



Prueba 3

29/01/2021

Objetivo:

- Consolidar los conocimientos adquiridos en clase sobre JPA.

Enunciado:

Realizar un sistema implementando todos los conceptos vistos en clases para gestionar la hipoteca de las casas con las siguientes características:

- Las personas compran casas y se convierten en propietarios.
- Para pagarlas es habitual que el propietario formalice un préstamo hipotecario con una entidad bancaria.
- El banco toma la casa en forma de aval en caso de impago de las mensualidades.
- En el caso de que el capital fiado supera el valor de tasación de la casa y el sueldo del propietario no es suficiente, el banco suele exigir la presencia de un avalista (garante).
- Para formalizar la hipoteca se necesitan los datos personales del propietario, además de su cédula, dirección de la casa, su dirección, nombres, apellidos y fecha de nacimiento y del garante de ser necesario.
- El capital de la hipoteca se ajusta teniendo en cuenta el valor de tasación de la casa y los datos de dirección.
- Toda hipoteca se formaliza detallando el capital, el interés (8,99 - 16,99%) y la duración (fecha de inicio y fecha de fin).
- A partir de estos datos se calcula el importe de cada mensualidad para el total del tiempo que pide el préstamo.
- No es necesario guardar los datos del banco pero sí un sistema de autenticación.
- Generar los datos con el sistema de amortización Alemán [1].

**Ejemplo** simulador de crédito para guía: <https://www.pichincha.com/portal/simuladores/simulador-de-creditos>

Se calificará de la siguiente forma:

- JPA: 40%
- Excepciones: 10%
- MVC: 20%
- Diagrama de clases: 10%
- Usabilidad - Vista: 15%
- Programación genérica, Java 8, reflexión: 15%

**Bibliografía**

- [1] <https://www.produbanco.com.ec/banca-minorista/cr%C3%A9ditos/hipotecario/simulador-de-cr%C3%A9dito-hipotecario/>

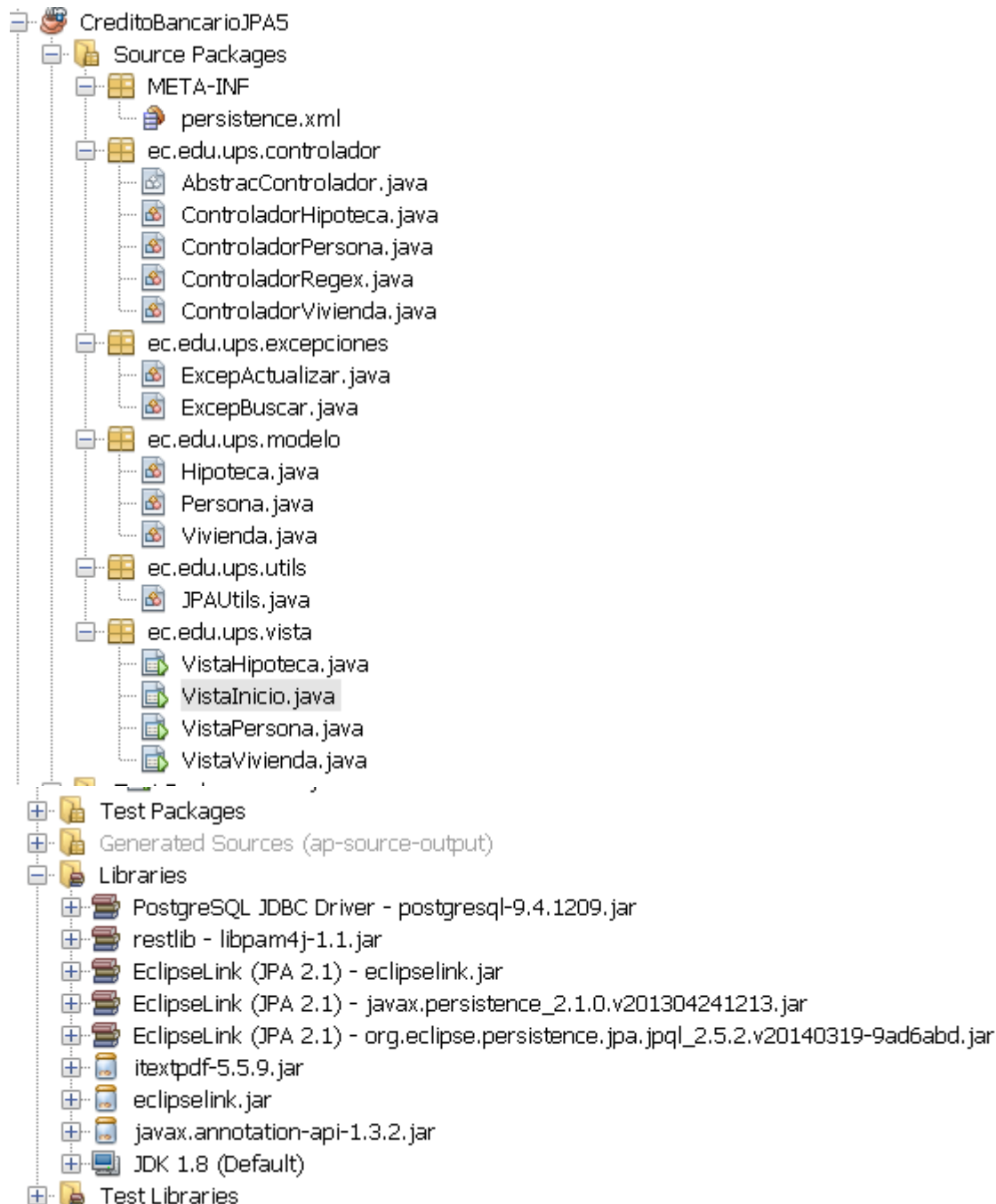
**Entrega:** Subir al Git el documento en formato PDF de los resultados y código hasta las 23:55 del



