

Unidad 2. Métodos y estructuras de control

**MATERIA: Programación orientada a objetos I**

26 de abril de 2020

Guadalupe García Vázquez

MATRICULA: ES1921008556

Desarrollo en Software

**Docente: Claudia Erika González Gómez**

Actividad 3. Retorno de parámetros e implementación de estructuras cíclicas

Contenido

[Introducción 3](#_Toc38801771)

[Indicaciones 6](#_Toc38801772)

[Desarrollo 7](#_Toc38801773)

[Conclusiones 20](#_Toc38801774)

# Introducción

En las siguientes hojas veremos más a detalle el tema de los métodos. El objetivo de este es poner en práctica la aplicación de los conceptos vistos, así como distinguir el funcionamiento de los distintos métodos utilizados en Java y sus aplicaciones para la creación de métodos que devuelven y que no devuelven valor; también abordaré la resolución de operaciones utilizando estructuras cíclicas, en el caso del siguiente ejercicio se utilizará While, y la condición if-else.

**Estructuras de control cíclicas.**

Palabras claves

*Las* ***estructuras de control cíclicas*** *son bloques de instrucciones que se repiten un número de veces mientras se cumple, o hasta que se cumpla una condición (García, Rodríguez, Mingo, Imaz, Brazález, Larzabal et ál., 2000).*

*Un* ***bucle*** *se utiliza para realizar un proceso repetidas veces. Se denomina también lazo o loop. El código incluido entre las llaves { } (opcionales si el proceso repetitivo consta de una sola línea), se ejecutará mientras se cumpla unas determinadas condiciones.*

Como regla general puede decirse que se utilizará:

* Ciclo for, cuando se conozca de antemano el número exacto de veces que ha de repetirse un determinado bloque de instrucciones. •
* Ciclo do-while, cuando no se conoce exactamente el número de veces que se ejecutará, pero se sabe que por lo menos se ha de ejecutar una.
* Ciclo while-do, cuando es posible que no deba ejecutarse ninguna vez.

**El ciclo while** tiene como característica que primero evalúa la condición y solo si se cumple realiza las instrucciones, así es que si la condición no es cumplida desde un inicio, el control del programa no entrará en el bloque de instrucciones que se encuentran dentro del ciclo.

*Su declaración es como se muestra a continuación:*

*while (condicion){*

*Bloque de Instrucciones a repetir*

*}*

**Do-while** Este ciclo permite la ejecución del bloque repetitivo al menos una vez, se cumpla o no la condición, pues ejecuta primero el bloque do y al finalizarlo evalúa la condición, si ésta se cumple regresa a ejecutar nuevamente el bloque do, de lo contrario termina la ejecución.

do{

Bloque de instrucciones a repetir

}

While (condicion);

**For** Este ciclo es especialmente útil cuando, de antemano, conoces el número de veces que quieres que se ejecute el mismo bloque de instrucciones, su declaración es como sigue:

*for(inicio ; condición; incremento){*

*Bloque de instrucciones repetitivas*

*}*

***Estructuras de control selectivas***

*Las estructuras alternativas son construcciones que permiten alterar el flujo secuencial de un programa, de forma que en función de una condición o el valor de una expresión, el mismo pueda ser desviado en una u otra alternativa de código (García, Rodríguez, Mingo, Imaz, Brazález, Larzabal et ál., 2000).*

If - else

La estructura de selección simple o if es aquella secuencia de un programa que se ve afectada por una condición, si ésta es cumplida la secuencia del programa entrará en el bloque de instrucciones de dicha estructura, de lo contrario la secuencia del programa se “saltará” ese bloque de código y entrará al bloque de instrucciones else, dado que la condición se cumplió.

Esta estructura de control se conforma como sigue: // Instrucciones previas a la condición

*if( condición) {*

*Instrucciones a ejecutarse solo si la condición se cumplió.*

*}*

If anidado

Esta estructura es un agrandamiento de la anterior, ya que se pueden evaluar diferentes condiciones ligadas.

**Los constructores** son unos métodos o funciones miembro muy importantes. Como se puede ver, no tienen valor de retorno (ni siquiera void) y su nombre coincide con el de la clase. Los constructores son un ejemplo típico de métodos sobrecargados (overloaded): en este caso hay dos constructores, el segundo de los cuales no tiene ningún argumento.

Un constructor es un método especial de una clase que se llama automáticamente siempre que se declara un objeto de esa clase.

