

# UNIVERSIDAD NACIONAL SAN CRISTOBAL DE HUAMANGA

***FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL***

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



## **INFORME 05**

**CURSO** : PROGRAMACIÓN ORIENTADA A OBJETOS.

**DOCENTE** : FREDY BARRIENTOS.

**ALUMNO** : DIEGO ÁNGELO CÁRDENAS MERCADO.

**SERIE** 100

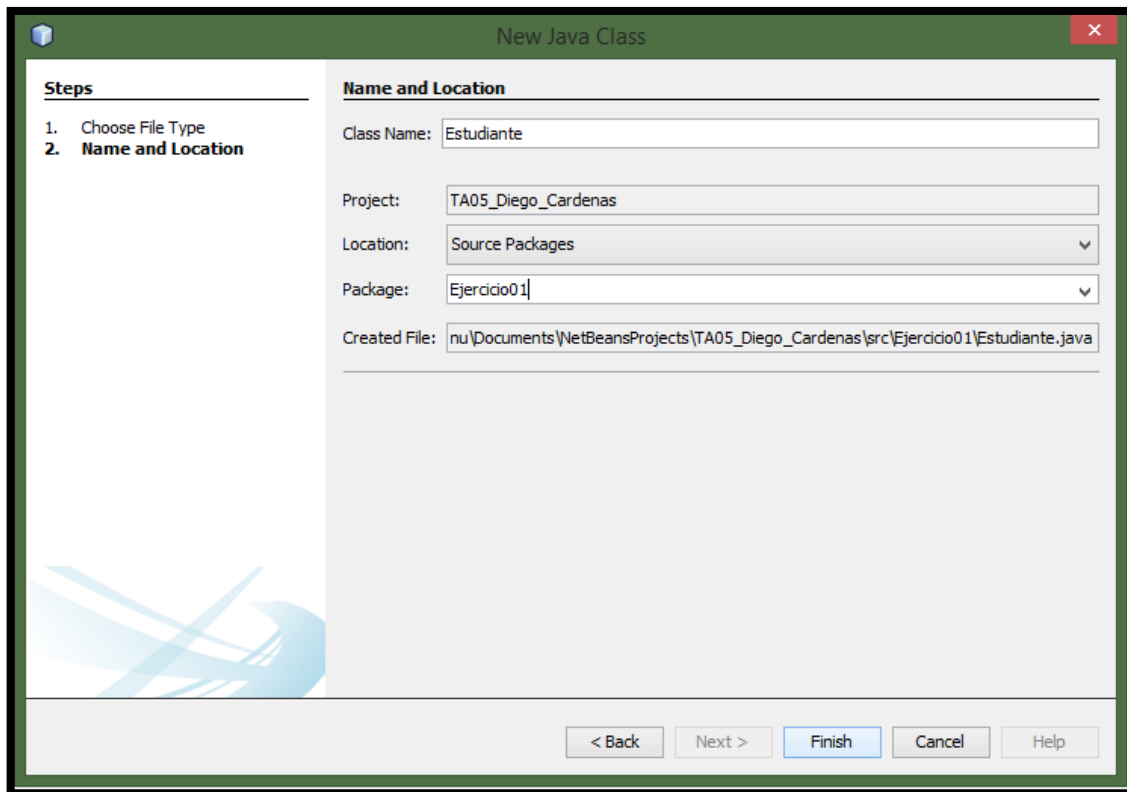
**FECHA** : SÁBADO 28 DE SEPTIEMBRE.

**AYACUCHO – PERÚ**

**2019**

## EJERCICIO 01

Creamos la clase estudiante, pero esta no será una clase principal, será almacenada en el paquete Ejercicio01.



Primeramente empezamos a crearle atributos, como el nombre, apellido, etc. También crearemos un ArrayList, ya que, en lo personal, trabajare con esto.

```
package Ejercicio01;

import java.util.ArrayList;
import javax.swing.JOptionPane;

public class Estudiante {

    //Creamos los atributos
    private String _nombres;
    private String _apellidos;
    private String _especialidad;
    private int _codigo;

    ArrayList<Integer> notas[];
```

Ahora creamos el constructor, este mismo sirve para que podamos crear un objeto y establecerle valores, por eso en su parámetro pongo las variables que necesitare pedir al usuario.

```
public class Estudiante {

    //Creamos los atributos
    private String _nombres;
    private String _apellidos;
    private String _especialidad;
    private int _codigo;

    ArrayList<Integer> notas[];

    //creamos el constructor
    public Estudiante(String _nombres,String _apellidos,String _especialidad,int _codigo){
        this._nombres = _nombres;
        this._apellidos = _apellidos;
        this._especialidad = _especialidad;
        this._codigo = _codigo;
    }
}
```

