

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

Escuela de Formación Profesional de Ingeniería de Sistema



INFORME DE LABORATORIO N° 6 (TA05)

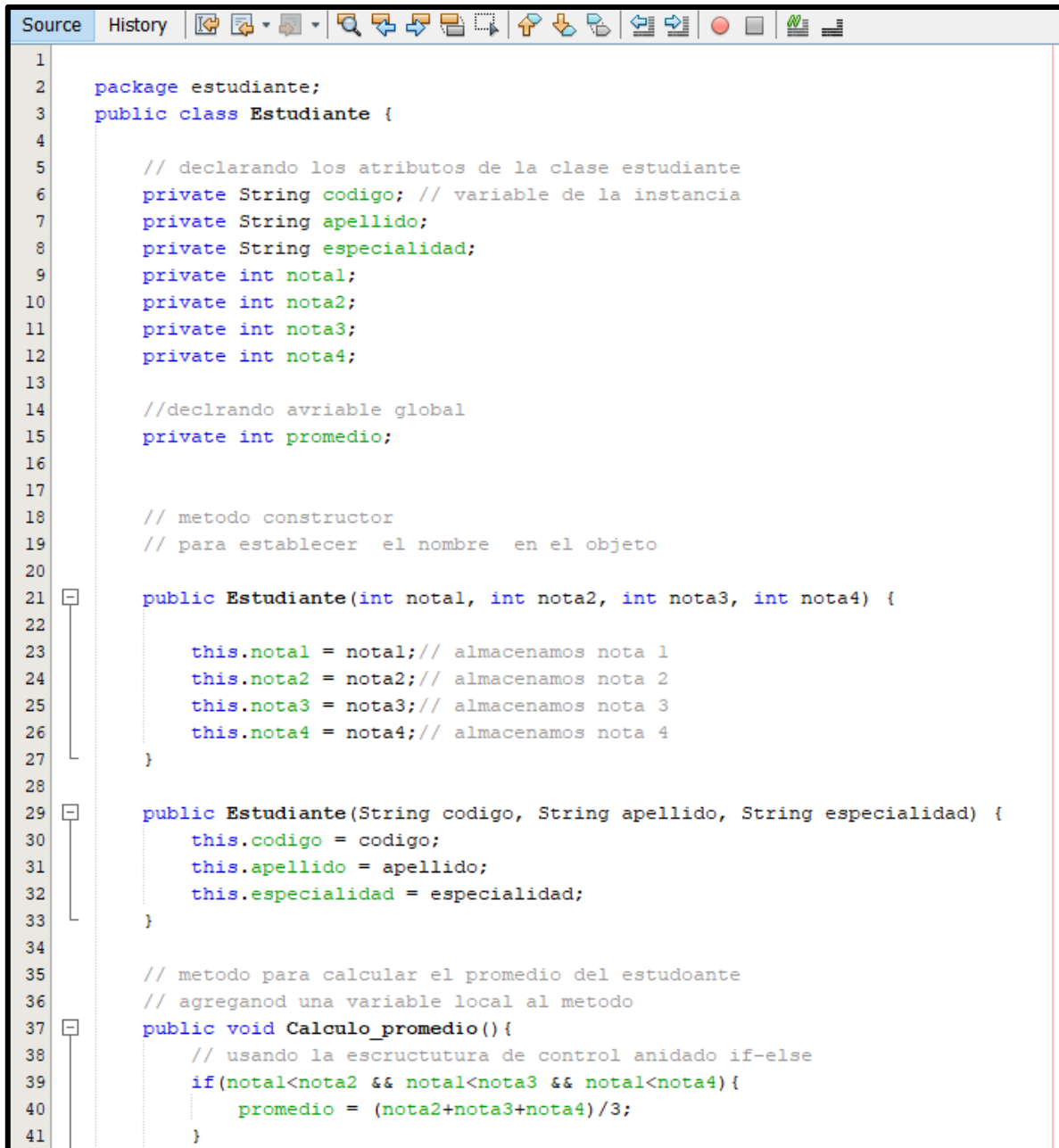
**ASIGNATURA** : PROGRAMACIÓN ORIENTADA A OBJETO  
**DOCENTE** : BARRIENTOS ESPILLCO, Fredy  
**ESTUDIANTE** : LLACTA ORE, Eduard

**AYACUCHO-PERÚ**  
**2019**

La **Tarea Académica N° 05** (TA05) consiste en los siguientes ejercicios:

1.- Implemente una clase de nombre Estudiante con los siguientes atributos privados: código, apellidos, especialidad, nota1, nota2, nota3, nota4. Considere un método para calcular el promedio, sin considerar la nota menor.