Su función es inicializar el objeto y sirve para asegurarnos que los objetos siempre contengan valores válidos.

Cuando se crea un objeto en Java se realizan las siguientes operaciones de forma automática:

1. Se asigna memoria para el objeto.

2. Se inicializan los atributos de ese objeto con los valores predeterminados por el sistema.

3. Se llama al constructor de la clase que puede ser uno entre varios.

## Indicaciones

I.- Lee detalladamente y analiza el siguiente planteamiento de problema:

Se requiere calcular el área un **Trapecio, Circulo o Pentágono**. En el cual el usuario debe de elegir en primera instancia que figura geométrica requiere realizar, posteriormente el usuario debe de introducir por teclado los datos necesarios para calcular el área de la figura elegida. Para poder desarrollar esta actividad se debe de utilizar un método por cada figura para calcular cada área, y este devolverá un número real.

Dicho resultado obtenido deberá mostrarse en pantalla. Aquí podrás observar las fórmulas que necesitas para cada figura:

• Trapecio: (BaseMayor + BaseMenor) x h

• Circulo: PI \* (radio^2)

• Pentágono: (Perímetro \* Apotema) / 2

II.- Una vez entendido completamente la solución del problema, realiza un análisis y diseño de solución considerando el uso de métodos con retorno de parámetro. Por cada uno de los métodos, realiza el análisis y determina:

• Describe qué hará el método dentro de la solución.

• Parámetros de entrada, es decir, qué datos necesitas enviarle al método.

• Dato de salida, qué se espera que devuelva el método como resultado final.

III.- Posteriormente, en la etapa de codificación:

Compílalo y ejecútalo para verificar que funciona correctamente.

En caso de que el compilador envíe errores, verifica dónde está el error y corrígelo.

Verifica los resultados ejecutando el programa al menos dos veces con diferentes valores de entrada, realiza las capturas de pantalla de ambas ejecuciones e intégralo al documento de Word.

En caso que no envíe los resultados esperados explica el porqué, corrige y ejecuta de nuevo.

IV.- Agrega los siguientes archivos a una carpeta llamada DPO1\_U2\_A3\_XXYZ:

• El documento de Word

• Carpeta del proyecto desarrollado

### 

### Desarrollo

Para la solución del problema debemos entender que es lo que buscamos obtener. En este caso es poder obtener el área de tres figuras diferentes, cada una de ellas tiene características diferentes y por lo tanto necesitan un trato individual. Por lo que ocuparé los constructores: Un Constructor es una función, método, etc, de las clases, la cual es llamada automáticamente cuando se crea un objeto de esa clase, estos no devuelven valores, pero ayudan a facilitar la distribución de las órdenes del programa.

Debido a que utilizaremos estructuras de control cíclicas (while-do), y estructuras de control selectivas (if anidado), el agrupar los métodos, permite un mejor flujo en el diseño y ejecución

**Constructor Figura 1**

**Trapecio:** distinguiendo que para el cálculo del área necesitamos tres valores, la medida de su base mayor, la medida de su base menor y la altura.

Para aplicar la fórmula: Trapecio: (BaseMayor + BaseMenor) x h

*//Trapecio*

*// Atributos*

*float BaseMayor = 0, BaseMenor = 0, h = 0;*

*//Llamado*

*BaseMayor= Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de la Base Mayor del Trapecio "));*

*BaseMenor= Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de la Base Menor del Trapecio "));*

*h= Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de la Altura del Trapecio "));*

*//Metodos,*

*Trapecio=(BaseMayor + BaseMenor)\* h;*

*Este método devolverá el valor de la operación, el área de la figura.*

**Constructor Figura 2**

**Círculo:** distinguiendo que para el cálculo del área necesitamos tres valores, la medida de su radio y Pi, siendo Pi una constante.

Para aplicar la fórmula: • Circulo: PI \* (radio^2)

*// Circulo*

*// Atributo*

*double Radio= 0,Radio2, Areacirculo; variables*

*final double PI =3.1416; constantes*

*//Llamado*

*Radio= Double.parseDouble(JOptionPane.showInputDialog("Ingrese el radio del circulo "));*

*//Metodos*

*Radio2=Radio\*Radio;*

*Areacirculo= (Double) (PI\* Radio2);*

*Este método devolverá el valor de la operación, el área de la figura.*

*JOptionPane.showMessageDialog(null,"El área del círculo es = " +Areacirculo+ " u2 ");*

**Constructor Figura 3**

**Pentágono:** distinguiendo que para el cálculo del área necesitamos el valor del perímetro, el cual se obtiene de la medida de un lado por 5 (el total que tiene) y la medida de su apotema.

