

Unidad 2. Métodos y estructuras de control

**MATERIA: Programación orientada a objetos I**

25 de abril de 2020

Guadalupe García Vázquez

MATRICULA: ES1921008556

Desarrollo de Software

**Docente: Claudia Erika González Gómez**

Contenido

[**Introducción** 3](#_Toc38802870)

[Desarrollo, indicaciones de la actividad 5](#_Toc38802871)

[**Conclusiones** 10](#_Toc38802872)

[**Bibliografía** 11](#_Toc38802873)

**Actividad: Actividad 2. Detección de errores**

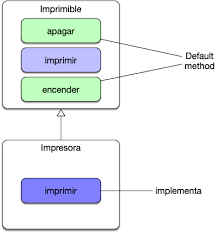
# **Introducción**

Recordemos que a programación orientada a objetos implementa cuatro conceptos:

* La abstracción es un proceso mental de extracción de las características esenciales.
* La encapsulación es ocultar los detalles que dan soporte a un conjunto de características esenciales de una abstracción.
* La modularidad es descomponer un sistema en un conjunto de partes.
* La jerarquía es un proceso de estructuración de varios elementos por niveles.

Estos conceptos contienen los siguientes elementos: clases y objetos, atributos y estado, métodos y mensajes, herencia y polimorfismo.

A continuación abordaremos el elemento de **Método.**



**MÉTODOS Y MENSAJES**

***Operación:*** *f. Mat. Conjunto de reglas que permiten, partiendo de una o varias cantidades o expresiones, llamadas datos, obtener otras cantidades o expresiones llamadas resultados.*

Un método define una **operación** sobre un objeto.

En general, realizan dos posibles acciones: consultar el estado del objeto o modificarlo.

Los métodos disponen de parámetros que permiten delimitar la acción del mismo. Nos podemos encontrar con diversos tipos de métodos:

♣ Consultan o modifican un atributo, normalmente nos referenciaremos a ellos como: getters & setters

♣ Realizan operaciones sobre el conjunto de atributos, calculando valores o realizando modificaciones

♣ Inicializan los atributos al principio del ciclo de vida, o liberan los recursos al final del ciclo; nos referiremos a ellos como constructores o destructores

**Existen dos tipos cuando se habla de ellos:**

**• Métodos que retornan valores:**

Se espera recibir alguna respuesta o resultado de alguna operación realizada por las sentencias que conforman su cuerpo.

Para que un método pueda regresar valores, deben estar involucrados varios factores que le ayuden o “alimenten”, para poder realizar dichas operaciones

**Recepción de parámetros:** Los parámetros de un método se pueden entender como los valores que éste recibe desde la parte del código donde es invocado. Los parámetros pueden ser de tipos primitivos (int, double, float, entre otros) u objetos.

*Ejemplo: public int obtener\_potencia(int base, int potencia){ …*

*}*

**Retorno de parámetros**: Cuando el método de nuestro interés ya ha recibido (o consigue) los elementos necesarios para poder realizar sus funciones, retornará el resultado de aplicarlas sobre los parámetros recibidos.

Se deberá indicar en la declaración del método qué tipo de dato retornará, y señalar en el cuerpo de sus sentencias la palabra reservada return (con esto se precisa que regresará).

*Ejemplo: public int obtener\_potencia(int base, int potencia){ int resultado = Math.pow(base, potencia); return resultado;*

*}*

**• Métodos que no retornan valores.**

Ejemplo: *void nombre\_método([parámetros]){*

*sentencia1;*

*sentencia2;*

*sentenciaN;*

*}*

Los métodos se invocan (llaman) por el nombre definido en su declaración, y se deberá pasar la lista de argumentos entre paréntesis.

Ejemplo, se definirá un método que imprima a pantalla un mensaje para el usuario.

*public void saludo\_usuario(){ String cadena\_Saludo = “Hola usuario, feliz viaje en Java”;*

*System.out.println(cadena\_Saludo);*

*System.out.println(“Mensaje impreso desde un método”);*

*}*

Cuando ya se ha definido el método y sus sentencias son las necesarias y adecuadas, se hace una invocación a él o, mejor dicho, se invocan las sentencias que conforman el cuerpo del método.

Se realiza de la siguiente manera:

*public static void main(String[ ] args){*

*saludo\_usuario();*

*}*

La salida será la siguiente:

Hola usuario, feliz viaje en Java

Mensaje impreso desde un método.

**Actividad: Actividad 2. Detección de errores**

Desarrollo, indicaciones de la actividad**:**

Se desea realizar una calculadora con las cinco operaciones estándar de Java (suma, resta, multiplicación, división y módulo) y tendrás que:

1. Analizar el planteamiento del caso e identificar qué se requiere lograr mediante los métodos

Para llevar a cabo el planteamiento hay que establecer todos los componentes del programa para el desarrollo de una calculadora, con 5 funciones; suma, resta, multiplicación, división y módulo).