Después empezaremos a crear los métodos, primero el método de llenar lista, me permitirá almacenar los distintas notas en una lista y nos retornara la lista ya llena, nos servirá también más adelante.

```
//creamos el metodo para crear una lista y llenarla
public ArrayList<Integer> llenarLista() {

    //creamos una lista
    ArrayList<Integer> notas;
    notas = new ArrayList();

    //llenamos la lista con las notas
    for(int i=0;i<4;i++){
        notas.add(Integer.parseInt(JOptionPane.showInputDialog("Ingrese nota" + (i+1) + ": ")));
    }

    return notas;
}
```

Ahora crearemos el método de calcular el menor, según el problema pide que debemos eliminar la menor de las notas, así que por eso lo metí en una lista para que se me haga un poco más fácil buscar al menor y una vez encontrado la nota más baja me retornara ese mínimo, claro, puede hacerse de otro modo.

```
//creamos el metodo que busque el menor
public int calcularMenor(ArrayList<Integer> notas){

    //consideramos a la primera posicion menor
    int _min = notas.get(0);

    //comparacion con cada uno de los numeros de la lista
    for(int i =1; i<notas.size();i++){
        if(_min > notas.get(i)){
            _min = notas.get(i);
        }
    }

    return _min;
}
```

Lo siguiente que haremos será crear un método para calcular el promedio, para eso tendremos que tener en nuestro parámetro un valor, ese valor será la nota menor que nos retornó el método anterior, con ese valor restaremos la suma de las cuatro notas, para después dividir las entre 4, último nos retornara el promedio excluyendo el menor de las notas.

```
//aqui calcularemos el promedio excluyendo al menor
public double calcularPromedio(int _min, ArrayList<Integer> notas){

    //acumulador
    int a = 0;

    //aqui iremos sumando cada uno de las notas
    for(int i=0; i<4;i++){
        a = a + notas.get(i);
    }

    //sacamos promedio excluyendo al menor
    double _promedio = (a - _min)/4;

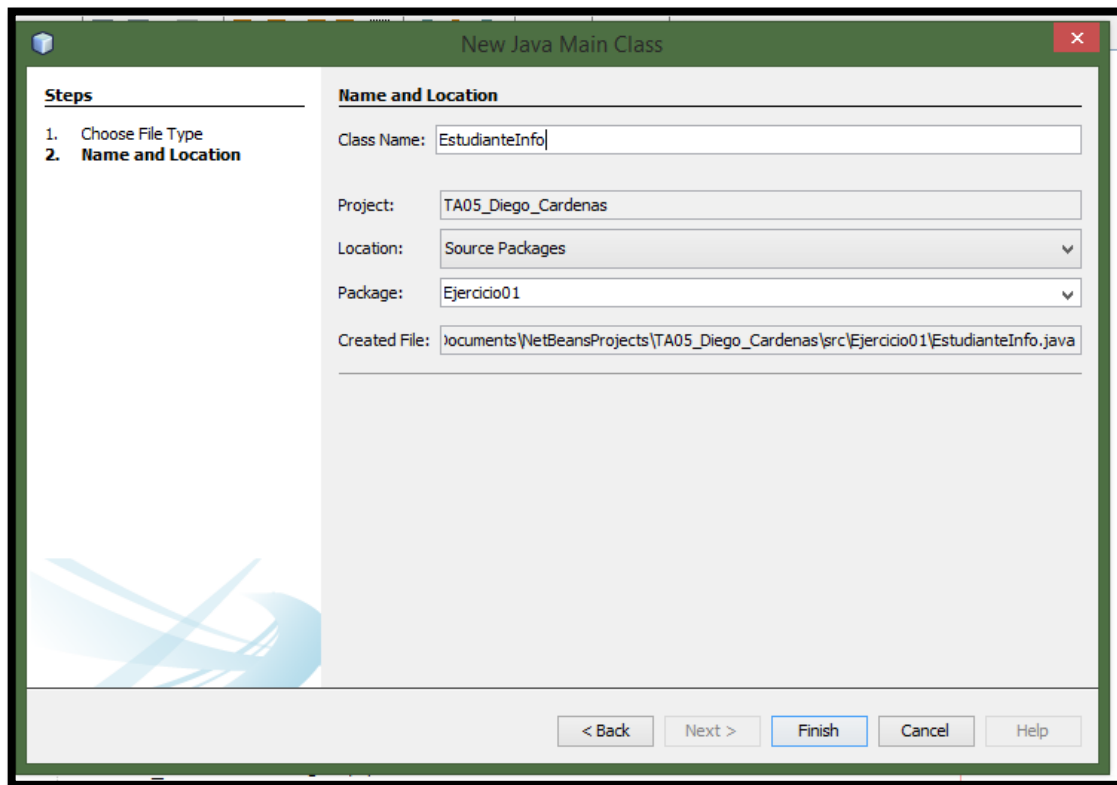
    return _promedio;
}
```