Para aplicar la fórmula: • Pentágono: (Perímetro \* Apotema) / 2

*// Pentágono*

*// Atributo*

float Pentagono, Perimetro, Lado = 0, Apotema = 0; variables

//Llamado

Lado= Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de un lado del Péntagono "));

Apotema= Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor del Apotema del Péntagono "));

//Metodo

Perimetro= Lado\*5;

Pentagono=(Perimetro \* Apotema) / 2;

*Estos dos métodos devolverán el valor de la operación, el área de la figura.*

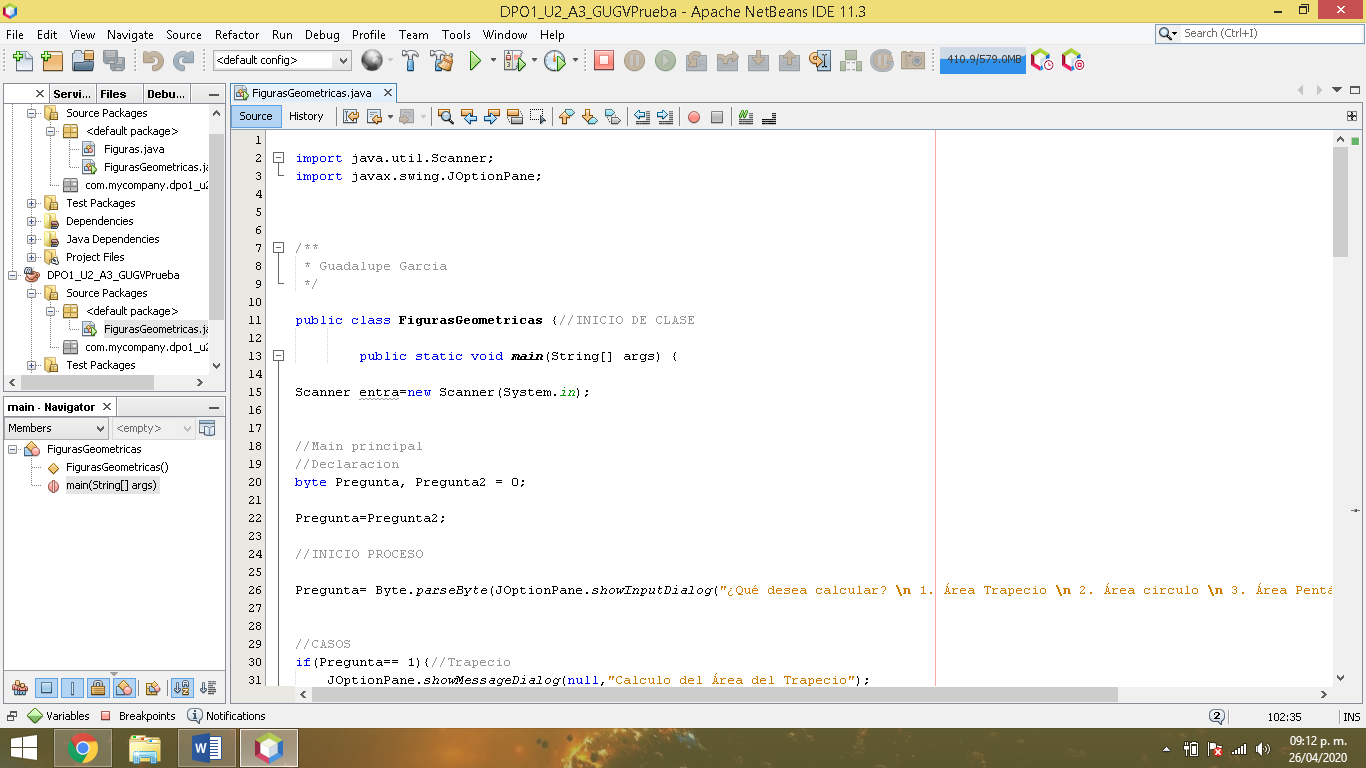
JOptionPane.showMessageDialog(null,"El área del Pentagono es = " + Pentagono+ " u2 ");

