

SEMANA 2 – TALLER DE DESARROLLO DE APLICACIONES 1

ENTREGADO POR: Mendoza Ricse Gimena Cristel

CÓDIGO: Q00076H

CATEDRÁTICO: Fernández Bejarano Raúl Enrique

EJEMPLO 1:

ENUNCIADO:

En una universidad de Lima, los alumnos están clasificados en cuatro categorías. A cada categoría le corresponde una pensión mensual distinta, según la siguiente tabla:

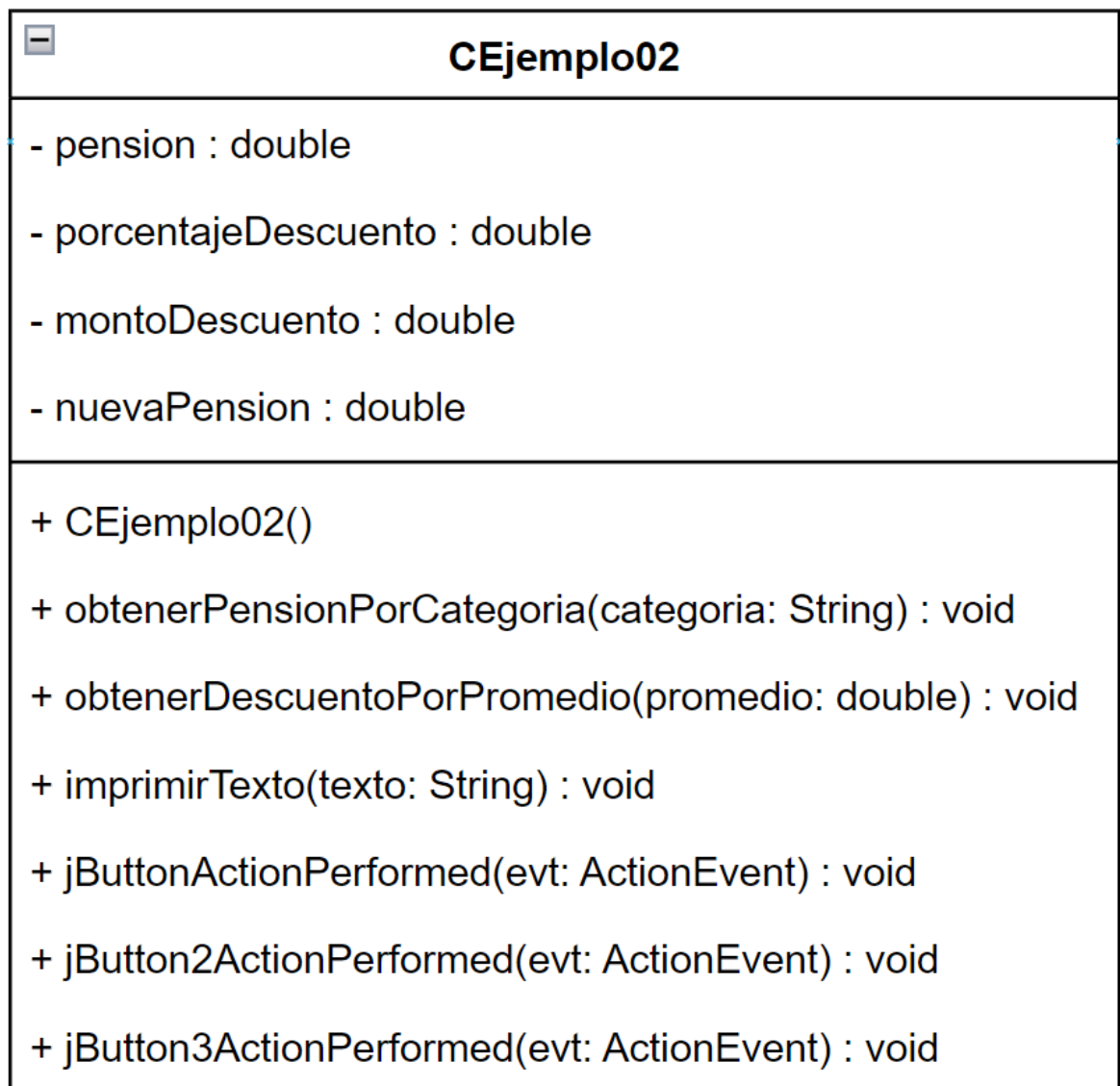
Categoría	Pensión
A	S/. 550
B	S/. 500
C	S/. 460
D	S/. 400

