Logotipo

Descripción generada automáticamente con confianza baja

**Servicio Nacional de Aprendizaje – SENA**

**Centro de Diseño y Metrología**

**InformeProgramasRealizados**

**Aprendiz:**

**Juan Esteban Castro Escamilla**

**No Ficha 2848530-A**

# Tabla de contenido

[**Tabla de contenido 2**](#_l4sr8za4iadx)

[**Estructurado y Orientado a objetos 3**](#_q7a4wb4imws3)

[**Programación Estructurada: 3**](#_ocei3e6tqiya)

[**Informe Programas Estructurados 4**](#_9n9rpq3l6aos)

[Programa n-1 : NotasEstructurada - PROGRAMA REALIZADO EN CLASE 4](#_kyvopzq72utl)

[Programa n-2 : EmpresaEstructurada - PROGRAMA REALIZADO EN CLASE 5](#_8wrt3fft1lmf)

[Programa n-3 : EstudianteEstructurada - PROGRAMA TRABAJO EN CASA 8](#_fnrpvh9jcbid)

[Programa n-4 : EmpleadoEstructurada - PROGRAMA TRABAJO EN CASA 10](#_hrpkydnux35k)

[**Informe Programas POO 12**](#_gk9ka7nszl7z)

[Programa n-1 : NotasObjetos - PROGRAMA REALIZADO EN CLASE 12](#_sbj2r7q05v01)

[Programa n-2 : EmpresaObjetos - PROGRAMA REALIZADO EN CLASE 14](#_1dppdbj04f0n)

[Programa n-3 : Biblioteca - PROGRAMA REALIZADO EN CLASE (NO SE TERMINÓ, FUE UN EJEMPLO) 18](#_mueyxmqcc3hm)

[Programa n-4 : EstudiantePOO - PROGRAMA REALIZADO EN CASA 20](#_z57eo1r4scaz)

[Programa n-5 : EmpleadosPOO - PROGRAMA REALIZADO EN CASA 23](#_uf54h4k64gc6)

# Estructurado y Orientado a objetos

# **Programación Estructurada:**

* Se basa en el uso de funciones o procedimientos, que son bloques de código organizados secuencialmente.
* El flujo de control sigue una estructura lineal o jerárquica, con instrucciones que se ejecutan de forma secuencial, con bifurcaciones (condicionales) y repeticiones (bucles).
* El objetivo principal es dividir el programa en subrutinas (funciones o procedimientos) que realicen tareas específicas.

**Programación Orientada a Objetos (POO):**

* Se organiza en objetos y clases. Una clase es una plantilla o molde que define las propiedades y comportamientos comunes de los objetos, mientras que los objetos son instancias de esas clases.
* El enfoque principal es modelar el mundo real mediante entidades (objetos) que tienen propiedades (atributos) y comportamientos (métodos).
* La idea es que los objetos interactúan entre sí mediante métodos, y el programa se estructura en torno a estos objetos y sus relaciones.

# 

# Informe Programas Estructurados

## Programa n-1 : NotasEstructurada - **PROGRAMA REALIZADO EN CLASE**

**(Antes de iniciar la clase en donde trabajaremos, vamos a importar la utilidad JOptionPanel la cual nos servirá para poder solicitar por pantalla al usuario que ingrese los diferentes datos que se pedirán mas adelante, cabe recalcar que esto aplicará para todos los ejercicios que se veran despues que importe esta utilidad)**

 // Usamos la palabra reservada import para utilizar la herramienta

(Creamos la clase pública con su respectivo nombre, luego se creará una estructura repetitiva FOR la cual será controlada por una variable llamada est que se le asigna el número 1 y se repetirá hasta que sea igual o menor a 2 es decir, 1 vez mas en este caso) // el ++ permite que se itere de 1 en 1)

(Se crean allí las variables de texto nombre y de decimal (Float) nota1, nota2 y notaFinal)

(Gracias a la utilidad OptionPane podremos mostrar un diálogo de entrada para asignar los valores que allí nos dice tales como nombre del estudiante y sus notas)