**Prueba 3**

29/01/2021

domingo 31 de Enero del 2021.





**Prueba 3**

29/01/2021



ID:

CEDULA:

NOMBRE:

APELLIDO:

FECHA NACIMIENTO:   
DIA / MES / AÑO

SALARIO:

TEEFONO

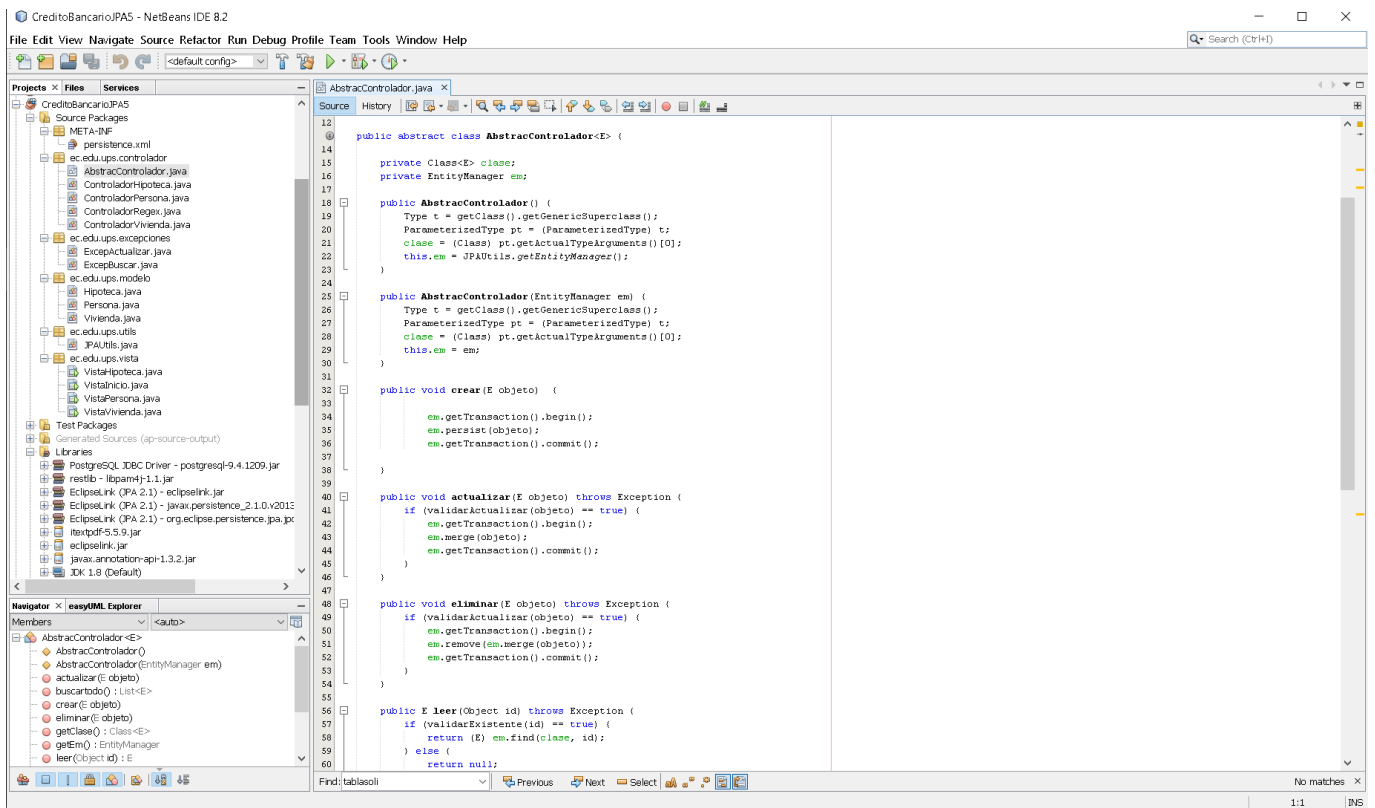
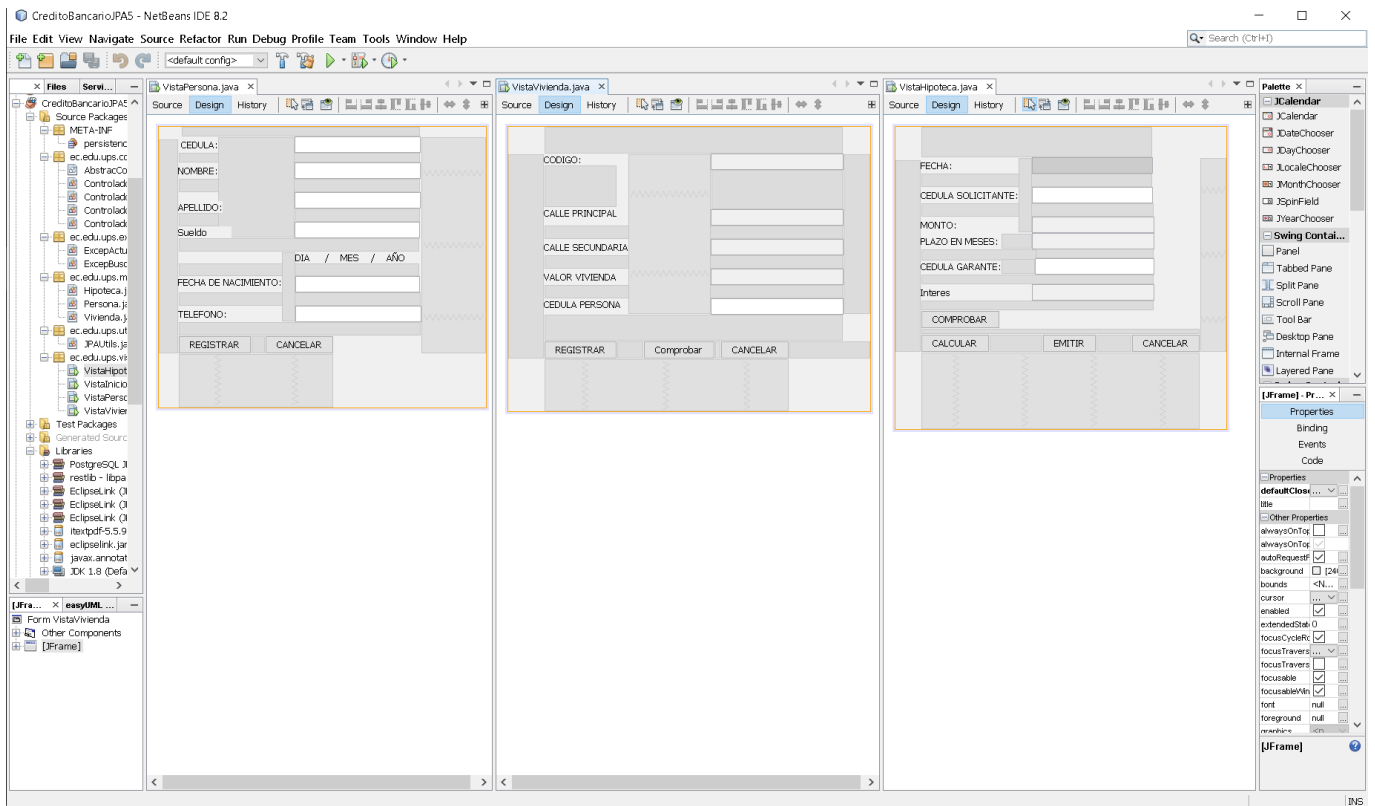