Compilación y ejecución del programa

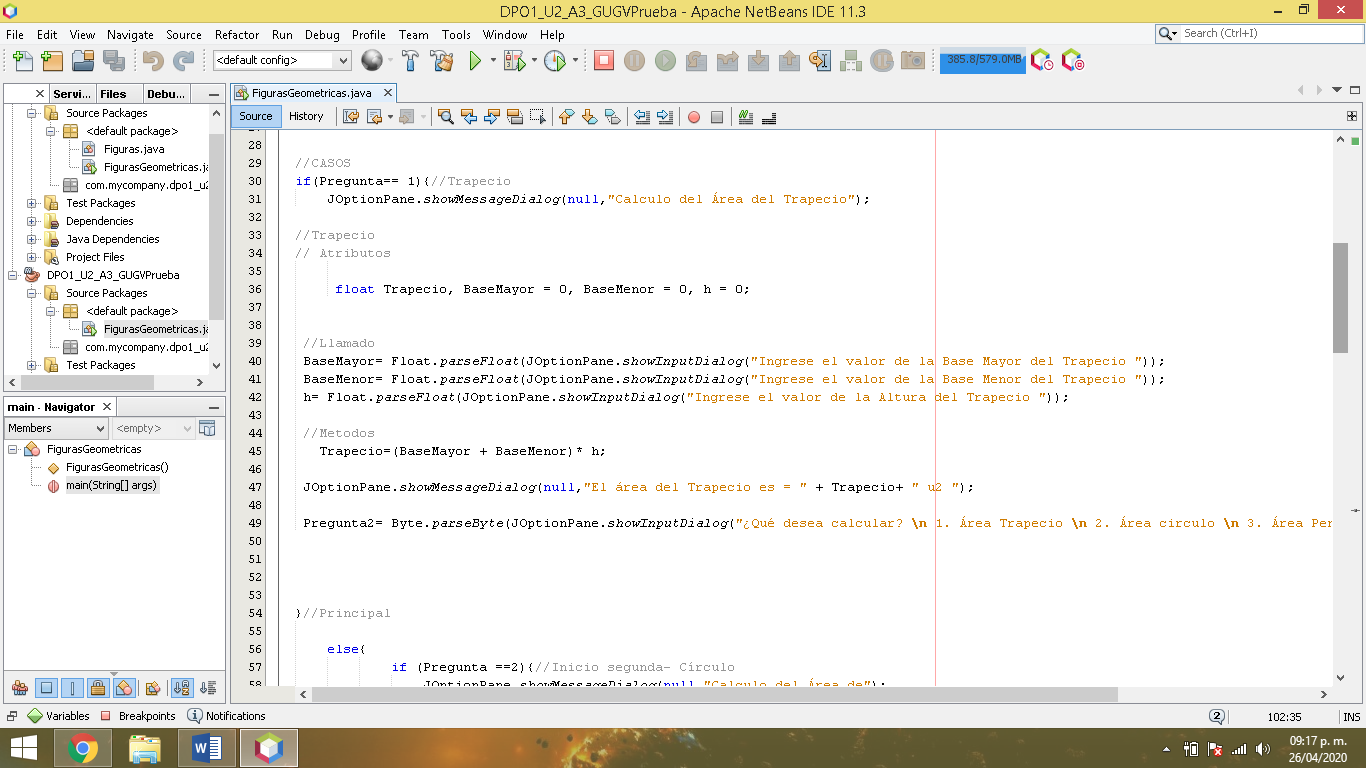
En un comienzo había realizado la diseñado que en un solo main, realizará todos los procesos.

