

UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE HUAMANGA
FACULTAD DE INGENIERIA DE MINAS, GEOLOGIAS Y CIVIL
ESCUELA DE FORMACION PROFESIONAL DE INGENIERIA DE SISTEMAS



Laboratorio 05

ALUMNO: PARIONA FIGUEROA, JUNIOR FLAVIO

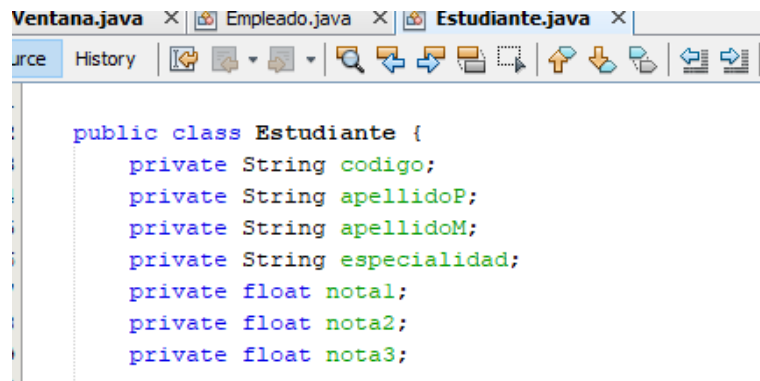
PROFESOR: FREDY BARRIENTOS

AYACUCHO-PERU

2019

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

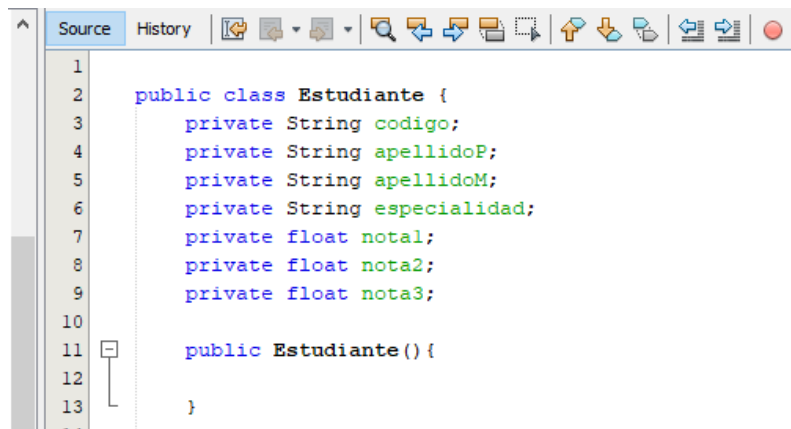
- En este ejercicio o código realizado primero creamos una clase estudiante con sus atributos privados (código, apellidos, especialidad, nota1, nota2, nota3, nota4). Tal como se muestra en la captura realizada.



The screenshot shows a code editor with three tabs: 'Ventana.java', 'Empleado.java', and 'Estudiante.java'. The 'Estudiante.java' tab is active, displaying the following Java code:

```
public class Estudiante {  
    private String codigo;  
    private String apellidoP;  
    private String apellidoM;  
    private String especialidad;  
    private float nota1;  
    private float nota2;  
    private float nota3;  
}
```

- Después de declarar los atributos, es necesario crear un constructor para inicializar la clase estudiante tal como se muestra a continuación.