# Programación Aplicada

## Tema: Base de Datos Java.

### Prueba 3

29/01/2021





# Programación Aplicada

## Tema: Base de Datos Java.

### Prueba 3

29/01/2021

CreditoBancarioJPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

AbstracControlador.java

```
40 public void actualizar(E objeto) throws Exception {
41     if (validarActualizar(objeto) == true) {
42         em.getTransaction().begin();
43         em.merge(objeto);
44         em.getTransaction().commit();
45     }
46 }
47
48 public void eliminar(E objeto) throws Exception {
49     if (validarEliminar(objeto) == true) {
50         em.getTransaction().begin();
51         em.remove(em.merge(objeto));
52         em.getTransaction().commit();
53     }
54 }
55
56 public E leer(Object id) throws Exception {
57     if (validarExistente(id) == true) {
58         return (E) em.find(class, id);
59     } else {
60         return null;
61     }
62 }
63
64 public List<E> buscarTodo() throws Exception {
65     return em.createQuery("Select t from " + class.getSimpleName() + " t").getResultList();
66 }
67
68 public abstract boolean validarActualizar(E objeto) throws Exception;
69
70 public abstract boolean validarExistente(Object id) throws Exception;
71
72 public Class<E> getClass() {
73     return class;
74 }
75
76 public void setClass(Class<E> class) {
77     this.class = class;
78 }
79
80 public EntityManager getEm() {
81     return em;
82 }
83
84 public void setEm(EntityManager em) {
85     this.em = em;
86 }
87
88 }
```

Find: tablasoli

No matches

32:35

CreditoBancarioJPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

AbstracControlador.java | ControladorPersona.java

```
8 /**
9  *
10  * @author japch
11  */
12 public class ControladorPersona extends AbstracControlador<Persona> {
13
14     Persona persona = new Persona();
15
16     @Override
17     public boolean validarActualizar(Persona objeto) throws ExcepcActualizar {
18         String queryString = "Select * from persona as p "
19             + "where p.cedula like ?";
20         Query query = getEm().createNativeQuery(queryString, Persona.class);
21         List<Persona> persona = query.setParameter(1, objeto.getCedula()).getResultList();
22         if (persona.size() == 1) {
23             return true;
24         } else {
25             return false;
26         }
27     }
28
29     @Override
30     public boolean validarExistente(Object id) throws ExcepcBuscar {
31         String queryString = "Select * from persona as p "
32             + "where p.cedula like ?";
33         Query query = getEm().createNativeQuery(queryString, Persona.class);
34         List<Persona> persona = query.setParameter(1, id).getResultList();
35         if (persona.size() == 1) {
36             return true;
37         } else {
38             return false;
39         }
40     }
41
42     public boolean consultacedula(String cedula) {
43
44         Query consulta = getEm().createNamedQuery("consultaCedula");
45         consulta.setParameter("cedula", cedula);
46         return true;
47     }
48
49 }
50
51 }
```

Find: tablasoli

No matches

1:1



# Programación Aplicada

## Tema: Base de Datos Java.

### Prueba 3

29/01/2021

CreditoBancarioJPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

Source Packages

- CreditoBancarioJPAS
  - META-INF
    - persistence.xml
  - ec.edu.ups.controlador
    - AbstractControlador.java
    - ControladorHipoteca.java
    - ControladorPersona.java
    - ControladorRegeu.java
    - ControladorVivienda.java
  - ec.edu.ups.excepciones
    - ExcepActualizar.java
    - ExcepBuscar.java
  - ec.edu.ups.modelo
    - Hipoteca.java
    - Persona.java
    - Vivienda.java
  - ec.edu.ups.utilis
    - JPautis.java
    - VistaHipoteca.java
    - VistaInicio.java
    - VistaPersona.java
    - VistaVivienda.java
  - Test Packages
  - Generated Sources (ap-source-output)
  - Libraries
    - PostgreSQL JDBC Driver - postgresql-9.4.1209.jar
    - restlib - libpam-1.1.jar
    - EclipseLink (PA 2.1) - eclipseLink.jar
    - EclipseLink (PA 2.1) - javax.persistence\_2.1.0.v2013.jar
    - EclipseLink (PA 2.1) - org.eclipse.persistence.jpa.jar
    - hibernate-5.5.9.jar
    - eclipseLink.jar
    - javax.annotation-api-1.3.2.jar
    - JDK 1.8 (Default)

cedula - Navigator | easyXML Explorer

Members

- setCasas(List<Vivienda> casas)
- setCedula(String cedula)
- setFechaDeNacimiento(String fechaDeNacimiento)
- setNombre(String nombre)
- setSalarioMensual(double salarioMensual)
- setTelefono(String telefono)
- toString() : String
- Object
- apellido : String
- casas : List<Vivienda>
- cedula : String