1. Determinar el tipo y cantidad de métodos que se deben desarrollar para resolver el problema dado. Posteriormente, elaborarlos por escrito.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Valor retorno | Método | Parámetro, tipo dato |
| 1 | Suma | *float* | *suma=numero1+numero2;* | *float numero1, numero2* |
| 2 | Resta | *float* | *resta=numero1-numero2;* | *float numero1, numero2* |
| 3 | Multiplicación | *float* | *mul=numero1\*numero2;* | *float numero1, numero2* |
| 4 | División | *float* | *div=numero1/numero2* | *float numero1, numero2* |
| 5 | Modulo | *float* | *modulo=numero1%numero2;* | *float numero1, numero2* |

1. Desarrolla el código de los métodos en Java e identifica los posibles errores al ejecutar tu programa

No declarar valores: De los errores principales está el no hacer la declaración correcta de las variables/constantes, ya que es a partir de este que podemos llamarlos para ejecutar los métodos, y si estos no están bien o no están declarados, el sistema nos mostrará error.

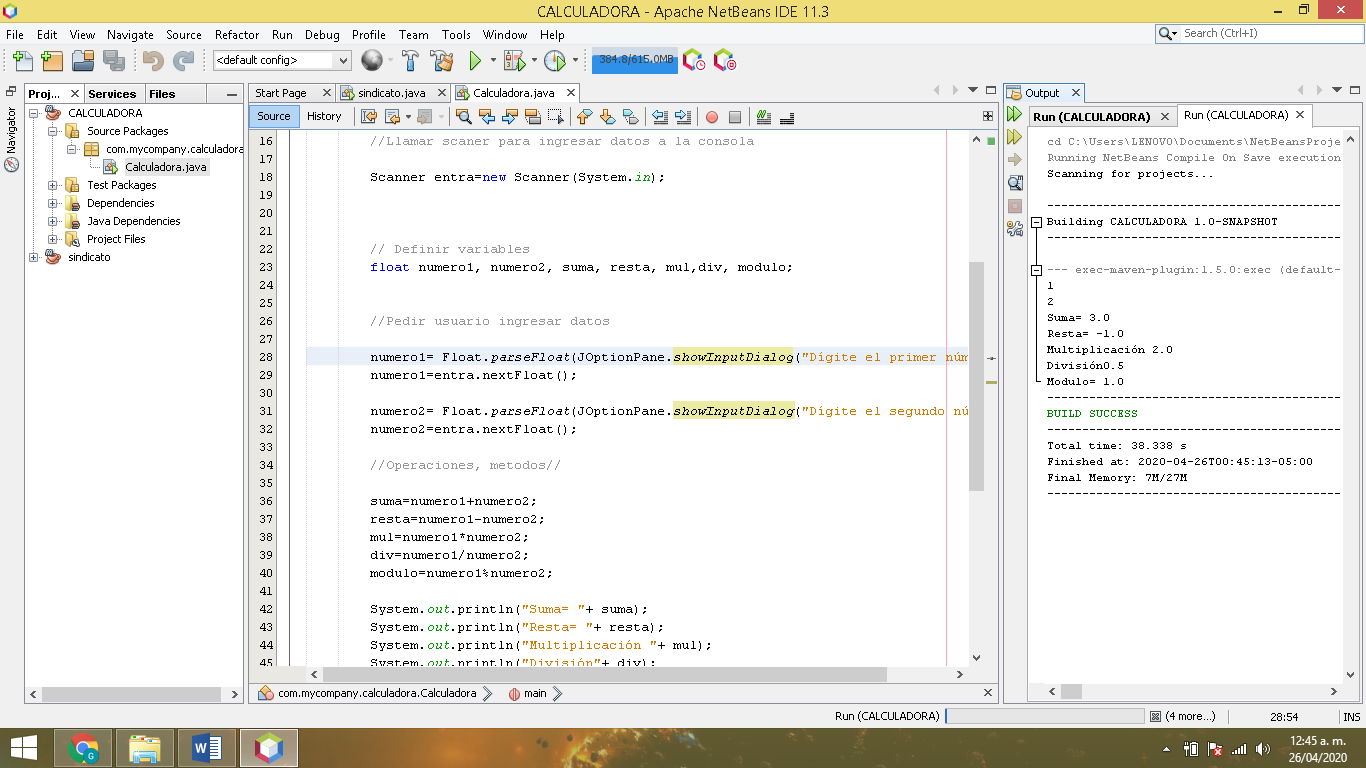
Tipo de dato: Diferencias entre el tipo de dato declarado y la forma de cómo es llamado en el método. Además de no elegir el tipo de dato correcto, por ejemplo en el caso de una división si estamos trabajando con datos enteros, no podremos utilizar el número racional, por lo que el resultado no sería el preciso.