Ahora por ultimo crearemos el método que nos permitirá mostrar en pantalla, esta no tendrá algún retorno así que se pondrá void.

```
//creamos este metodo para imprimir los valores
public void mostrarInfo(double promedio, ArrayList<Integer> notas, int _min){
    JOptionPane.showMessageDialog(null, "NOMBRES: " + _nombres + "\nAPELLIDOS: " + _apellidos + "\nESPECIALIDAD: " +
        _especialidad + "\nCODIGO: " + _codigo + "\nNOTAS = " + notas + "\nNOTA MAS BAJA: " + _min + "\nPROMEDIO: " +
        promedio);
}

}
```

Ahora creamos la clase principal llamada EstudianteInfo, aquí se ejecutara las diferentes métodos.



Pedimos al usuario que ingrese los siguientes datos, estas serán utilizadas al llamar al crear el objeto.

```
package Ejercicio01;

import java.util.ArrayList;
import javax.swing.JOptionPane;

public class EstudianteInfo {

    public static void main(String[] args) {

        //Ingresamos por teclado
        String _nom = JOptionPane.showInputDialog("Ingrese nombres: ");
        String _ape = JOptionPane.showInputDialog("Ingrese apellidos: ");
        String _esp = JOptionPane.showInputDialog("Ingrese especialidad: ");
        int _cod = Integer.parseInt(JOptionPane.showInputDialog("Ingrese codigo: "));
```

Ahora creamos el objeto, y los valores ingresados por teclado serán puestos en el argumento del objeto creado.

```
package Ejercicio01;

import java.util.ArrayList;
import javax.swing.JOptionPane;

public class EstudianteInfo {

    public static void main(String[] args) {

        //Ingresamos por teclado
        String _nom = JOptionPane.showInputDialog("Ingrese nombres: ");
        String _ape = JOptionPane.showInputDialog("Ingrese apellidos: ");
        String _esp = JOptionPane.showInputDialog("Ingrese especialidad: ");
        int _cod = Integer.parseInt(JOptionPane.showInputDialog("Ingrese codigo: "));

        //creamos el objeto el
        Estudiante e1 = new Estudiante(_nom, _ape, _esp, _cod);
```

Después de haber creado el objeto, esta tendrá los métodos que creamos en la clase Estudiante, primeramente llamamos al método llenar lista, y lo almacenaremos en una lista su retorno, este último servirá para el método del calcular el menor, ya que necesitara de la lista para buscar al menor y su retorno los guardamos en una variable, ahora con la lista y el mínimo será posible ejecutar el método calcular el promedio, ya que será necesario sumar cada elemento de la lista y luego restarle el mínimo para después calcular su promedio.

```
//llamamos aal metodo de llenarLista
ArrayList<Integer> notas = e1.llenarLista();

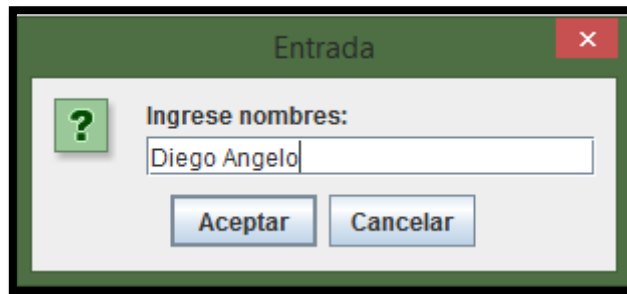
//llamamos al metodo calcular menor
int _min = e1.calcularMenor(notas);

//llamamos al metodo calcularPromedio
double _prom = e1.calcularPromedio(_min, notas);
```

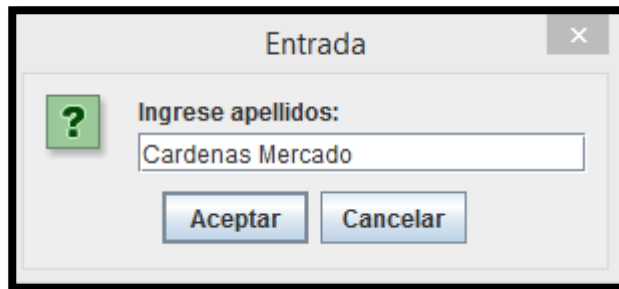
Por ultimo llamamos al método de mostrar todo y así lo hacemos.

```
//llamamos al metodo
e1.mostrarInfo(_prom, notas, _min);
```

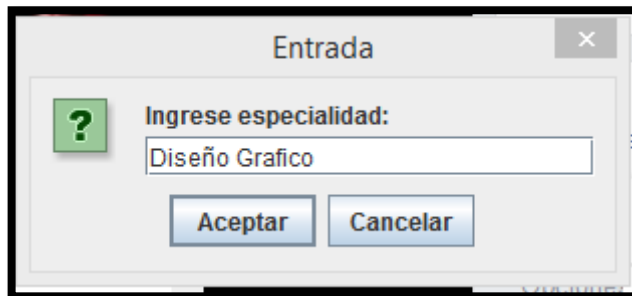
Ejecutamos el código...