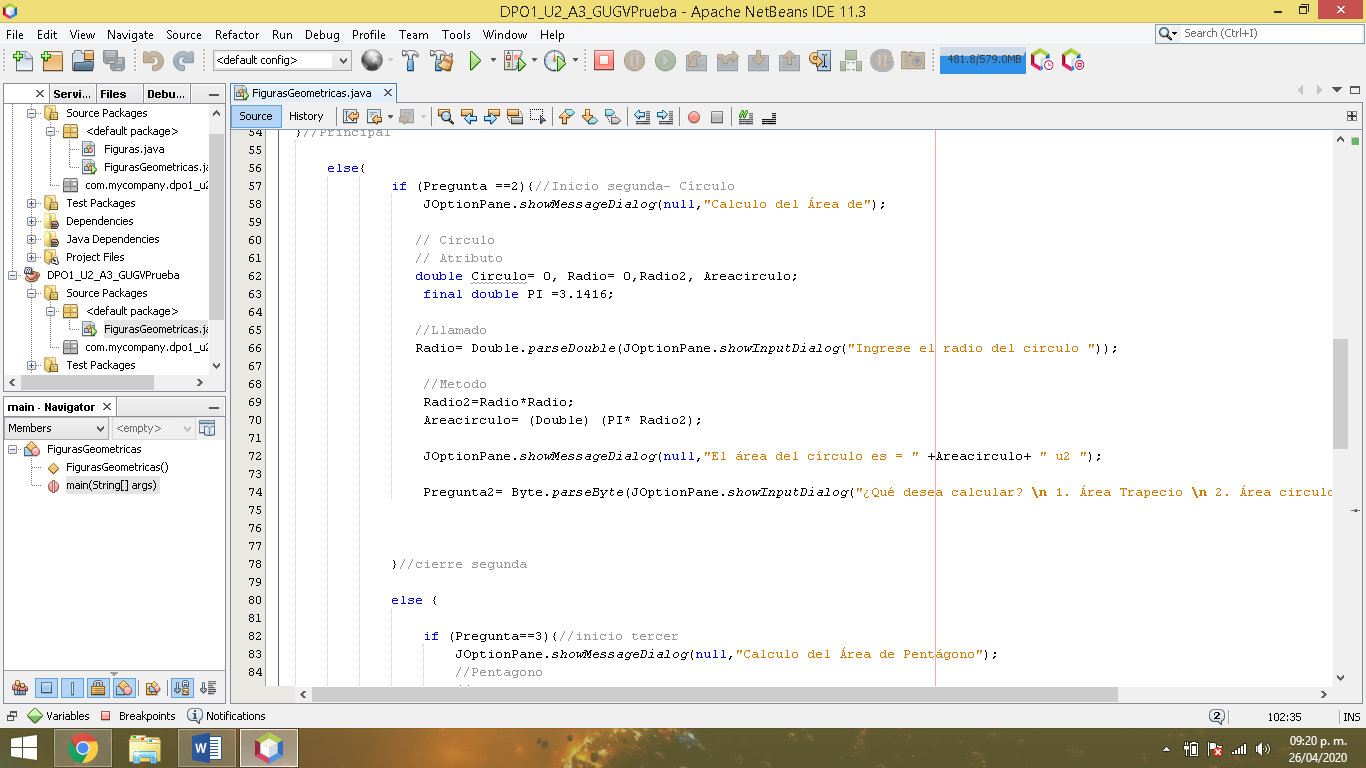
Comencé con llamar las librerías a ocupar, otras el sistema las fue requerimiento mientras escribía el código.

Después viene las declaraciones de las variables y las constantes, como todo estaba concentrado en uno solo, se hicieron mención de todas, la mayoría de ellas eran variables, ya que dependían de los datos ingresados por el usuario final.



Debido a que uno de los requerimientos del problema planteado es que el usuario eligiera que tipo de cálculo quería realizar, el sistema debía tener la opción de que de acuerdo a la respuesta del usuario realizaría un conjunto de métodos (por figura). Decidí utilizar If anidadas, ya que eran más de dos opciones.