Métodos: no llevar a cabo de forma adecuada el planteamiento del método, provocados por errores de sintaxis o argumentos.

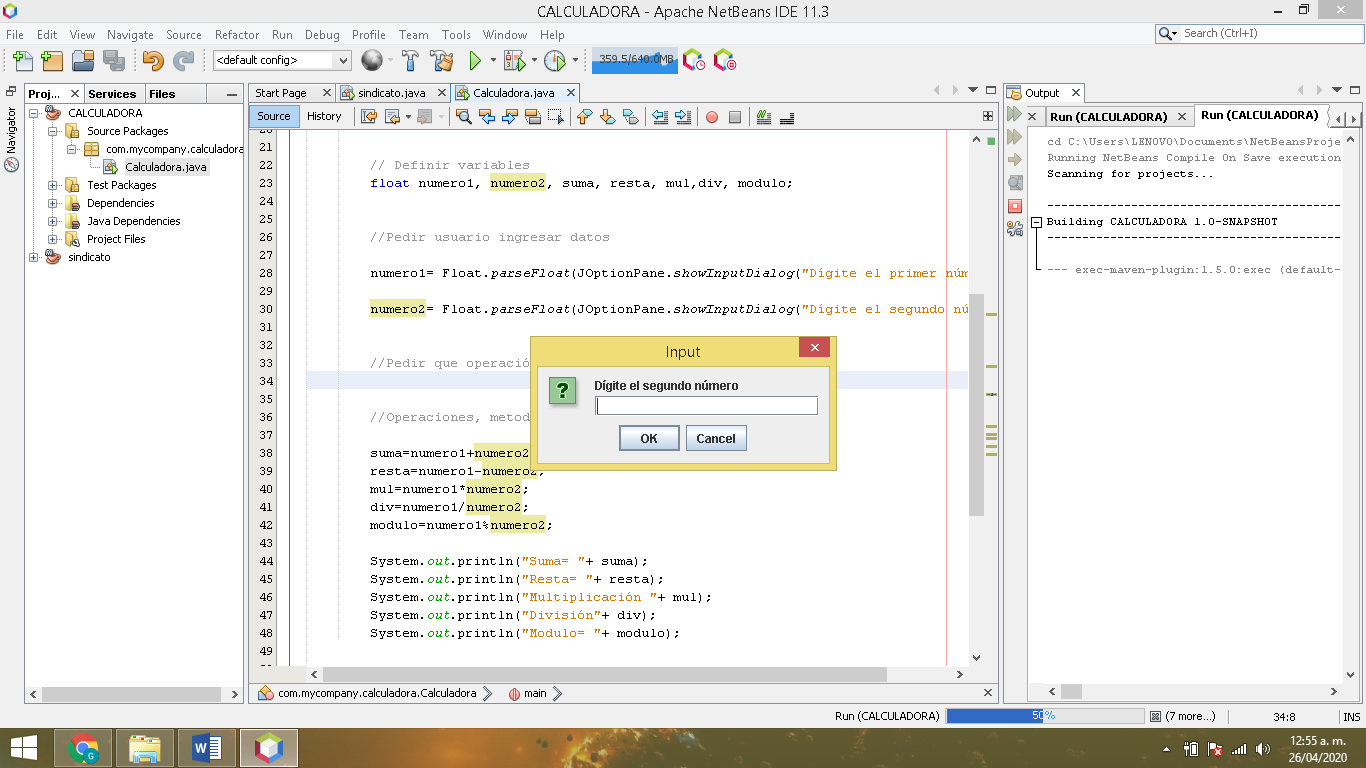
1. Una vez identificados y solucionados los errores, comenta en las líneas de código si estas utilizando la recepción, retorno o paso de parámetros.