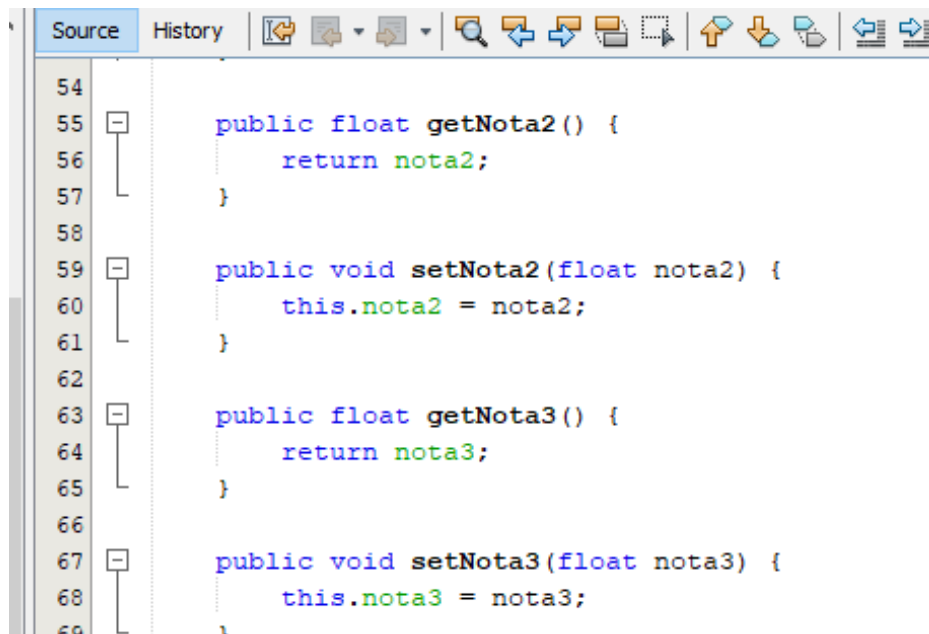
The screenshot shows the same code editor with the 'Estudiante.java' tab active. The code now includes a constructor:

```
1  
2 public class Estudiante {  
3     private String codigo;  
4     private String apellidoP;  
5     private String apellidoM;  
6     private String especialidad;  
7     private float nota1;  
8     private float nota2;  
9     private float nota3;  
10  
11     public Estudiante() {  
12  
13     }  
14 }
```

- Después de realizar la creación o la declaración de atributos para poder acceder a los atributos mencionados creamos sus respectivos métodos accesorios(get) y métodos mutadores(set). Se utiliza un método accesor para devolver el valor de un campo privado. Y método mutador para establecer el valor de un campo privado, tal como se muestra en la siguiente captura.

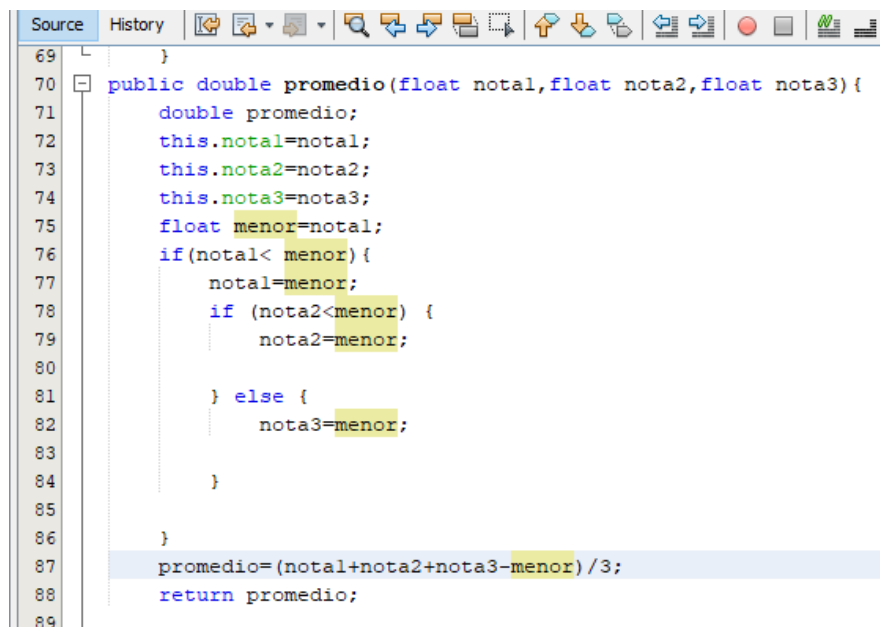
```
Source History [Icons]
14
15 public String getCodigo() {
16     return codigo;
17 }
18
19 public void setCodigo(String codigo) {
20     this.codigo = codigo;
21 }
22
23 public String getApellidoP() {
24     return apellidoP;
25 }
26
27 public void setApellidoP(String apellidoP) {
28     this.apellidoP = apellidoP;
29 }
30
31 public String getApellidoM() {
32     return apellidoM;
33 }
34
```

```
Source History [Icons]
34
35 public void setApellidoM(String apellidoM) {
36     this.apellidoM = apellidoM;
37 }
38
39 public String getEspecialidad() {
40     return especialidad;
41 }
42
43 public void setEspecialidad(String especialidad) {
44     this.especialidad = especialidad;
45 }
46
47 public float getNota1() {
48     return nota1;
49 }
50
51 public void setNota1(float nota1) {
52     this.nota1 = nota1;
53 }
54
```



```
54
55 public float getNota2() {
56     return nota2;
57 }
58
59 public void setNota2(float nota2) {
60     this.nota2 = nota2;
61 }
62
63 public float getNota3() {
64     return nota3;
65 }
66
67 public void setNota3(float nota3) {
68     this.nota3 = nota3;
69 }
```