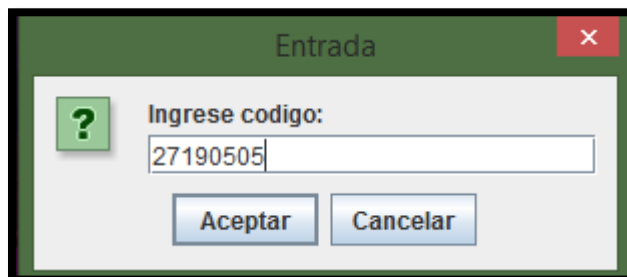
A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese nombres:", a text input field containing "Diego Angelo", and two buttons: "Aceptar" and "Cancelar".



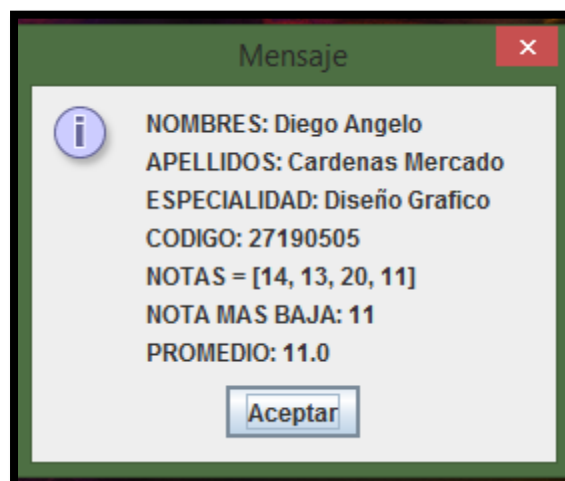
A dialog box titled "Entrada" with a light gray header bar and a gray close button. It contains a green question mark icon, the label "Ingrese apellidos:", a text input field containing "Cardenas Mercado", and two buttons: "Aceptar" and "Cancelar".



A dialog box titled "Entrada" with a light gray header bar and a gray close button. It contains a green question mark icon, the label "Ingrese especialidad:", a text input field containing "Diseño Grafico", and two buttons: "Aceptar" and "Cancelar".



A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese codigo:", a text input field containing "27190505", and two buttons: "Aceptar" and "Cancelar".

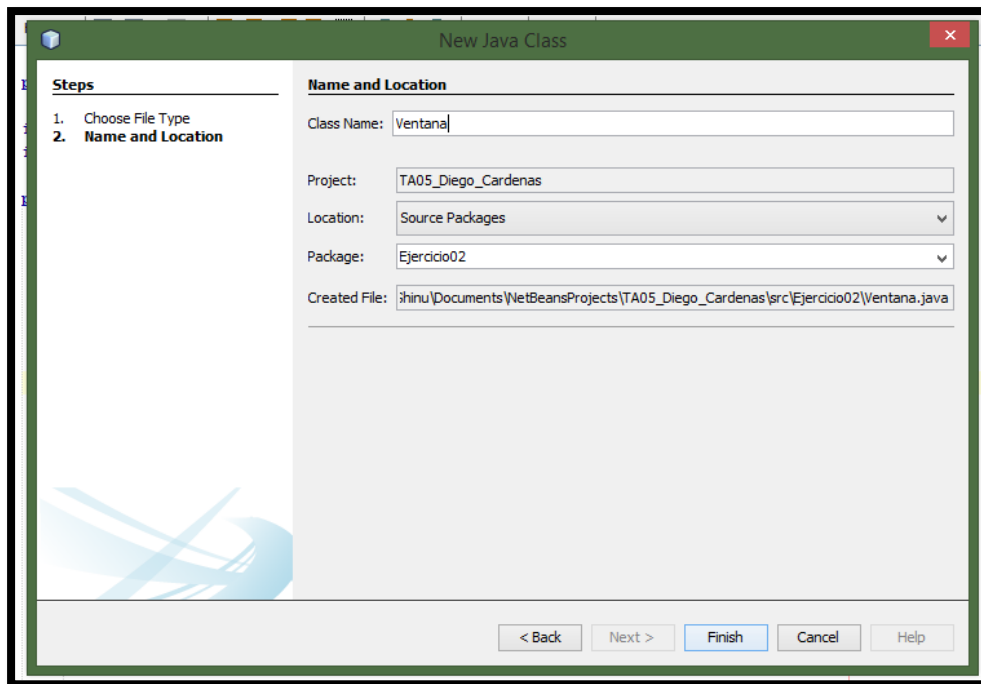


A dialog box titled "Mensaje" with a green header bar and a red close button. It contains a blue information icon and the following text:  
NOMBRES: Diego Angelo  
APELLIDOS: Cardenas Mercado  
ESPECIALIDAD: Diseño Grafico  
CODIGO: 27190505  
NOTAS = [14, 13, 20, 11]  
NOTA MAS BAJA: 11  
PROMEDIO: 11.0  
At the bottom is an "Aceptar" button.