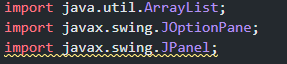
(Finalizando la asignación de los datos va a realizar una función (promedio de las notas 2 obtenidas) en la variable notaFinal y se guardará en esta)

(Por último gracias al estructura de un IF básico, nos permitirá evaluar si la nota final es mayor o igual a 3 y de esta manera si se cumple dicha función mostrará un mensaje de aprobado para el alumno y si no, mostrara un mensaje que el estudiante reprobó)



## Programa n-2 : EmpresaEstructurada - **PROGRAMA REALIZADO EN CLASE**

(Antes de iniciar el programa como se comentó anteriormente, se utilizara la utilidad JOptionPane y JPanel de la misma manera, en este caso en concreto, se agregó la utilidad ArrayList para poder crear un arreglo que se podrá observar después de esta imagen)



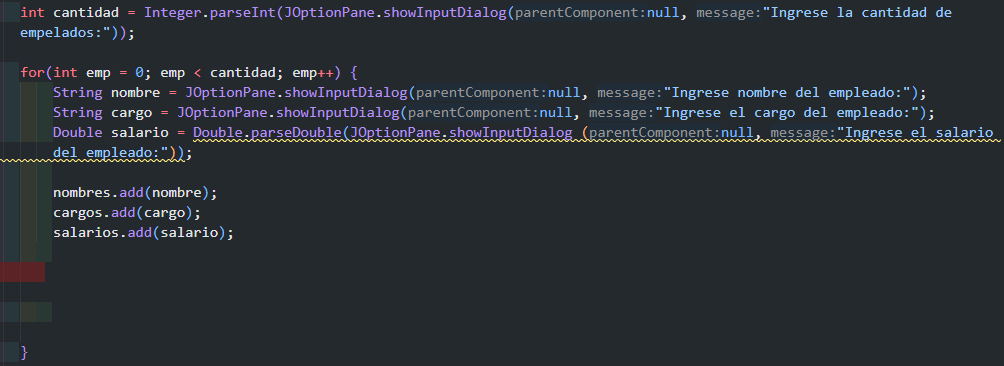
(Utilidad Arraylist permite crear diferentes arreglos para lo que necesitemos, en este caso, se creó 3 diferentes arreglos para nombres, cargos y salarios de los empleados con sus respectivos tipo de dato)



(Creamos una variable de tipo int que se llamó cantidad de la misma forma con la utilidad OptionPane nos permitirá asignar el valor de la cantidad en este caso de los empleados)

(Luego se creo la estructura repetitiva FOR con su variable emp iniciada en 0 y que esta se cumplira hasta que sea menor a la cantidad es decir, al valor que ya se asignó anteriormente)

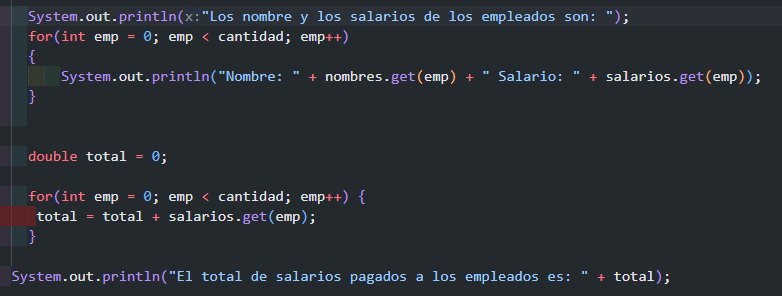
(Mismo proceso de pedir la asignación de los datos de las variables nombre, cargo y salario, con la diferencia que al esto ser un programa con Arreglos, al final del FOR se agregara dichos valores en sus respectivos arreglos y posiciones=



// UN ARREGLO SE CUENTA DESDE EL NUMERO 0, TENER MUY EN CUENTA ESO

(Se muestra por pantalla con el println los nombres y salarios de cada empleado, gracias a una estructura FOR la cual recorrerá los arreglos de nombres y salarios)