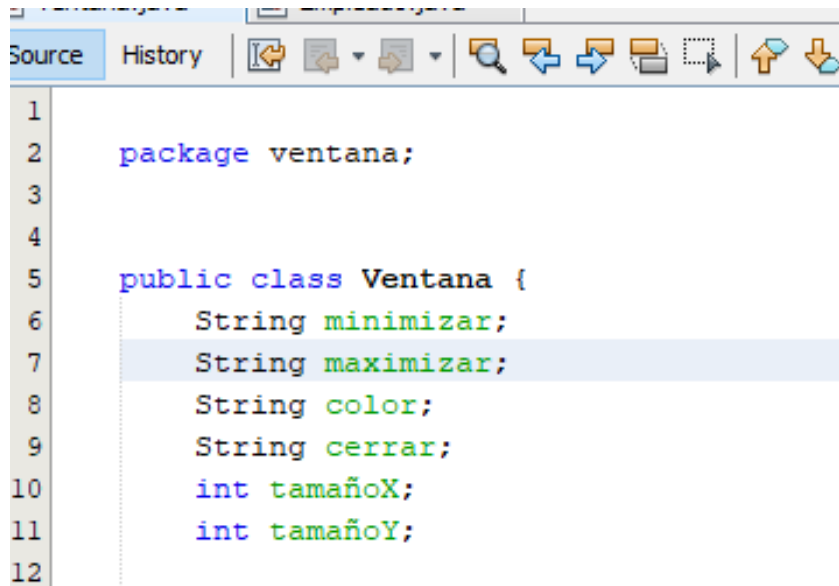
- Como en el ejercicio nos pide crear un método para calcular el promedio de notas del estudiante sin considerar la menor nota del estudiante lo cual creamos un método con el nombre promedio. Tal como se muestra en la captura. pero como nos dice sin consideras la nota menor utilizamos un if para saber cual es la nota menor declarando a la nota1 como el menor, y luego para sacar el promedio de las notas lo sumamos las notas menos el menor y dividiéndolo entre 3.



```
69 }
70 public double promedio(float notal, float nota2, float nota3) {
71     double promedio;
72     this.notal=notal;
73     this.nota2=nota2;
74     this.nota3=nota3;
75     float menor=notal;
76     if(notal< menor){
77         notal=menor;
78         if (nota2<menor) {
79             nota2=menor;
80         } else {
81             nota3=menor;
82         }
83     }
84 }
85
86 }
87 promedio=(notal+nota2+nota3-menor)/3;
88 return promedio;
89 }
```