## EJERCICIO 02

Creamos la clase ventana...



Que tipos de atributos tendría la clase ventana?, bueno, para mi fueron, el nombre ya que todos las ventanas tienen, al igual que las opciones, la de borrar, o crear; también puede ser la cantidad de archivos.

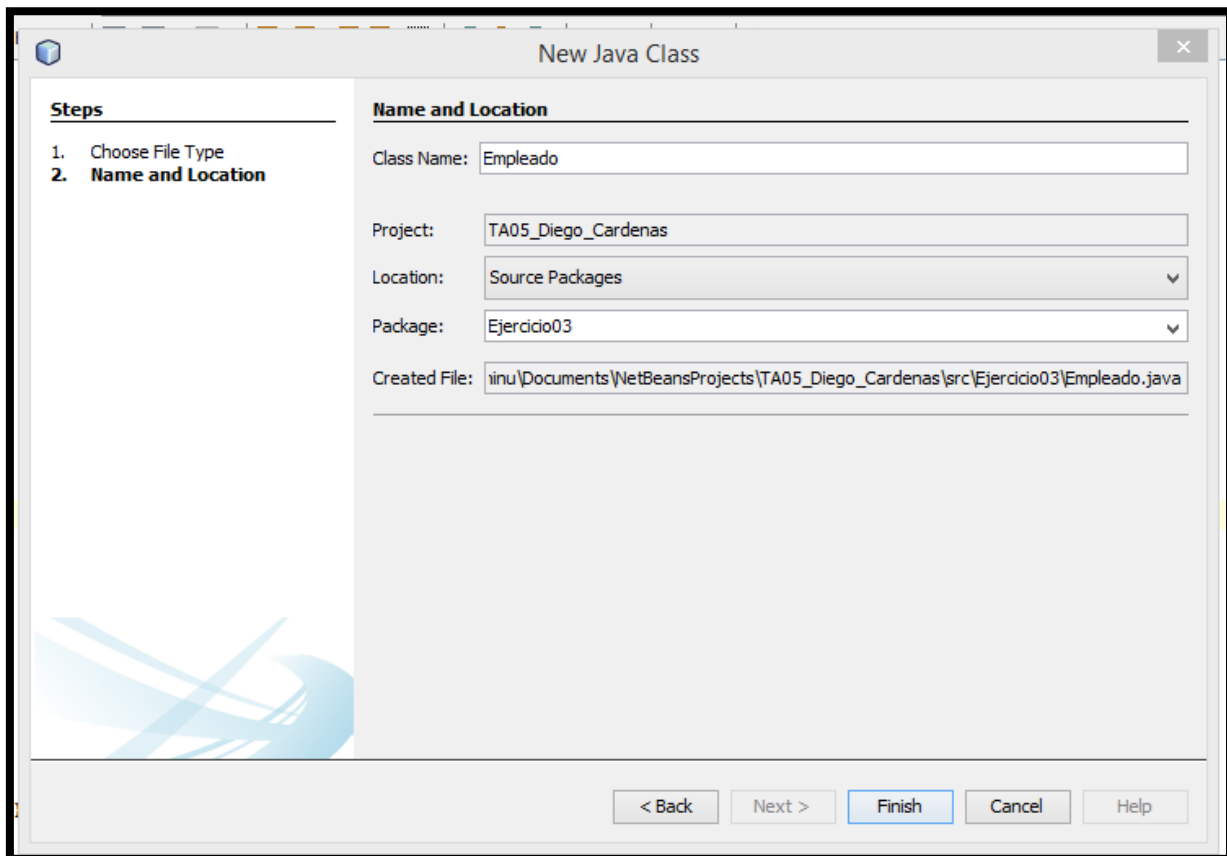
```
public class Ventana {  
  
    //creamos lo atributos  
    private String _nombre;  
    private String _opcion;  
    private int _can_archivos;
```

Como métodos o funcionalidades serían las siguientes: abrir la ventana, cerrar la ventana, eliminar archivos.

```
    //creamos los siguientes metodos  
  
    public void abrirVentana() {  
        System.out.println("Ventana abierta");  
    }  
  
    public void cerrarVentana() {  
        System.out.println("Ventana cerrada");  
    }  
  
    public void eliminarArchivos() {  
        System.out.println("Archivo eliminado");  
    }  
}
```

## EJERCICIO 03

Creamos la clase Empleado como se muestra en la imagen.