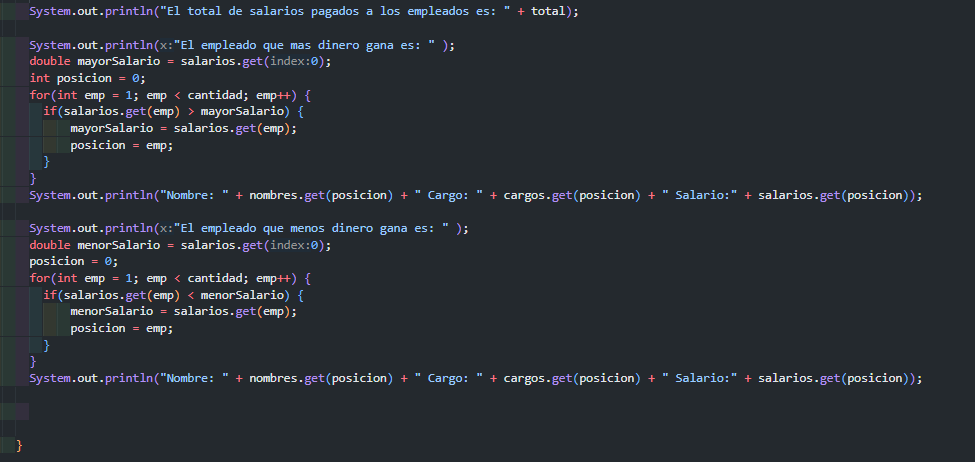
(Luego vamos a mostrar por pantalla y/o consola el total de los salarios pagados, esto se nos ha permitido gracias a la función FOR que almacenera en la variable total que se ha creado cada uno de los salarios y estos se sumarán entre si, gracias a la auxiliar total que se suma así misma cargando cada salario y sumandolo)



(Finalmente mostraremos quien fue el empleado que ganó más dinero, es decir el de mayor salario y también el que gano menos, es decir el de menor salario)

(Creamos una variable mayorSalario que nos permite cargar la posición del primer salario es decir el índice 0 del arreglo salarios y luego otra variable auxiliar llamada posición que nos permitirá dentro del for saber si dicho salario guardado es mayor a esta sucesivamente hasta encontrar el mayor de todo y guardarlo finalmente en el variable mayorSalario)

(Asi mismo para el de menor salario pero en este caso, evaluará el menor de estos)



## Programa n-3 : EstudianteEstructurada - **PROGRAMA TRABAJO EN CASA**

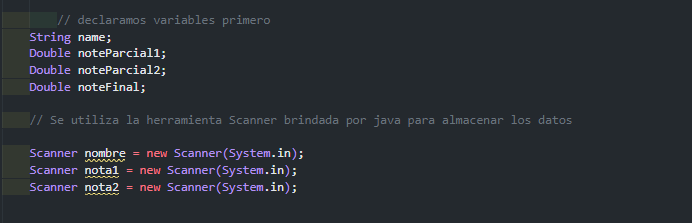
(Mismo programa mostrado en NotasEstructurada con la diferencia única que se utilizó la herramienta Scanner para asignar los valores a las variables)



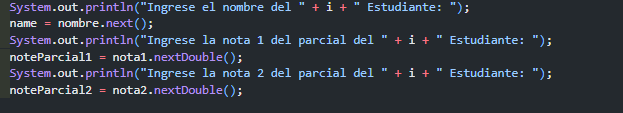
// Se declaro las variables a parte para usar dicha utilidad Scanner

// Scanner permite asignar los valores desde la propia terminal que nos brinda el IDE y no desde una interfaz, permitiendo realizar el trabajo de manera más eficaz

// El scanner tendrá su propias variables las cuales guardaran la asignación de las variables principales