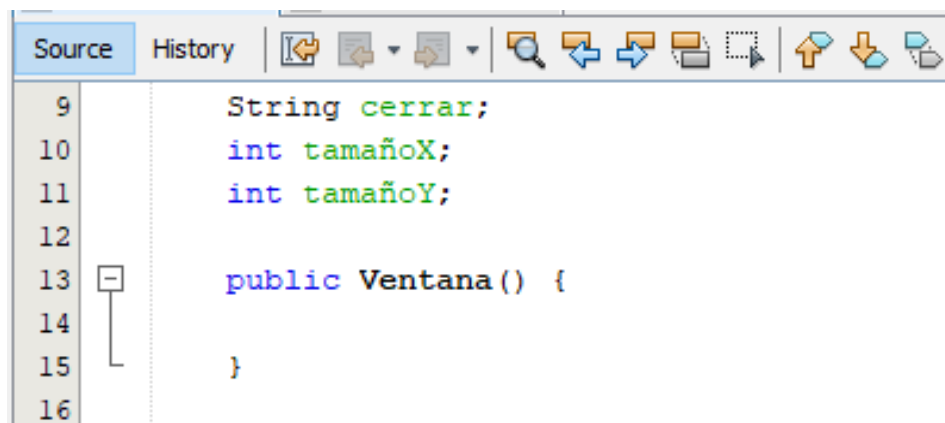
2. ¿Cuáles serían los atributos de la clase ventana (de ordenador)? ¿cuáles serían los métodos? Piensa en las propiedades y en el comportamiento de una ventana de cualquier programa.

- En este ejercicio o código realizado primero creamos una clase de ventana (de un ordenador) con sus atributos públicos(minimizar, maximizar, color, cerrar, tamañoX, tamañoY).tal como se muestra en la captura.



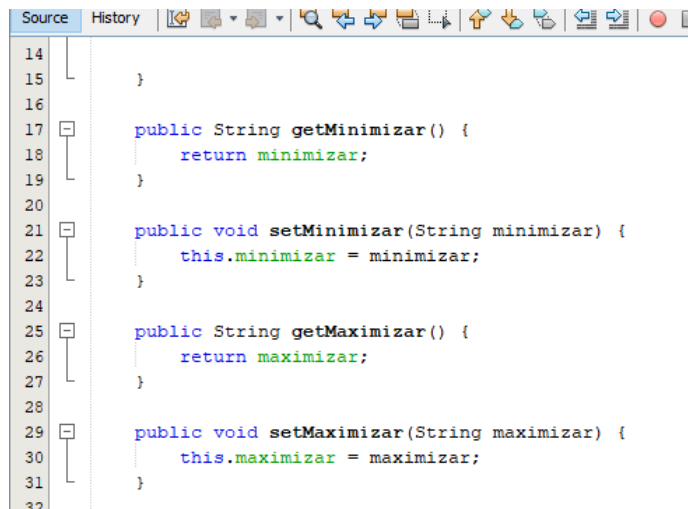
```
1
2 package ventana;
3
4
5 public class Ventana {
6     String minimizar;
7     String maximizar;
8     String color;
9     String cerrar;
10    int tamañoX;
11    int tamañoY;
12 }
```

- Después de declarar los atributos de la clase ventana, es necesario crear un constructor para inicializar la clase ventana, tal como se muestra a continuación.



```
9     String cerrar;
10    int tamañoX;
11    int tamañoY;
12
13    public Ventana() {
14    }
15
16 }
```

- Después de realizar la creación o la declaración de atributos para poder acceder a los atributos mencionados creamos sus respectivos métodos accesorios(get) y métodos mutadores(set). tal como se muestra en la siguiente captura.
- Para el atributo minimizar.
Creamos su metodo accesor y luego su metodo mutador.
- Para el atributo maximizar.
Creamos tambien su metodo accesor y luego su metodo mutador.
Tal como se muestra en la siguiente imagen.