Creamos los siguientes atributos según la información que nos da el problema.

```
package Ejercicio03;

public class Empleado {

    //creamos los atributos
    private String _nombres;
    private String _apellidos;
    private int _codigo;
    private String _aLaboral;
    private int _sueldo;
    private int _horas_extra;
    private int _afiliacion_AFP;
```

Creamos el constructor que servirá para crear el objeto en la clase principal.

```
public class Empleado {

    //creamos los atributos
    private String _nombres;
    private String _apellidos;
    private int _codigo;
    private String _aLaboral;
    private int _sueldo;
    private int _horas_extra;
    private int _afiliacion_AFP;

    //creamos el constructor
    public Empleado(String _nombres, String _apellidos, int _codigo,
                    String _aLaboral, int _sueldo, int _horas_extra, int _afiliacion_AFP) {

        this._nombres = _nombres;
        this._apellidos = _apellidos;
        this._codigo = _codigo;
        this._aLaboral = _aLaboral;
        this._sueldo = _sueldo;
        this._horas_extra = _horas_extra;
        this._afiliacion_AFP = _afiliacion_AFP;
    }
}
```

Luego empezaremos creando el método montoHorasExtra, esta nos permitirá calcular cuando es la cantidad que se gana por horas extra, según la fórmula que está en el pdf calcularemos esto y nos retornara.

```
//creamos los metodos:

//monto de horas extra
public double montoHorasExtra() {
    double _m;
    _m = (_sueldo/240) * _horas_extra; //segun la formula del pdf
    return _m;
}
```

Ahora creamos otro método llamado montoDescuento, nos servirá para el cálculo de los descuento según la afiliación, ya sea por AFP, SNP o por salud, y el total de estos tres. Este método no nos retornara nada ya que solo mostrara los valores, por eso ira un void.

```
//monto de los descuentos de AFP y el total de estos
public void montoDescuento() {
    double _desc1 = _sueldo * 11/100; //descuento por AFP
    double _desc2 = _sueldo * 13/100; //descuento por SNP
    double _desc3 = _sueldo * 3/100; //descuento por salud

    System.out.println("Descuento por AFP: " + _desc1 + "\nDescuento por SNP: " + _desc2 + "\nDescuento por salud: " + _desc3);

    double _descT = _desc1 + _desc2 + _desc3; //suma de todos los descuentos

    System.out.println("monto total de descuentos: " + _descT);
}
```

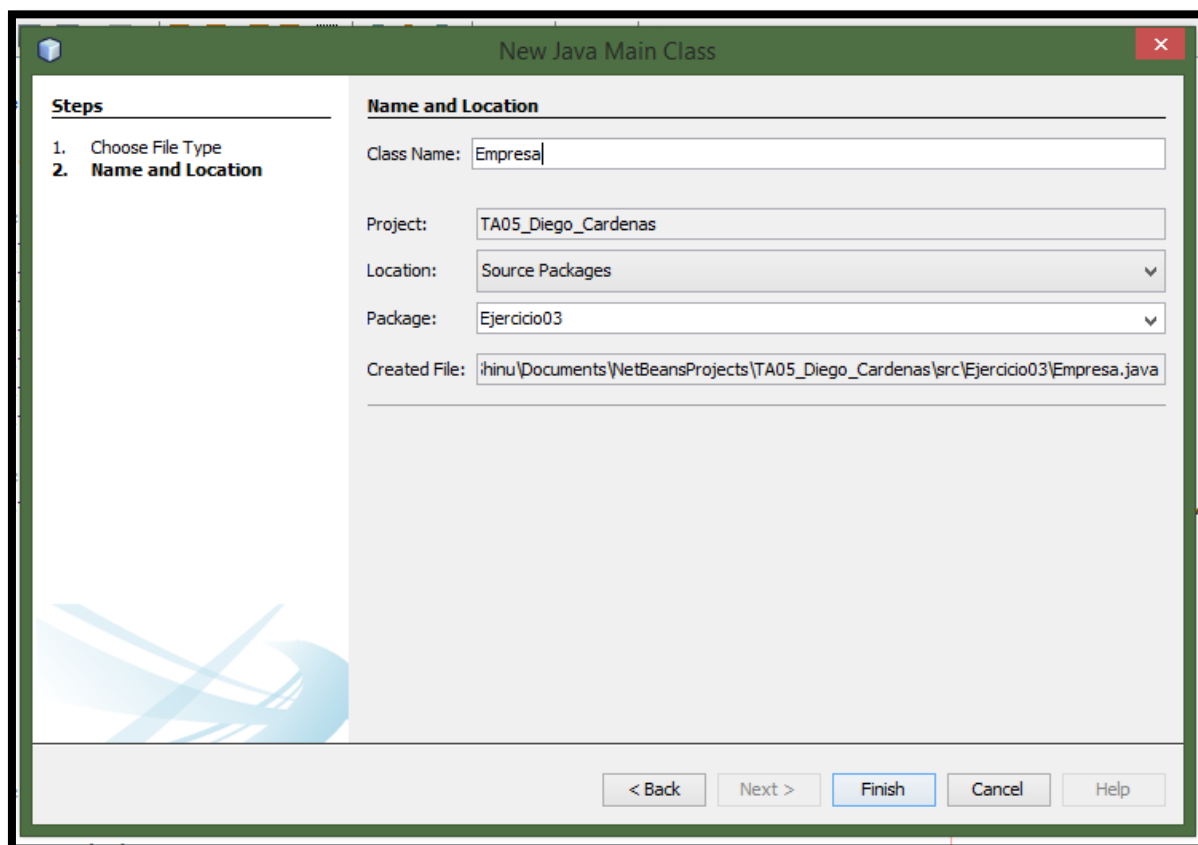
Después creamos otro método, este servirá para calcular el total del sueldo, ósea, la suma del sueldo inicial o neto, con el descuento de horas junto con el descuento según la afiliación, este necesita un parámetro, la cual sería la del monto de las horas extra. Como el anterior, este solo mostrara mas no retornara nada, por eso va un void.

```
//metodo creado para el total del dsueldo
public void totalSueldo(double _m){
    //segun la opcion del la afiliacion
    if(_afiliacion_AFP == 1){
        double _st = _sueldo + _m + (_sueldo * 11/100);
        System.out.println("Sueldo Total: " + _st);
    }else if(_afiliacion_AFP == 2){
        double _st = _sueldo + _m + (_sueldo * 13/100);
        System.out.println("Sueldo Total: " + _st);
    }else if(_afiliacion_AFP == 3){
        double _st = _sueldo + _m + (_sueldo * 3/100);
        System.out.println("Sueldo Total: " + _st);
    }
}
```