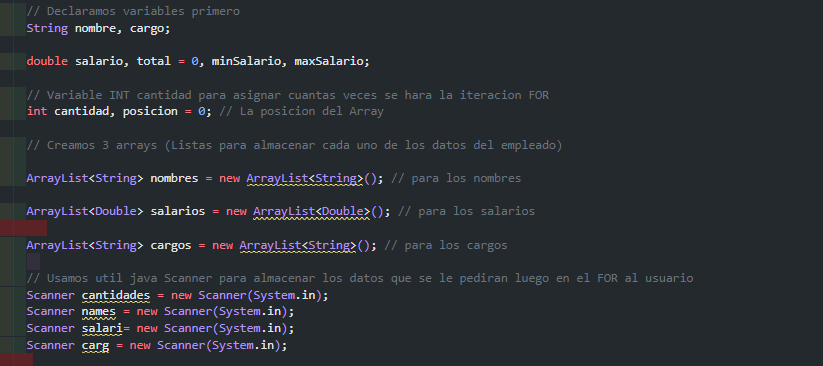


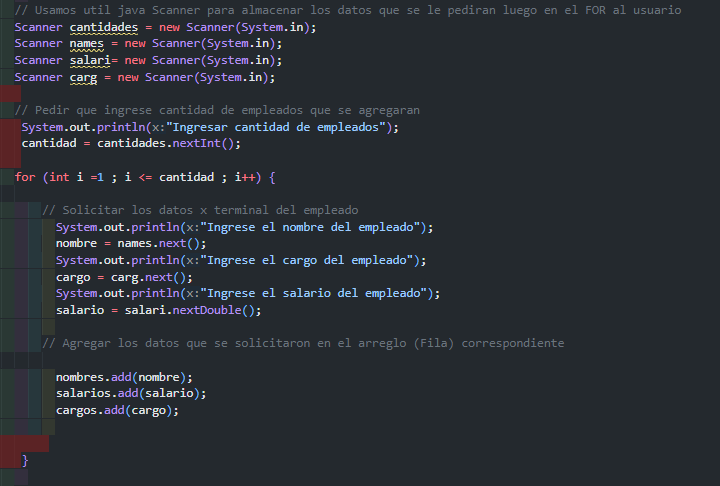
// A diferencia del JOptionPane, aquí utilizamos println y luego abajo de este la variable principal y esto será igual al scanner nombre y el next que nos permite indicar qué tipo de datos se asigna y siempre debe ponerse para evitar errores o que el programa no funciona)