*Código del programa: solo la clase estudiante*

A screenshot of a code editor window with a toolbar at the top. The code is in Java and defines a class named 'Estudiante' within the 'estudiante' package. The class has private attributes for 'codigo', 'apellido', 'especialidad', and four 'nota' values (nota1 to nota4), plus a private 'promedio' attribute. It includes two constructors: one taking four integers (notes) and another taking three strings (code, surname, specialty). A method named 'Calculo\_promedio()' is also present, which uses an if-else structure to calculate the average of the four notes, excluding the lowest one. The code is line-numbered from 1 to 41.

```
1
2 package estudiante;
3 public class Estudiante {
4
5     // declarando los atributos de la clase estudiante
6     private String codigo; // variable de la instancia
7     private String apellido;
8     private String especialidad;
9     private int nota1;
10    private int nota2;
11    private int nota3;
12    private int nota4;
13
14    //declrando avriable global
15    private int promedio;
16
17
18    // metodo constructor
19    // para establecer el nombre en el objeto
20
21    public Estudiante(int nota1, int nota2, int nota3, int nota4) {
22
23        this.nota1 = nota1; // almacenamos nota 1
24        this.nota2 = nota2; // almacenamos nota 2
25        this.nota3 = nota3; // almacenamos nota 3
26        this.nota4 = nota4; // almacenamos nota 4
27    }
28
29    public Estudiante(String codigo, String apellido, String especialidad) {
30        this.codigo = codigo;
31        this.apellido = apellido;
32        this.especialidad = especialidad;
33    }
34
35    // metodo para calcular el promedio del estudoante
36    // agreganod una variable local al metodo
37    public void Calculo_promedio() {
38        // usando la escructutura de control anidado if-else
39        if(nota1<nota2 && nota1<nota3 && nota1<nota4){
40            promedio = (nota2+nota3+nota4)/3;
41        }
```

```

42     else{
43         if(nota2<notal && nota2<nota3 && nota2<nota4){
44             promedio=(notal+nota3+nota4)/3;
45         }
46     }
47     else{
48         if(nota3<notal && nota3<nota2 && nota3<nota4){
49             promedio = (notal+nota2+nota4)/3;
50         }
51         else{
52             promedio = (notal+nota2+nota3)/3;
53         }
54     }
55 }
56 //fin del metodos promedio
57
58 // metodo para mostrar sus datos
59 public void datos() {
60     System.out.println("\033[32mCODIGO: "+codigo+"\n\033[32mAPELLIDO: "+apellido+"\n\033[32mESPECIALIDAD: "+especialidad);
61 }
62 // fin de l metodo dato
63
64 // metodo para mostrar el promedio
65 public void imprimir(){
66     Calculo_promedio();
67     System.out.println("\033[32m*****");
68     System.out.println("\033[32mSUS NOTAS: "+notal+" "+nota2+" "+nota3+" "+nota4);
69     System.out.println();
70     System.out.println("\033[32m*****PROMEDIO FINAL*****");
71     System.out.println("\033[32mRESULTADO DEL PROMEDIO FINAL: "+promedio);
72 }
73 // fin del metodo imprimir
74
75
76
77

