    }

                }

            }

        }

    }

}
```

```
        } else {

            total += Integer.parseInt(cedula[i]);

        }

        if (total == 10 && cedula[cedula.length - 1].equals("9")) {

            return true;

        } else {

            total = 10 - total % 10;

            if (cedula[cedula.length - 1].equals("" + total)) {

                return true;

            }

        }

    }

    throw new ExcepcionCedula();

}

JOptionPane.showMessageDialog(null, "La cedula es incorrecta");

return false;

}

public boolean iniciarSesion(String correo, String contrasena) {

    System.out.println(correo);
    System.out.println(contrasena);

    EntityManager em = getEntityManager();

    try {

        String jpql = "Select p from Persona p where p.correo='" + correo + "'";
        Persona persona = (Persona) em.createQuery(jpql).getSingleResult();

        System.out.println(persona);

        if (persona.getContrasena().equals(contrasena)) {

            return true;

            //return persona;

        }

    }
```



```
        //return persona;
    } catch (NoResultException e) {

        System.out.println("No ha sido posible iniciar Sesión");
        return false;

    }

    return false;
}

public Persona buscar(String cedula) {

    EntityManager em = getEntityManager();

    try {

        String jpql = "Select p from Persona p where p.cedula='"+cedula+"'";
        Persona p = (Persona) em.createQuery(jpql).getSingleResult();

        return p;

    } catch (NoResultException e) {


        System.out.println("Error " + e);