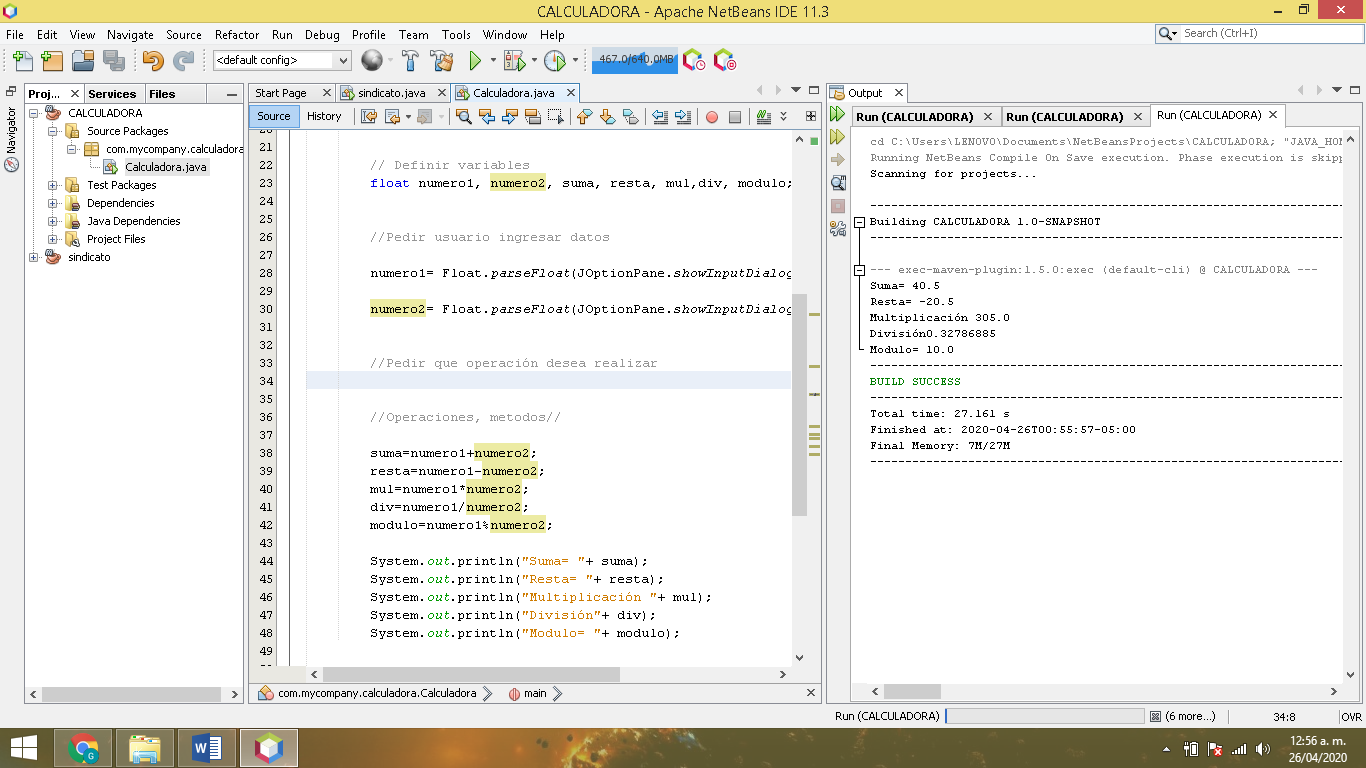
A continuación mostraré los errores que tuve en la ejecución del programa.

Estaba mezclando dos tipos de llamados de impresión, tanto el scanner como el panel, por lo que los resultados de las operaciones no se mostraban correctamente.

