

## TALLER DE DESARROLLO DE APLICACIONES 1

**SEMANA: 2**

**ENTREGADO POR:** Mescua Segovia Marino

**CÓDIGO:** N04058C

**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:**

CEjemplo02
<ul style="list-style-type: none"><li>- pension : double</li><li>- porcentajeDescuento : double</li><li>- montoDescuento : double</li><li>- nuevaPension : double</li></ul>
<ul style="list-style-type: none"><li>+ CEjemplo02()</li><li>+ obtenerPensionPorCategoria(categoria: String) : void</li><li>+ obtenerDescuentoPorPromedio(promedio: double) : void</li><li>+ imprimirTexto(texto: String) : void</li><li>+ jButtonActionPerformed(evt: ActionEvent) : void</li><li>+ jButton2ActionPerformed(evt: ActionEvent) : void</li><li>+ jButton3ActionPerformed(evt: ActionEvent) : void</li></ul>

**DISEÑO DE LA APLICACIÓN:**

Seleccione Categoria : A

Ingresa el promedio:

CALCULAR NUEVO SALIR

**CÓDIGO DE LA APLICACIÓN:**

```

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
144 private void btn_calcularActionPerformed(java.awt.event.ActionEvent evt) {
145     try {
146         String categoria = cmb_categoria.getSelectedItem().toString();
147         String promedioTexto = txt_promedio.getText();
148
149         if (promedioTexto.isEmpty()) {
150             JOptionPane.showMessageDialog(this, "Por favor ingresa el promedio.", "Error", JOptionPane.ERROR_MESSAGE);
151             return;
152         }
153
154         double promedio = Double.parseDouble(promedioTexto);
155
156         obtenerPensionPorCategoria(categoria);
157         obtenerDescuentoPorPromedio(promedio);
158
159         montoDescuento = pension * porcentajeDescuento / 100;
160         nuevaPension = pension - montoDescuento;
161
162         txt_resultado.setText(""); // Limpiar
163         imprimirTexto("Categoria: " + categoria);
164         imprimirTexto("Promedio: " + promedio);
165         imprimirTexto("Pensión original: S/ " + pension);
166         imprimirTexto("Descuento aplicado: " + porcentajeDescuento + " %");
167         imprimirTexto("Monto de rebaja: S/ " + montoDescuento);
168         imprimirTexto("Nueva pensión: S/ " + nuevaPension);
169     } catch (NumberFormatException e) {
170         JOptionPane.showMessageDialog(this, "Por favor ingresa un promedio válido (número decimal).", "Error", JOptionPane.ERROR_MESSAGE);
171     }
172 }
173
174 private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {
175     cmb_categoria.setSelectedIndex(0);
176     txt_promedio.setText("");
177     txt_resultado.setText("");
178 }
179
180 private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
181     System.exit(0);
182 }
183

```

```

13 public class Ejemplo2 extends javax.swing.JFrame {
14
15     /**
16      * Creates new form Ejemplo2
17      */
18     double pension = 0;
19     double porcentajeDescuento = 0;
20     double montoDescuento = 0;
21     double nuevaPension = 0;
22
23     public Ejemplo2() {
24         initComponents();
25     }
26
27     private void imprimirTexto(String texto) {
28         txt_resultado.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     }

```

### **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
$\geq 10$	20%
$< 10$	10%

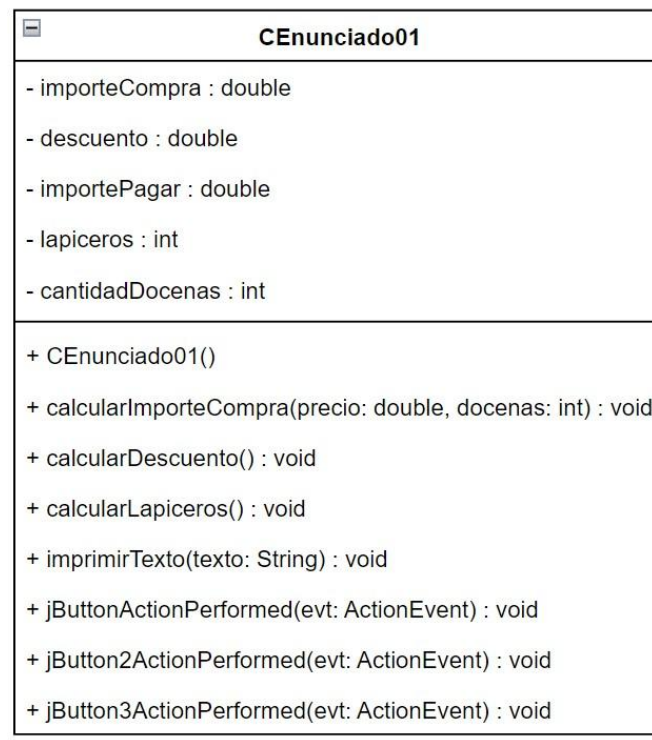
Adicionalmente, la tienda obsequia lapiceros de acuerdo con la siguiente tabla:

Importe a pagar	Lapiceros
$\geq 200$	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:**

precio por docena:

cantidad por docena:

**CÓDIGO DE LA APLICACIÓN:**

```

13  */
14  public class Enunciado1 extends javax.swing.JFrame {
15
16      /**
17       * Creates new form Enunciado1
18       */
19      public Enunciado1() {
20          initComponents();
21      }
22
23      double importeCompra = 0;
24      double descuento = 0;
25      double importePagar = 0;
26      int lapiceros = 0;
27      int cantidadDocenas = 0;
28
29      private void imprimirTexto(String texto) {
30          txt_resultado.append(texto + "\n");
31      }
32
33      private void calcularImporteCompra(double precio, int docenas) {
34          importeCompra = precio * docenas;
35      }
36
37      private void calcularDescuento() {
38          if (cantidadDocenas >= 10) {
39              descuento = importeCompra * 0.20;
40          } else {
41              descuento = importeCompra * 0.10;
42          }
43      }
44
45      private void calcularLapiceros() {
46          if (importePagar >= 200) {
47              lapiceros = 2 * cantidadDocenas;
48          } else {
49              lapiceros = 0;
50          }
51      }
52

```

```

25      private void btn_calculasActionPerformed(java.awt.event.ActionEvent evt) {
26          try {
27              double precioDocena = Double.parseDouble(txt_precio.getText());
28              cantidadDocenas = Integer.parseInt(txt_docena.getText());
29
30              calcularImporteCompra(precioDocena, cantidadDocenas);
31              calcularDescuento();
32              calcularLapiceros();
33
34              importePagar = importeCompra - descuento;
35
36              txt_resultado.setText(""); // Limpiar área
37              imprimirTexto("Importe de la compra: S/ " + importeCompra);
38              imprimirTexto("Importe del descuento: S/ " + descuento);
39              imprimirTexto("Importe a pagar: S/ " + importePagar);
40              imprimirTexto("Lapiceros de obsequio: " + lapiceros);
41          } catch (NumberFormatException e) {
42              JOptionPane.showMessageDialog(this, "Por favor ingresa valores válidos", "Error", JOptionPane.ERROR_MESSAGE);
43          }
44      }
45
46      private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {
47          txt_precio.setText("");
48          txt_docena.setText("");
49          txt_resultado.setText("");
50      }
51
52      private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
53          System.exit(0);
54      }
55

```



## **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:

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

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

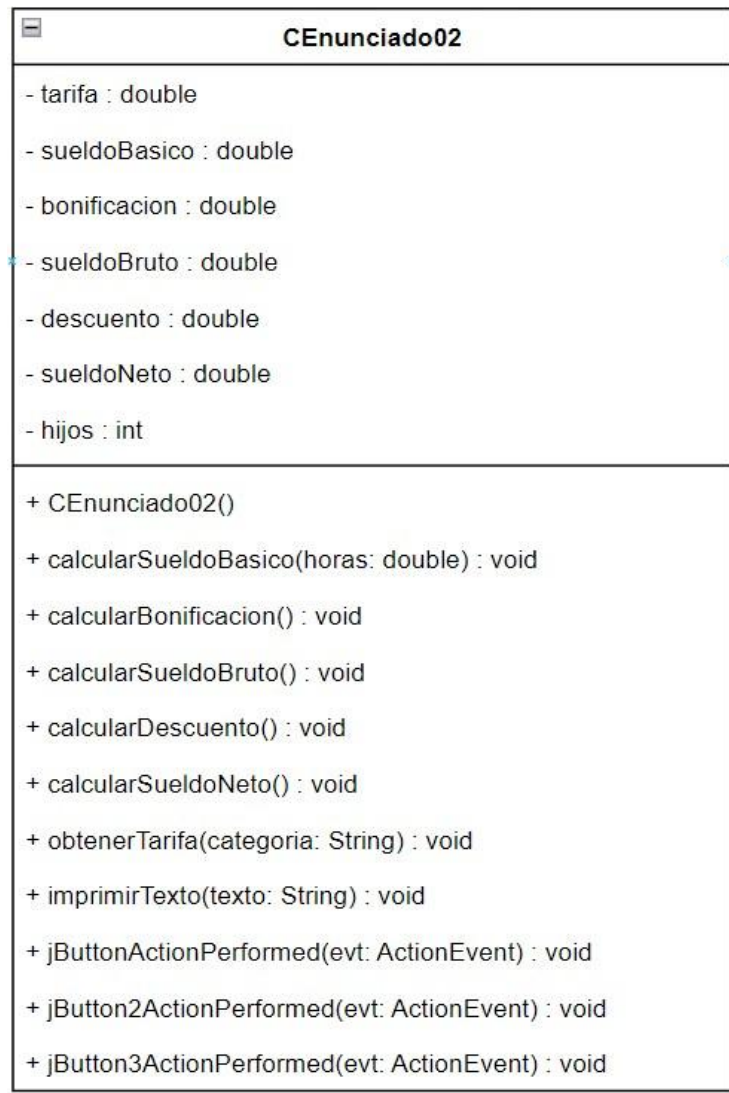
<b>Número de hijos</b>	<b>Bonificación</b>
Hasta 3	S/. 40.5 por cada hijo
Más de 3	S/. 35.0 por cada hijo