Semestralmente, la universidad efectúa rebajas en las pensiones de sus estudiantes a partir del segundo ciclo, basándose en el promedio ponderado del ciclo anterior. El porcentaje de descuento está determinado según la siguiente tabla:

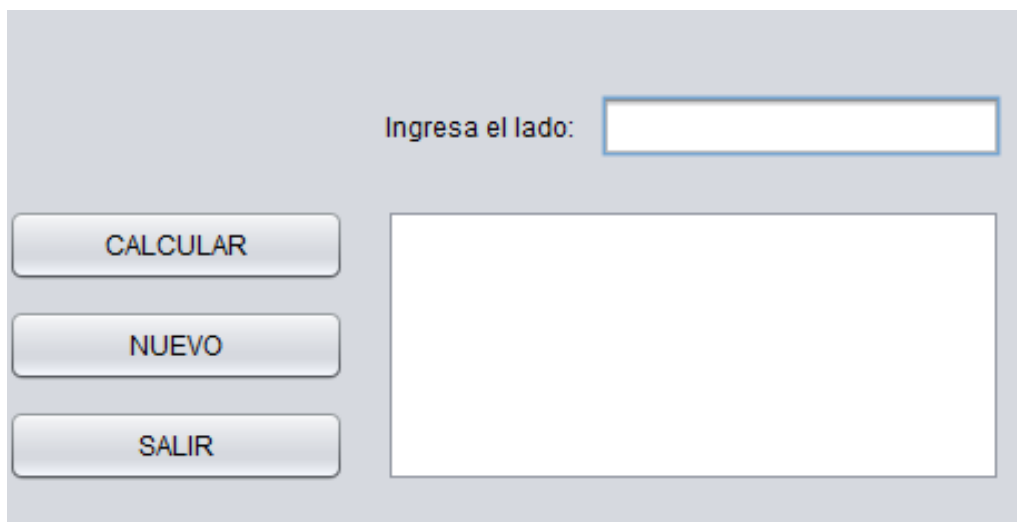
Promedio	Descuento
00 a 13.99	No hay descuento
14.00 a 15.99	10 %
16.00 a 17.99	12 %
18.00 a 20.00	15 %

Dado el promedio ponderado y la categoría de un estudiante, diseñe una aplicación que determine cuánto de rebaja recibirá sobre su pensión actual y a cuánto asciende su nueva pensión.

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:



The application design mockup features a light gray background. At the top right, the text "Ingresa el lado:" is followed by a white text input field with a blue border. On the left side, there are three stacked buttons with rounded corners and a light gray gradient: "CALCULAR", "NUEVO", and "SALIR". To the right of these buttons is a large, empty white rectangular area with a thin gray border.