```

*Código del programa: el método main*

```

1
2 package prueba_estudiante;
3
4 import estudiante.Estudiante; // importando el paquete estudiante de la clase Estudiante
5 import java.util.Scanner; // importando la clase scanner
6 public class Prueba_estudiante {
7
8     /**
9      * @param args the command line arguments
10     */
11     public static void main(String[] args) {
12         //declaranod la clase scanner
13         Scanner teclado = new Scanner(System.in);
14
15         // pidiendo al usuario que ingrese sus datos
16         System.out.printf("\033[36mINGRESE SU CODIGO: ");
17         String codigo = teclado.nextLine();
18
19         System.out.printf("\033[36mINGRESE SU APELLIDO: ");
20         String apellido = teclado.nextLine();
21
22         System.out.printf("\033[36mINGRESE SU ESPECIALIDAD: ");
23         String especialidad = teclado.nextLine();
24
25         System.out.printf("\033[36mINGRSE SU PRIMERA NOTA 1: ");
26         int notal = teclado.nextInt();
27
28         System.out.printf("\033[36mINGRESE SU SEGUNDA NOTA 2: ");
29         int nota2 = teclado.nextInt();
30
31         System.out.printf("\033[36mINGRESE SU TEERCERA NOTA: ");
32         int nota3 = teclado.nextInt();
33
34         System.out.printf("\033[36mINGRESE SU CUARTA NOTA: ");
35         int nota4 = teclado.nextInt();

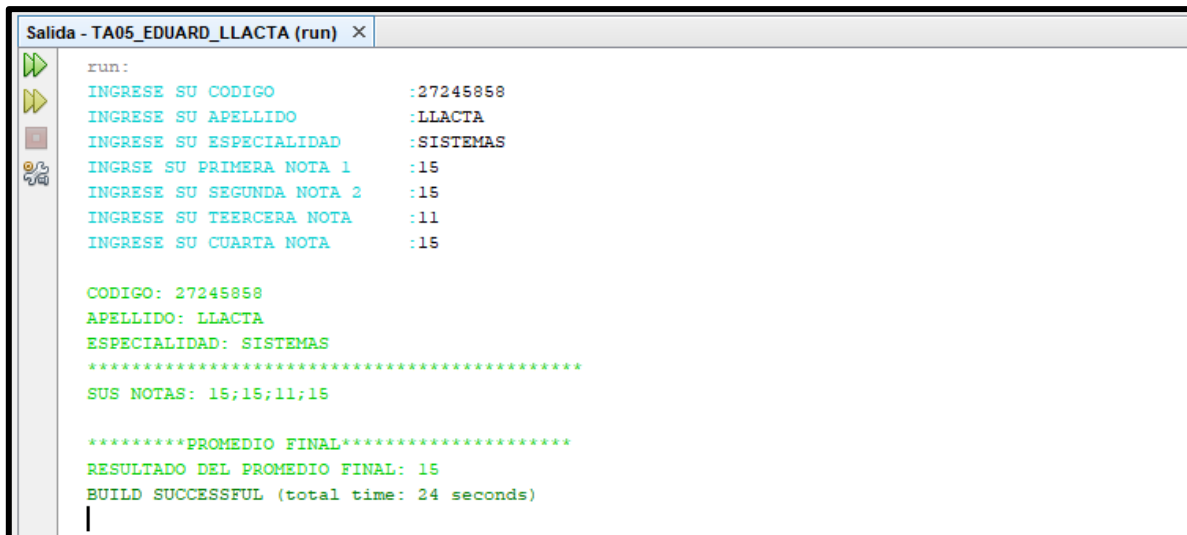
```

```

36
37         // declarando y creando el objeto de la clase
38
39         Estudiante obj = new Estudiante(codigo,apellido,especialidad);
40         Estudiante obj1 = new Estudiante(notal,nota2,nota3,nota4);
41         System.out.println();
42         obj.datos();
43         obj1.imprimir();
44
45     }
46 }
47

```

*Ejecutando el programa:*



```

run:
INGRESE SU CODIGO           :27245858
INGRESE SU APELLIDO         :LLACTA
INGRESE SU ESPECIALIDAD     :SISTEMAS
INGRESE SU PRIMERA NOTA 1   :15
INGRESE SU SEGUNDA NOTA 2   :15
INGRESE SU TEERCERA NOTA    :11
INGRESE SU CUARTA NOTA     :15

CODIGO: 27245858
APELLIDO: LLACTA
ESPECIALIDAD: SISTEMAS
*****
SUS NOTAS: 15;15;11;15

*****PROMEDIO FINAL*****
RESULTADO DEL PROMEDIO FINAL: 15
BUILD SUCCESSFUL (total time: 24 seconds)

```

*este programa nos permite evaluar el promedio del estudiante, pero solo tomando en cuenta las tres primeras notas altas, dejando la nota baja.*

2.- ¿Cuáles serían los atributos de la clase Ventana (de ordenador)? ¿cuáles serían los métodos? Piensa en las propiedades y en el comportamiento de una ventana de cualquier programa.