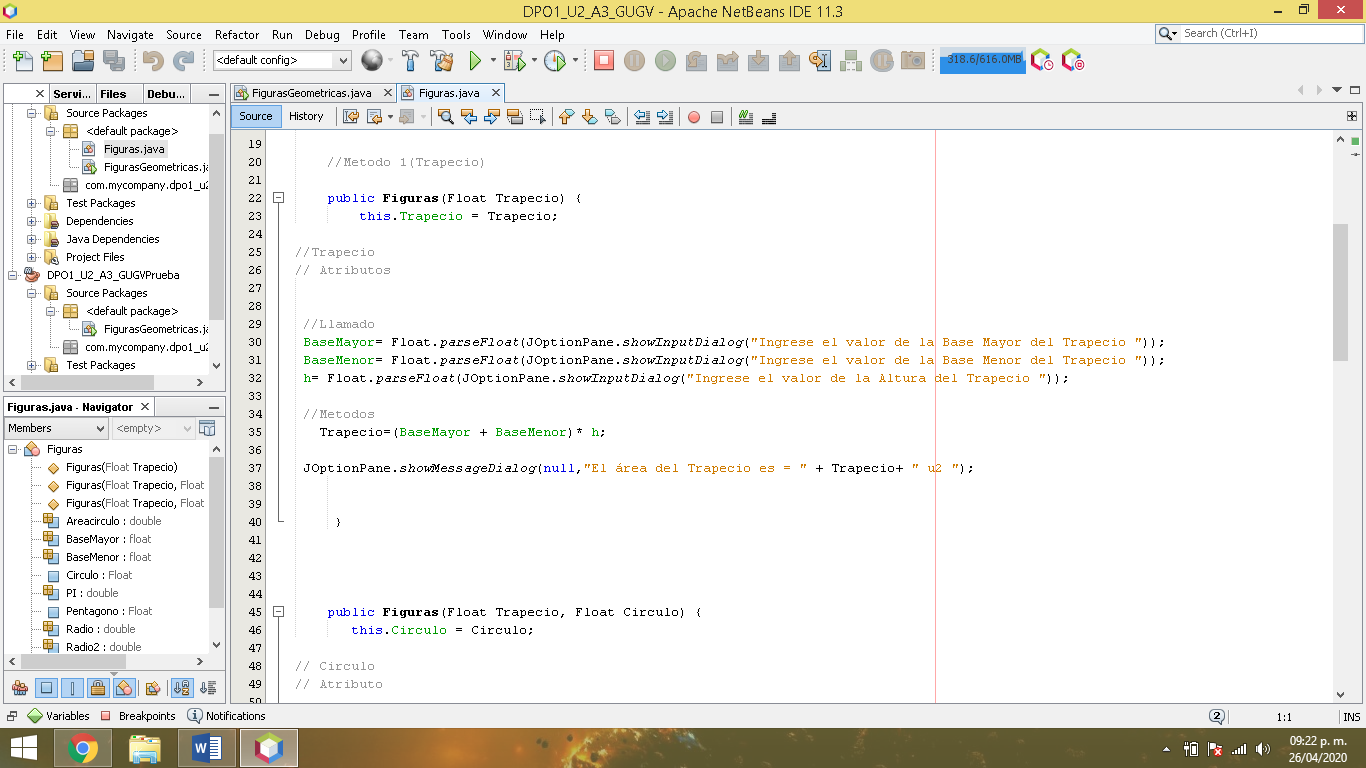
Pero existía otro factor, darle la opción al usuario de poder seguir calculando otras áreas, por lo cual había que hacer uso de los ciclos, para el caso se utilizó While. 



Los principales errores que me arrojaba el sistema, eran por algunos tipos de datos que no hice la elección