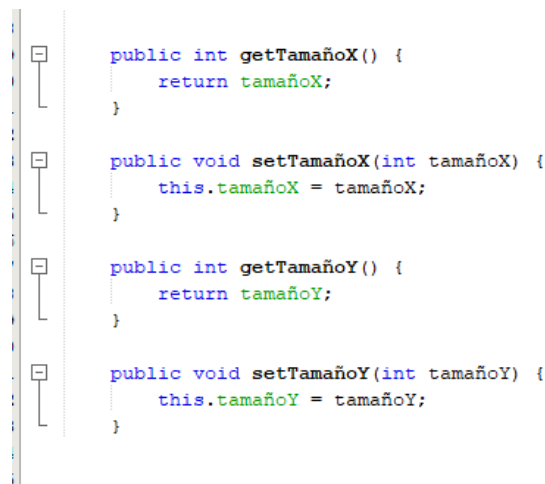


```

14  }
15  }
16
17  public String getMinimizar() {
18      return minimizar;
19  }
20
21  public void setMinimizar(String minimizar) {
22      this.minimizar = minimizar;
23  }
24
25  public String getMaximizar() {
26      return maximizar;
27  }
28
29  public void setMaximizar(String maximizar) {
30      this.maximizar = maximizar;
31  }
32

```

- Para el atributo tamañoX.
También creamos su método accesor y su método mutador
- Para el atributo tamaño
Primero creamos su método accesor y su método mutador.



```

    public int getTamañoX() {
        return tamañoX;
    }

    public void setTamañoX(int tamañoX) {
        this.tamañoX = tamañoX;
    }

    public int getTamañoY() {
        return tamañoY;
    }

    public void setTamañoY(int tamañoY) {
        this.tamañoY = tamañoY;
    }

```

- Para el color.
Su método accesor y su método mutador
- Para cerrar.
También su método accesor y su método mutador.

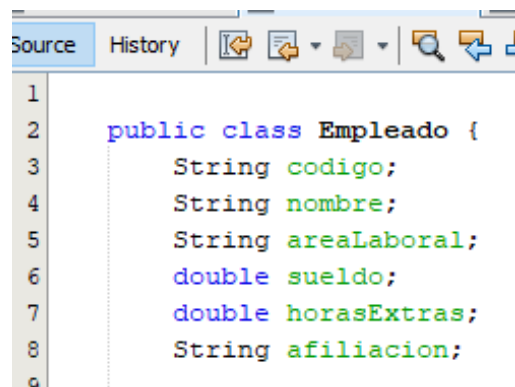
```

32
33 public String getColor() {
34     return color;
35 }
36
37 public void setColor(String color) {
38     this.color = color;
39 }
40
41 public String getCerrar() {
42     return cerrar;
43 }
44
45 public void setCerrar(String cerrar) {
46     this.cerrar = cerrar;
47 }

```

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.

- En este ejercicio o código realizado primero creamos una clase empleado con sus atributos (código, nombre, área laboral, sueldo, horas extras, afiliación a una AFP). Tal como se muestra en la captura realizada.

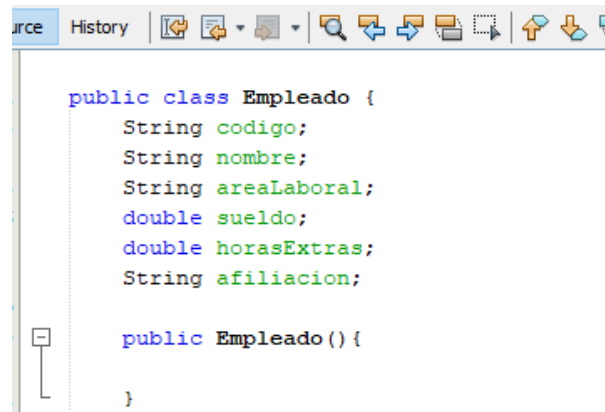


```

Source History
1
2 public class Empleado {
3     String codigo;
4     String nombre;
5     String areaLaboral;
6     double sueldo;
7     double horasExtras;
8     String afiliacion;
9

```

- Después de declarar los atributos de la clase empleado, es necesario crear un constructor para inicializar la clase empleado, tal como se muestra a continuación.