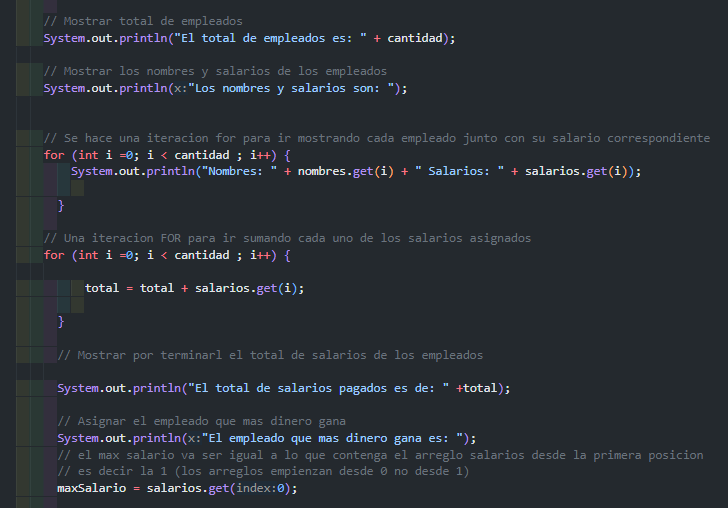


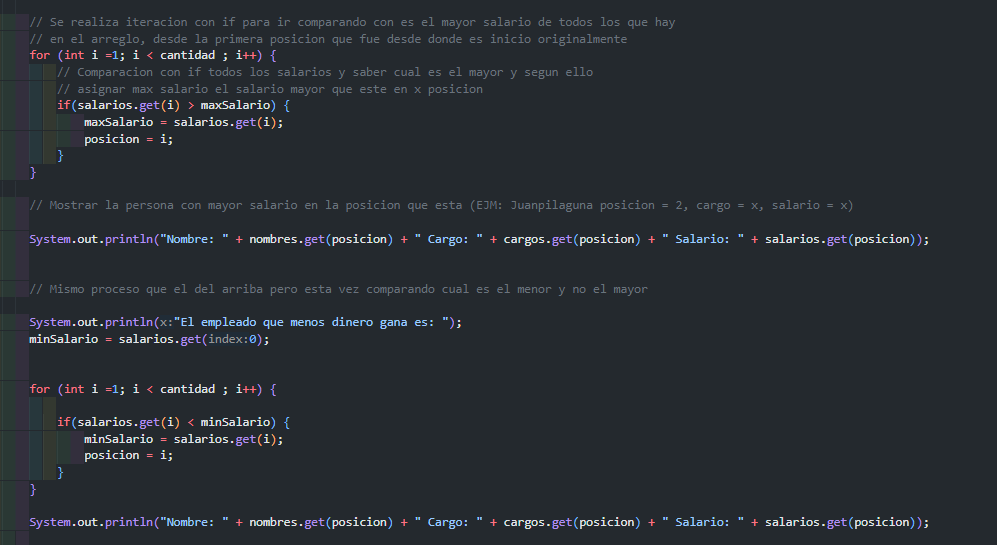
## 

## Programa n-4 : EmpleadoEstructurada - **PROGRAMA TRABAJO EN CASA**







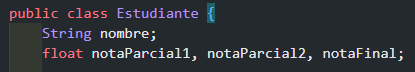


# Informe Programas POO

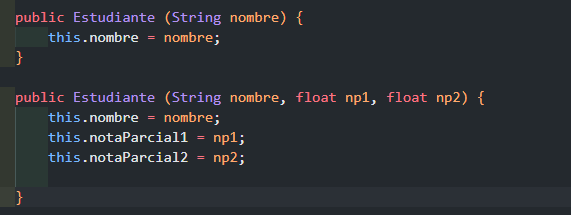
## Programa n-1 : NotasObjetos - **PROGRAMA REALIZADO EN CLASE**

(En este caso el primer paso fue crear una clase la cual tuvo de nombre Estudiante)

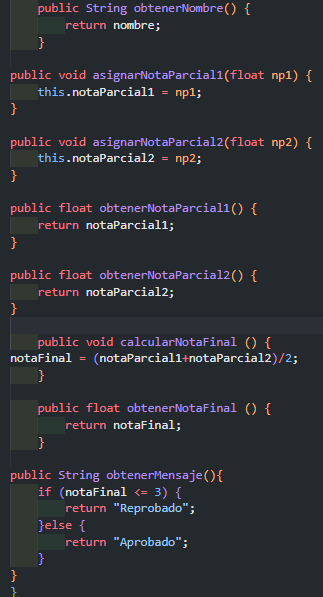
(Dicha clase tuvo los siguientes atributos: nombre, notaParcial1, notaParcial2, notaFinal)



(Se crea los respectivos constructores y se indica cuáles datos o atributos se van a utilizar específicamente con el this)

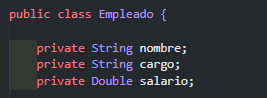


(Creamos los respectivos GET y SET en este caso con la palabra en clave obtener y asigna para cada uno de los atributos que hemos definido desde el principio luego, creamos una función para calcular el promedio es decir la notaFinal del estudiante y dicha función se guardara en la variable notaFinal y del mismo modo se obtiene dicho resultado y finalmente se creó el obtenerMensaje que nos permite saber por medio de una condición quien aprobo o no)