de manera óptima, errores de caracteres faltantes al escribir, y en el IF anidado, la limitación de cada una de las secciones.

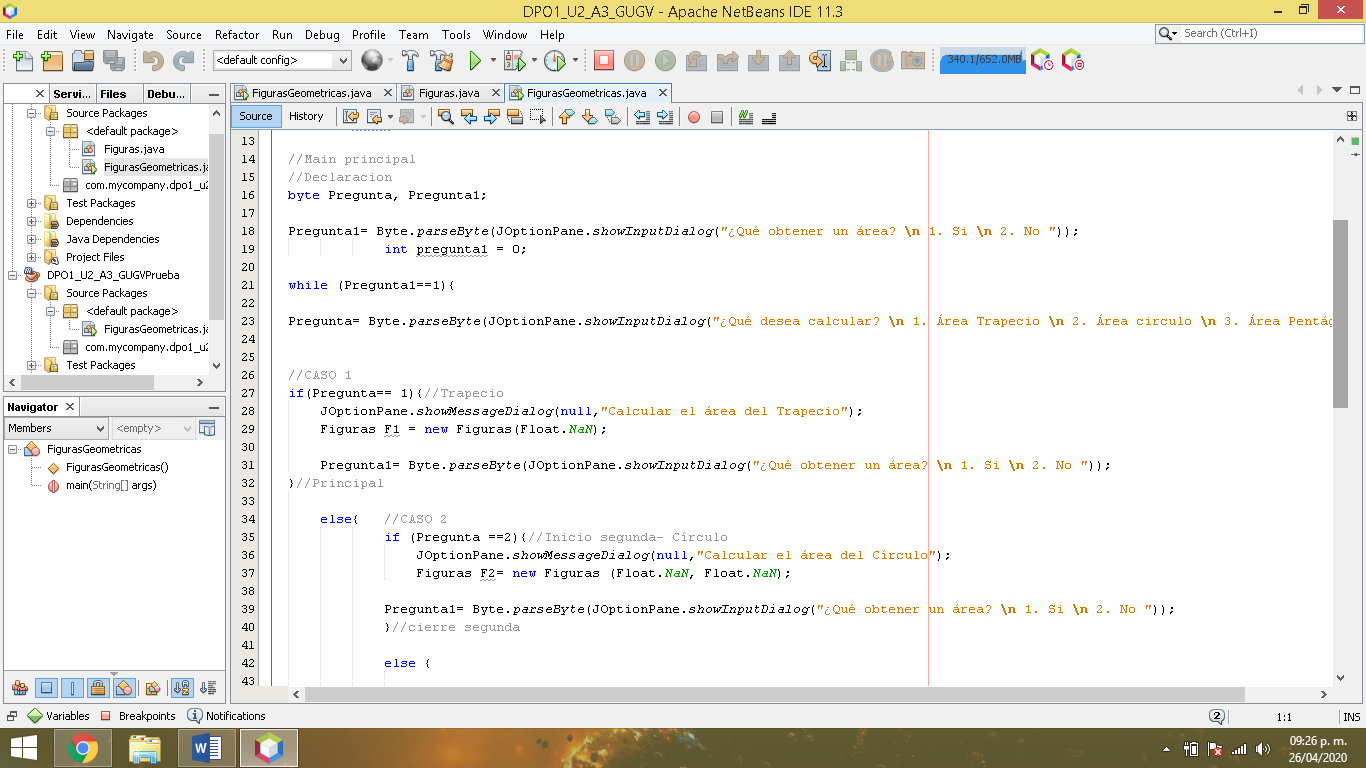
Posteriormente encontré la opción de utilizar constructores. Los cuáles con el avance que tenía hasta ese momento ayudaría a terminarlo de la mejor manera.