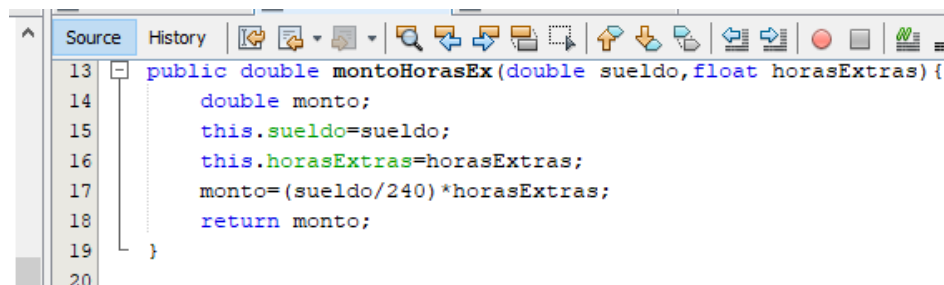
```

public class Empleado {
    String codigo;
    String nombre;
    String areaLaboral;
    double sueldo;
    double horasExtras;
    String afiliacion;

    public Empleado() {
    }
}

```

- En este ejercicio nos pide hallar el monto de las horas extras lo cual es necesario crear su método de monto horas extras con sus parámetros (); Es un método de accesor (get). También declare un variable monto tal como se muestra en la captura.

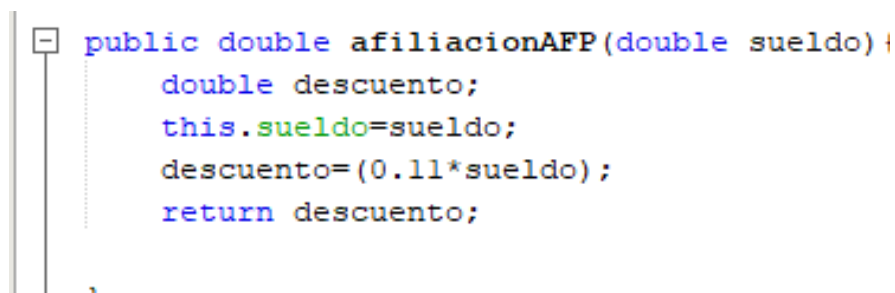


```

13 public double montoHorasEx(double sueldo, float horasExtras) {
14     double monto;
15     this.sueldo=sueldo;
16     this.horasExtras=horasExtras;
17     monto=(sueldo/240)*horasExtras;
18     return monto;
19 }
20

```

- También nos pide hallar el descuento AFP lo es también necesario crear su método accesor (get) con sus respectivos parámetros, también declare un variable descuento y un return del descuento.

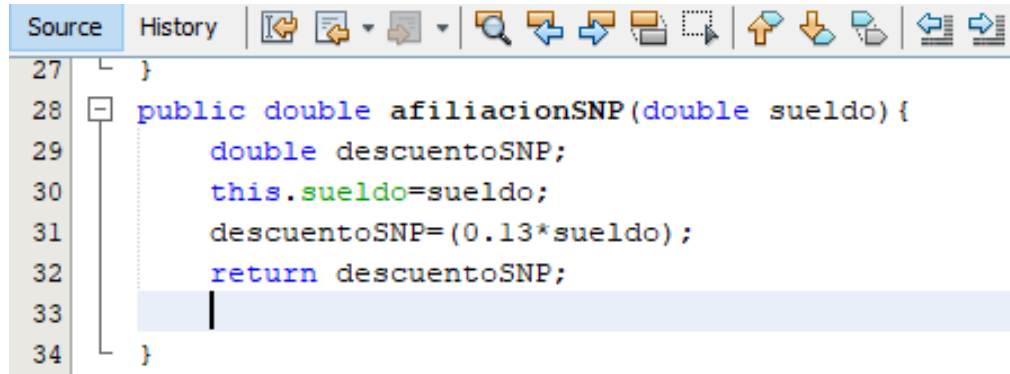


```

public double afiliacionAFP(double sueldo) {
    double descuento;
    this.sueldo=sueldo;
    descuento=(0.11*sueldo);
    return descuento;
}

```


- También nos pide hallar el descuento SNP para eso es necesario crear su método accesor (get), con su respectivo atributo, declarando un variable descuentoSNP.