*Codigo del programa: clase ventana*

```
Source History
1
2 package ventana;
3 public class Ventana {
4     // declarando los atributos de la clase ventana
5     public String nombre_ventana;
6     public double tamaño;
7     public String color;
8
9     // VARIABLE PARA INSTANCIAR EL OBJETO
10    String palabra;
11
12    // CONSTRUCTOR CON TRES ARGUMENTOS
13
14    public Ventana(String nombre_ventana, double tamaño, String color) {
15        this.nombre_ventana = nombre_ventana; // almacena nombre de ventana
16        this.tamaño = tamaño; // almacenar tamaño
17        this.color = color; // almacenar color
18    }
19    //CONSTRUCTOR PARA LA VARIABLE PALABRA
20
21    public Ventana(String palabra) {
22        this.palabra = palabra;
23    } //FIN DEL CONSTRUCTOR
24
25    // metodo para cerrar ventana
26    public void cerrar_ventana() {
27
28        System.out.print("\033[33mDESEA CERRAR LA VENTANA: ");
29        System.out.println("\033[33m[SI]\t\t[NO]");
30        System.out.println("\033[34m _____ ");
31        System.out.println(" | \t\t\033[32mESCRITORIO \t\t\033[31m- ■ X \033[34m | ");
32        System.out.println("\033[34m _____ ");
33        System.out.println("\033[34m | ");
34        System.out.println("\033[34m | ");
35        System.out.println("\033[34m | ");
36        System.out.println("\033[34m | ");
37        System.out.println("\033[34m _____ ");
38
39
40        if("SI".equals(palabra)) {
41            System.out.println("*****");
42            System.out.println();
43            System.out.println("LA VENTANA SE CERRADO");
44
45        }
46        else{
47            System.out.println("*****");
48            System.out.println();
49            System.out.println("LA VENTANA ESTA ABIERTO");
50            System.out.println("\033[34m _____ ");
51            System.out.println(" | \t\t\033[32mESCRITORIO \t\t\033[31m- ■ X \033[34m | ");
52            System.out.println("\033[34m _____ ");
53            System.out.println("\033[34m | ");
54            System.out.println("\033[34m | ");
55            System.out.println("\033[34m | ");
56            System.out.println("\033[34m | ");
57            System.out.println("\033[34m _____ ");
58
59        }
60    }
61    //FIN DEL METODO
62 }
```

```

62
63 //METODO PARA ABRIR VENTANA
64 public void abrir_ventana() {
65
66     if("SI".equals(palabra)) {
67         System.out.println("VENTANA ESTA ABIERTO");
68         System.out.println("*****");
69         System.out.println();
70         System.out.println("\033[34m _____ ");
71         System.out.println(" | \t\t\033[32mESCRITORIO \t\t\033[31m- ■ X \033[34m | ");
72         System.out.println("\033[34m _____ ");
73         System.out.println("\033[34m | ");
74         System.out.println("\033[34m | ");
75         System.out.println("\033[34m | ");
76         System.out.println("\033[34m | ");
77         System.out.println("\033[34m _____ ");
78     }
79     //fin de if
80     else{
81         System.out.println("*****");
82         System.out.println("LA VENTANA ESTA CERRADO PORQUE UD NO LA ABRIÓ");
83     }
84     //fin de else
85 }
86 //FIN DEL METODO ABRI VENTANA
87
88 //METODO MINIMIZAR VENTANA
89 public void minimizar_ventana() {
90     System.out.println("MINIMIZAR LA VENTANA");
91 }
92
93 //FIN DEL METODO MINIMIZAR VENTANA
94
95 //METODO PARA MAXIMIZAR VENTANA
96 public void maximizar_ventana() {
97     System.out.println("MAXIMIZAR LA VENTANA");
98 }
99 //FIN DEL METODO MAXIMIZAR VENTANA
100 }