CÓDIGO DE LA APLICACIÓN:

```
13 public class CEjemplo02 extends javax.swing.JFrame {
14
15     /**
16      * Creates new form CEjemplo02
17      */
18     double pension = 0;
19     double porcentajeDescuento = 0;
20     double montoDescuento = 0;
21     double nuevaPension = 0;
22
23     public CEjemplo02() {
24         initComponents();
25     }
26
27     private void imprimirTexto(String texto) {
28         txtResultado.append(texto + "\n");
29     }
30
31     private void obtenerPensionPorCategoria(String categoria) {
32         switch (categoria) {
33             case "A": {
34                 pension = 550;
35                 break;
36             }
37             case "B": {
38                 pension = 500;
39                 break;
40             }
41             case "C": {
42                 pension = 460;
43                 break;
44             }
45             case "D": {
46                 pension = 400;
47                 break;
48             }
49             default: {
50                 pension = 0;
51                 break;
52             }
53         }
54     }
55 }
```

```

56 private void obtenerDescuentoPorPromedio(double promedio) {
57     if (promedio >= 0 && promedio <= 13.99) {
58         porcentajeDescuento = 0;
59     } else if (promedio >= 14 && promedio <= 15.99) {
60         porcentajeDescuento = 10;
61     } else if (promedio >= 16 && promedio <= 17.99) {
62         porcentajeDescuento = 12;
63     } else if (promedio >= 18 && promedio <= 20) {
64         porcentajeDescuento = 15;
65     } else {
66         porcentajeDescuento = 0;
67         JOptionPane.showMessageDialog(this, "Promedio fuera de rango (0 a 20).", "Advertencia", JOptionPane.WARNING_MESSAGE);
68     }
69 }
70
71 /**
72  * This method is called from within the constructor to initialize the form.
73  * WARNING: Do NOT modify this code. The content of this method is always
74  * regenerated by the Form Editor.
75  */
76 @SuppressWarnings("unchecked")
77 Generated Code
78
79 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
80     System.exit(0);
81 }
82
83 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
84     cmbCategoria.setSelectedIndex(0);
85     txtPromedio.setText("");
86     txtResultado.setText("");
87 }

```

```

181 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
182     try {
183         String categoria = cmbCategoria.getSelectedItem().toString();
184         String promedioTexto = txtPromedio.getText();
185
186         if (promedioTexto.isEmpty()) {
187             JOptionPane.showMessageDialog(this, "Por favor ingresa el promedio.", "Error", JOptionPane.ERROR_MESSAGE);
188             return;
189         }
190
191         double promedio = Double.parseDouble(promedioTexto);
192
193         obtenerPensionPorCategoria(categoria);
194         obtenerDescuentoPorPromedio(promedio);
195
196         montoDescuento = pension * porcentajeDescuento / 100;
197         nuevaPension = pension - montoDescuento;
198
199         txtResultado.setText(""); // Limpiar
200         imprimirTexto("Categoria: " + categoria);
201         imprimirTexto("Promedio: " + promedio);
202         imprimirTexto("Pensión original: S/ " + pension);
203         imprimirTexto("Descuento aplicado: " + porcentajeDescuento + " %");
204         imprimirTexto("Monto de rebaja: S/ " + montoDescuento);
205         imprimirTexto("Nueva pensión: S/ " + nuevaPension);
206
207     } catch (NumberFormatException e) {
208         JOptionPane.showMessageDialog(this, "Por favor ingresa un promedio válido (número decimal).", "Error", JOptionPane.ERROR_MESSAGE);
209     }
210 }
211

```

ACTIVIDAD 1:

ENUNCIADO:

Una tienda ha puesto en oferta la venta de un producto ofreciendo un porcentaje de descuento sobre el importe de la compra, de acuerdo con la siguiente tabla:

Docenas adquiridas	Descuento
≥ 10	20%
< 10	10%

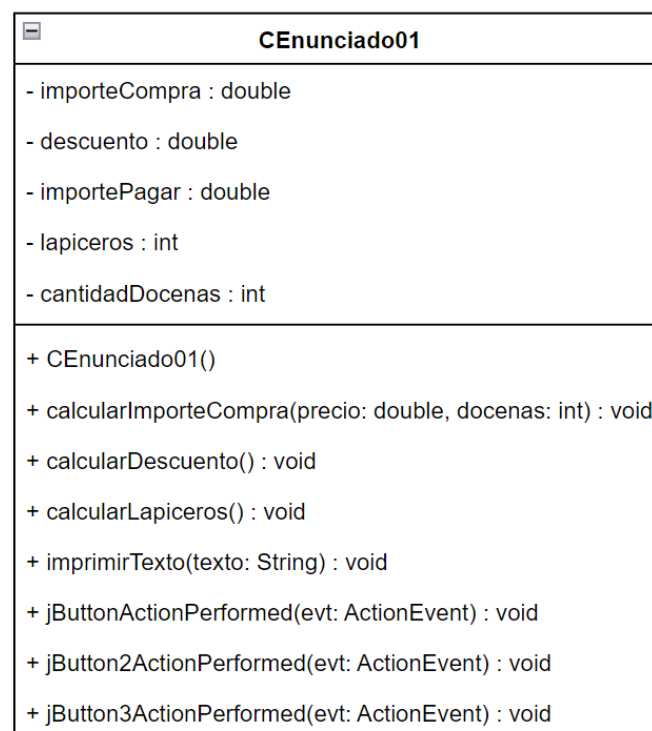
Adicionalmente, la tienda obsequia lapiceros de acuerdo con la siguiente tabla:

Importe a pagar	Lapiceros
≥ 200	2 por cada docena
< 200	0

Dado el precio de la docena y la cantidad de docenas adquiridas, diseñe un programa que determine:

- El **importe de la compra**
- El **importe del descuento**
- El **importe a pagar**
- La **cantidad de lapiceros de obsequio**

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:

The screenshot shows a Java Swing window with a light gray background. At the top right, there is a text label 'precio por docena:' followed by a text input field. Below it, on the left, is a text label 'cantidad por docena:' followed by another text input field. To the left of the input fields, there are three buttons stacked vertically: 'CALCULAR', 'NUEVO', and 'SALIR'. To the right of these buttons is a large, empty rectangular text area for displaying results.

CÓDIGO DE LA APLICACIÓN:

```
13 public class CEnunciado01 extends javax.swing.JFrame {
14
15     /**
16      * Creates new form CEnunciado01
17      */
18     public CEnunciado01() {
19         initComponents();
20     }
21
22     double importeCompra = 0;
23     double descuento = 0;
24     double importePagar = 0;
25     int lapiceros = 0;
26     int cantidadDocenas = 0;
27
28     private void imprimirTexto(String texto) {
29         txtResultado.append(texto + "\n");
30     }
31
32     private void calcularImporteCompra(double precio, int docenas) {
33         importeCompra = precio * docenas;
34     }
35
36     private void calcularDescuento() {
37         if (cantidadDocenas >= 10) {
38             descuento = importeCompra * 0.20;
39         } else {
40             descuento = importeCompra * 0.10;
41         }
42     }
43
44     private void calcularLapiceros() {
45         if (importePagar >= 200) {
46             lapiceros = 2 * cantidadDocenas;
47         } else {
48             lapiceros = 0;
49         }
50     }
```

```

155 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
156     System.exit(0);
157 }
158
159 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
160     txtPrecio.setText("");
161     txtDocenas.setText("");
162     txtResultado.setText("");
163 }
164
165 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
166     try {
167         double precioDocena = Double.parseDouble(txtPrecio.getText());
168         cantidadDocenas = Integer.parseInt(txtDocenas.getText());
169
170         calcularImporteCompra(precioDocena, cantidadDocenas);
171         calcularDescuento();
172         calcularLapiceros();
173
174         importePagar = importeCompra - descuento;
175
176         txtResultado.setText(""); // Limpiar área
177         imprimirTexto("Importe de la compra: S/ " + importeCompra);
178         imprimirTexto("Importe del descuento: S/ " + descuento);
179         imprimirTexto("Importe a pagar: S/ " + importePagar);
180         imprimirTexto("Lapiceros de obsequio: " + lapiceros);
181     } catch (NumberFormatException e) {
182         JOptionPane.showMessageDialog(this, "Por favor ingresa valores válidos", "Error", JOptionPane.ERROR_MESSAGE);
183     }
184 }