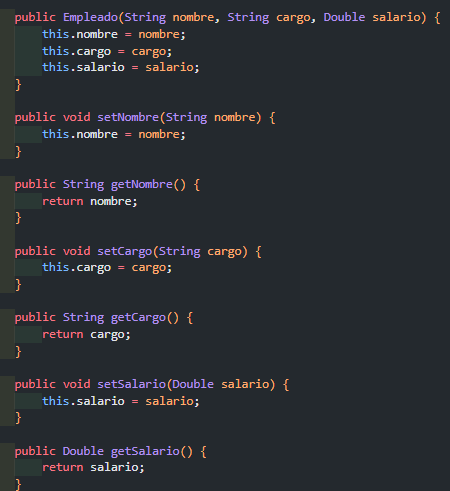


## Programa n-2 : EmpresaObjetos - **PROGRAMA REALIZADO EN CLASE**

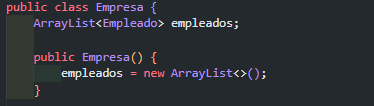
(Se creó principalmente la clase Empleado la cual contiene los siguientes atributos: nombre, cargo, salario)



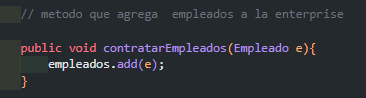
(Creación del respectivo constructor de empleado y sus SETTERS y GETTERS los cuales nos permite obtener y asignar dichos valores)



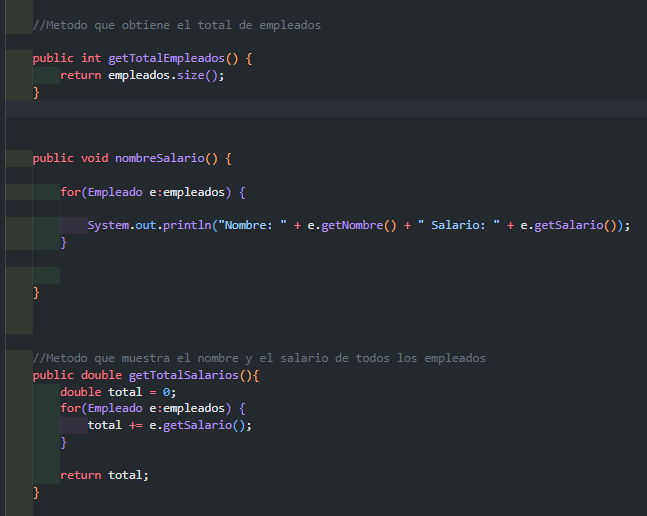
(Luego se crea la siguiente clase para la contratación de dicho empleado a la empresa, dicha clase no contiene ningún atributo y principalmente se creó un ArrayList de tipo empleados para guardar dicha información en un Array de tipo ArrayList)



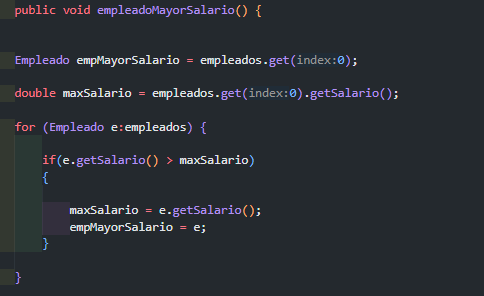
(Luego de declarar el ArrayList se crea un método que nos permite agregar a los empleados la cual se abrevia con la letra e de empleado y se agregaran en el array empleados)



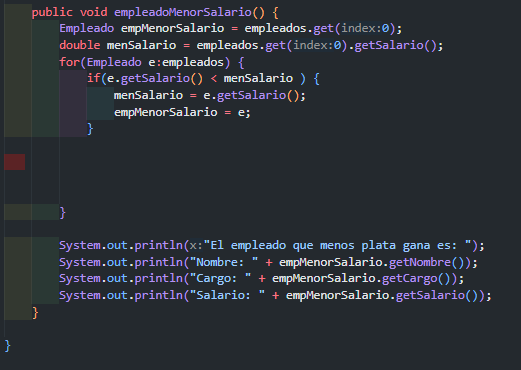
(Creamos los metodos getTotalEmpleados la cual nos permite conocer el tamaño total del array es decir la cantidad de lo empleados, nombreSalario para mostrar los nombres y salarios de cada empleado digitado y el getTotalSalarios que nos permite conocer por medio de una variable auxiliar llamada total que comienza en cero y se va sumando simultáneamente con cada uno de lo salarios y se guarda allí en total y se retornal)