Find: tablasoli

No matches

```
15 * @author jaspch
16 */
17 @Entity
18 @NamedQuery(name = "consultaCedula", query = "Select p from Persona p where p.cedula = :cedula")
19
20 public class Persona implements Serializable {
21
22     @Id
23     @Column
24     private String cedula;
25
26     @Column
27     private String nombre;
28
29     @Column
30     private String apellido;
31
32     @Column
33     private String telefono;
34
35     @Column
36     private String fechaDeNacimiento;
37
38     @Column
39     private double salarioMensual;
40
41     @OneToMany(mappedBy = "persona", cascade = CascadeType.ALL)
42     private List<Vivienda> casas;
43
44     public List<Vivienda> getCasas() {
45         return casas;
46     }
47
48     public void setCasas(List<Vivienda> casas) {
49         this.casas = casas;
50     }
51
52     public String getCedula() {
53         return cedula;
54     }
55
56     public void setCedula(String cedula) {
57         this.cedula = cedula;
58     }
59
60     public String getNombre() {
61         return nombre;
62     }
63
64     public void setNombre(String nombre) {
65         this.nombre = nombre;
66     }
67
68     public String getApellido() {
69         return apellido;
70     }
71
72     public void setApellido(String apellido) {
73         this.apellido = apellido;
74     }
75
76     public String getTelefono() {
77         return telefono;
78     }
79
80     public void setTelefono(String telefono) {
81         this.telefono = telefono;
82     }
83
84     public String getFechaDeNacimiento() {
85         return fechaDeNacimiento;
86     }
87
88     public void setFechaDeNacimiento(String fechaDeNacimiento) {
89         this.fechaDeNacimiento = fechaDeNacimiento;
90     }
91
92     public double getSalarioMensual() {
93         return salarioMensual;
94     }
95
96     public void setSalarioMensual(double salarioMensual) {
97         this.salarioMensual = salarioMensual;
98     }
99
100     public String toString() {
101         return "Persona [cedula=" + cedula + ", nombre=" + nombre + ", apellido=" + apellido + ", telefono=" + telefono + ", fechaDeNacimiento=" + fechaDeNacimiento + ", salarioMensual=" + salarioMensual + "]";
102     }
103
104     public Object[] toArray() {
105         return new Object[] { cedula, nombre, apellido, telefono, fechaDeNacimiento, salarioMensual };
106     }
107
108     public void toArray(Object[] array) {
109         array[0] = cedula;
110         array[1] = nombre;
111         array[2] = apellido;
112         array[3] = telefono;
113         array[4] = fechaDeNacimiento;
114         array[5] = salarioMensual;
115     }
116 }
```

CreditoBancarioJPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

Source Packages

- CreditoBancarioJPAS
  - META-INF
    - persistence.xml
  - ec.edu.ups.controlador
    - AbstractControlador.java
    - ControladorHipoteca.java
    - ControladorPersona.java
    - ControladorRegeu.java
    - ControladorVivienda.java
  - ec.edu.ups.excepciones
    - ExcepActualizar.java
    - ExcepBuscar.java
  - ec.edu.ups.modelo
    - Hipoteca.java
    - Persona.java
    - Vivienda.java
  - ec.edu.ups.utilis
    - JPautis.java
    - VistaHipoteca.java
    - VistaInicio.java
    - VistaPersona.java
    - VistaVivienda.java
  - Test Packages
  - Generated Sources (ap-source-output)
  - Libraries
    - PostgreSQL JDBC Driver - postgresql-9.4.1209.jar
    - restlib - libpam-1.1.jar
    - EclipseLink (PA 2.1) - eclipseLink.jar
    - EclipseLink (PA 2.1) - javax.persistence\_2.1.0.v2013.jar
    - EclipseLink (PA 2.1) - org.eclipse.persistence.jpa.jar
    - hibernate-5.5.9.jar
    - eclipseLink.jar
    - javax.annotation-api-1.3.2.jar
    - JDK 1.8 (Default)

Navigator | easyXML Explorer

Members

- Vivienda : Serializable
- getCodigoCasa() : int
- getDireccionCalle1() : String
- getDireccionCalle2() : String
- getPersona() : Persona
- getValorDeCasa() : double
- setCodigoCasa(int codigoCasa)
- setDireccionCalle1(String direccionCalle1)
- setDireccionCalle2(String direccionCalle2)
- setPersona(Persona persona)