Ahora crearemos un último método, este último nos servirá para mostrar todo los datos al usuario.

```
//creacion del metodo para mostrar todo
public void Mostrar(){
    System.out.println("Nombres: " + _nombres + "\nApellidos: " + _apellidos + "\nCodigo: " +
        _codigo + "\nArea Laboral: " + _aLaboral + "\nSueldo: " + _sueldo + "\nHoras Extra: " +
        _horas_extra + "\nAfiliacion(1:AFP, 2:SNP, 3:Salud:) " + _afiliacion_AFP);
}
```

Ahora creamos una clase principal, en mi caso yo la llamare empresa.



Ahora ingresamos los siguiente datos, servirán para el argumento del objeto.

```
package Ejercicio03;

import javax.swing.JOptionPane;

public class Empresa {

    public static void main(String[] args) {
        //ingresamos por teclado todos estos valores
        String _nom = JOptionPane.showInputDialog("Ingrese nombres: ");
        String _ape = JOptionPane.showInputDialog("Ingrese apellidos: ");
        int _cod = Integer.parseInt(JOptionPane.showInputDialog("Ingrese codigo: "));
        String _alab = JOptionPane.showInputDialog("Ingrese area laboral");
        int _suel = Integer.parseInt(JOptionPane.showInputDialog("Ingrese sueldo: "));
        int _hor = Integer.parseInt(JOptionPane.showInputDialog("Ingrese horas extra: "));
        int _afil = Integer.parseInt(JOptionPane.showInputDialog("Ingrese afiliacion(1:AFP, 2:SNP, 3:Por salud): "));
    }
}
```

Creamos el objeto e1 con sus respectivos argumentos, que vendrían a ser los que ingresamos por teclado.

```
package Ejercicio03;

import javax.swing.JOptionPane;

public class Empresa {

    public static void main(String[] args) {
        //ingresamos por teclado todos estos valores
        String _nom = JOptionPane.showInputDialog("Ingrese nombres: ");
        String _ape = JOptionPane.showInputDialog("Ingrese apellidos: ");
        int _cod = Integer.parseInt(JOptionPane.showInputDialog("Ingrese codigo: "));
        String _alab = JOptionPane.showInputDialog("Ingrese area laboral");
        int _suel = Integer.parseInt(JOptionPane.showInputDialog("Ingrese sueldo: "));
        int _hor = Integer.parseInt(JOptionPane.showInputDialog("Ingrese horas extra: "));
        int _afil = Integer.parseInt(JOptionPane.showInputDialog("Ingrese afiliacion(1:AFP, 2:SNP, 3:Por salud): "));

        //creamos el objeto e1
        Empleado e1 = new Empleado(_nom,_ape,_cod,_alab,_suel,_hor,_afil);
    }
}
```