(Misma función mostrada en los ejercicios de estructurada, básicamente se asigna al empMayorSalario la posición del empleado luego la variable auxiliar maxSalario guardará la posición y el salario de este, luego en la condición if evalúa si ese salario en dicha posición es mayor a maxSalario y así se recorre hasta que encuentre el mayor de todos y lo deje asignado en la variable empMayorSalario)

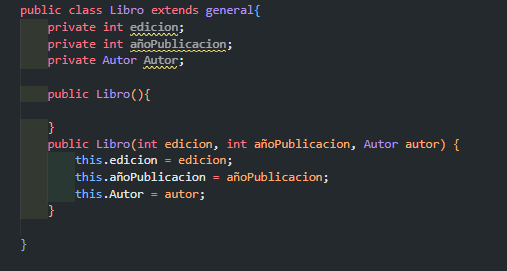


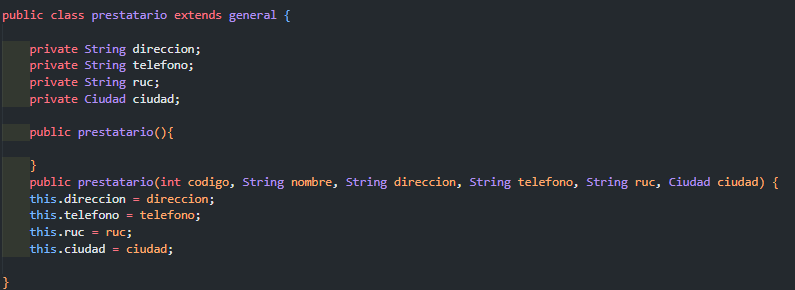
(Misma función que la anterior, solamente cambia las variables y que se evalúa si es menor)

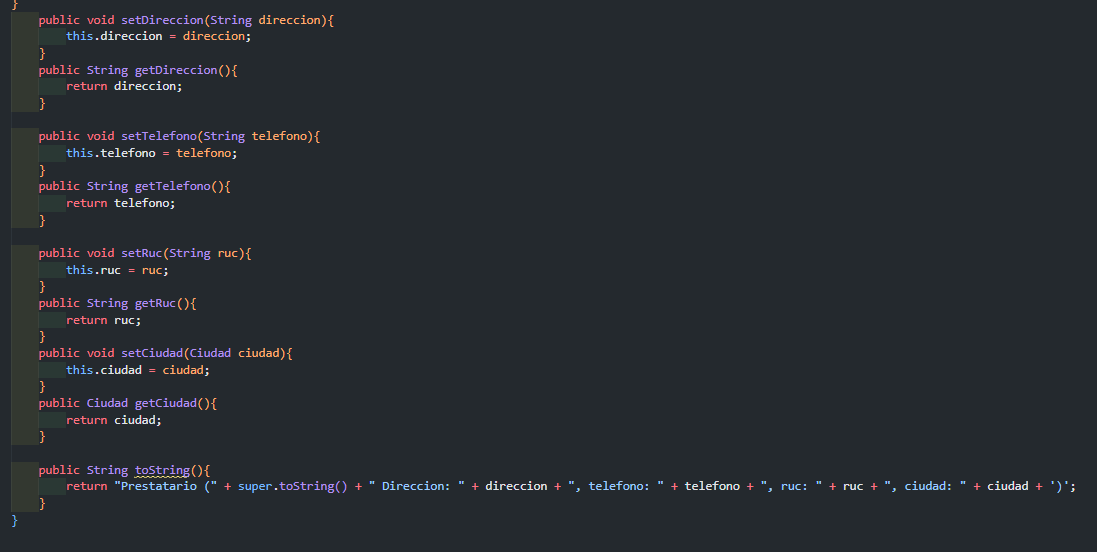


## Programa n-3 : Biblioteca - **PROGRAMA REALIZADO EN CLASE (NO SE TERMINÓ, FUE UN EJEMPLO)**

(En este ejemplo se creó las clases Libro y Prestatario con los siguientes métodos y atributos)