Find: tablasoli

No matches

```
15 * @author jaspch
16 */
17 @Entity
18 public class Vivienda implements Serializable {
19
20     @Id
21     @Column
22     private int codigoCasa;
23
24     @Column
25     private String direccionCalle1;
26     @Column
27     private String direccionCalle2;
28     @Column
29     private double valorDeCasa;
30
31     @ManyToOne
32     @JoinColumn(name = "fk_persona")
33     private Persona persona;
34
35     public int getCodigoCasa() {
36         return codigoCasa;
37     }
38
39     public void setCodigoCasa(int codigoCasa) {
40         this.codigoCasa = codigoCasa;
41     }
42
43     public String getDireccionCalle1() {
44         return direccionCalle1;
45     }
46
47     public void setDireccionCalle1(String direccionCalle1) {
48         this.direccionCalle1 = direccionCalle1;
49     }
50
51     public String getDireccionCalle2() {
52         return direccionCalle2;
53     }
54
55     public void setDireccionCalle2(String direccionCalle2) {
56         this.direccionCalle2 = direccionCalle2;
57     }
58
59     public double getValorDeCasa() {
60         return valorDeCasa;
61     }
62
63     public void setValorDeCasa(double valorDeCasa) {
64         this.valorDeCasa = valorDeCasa;
65     }
66
67     public Persona getPersona() {
68         return persona;
69     }
70
71     public void setPersona(Persona persona) {
72         this.persona = persona;
73     }
74 }
```



# Programación Aplicada

## Tema: Base de Datos Java.

### Prueba 3

29/01/2021

CreditoBancarioJPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

Source Packages

- CreditoBancarioJPAS
  - META-INF
  - persistencia.xml
  - ec.edu.ups.controlador
    - AbstractControlador.java
    - ControladorHipoteca.java
    - ControladorPersona.java
    - ControladorRegex.java
    - ControladorVivienda.java
  - ec.edu.ups.excepciones
    - ExcepActualizar.java
    - ExcepBuscar.java
  - ec.edu.ups.modelo
    - Hipoteca.java
    - Persona.java
    - Vivienda.java
  - ec.edu.ups.utils
    - JPUtils.java
  - ec.edu.ups.vista
    - VistaHipoteca.java
    - VistaInicio.java
    - VistaPersona.java
    - VistaVivienda.java
- Test Packages
- Generated Sources (ap-source-output)
- Libraries
  - PostgreSQL JDBC Driver - postgresql-9.4.1209.jar
  - restlib - libpam4-1.1.jar
  - EclipseLink (PA 2.1) - eclipseLink.jar
  - EclipseLink (PA 2.1) - javax.persistence\_2.1.0.v2013.jar
  - EclipseLink (PA 2.1) - org.eclipse.persistence.jpa.jar
  - restlib-5.5.9.jar
  - ecipseLink.jar
  - javax.annotation-api-1.3.2.jar
  - JDK 1.8 (Default)

Members

VistaPersona :: JFrame

- VistaPersona()
- bttCancelarActionPerformed(ActionEvent evt)
- bttRegistrarActionPerformed(ActionEvent evt)
- crearPersona()
- initComponents()
- limpiar()
- main(String[] args)
- bttCancelar : JButton
- bttRegistrar : JButton

Find: tablasoli

```
15 import java.util.List;
16 import javax.swing.JOptionPane;
17
18 /**
19  *
20  * @author japho
21  */
22 public class VistaPersona extends javax.swing.JFrame {
23     private ControladorVivienda cv = new ControladorVivienda();
24     private ControladorHipoteca ch = new ControladorHipoteca();
25     private ControladorPersona cp = new ControladorPersona();
26     private ControladorRegex cr = new ControladorRegex();
27
28     /**
29      * Creates new form VistaPersonas
30      */
31     public VistaPersona() {
32         initComponents();
33     }
34
35     public void crearPersona() {
36         if (!txtCedula.getText().isEmpty() && !txtNombre.getText().isEmpty() && !txtApellido.getText().isEmpty()
37             && !txtTelefono.getText().isEmpty() && !txtCausaIdo.getText().isEmpty() && !txtFechaNac.getText().isEmpty()) {
38
39             cr.IngreseRegex("(\\d{9})-?(\\d{1})?");
40             boolean validardedula = cr.validar(txtCedula.getText());
41             if (validardedula) {
42                 try {
43                     Persona p = new Persona();
44                     List<Vivienda> casas = new ArrayList<>();
45                     p.setCedula(txtCedula.getText());
46                     p.setNombre(txtNombre.getText());
47                     p.setApellido(txtApellido.getText());
48                     p.setTelefono(txtTelefono.getText());
49                     p.setFechaNacimiento(txtFechaNac.getText());
50                     p.setSalarioAnual(Double.parseDouble(txtCausaIdo.getText()));
51                     p.setCasas(casas);
52                     cp.crear(p);
53                     JOptionPane.showMessageDialog(null, "Persona registrada");
54
55                     limpiar();
56                 } catch (Exception ex) {
57                     ex.printStackTrace();
58                 }
59             } else {
60                 JOptionPane.showMessageDialog(null, "Cedula invalida");
61             }
62         } else {
63             JOptionPane.showMessageDialog(null, "Llenar todos los campos");
64         }
65     }
66 }
```

CreditoBancarioJPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

Source Packages

- CreditoBancarioJPAS
  - META-INF
  - persistencia.xml
  - ec.edu.ups.controlador
    - AbstractControlador.java
    - ControladorHipoteca.java
    - ControladorPersona.java
    - ControladorRegex.java
    - ControladorVivienda.java
  - ec.edu.ups.excepciones
    - ExcepActualizar.java
    - ExcepBuscar.java
  - ec.edu.ups.modelo
    - Hipoteca.java
    - Persona.java
    - Vivienda.java
  - ec.edu.ups.utils
    - JPUtils.java
  - ec.edu.ups.vista
    - VistaHipoteca.java
    - VistaInicio.java
    - VistaPersona.java
    - VistaVivienda.java
- Test Packages
- Generated Sources (ap-source-output)
- Libraries
  - PostgreSQL JDBC Driver - postgresql-9.4.1209.jar
  - restlib - libpam4-1.1.jar
  - EclipseLink (PA 2.1) - eclipseLink.jar
  - EclipseLink (PA 2.1) - javax.persistence\_2.1.0.v2013.jar
  - EclipseLink (PA 2.1) - org.eclipse.persistence.jpa.jar
  - restlib-5.5.9.jar
  - ecipseLink.jar
  - javax.annotation-api-1.3.2.jar
  - JDK 1.8 (Default)

Members

VistaHipoteca :: JFrame

- VistaHipoteca()
- bttCalcularActionPerformed(ActionEvent evt)
- bttCancelarActionPerformed(ActionEvent evt)
- bttEmitirActionPerformed(ActionEvent evt)
- bttComprobarActionPerformed(ActionEvent evt)
- comprobarCedulas()
- deshabilitarCajones()
- emitirRegistro()
- habilitarCajones()

Find: tablasoli

```
50 }
51
52 public void habilitarCajones() {
53     txtMontto.setEditable(true);
54     txtPlazo.setEditable(true);
55     txtCedulaGarante.setEditable(true);
56     txtInteres.setEditable(true);
57 }
58
59 public void deshabilitarCajones() {
60     txtMontto.setEditable(false);
61     txtPlazo.setEditable(false);
62     txtCedulaGarante.setEditable(false);
63 }
64
65 public void comprobarCedulas() {
66     try {
67         if (controladorP.leer(txtCedulaGarante.getText()) != null) {
68             if (controladorP.leer(txtCedula.getText()) != null) {
69                 JOptionPane.showMessageDialog(null, "Las personas estan registradas");
70                 habilitarCajones();
71                 btnComprobar.setEnabled(false);
72                 btnCalcular.setEnabled(true);
73                 btnEmitir.setEnabled(true);
74             }
75         } else {
76             JOptionPane.showMessageDialog(null, "cedula del solicitante no existe ");
77         }
78     } else {
79         JOptionPane.showMessageDialog(null, "cedula del garante no existe ");
80     }
81 } catch (Exception ex) {
82     Logger.getLogger(VistaHipoteca.class.getName()).log(Level.SEVERE, null, ex);
83 }
84
85 }
86
87 /**
88  * This method is called from within the constructor to initialize the form.
89  * WARNING: Do NOT modify this code. The content of this method is always
90  * regenerated by the Form Editor.
91  */
92 @SuppressWarnings("unchecked")
93 Generated Code
94
95 private void txtCedulaGaranteKeyTyped(java.awt.event.KeyEvent evt) {
96     char type = evt.getKeyChar();
97     if (type == '\n') {
98         comprobarCedulas();
99     }
100 }
```