```

### *Código del programa: método main*

```

1
2 package prueba_ventana;
3
4 import ventana.Ventana; // importando el paquete ventana de la clase Ventana
5 import java.util.Scanner;
6 public class Prueba_ventana {
7
8
9     public static void main(String[] args) {
10
11         // declarando la clase Scanner
12         Scanner teclado = new Scanner(System.in);
13         System.out.println("QUE DESEA HACER");
14         System.out.println("SI= ABRIR VENTANA\tNO= NO SE ABRIÓ LA VENTANA");
15         System.out.println("SI= CERRAR VENTANA VENTANA\tNO= DEJAR VENTANA ABIERTO");
16         System.out.println("\033[33m\t[SI]\t[NO]");
17         String palabra=teclado.nextLine();
18
19         // declarando e inicializando el objeto
20         Ventana obl = new Ventana(palabra);
21         obl.abrir_ventana();
22         System.out.println();
23         obl.cerrar_ventana();
24     }
25 }

```

*Primero ejecutaremos para poder abrir la ventana:*

```
Salida - TA05_EDUARD_LLACTA (run) X
run:
QUE DESEA HACER
SI= ABRIR VENTANA      NO= NO SE ABRIO LA VENTANA
    [SI]    [NO]

SI
VENTANA ESTA ABIERTO
*****

  ESCRITORIO  -  X
  
```

*Segundo no abrir la ventana:*

```
Salida - TA05_EDUARD_LLACTA (run) X
run:
QUE DESEA HACER
SI= ABRIR VENTANA      NO= NO SE ABRIO LA VENTANA
    [SI]    [NO]

NO
*****
LA VENTANA ESTA CERRADO PORQUE UD NO LA ABRIO

```

*Tercero queremos cerrar ventana:*

```
Salida - TA05_EDUARD_LLACTA (run) X
run:
QUE DESEA HACER
SI= CERRAR VENTANA VENTANA      NO= DEJAR VENTANA VENTANA
    [SI]    [NO]

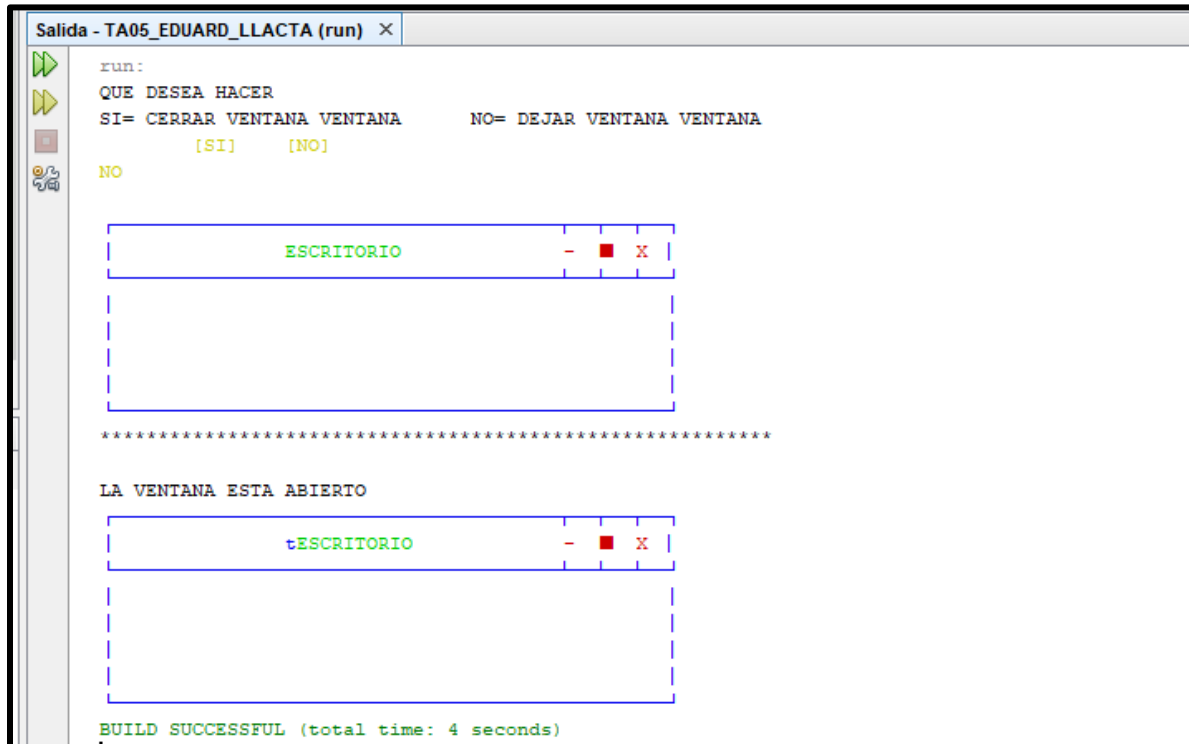
SI

  ESCRITORIO  -  X
  