Empezaremos a llamar los métodos de la clase Empleado, el primero será el de monto de horas extra y como esta tiene retorno lo guardamos en una variable.

```
//llamamos este metodo para clacular el monto de horas extra
double _m = e1.montoHorasExtra();
```

Ahora llamamos el método mostrar, que permite mostrar todos los datos que pusimos en el inicio.

```
//llamamos este metodo para mostrar los valores que no requieren calculo o que son puesto directamente
e1.Mostrar();
```

Después de este método se pondrá el siguiente que será el de montoDescuento, este nos mostrara cuales son los descuentos por afiliación seleccionada, y el total de la suma de estos tres.

```
//llamamos al metodo monto de descuento
e1.montoDescuento();
```

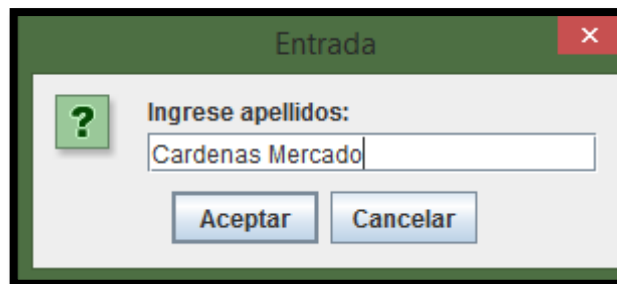
Seguido de este será el de totalSueldo, aquel nos permitirá saber el monto total con descuento y todo.

```
//llamamos al metodo de total de sueldo
e1.totalSueldo(_m);
```

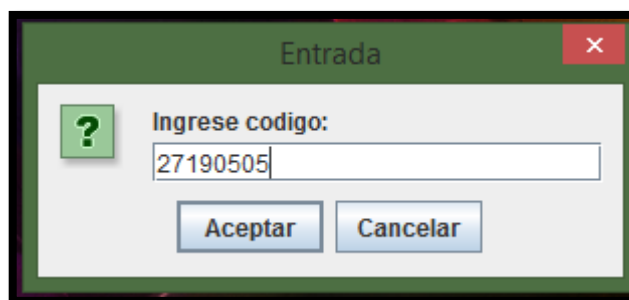
Ejecutamos el proyecto...



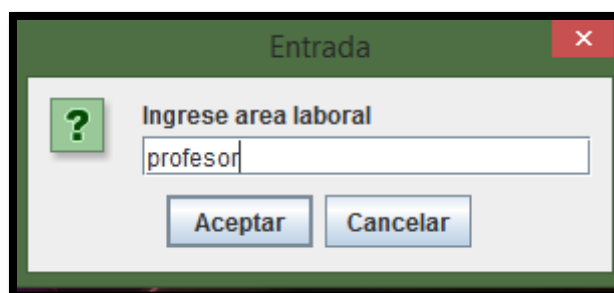
A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese nombres:", a text input field containing "Diego Angelo", and two buttons labeled "Aceptar" and "Cancelar".



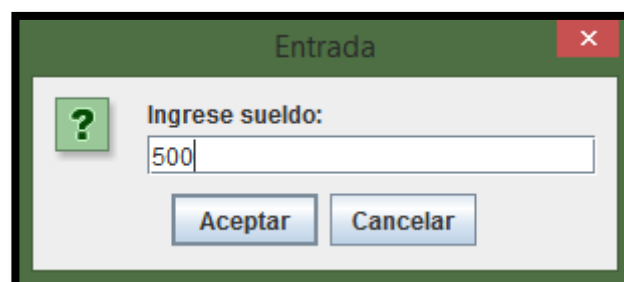
A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese apellidos:", a text input field containing "Cardenas Mercado", and two buttons labeled "Aceptar" and "Cancelar".



A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese codigo:", a text input field containing "27190505", and two buttons labeled "Aceptar" and "Cancelar".



A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese area laboral", a text input field containing "profesor", and two buttons labeled "Aceptar" and "Cancelar".



A dialog box titled "Entrada" with a green header bar and a red close button. It contains a green question mark icon, the label "Ingrese sueldo:", a text input field containing "500", and two buttons labeled "Aceptar" and "Cancelar".

Entrada

?

Ingrese horas extra:

2

Aceptar Cancelar

Entrada

?

Ingrese afiliacion(1:AFP, 2:SNP, 3:Por salud):

3

Aceptar Cancelar

Nombres: Diego Angelo  
Apellidos: Cardenas Mercado  
Codigo: 27190505  
Area Laboral: profesor  
Sueldo: 500  
Horas Extra: 2  
Afiliacion(1:AFP, 2:SNP, 3:Salud:) 3  
Descuento por AFP: 55.0  
Descuento por SNP: 65.0  
Descuento por salud: 15.0  
monto total de descuentos: 135.0  
Sueldo Total: 519.0