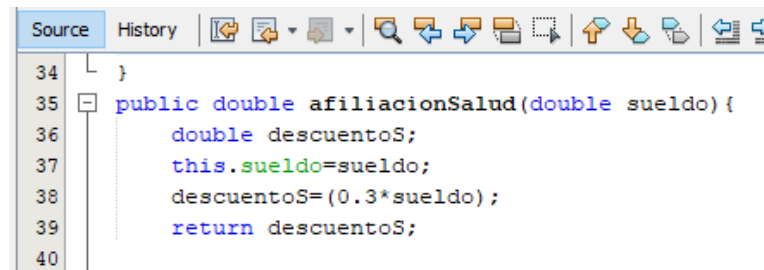


```

27 }
28 public double afiliacionSNP(double sueldo){
29     double descuentoSNP;
30     this.sueldo=sueldo;
31     descuentoSNP=(0.13*sueldo);
32     return descuentoSNP;
33 }
34

```

- También nos pide hallar el descuento salud para eso es necesario crear su método accesor (get), con su respectivo atributo, declarando un variable descuentoS.

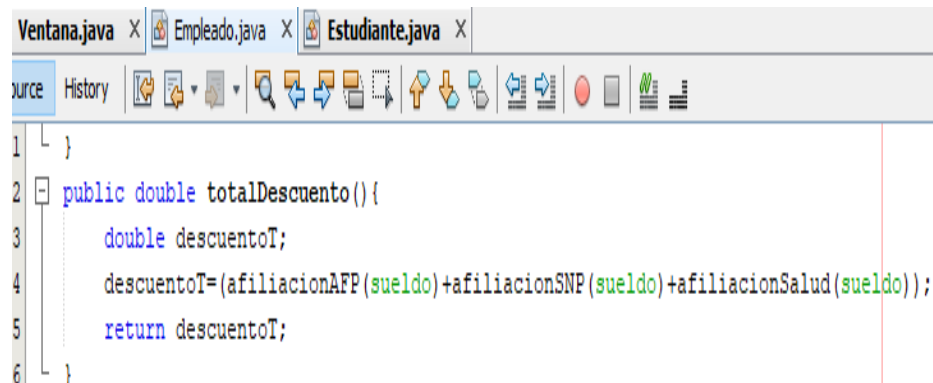


```

34 }
35 public double afiliacionSalud(double sueldo){
36     double descuentoS;
37     this.sueldo=sueldo;
38     descuentoS=(0.3*sueldo);
39     return descuentoS;
40

```

- En este ejercicio nos pide hallar el descuento total, lo cual cree un método accesor con el nombre totalDescuento, con un parámetro vacío. Y declarando un variable descuentoT, para poder hallar el descuento total tenemos que sumar las afiliaciones con sus atributos(sueldos). Tal como se muestra en la captura.