```

\*\*\*\*\*

LA VENTANA SE CERRADO  
BUILD SUCCESSFUL (total time: 2 seconds)

*Cuarto dejar la ventana abierto:*



```
Salida - TA05_EDUARD_LLACTA (run) ×
run:
QUE DESEA HACER
SI= CERRAR VENTANA VENTANA      NO= DEJAR VENTANA VENTANA
    [SI]    [NO]
NO

  +-----+ +-----+
  | ESCRITORIO | - [X] |
  +-----+ +-----+

  +-----+ +-----+
  |             | - [X] |
  +-----+ +-----+

*****

LA VENTANA ESTA ABIERTO

  +-----+ +-----+
  | tESCRITORIO | - [X] |
  +-----+ +-----+

  +-----+ +-----+
  |             | - [X] |
  +-----+ +-----+

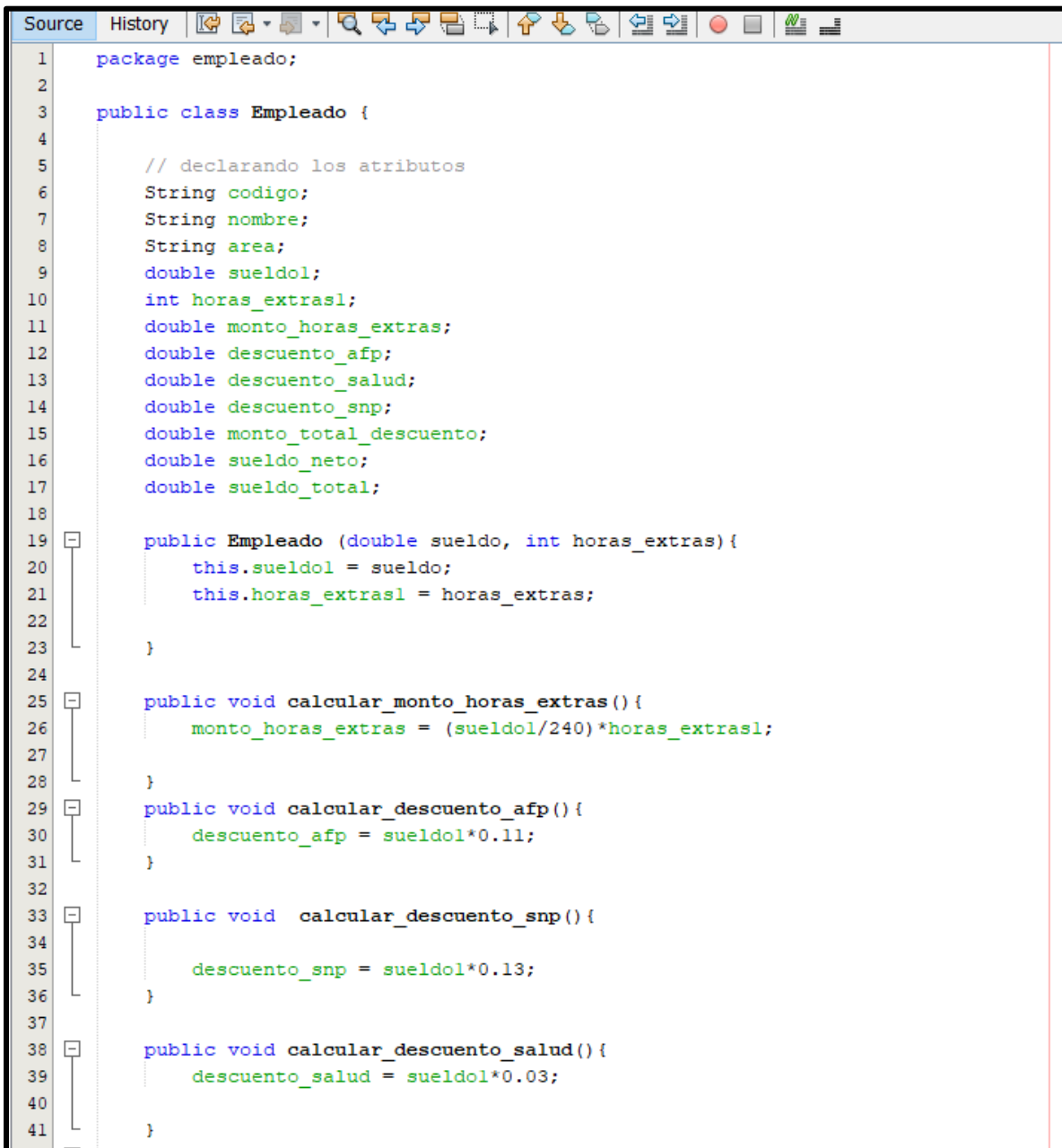
BUILD SUCCESSFUL (total time: 4 seconds)
```