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

<b>Sueldo bruto (S/.)</b>	<b>Descuento</b>
$\geq 3500$	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:**

Categoría:

A

▼

Horas trabajadas :

Número de hijos :

CALCULAR

NUEVO

SALIR

**CÓDIGO DE LA APLICACIÓN:**

```

14 public class Enunciado2 extends javax.swing.JFrame {
15
16     /**
17      * Creates new form Enunciado2
18      */
19     public Enunciado2() {
20         initComponents();
21     }
22
23     double tarifa = 0;
24     double sueldoBasico = 0;
25     double bonificacion = 0;
26     double sueldoBruto = 0;
27     double descuento = 0;
28     double sueldoNeto = 0;
29     int hijos = 0;
30
31     private void imprimirTexto(String texto) {
32         txt_resultados.append(texto + "\n");
33     }
34
35     private void calcularSueldoBasico(double horas) {
36         sueldoBasico = tarifa * horas;
37     }
38
39     private void calcularBonificacion() {
40         if (hijos <= 3) {
41             bonificacion = hijos * 40.5;
42         } else {
43             bonificacion = hijos * 35.0;
44         }
45     }
46
47     private void calcularSueldoBruto() {
48         sueldoBruto = sueldoBasico + bonificacion;
49     }
50
51     private void calcularDescuento() {
52         if (sueldoBruto >= 3500) {
53             descuento = sueldoBruto * 0.135;
54         } else {
55             descuento = sueldoBruto * 0.10;
56         }
57     }
58

```

```

59 private void calcularSueldoNeto() {
60     sueldoNeto = sueldoBruto - descuento;
61 }
62
63 private void obtenerTarifa(String categoria) {
64     switch (categoria) {
65         case "A": {
66             tarifa = 45.0;
67             break;
68         }
69         case "B": {
70             tarifa = 37.5;
71             break;
72         }
73     }
74 }

```

```

160 private void btn_calcularActionPerformed(java.awt.event.ActionEvent evt) {
161     try {
162         String categoria = cmb_categoria.getSelectedItem().toString();
163         double horas = Double.parseDouble(txt_horas.getText());
164         hijos = Integer.parseInt(txt_hijos.getText());
165
166         obtenerTarifa(categoria);
167         calcularSueldoBasico(horas);
168         calcularBonificacion();
169         calcularSueldoBruto();
170         calcularDescuento();
171         calcularSueldoNeto();
172
173         txt_resultados.setText(""); // Limpiar Área
174         imprimirTexto("Categoría: " + categoria);
175         imprimirTexto("Horas trabajadas: " + horas);
176         imprimirTexto("Número de hijos: " + hijos);
177         imprimirTexto("Sueldo básico: S/ " + sueldoBasico);
178         imprimirTexto("Bonificación: S/ " + bonificacion);
179         imprimirTexto("Sueldo bruto: S/ " + sueldoBruto);
180         imprimirTexto("Descuento aplicado: S/ " + descuento);
181         imprimirTexto("Sueldo neto: S/ " + sueldoNeto);
182     } catch (NumberFormatException e) {
183         JOptionPane.showMessageDialog(this, "Por favor ingresa valores válidos (números).", "Error", JOptionPane.ERROR_MESSAGE);
184     }
185 }
186
187 private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {
188     cmb_categoria.setSelectedIndex(0);
189     txt_horas.setText("");
190     txt_hijos.setText("");
191     txt_resultados.setText("");
192 }
193
194 private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
195     System.exit(0);
196 }

```