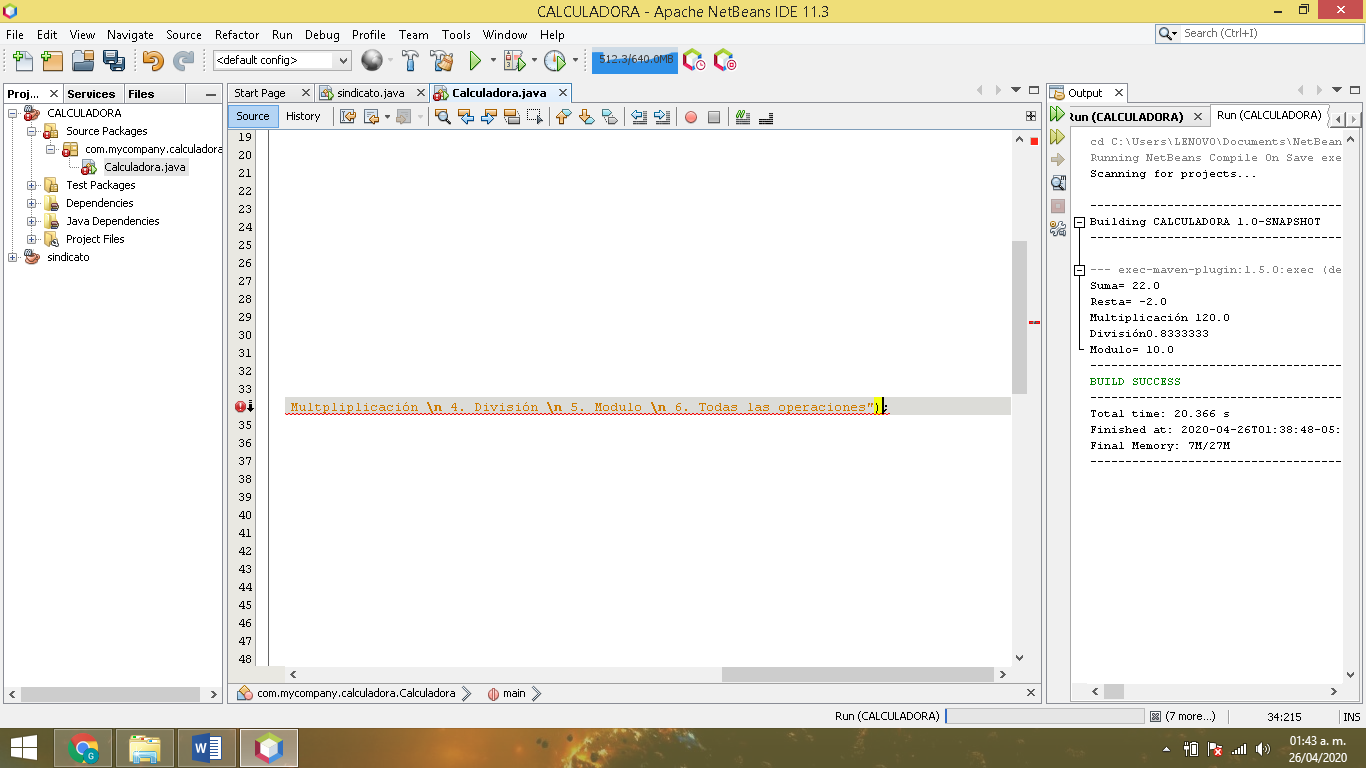


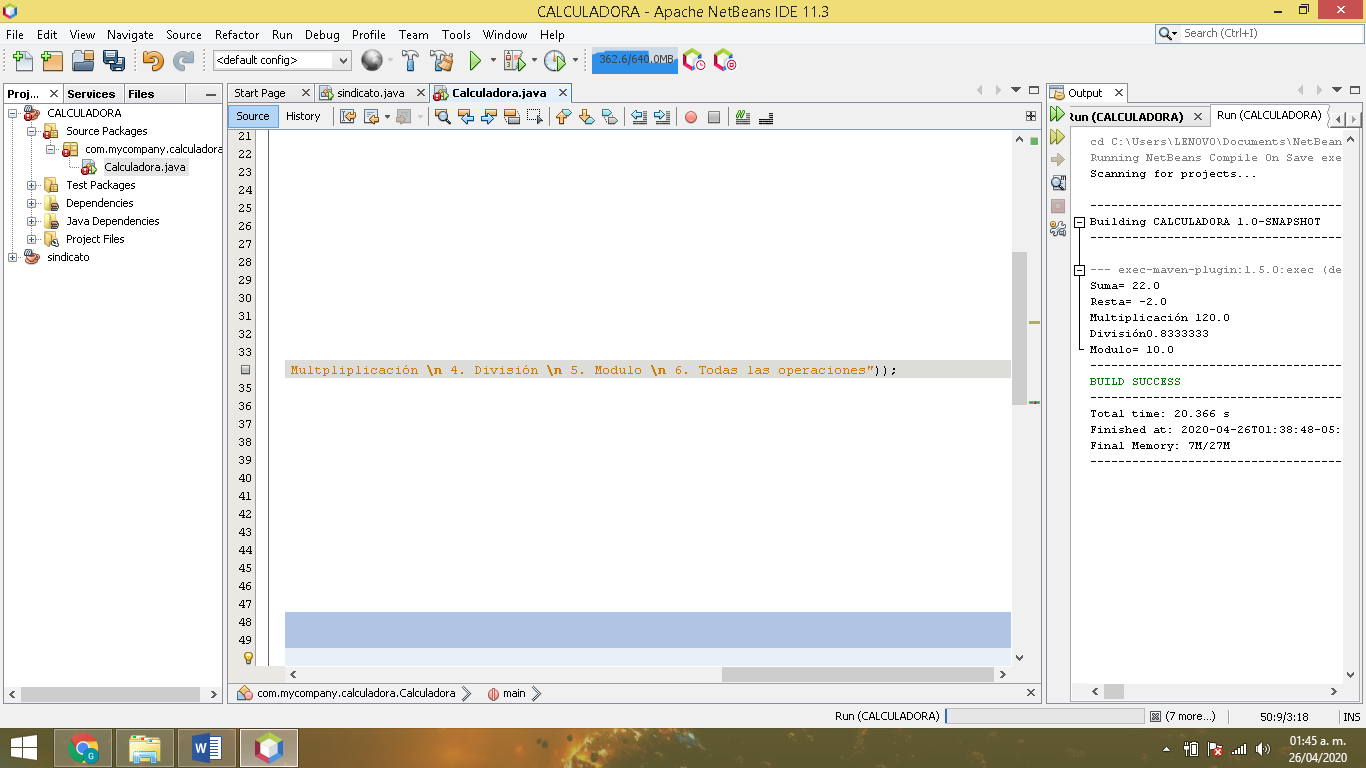
Eligiendo mostrarlo por el panel.