*En este programa realizamos los métodos de una ventana de computadora las cuales las básicas serian cerrar y abrir la ventana de un nombre en particular*



3.- Implemente una clase de nombre Empleado con los siguientes atributos: código, nombre, área laboral, sueldo, horas extras, afiliación a una AFP. Además, considere atributos de valores comunes para todos los empleados, para los porcentajes de descuento por afiliación a una AFP (11% del sueldo), por afiliación al sistema nacional de pensiones (13% del sueldo) y por salud (3% del sueldo). Considere método de cálculo para el monto de horas extras (sueldo básico/240 \*horas extras), monto de los descuentos por AFP, por SNP, por salud, monto total de descuentos, sueldo total, sueldo neto

*Código del programa: clase empleado*

The image shows a screenshot of a Java IDE window. The window has a title bar with 'Source' and 'History' tabs. Below the tabs is a toolbar with various icons for file operations, search, and execution. The main area displays the source code for a Java class named 'Empleado'. The code is as follows:

```
1 package empleado;
2
3 public class Empleado {
4
5     // declarando los atributos
6     String codigo;
7     String nombre;
8     String area;
9     double sueldo1;
10    int horas_extras1;
11    double monto_horas_extras;
12    double descuento_afp;
13    double descuento_salud;
14    double descuento_snp;
15    double monto_total_descuento;
16    double sueldo_netto;
17    double sueldo_total;
18
19    public Empleado (double sueldo, int horas_extras){
20        this.sueldo1 = sueldo;
21        this.horas_extras1 = horas_extras;
22    }
23
24
25    public void calcular_monto_horas_extras(){
26        monto_horas_extras = (sueldo1/240)*horas_extras1;
27    }
28
29    public void calcular_descuento_afp(){
30        descuento_afp = sueldo1*0.11;
31    }
32
33    public void calcular_descuento_snp(){
34
35        descuento_snp = sueldo1*0.13;
36    }
37
38    public void calcular_descuento_salud(){
39        descuento_salud = sueldo1*0.03;
40    }
41 }
```

```

42 public void calcular_total_descuento(){
43     monto_total_descuento = descuento_salud + descuento_snp;
44 }
45
46 public void calcular_sueldo_neto(){
47     sueldo_neto = sueldo1-monto_total_descuento;
48 }
49 public void calcular_sueldo_total(){
50     sueldo_total = sueldo_neto + monto_horas_extras;
51 }
52 public void imprimir(){
53     System.out.println("*****");
54     System.out.println("***programa para calcular el sueldo total del trabajador***");
55     System.out.println("su sueldo total es: "+sueldo_total);
56 }
57
58 }
59