```

ACTIVIDAD 2:

ENUNCIADO:

El sueldo bruto de los empleados de una empresa se calcula sumando el sueldo básico más la bonificación por hijos. El sueldo básico se calcula multiplicando las horas trabajadas por la tarifa horaria. La tarifa horaria depende de la categoría del empleado de acuerdo con la siguiente tabla:

Categoría	Tarifa horaria (S/.)
A	45.0
B	37.5

La bonificación por hijos se calcula de acuerdo con la siguiente tabla:

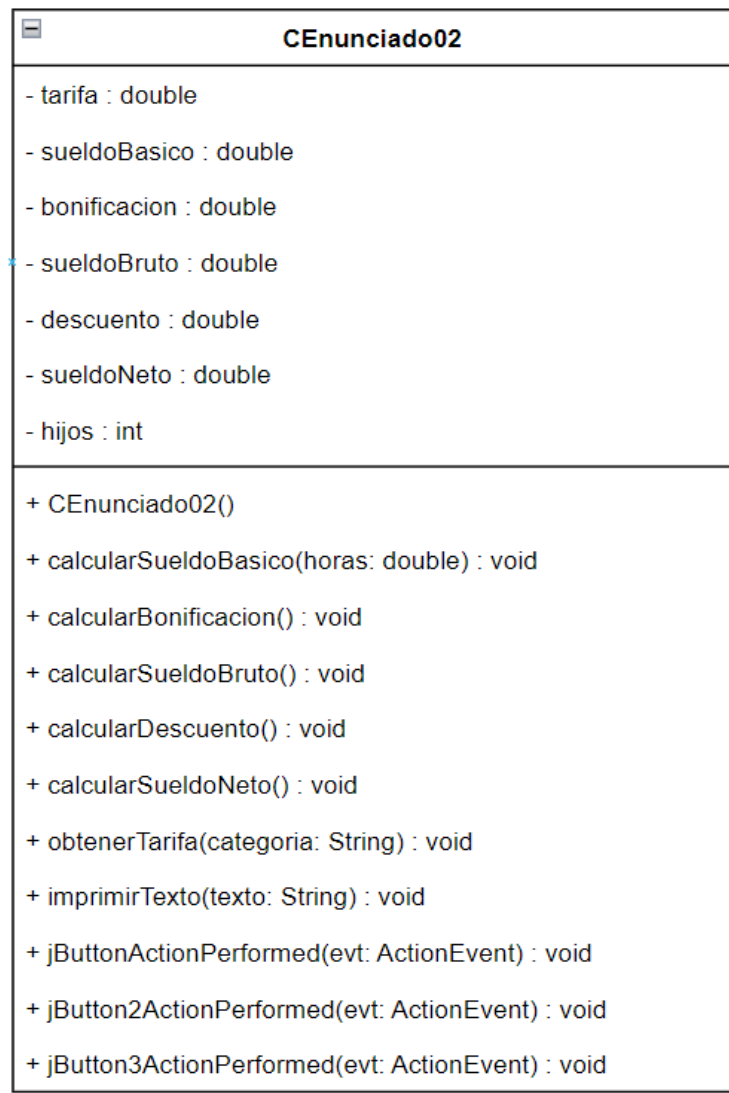
Número de hijos	Bonificación
Hasta 3	S/. 40.5 por cada hijo
Más de 3	S/. 35.0 por cada hijo

Por ley, todo empleado está sujeto a un porcentaje de descuento sobre el sueldo bruto de acuerdo con la siguiente tabla:

Sueldo bruto (S/.)	Descuento
≥ 3500	13.5%
< 3500	10.0%

Dadas la categoría y la cantidad de horas trabajadas de un empleado, diseñe un programa que determine el sueldo básico, el sueldo bruto, el descuento y el sueldo neto que le corresponden.