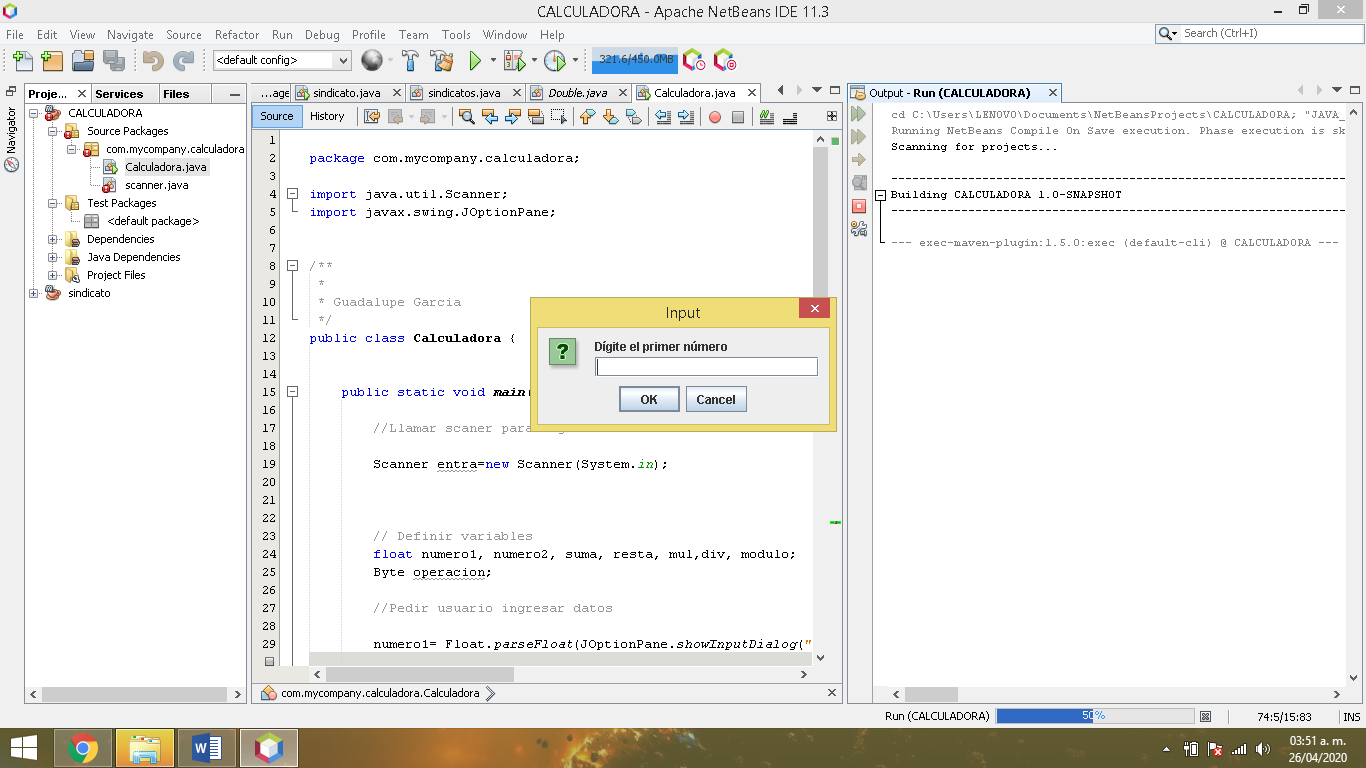


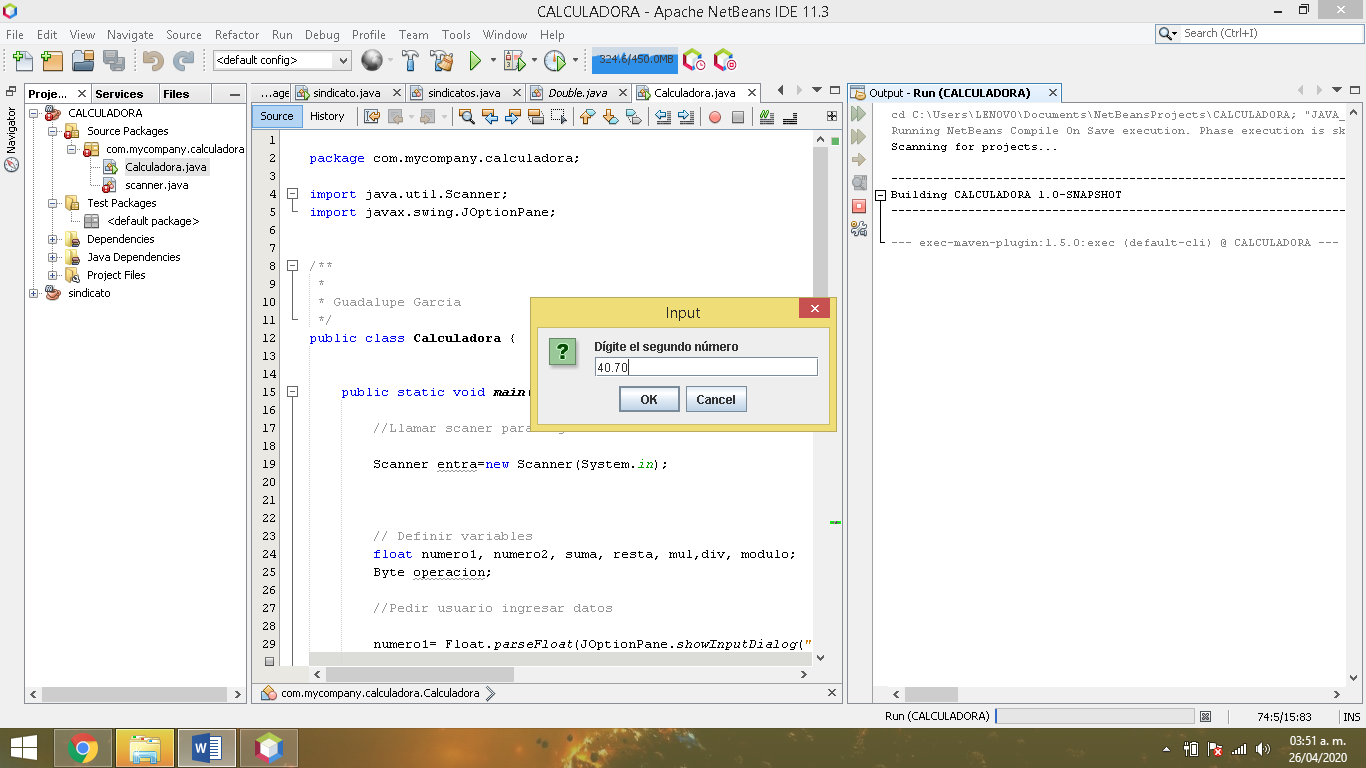
Para un mensaje al usuario, el sistema me marcaba error, debido a que me hacía falta un “)” para cerrar el argumento.