# Programación Aplicada

## Tema: Base de Datos Java.

### Prueba 3

29/01/2021

CreditoBancarioPAS - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

Projects | Files | Services

...ave | ControladorPersona.java | Persona.java | Vivienda.java | Hipoteca.java | VistaPersona.java | VistaVivienda.java | VistaInicio.java | VistaHipoteca.java

Source | Design | History

```
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\CreditoBancarioPAS
            document.open();

            Paragraph into = new Paragraph("Registro de pagos de la Hipoteca ");
            Paragraph space = new Paragraph(" ");

            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);
        }
    }
}
```

emitterregistro - Navigator | easyXML Explorer

Members

VistaHipoteca :: JFrame

- btrCalcularActionPerformed(ActionEvent evt)
- btrCancelarActionPerformed(ActionEvent evt)
- btrEntrarActionPerformed(ActionEvent evt)
- btrComprobarActionPerformed(ActionEvent evt)
- comprobarCedulas()
- deshabilitarCajones()
- emitirregistro()
- habilitarCajones()

Find: Previous Next Select

434:1

Montos.pdf

Archivo | D:\Metodos%20Numericos\CreditoBancarioPAS\Montos.pdf

Google WhatsApp youtube Facebook New Tab Instagram Chris Japon (@Chri... Proyecto Integrador... Anime online HD... AnimeYT2 - Anime... Ver Anime Online ... Otros favoritos

1 de 2

Vista de página | Lectura en voz alta | Dibujar | Resaltar | Borrar

### Registro de pagos de la Hipoteca

CEDULA SOLICITANTE	NOMBRE SOLICITANTE	APELLIDO SOLICITANTE	TELEFONO SOLICITANTE
0107480840	Christian	Japon	0967389328

CEDULA GARANTE	NOMBRE GARANTE	APELLIDO GARANTE	TELEFONO GARANTE
0107480845	Paola	Plaza	0965986547

MES	CAPITAL	SALDO	INTERES	TOTAL
1	520.83	25.000.00	354.17	875.00
2	520.83	24.479.17	346.79	867.62
3	520.83	23.958.33	339.41	860.24
4	520.83	23.437.50	332.03	852.86
5	520.83	22.916.67	324.65	845.49
6	520.83	22.395.83	317.27	838.11
7	520.83	21.875.00	309.90	830.73
8	520.83	21.354.17	302.52	823.35
9	520.83	20.833.33	295.14	815.97
10	520.83	20.312.50	287.76	808.59
11	520.83	19.791.67	280.38	801.22
12	520.83	19.270.83	273.00	793.84
13	520.83	18.750.00	265.63	786.46
14	520.83	18.229.17	258.25	779.08
15	520.83	17.708.33	250.87	771.70
16	520.83	17.187.50	243.49	764.32
17	520.83	16.666.67	236.11	756.94
18	520.83	16.145.83	228.73	749.57
19	520.83	15.625.00	221.35	742.19
20	520.83	15.104.17	213.98	734.81
21	520.83	14.583.33	206.60	727.43





# Programación Aplicada

## Tema: Base de Datos Java.

### Prueba 3

29/01/2021

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+H)

SQL 1 [jdbc:postgresql://localhost:5432/bdpruebaPA [postgres on public]

```
1 SELECT * FROM "public"."persona" LIMIT 100;
```

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

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

Max. rows: 100 | Fetched Rows: 3 | Matching Rows:

#	id	apellido	cedula	fecha_nacimiento	nombre	salario
1	401	Cjhtewr	01021545	12/10/1999	Cjht	
2	451	Japon	0107480940	20/10/1999	Christian	
3	501	Perez	01074809401	02/11/1995	Andres	

Output x | Vivienda.java x | ControladorVivienda.java x | VentanaPrincipal.java x | RegistrarPersona.java x | VistaVivienda.java x

CreditoBancario (run) x | SQL 1 execution x

Members | VistaPersona :: JFrame

- VistaPersona()
- clean()
- crearPersona()
- initComponents()
- button1ActionPerformed(ActionEvent)
- button2ActionPerformed(ActionEvent)
- main(String[] args)
- bttelefonoActionPerformed(ActionEvent)

Executed successfully in 0,003 s.  
Fetching resultset took 0,003 s.  
Line 1, column 1

Execution finished after 0,326 s, no errors occurred.

Source | History

```
85 }
86
87
88 public void
89 this
90
91
92 public String
93 retu
94
95
96 public void
97 this
98
99
100 public double
101 retu
102
103
104 public void
105 this
106
107
108 public Person
109 retu
110
111
112 public void
113 this
114
115
116
117
```

1:1 | DNS