DIAGRAMA DE CLASES:



DISEÑO DE LA APLICACIÓN:

The screenshot shows a Java Swing window with a light gray background. At the top right, there is a label "Categoría:" followed by a dropdown menu showing "A". Below this, there are two text input fields: "Horas trabajadas :" and "Número de hijos :". On the left side, there are three buttons stacked vertically: "CALCULAR", "NUEVO", and "SALIR". To the right of these buttons is a large, empty rectangular text area.

CÓDIGO DE LA APLICACIÓN:

```
13 public class CEnunciado02 extends javax.swing.JFrame {
14
15     /**
16      * Creates new form CEnunciado02
17      */
18     double tarifa = 0;
19     double sueldoBasico = 0;
20     double bonificacion = 0;
21     double sueldoBruto = 0;
22     double descuento = 0;
23     double sueldoNeto = 0;
24     int hijos = 0;
25
26     public CEnunciado02() {
27         initComponents();
28     }
29
30     private void imprimirTexto(String texto) {
31         txtResultado.append(texto + "\n");
32     }
33
34     private void calcularSueldoBasico(double horas) {
35         sueldoBasico = tarifa * horas;
36     }
37
38     private void calcularBonificacion() {
39         if (hijos <= 3) {
40             bonificacion = hijos * 40.5;
41         } else {
42             bonificacion = hijos * 35.0;
43         }
44     }
45
46     private void calcularSueldoBruto() {
47         sueldoBruto = sueldoBasico + bonificacion;
48     }
49
50     private void calcularDescuento() {
51         if (sueldoBruto >= 3500) {
52             descuento = sueldoBruto * 0.135;
53         } else {
54             descuento = sueldoBruto * 0.10;
55         }
56     }
57 }
```

```

58 private void calcularSueldoNeto() {
59     sueldoNeto = sueldoBruto - descuento;
60 }
61
62 private void obtenerTarifa(String categoria) {
63     switch (categoria) {
64         case "A": {
65             tarifa = 45.0;
66             break;
67         }
68         case "B": {
69             tarifa = 37.5;
70             break;
71         }
72     }
73 }
74
75 /**
76  * This method is called from within the constructor to initialize the form.
77  * WARNING: Do NOT modify this code. The content of this method is always
78  * regenerated by the Form Editor.
79  */
80 @SuppressWarnings("unchecked")
81 Generated Code
192
193 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
194     System.exit(0);
195 }
196
197 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
198     cmbCategoria.setSelectedIndex(0);
199     txtHoras.setText("");
200     txtHijos.setText("");
201     txtResultado.setText("");
202 }
203

```

```

203 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
204     try {
205         String categoria = cmbCategoria.getSelectedItem().toString();
206         double horas = Double.parseDouble(txtHoras.getText());
207         hijos = Integer.parseInt(txtHijos.getText());
208
209         obtenerTarifa(categoria);
210         calcularSueldoBasico(horas);
211         calcularBonificacion();
212         calcularSueldoBruto();
213         calcularDescuento();
214         calcularSueldoNeto();
215
216         txtResultado.setText(""); // Limpiar área
217         imprimirTexto("Categoría: " + categoria);
218         imprimirTexto("Horas trabajadas: " + horas);
219         imprimirTexto("Número de hijos: " + hijos);
220         imprimirTexto("Sueldo básico: S/ " + sueldoBasico);
221         imprimirTexto("Bonificación: S/ " + bonificacion);
222         imprimirTexto("Sueldo bruto: S/ " + sueldoBruto);
223         imprimirTexto("Descuento aplicado: S/ " + descuento);
224         imprimirTexto("Sueldo neto: S/ " + sueldoNeto);
225
226     } catch (NumberFormatException e) {
227         JOptionPane.showMessageDialog(this, "Por favor ingresa valores válidos (números).", "Error", JOptionPane.ERROR_MESSAGE);
228     }
229 }
230

```