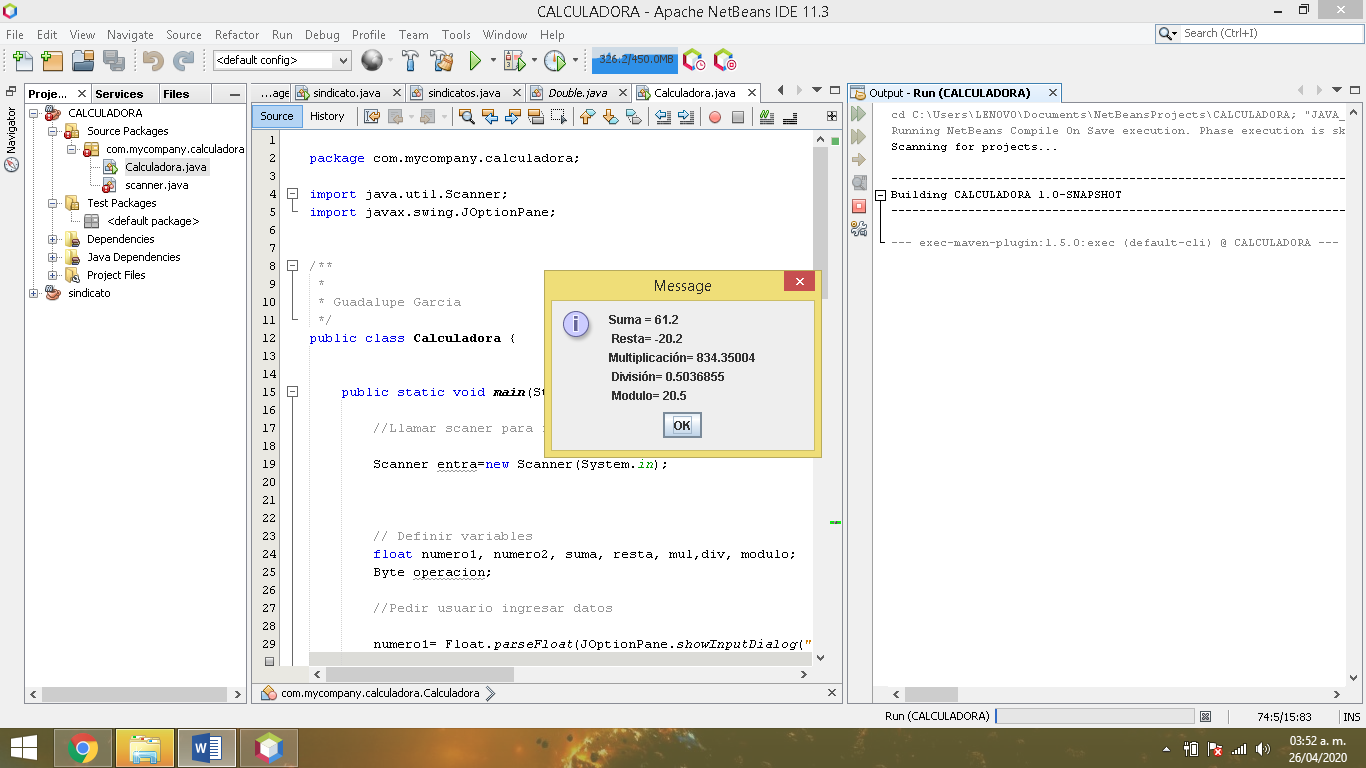




Después de corregir los errores, el desarrollo del programa quedó de la siguiente forma.







### **Conclusiones**

En este siguiente tema hemos abordado los métodos, que forman parte importante de la programación orientada a objetos. Con la práctica me voy asociando más a los conceptos. Como su nombre bien lo indica, la programación orientada a objetos tiene como pieza clave las declaraciones de los valores de las variables y constantes (objetos) que hay a lo largo del sistema para que puedan ser llamadas por los métodos y ejecuten el proceso u operación adecuados.

Se desarrolló una calculadora, pero para comenzar los procesos había que definir todos los elementos que hay en el caso e identificar qué tipo de dato son los más adecuados para los resultados que queremos. En este caso queremos la precisión de resultados, y la posibilidad de que el usuario pueda ingresar cualquier número, por lo que utilice el tipo Float.

Declarar todas las variables y constantes son importante, así como su tipo, para evitar errores por no declararlos.

Uno de los problemas con los que encontré es errores en la escritura del lenguaje y la mezcla entre dos tipo de llamado para imprimir resultados. Así que solo utilice uno solo, el llamado al Panel, para hacerlo visualmente más atractivo.

#### **Bibliografía**

Real Academia Española. (2020). *Diccionario de la lengua española*. Obtenido de https://dle.rae.es/

ATS, P. (Dirección). (2017). *25. Programación en Java || Condicionales || Ejercicio - Calculadora aritmética Obtenido de: https://www.youtube.com/watch?v=YfW6pXiFoWI* [Película].

novato, P. (Dirección). (2019). *Curso de Java Netbeans Completo [09.- Operadores matemáticos] Obtenido de https://www.youtube.com/watch?v=bjLx0bhm5eQ&list=PLCTD\_CpMeEKTT-qEHGqZH3fkBgXH4GOTF&index=12* [Película].