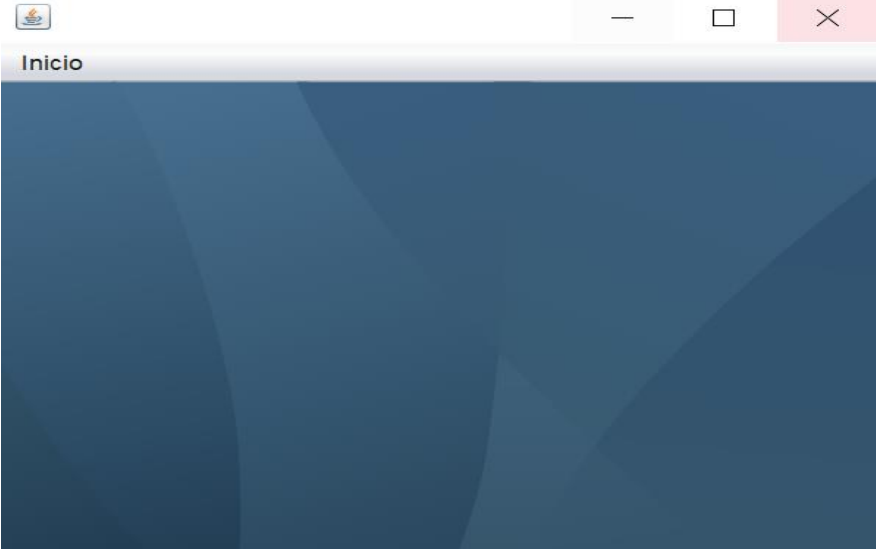
    }

    return null;
}
}
```

### 3. Interfaz grafica

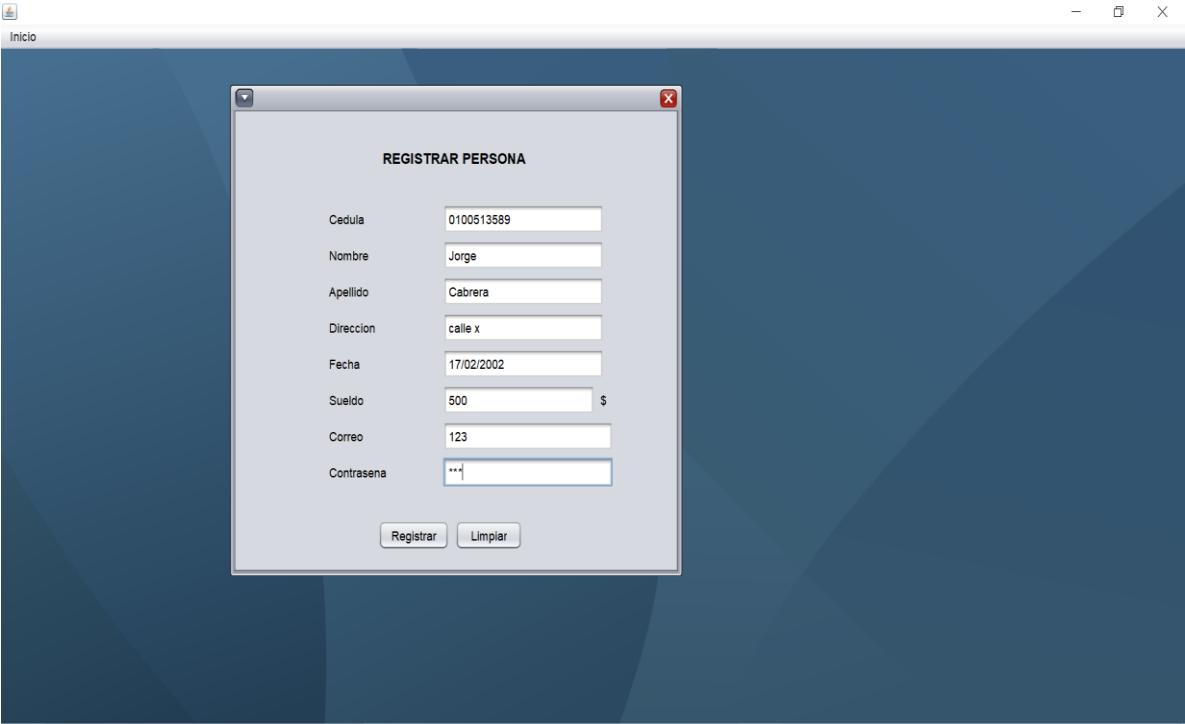
Creación de la clase Principal donde el usuario eligira una opción a realizar

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		




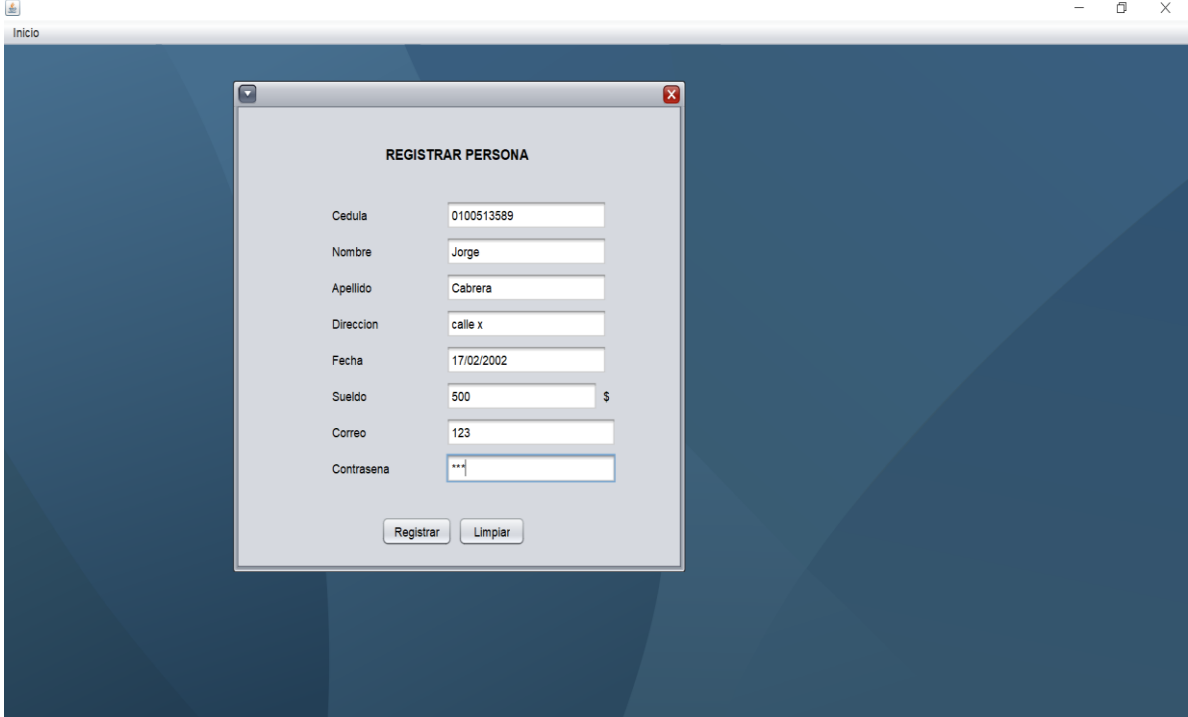
**Ventana Registrar Persona**

**Ingresara los datos de la persona**



**Al registrarse nos enviara un mensaje**


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Una ventana para iniciar Sesión donde compara los datos guardados en la base de datos



Una vez que el usuario haya iniciado sesión le dejara registrar casas y realizar las demás funcionalidades del programa

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



### Ventana para pedir prestamos

El usuario pedirá un préstamo y si el valor de la casa excede el monto que est pidiendo este nos pedirá un garante



Mes	Cuota	Interes	Amortizacion	Capital
0.0	0.0	0.0	0.0	4.0

Mostrar

### Tabla casa

SELECT \* FROM "public".ca... X

Max. rows: 100 | Fetched Rows: 1 | Matching Rows:

#	id	calleprimaria	callesecundaria	numeropisos	ubicacion
1		2102 Manuel Guillen	Antonio Delgado	2	Gualace

### Tabla credito

SELECT \* FROM "public".cr... X

Max. rows: 100 | Fetched Rows: 5 | Matching Rows:


#	id	fechafin	fechainicio	fechaprestamo	monto
1		2103 2024-07-01 00:00:00.000	2023-07-01 00:00:00.000	2021-01-02 00:00:00.000	
2		2151 2024-07-01 00:00:00.000	2023-07-01 00:00:00.000	2021-01-02 00:00:00.000	
3		2201 2024-07-01 00:00:00.000	2023-07-01 00:00:00.000	2021-01-02 00:00:00.000	

### Tabla persona

SELECT \* FROM "public".pe... X

Max. rows: 100 | Fetched Rows: 2 | Matching Rows:

#	id	apellido	cedula	contrasena	correo
1		2101 cabrera	0100513589	123	123

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

### RESULTADO(S) OBTENIDO(S):

Utilización de JPA utilizando patrones de diseño como Singleton y utilizando MVC.

### CONCLUSIONES:

Como conclusión podemos decir que aplicar JPA nos facilita y nos ayuda a ahorrar código y no es difícil de aplicarlo, utilizar esto nos ayuda para trabajar con bases de datos

### RECOMENDACIONES:

Recomiendo ver el material otorgado por el ingeniero puesto que hay se encuentra una gran variedad de información que nos ayudaran a trabajar en esta práctica.

**Nombre de estudiante:** Jorge Santiago Cabrera Arias

**Firma de estudiante:**



Escaneado con CamScanner