```

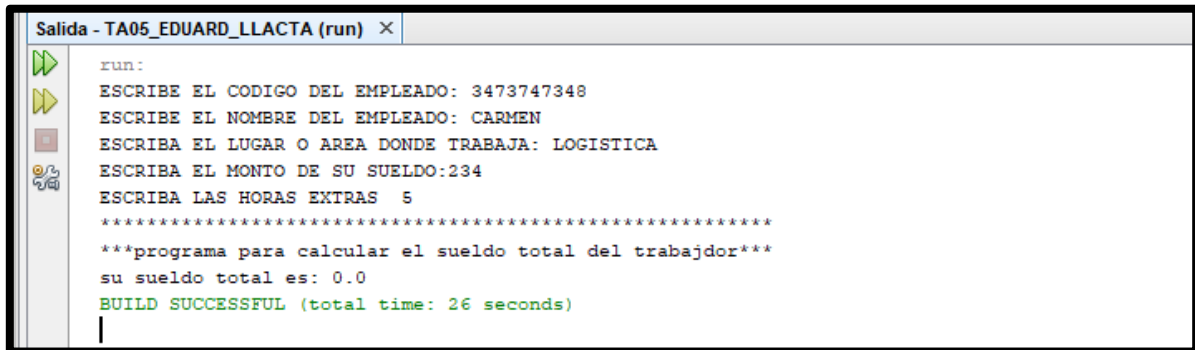
### *Código del programa: el método main*

```

1
2 package prueba_empleado;
3
4 import empleado.Empleado;
5 import java.util.Scanner;
6 public class Prueba_empleado {
7
8     /**
9      * @param args the command line arguments
10     */
11     public static void main(String[] args) {
12         Scanner teclado = new Scanner(System.in);
13
14         System.out.printf("ESCRIBE EL CODIGO DEL EMPLEADO: ");
15         String codigo = teclado.nextLine();
16
17         System.out.printf("ESCRIBE EL NOMBRE DEL EMPLEADO: ");
18         String nombre = teclado.nextLine();
19
20         System.out.printf("ESCRIBA EL LUGAR O AREA DONDE TRABAJA: ");
21         String area_laboral = teclado.nextLine();
22
23         System.out.printf("ESCRIBA EL MONTO DE SU SUELDO:");
24         double sueldo = teclado.nextDouble();
25
26         System.out.printf("ESCRIBA LAS HORAS EXTRAS ");
27         int horas_extras = teclado.nextInt();
28
29         Empleado datos = new Empleado(sueldo,horas_extras);
30         datos.imprimir();
31
32     }

```

### *Ejecutando el programa:*



```
Salida - TA05_EDUARD_LLACTA (run) ×
run:
ESCRIBE EL CODIGO DEL EMPLEADO: 3473747348
ESCRIBE EL NOMBRE DEL EMPLEADO: CARMEN
ESCRIBA EL LUGAR O AREA DONDE TRABAJA: LOGISTICA
ESCRIBA EL MONTO DE SU SUELDO:234
ESCRIBA LAS HORAS EXTRAS 5
*****
***programa para calcular el sueldo total del trabajador***
su sueldo total es: 0.0
BUILD SUCCESSFUL (total time: 26 seconds)
```

*Programa que no muestra el sueldo del trabajador con sus respectivos descuentos*

### *Conclusiones*

- Resolver un problema bajo el paradigma de la programación orientada a objetos implica determinar y caracterizar los diferentes objetos que intervienen en el problema, definir sus propiedades y métodos y ponerlos a interactuar
- Así mismo Para entender cómo funciona el paradigma de la programación orientada a objetos es necesario ver un programa como una colección de objetos que interactúan entre sí enviándose mensajes y cambiando su estado durante la ejecución.