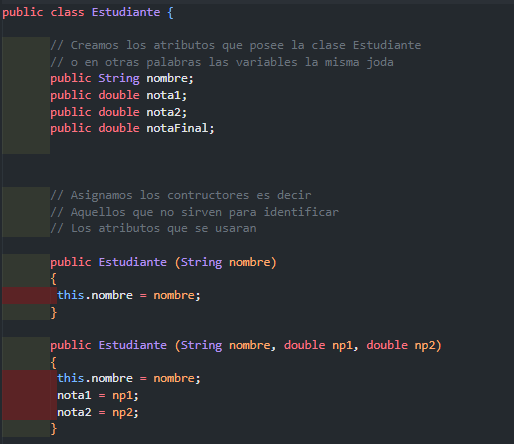


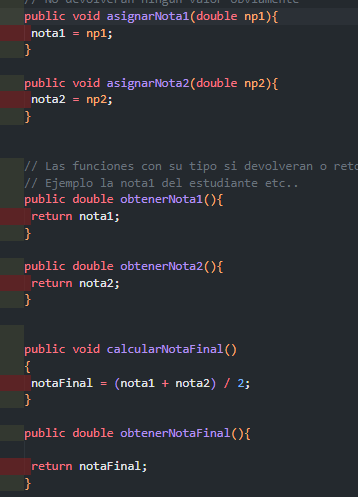


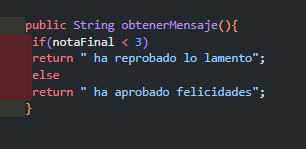


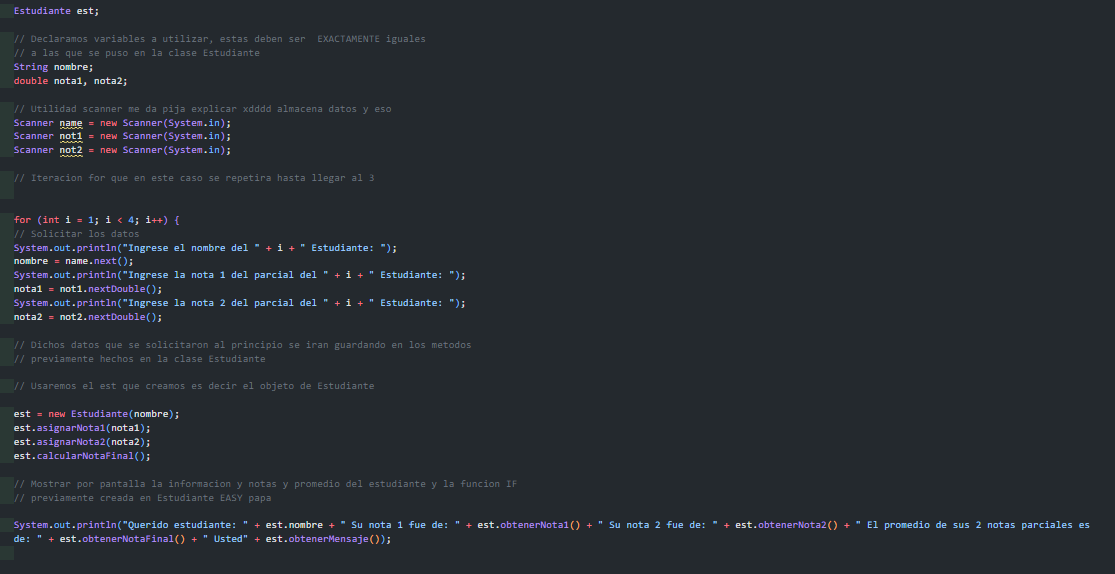
## Programa n-4 : EstudiantePOO - **PROGRAMA REALIZADO EN CASA**

(Mismo ejercicio realizado en clase solo con algunos diferencias, pero básicamente hace la misma función)









## Programa n-5 : EmpleadosPOO - **PROGRAMA REALIZADO EN CASA**

(Mismo ejercicio que el de empresa, con algunos cambios mínimos. Pero proporciona la misma función y ejercicio)