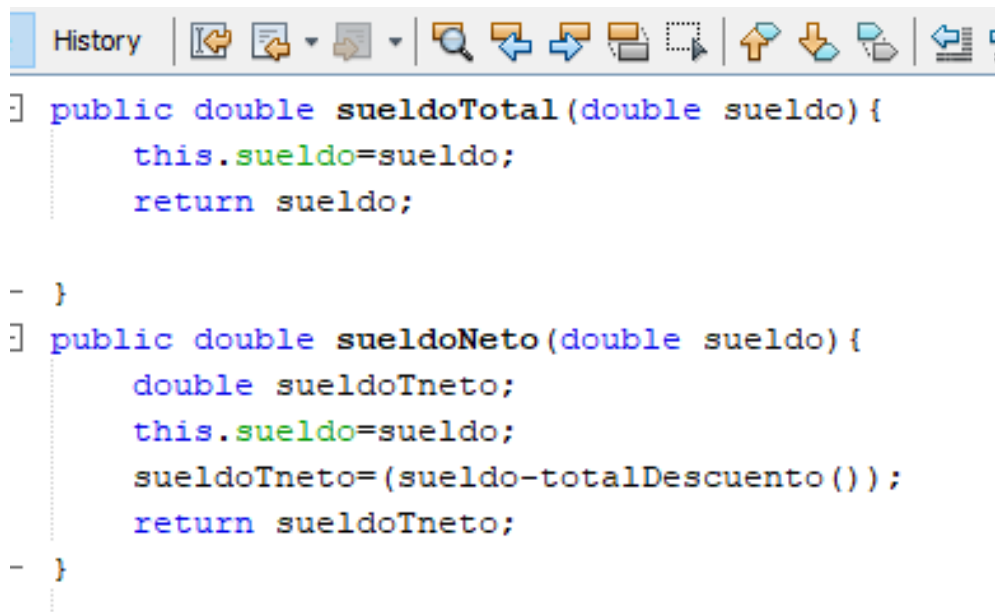


```

Ventana.java x Empleado.java x Estudiante.java x
Source History
1 }
2 public double totalDescuento(){
3     double descuentoT;
4     descuentoT=(afiliacionAFP(sueldo)+afiliacionSNP(sueldo)+afiliacionSalud(sueldo));
5     return descuentoT;
6 }

```

- En este ejercicio primero es necesario crear un método sueldo total con su parámetro double sueldo. En este ejercicio nos pide hallar el sueldo total, lo cual cree un método accesor con el nombre sueldo neto, con su parámetro (double sueldo). Y declarando un variable sueldoTneto, para poder hallar el sueldoTneto tenemos que restar el sueldo-total de descuento con su parámetro vacío, Tal como se muestra en la captura.



```

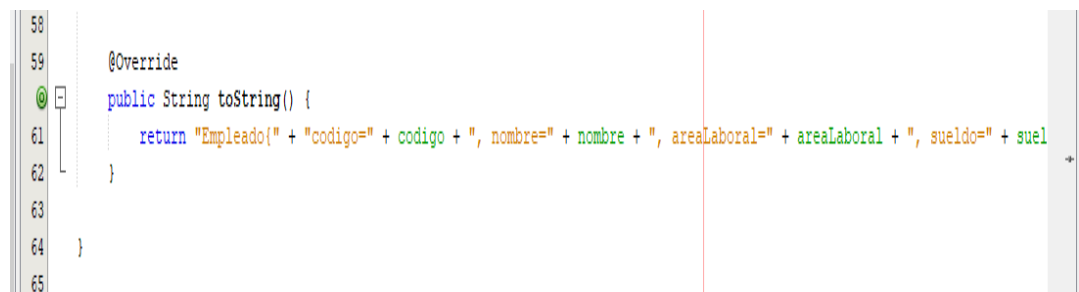
History | [Icons]
] public double sueldoTotal(double sueldo){
    this.sueldo=sueldo;
    return sueldo;

- }
] public double sueldoNeto(double sueldo){
    double sueldoTneto;
    this.sueldo=sueldo;
    sueldoTneto=(sueldo-totalDescuento());
    return sueldoTneto;

- }

```

- Luego lo concatene con to String.



```

58
59 @Override
60 public String toString() {
61     return "Empleado{" + "codigo=" + codigo + ", nombre=" + nombre + ", areaLaboral=" + areaLaboral + ", sueldo=" + sueldo + "
62 }
63
64 }
65

```

En forma general.

El this sirve para hacer referencia a un método o propiedad del objeto actual.