Universidad Abierta y a Distancia de México. (s.f.). *Programación orientada a objetos I*. Obtenido de Unidad 2. Métodos y estructuras de control: https://ceit.unadmexico.mx/contenidos/DCEIT/BLOQUE2/DS/02/DPO1/U2/descargables/DPO1\_U2\_Contenido.pdf (Consultado 23/03/20)

García, J. R. (20000). *Aprenda Java como si estuviera en primero*. Obtenido de Recuperado de: http://ocw.uc3m.es/cursos-archivados/programacion-java/manuales/java2-U-Navarra.pdf/view (26/04/20)

*package com.mycompany.calculadora;*

*import java.util.Scanner;*

*import javax.swing.JOptionPane;*

*/\*\**

*\**

*\* Guadalupe Garcia*

*\*/*

*public class Calculadora {*

*public static void main(String[] args) {*

*//Llamar scaner para ingresar datos a la consola*

*Scanner entra=new Scanner(System.in);*

*// Definir variables*

*float numero1, numero2, suma, resta, mul,div, modulo;*

*Byte operacion;*

*//Pedir usuario ingresar datos*

*numero1= Float.parseFloat(JOptionPane.showInputDialog("Dígite el primer número"));*

*numero2= Float.parseFloat(JOptionPane.showInputDialog("Dígite el segundo número"));*

*//Operaciones, metodos//*

*suma=numero1+numero2;*

*resta=numero1-numero2;*

*mul=numero1\*numero2;*

*div=numero1/numero2;*

*modulo=numero1%numero2;*

*//Imprimir resultados*

*JOptionPane.showMessageDialog(null,"Suma = " + suma+ "\n Resta= "+ resta+ "\n Multiplicación= "+ mul+ "\n División= " + div+ "\n Modulo= "+ modulo);*

*//Según operación deseada*

*System.out.println("Suma= "+ suma);*

*System.out.println("Resta= "+ resta);*

*System.out.println("Multiplicación= "+ mul);*

*System.out.println("División= "+ div);*

*System.out.println("Modulo= "+ modulo);*

*}*

*}*