Se agregó una nueva clase, Figuras a pesar de tener diferentes características, pertenecían a este grupo .

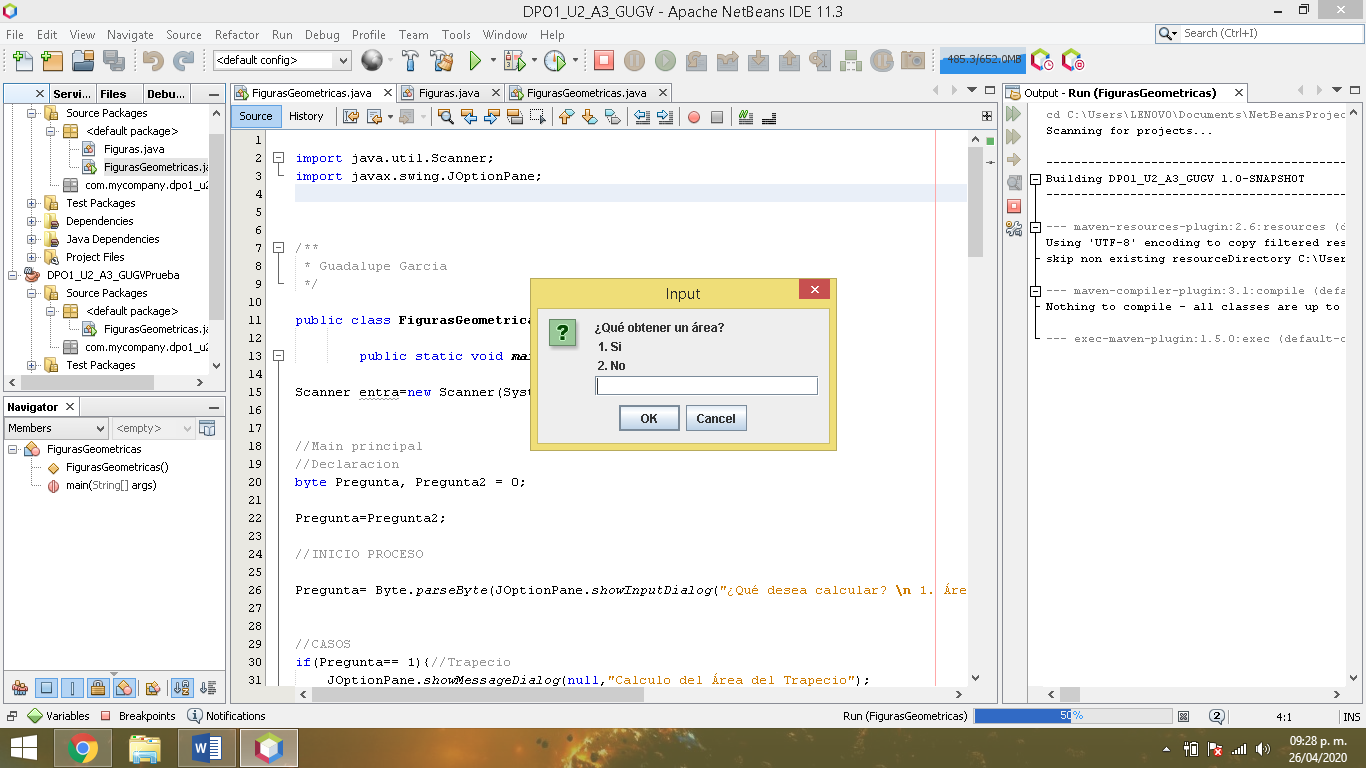


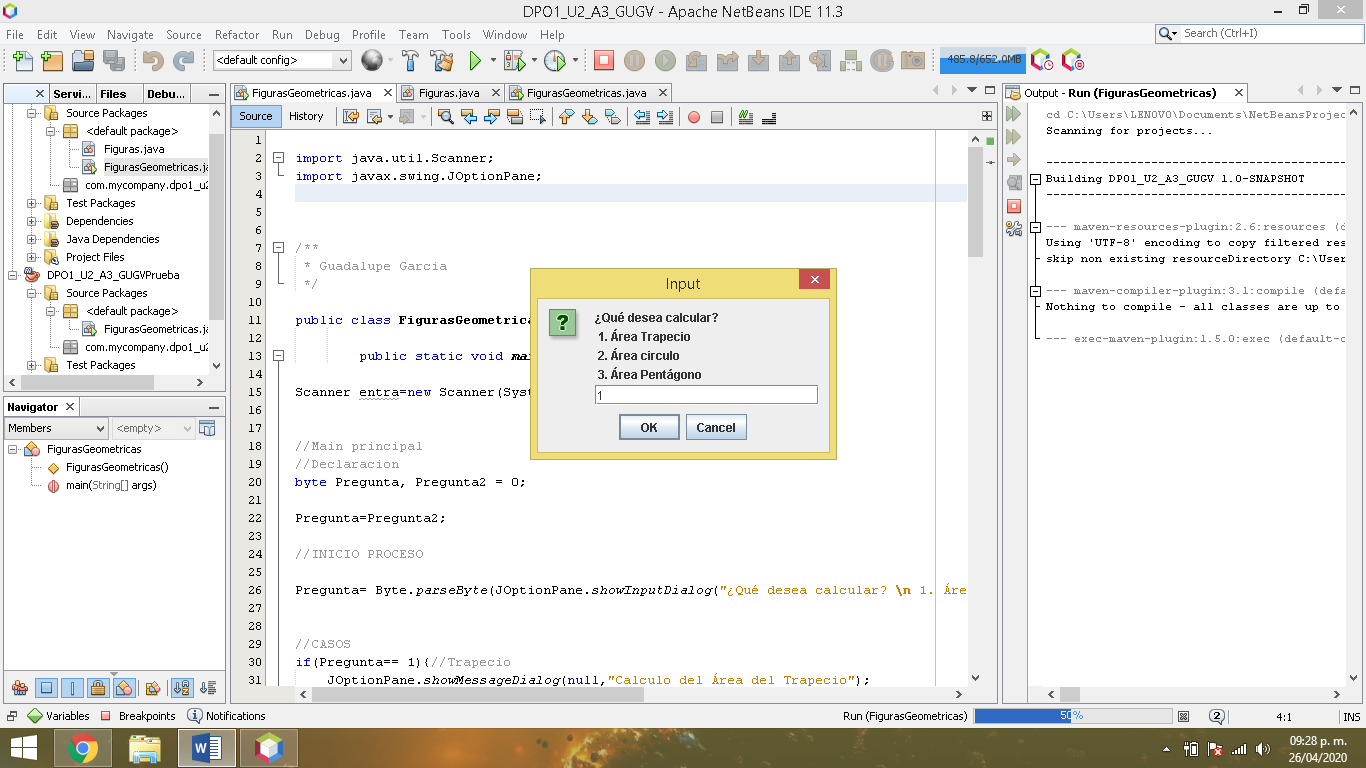
En los constructores, había que ponerles un identificador, declarar sus atributos y los métodos. Estos constructores permitieron enlazar las condiciones de mejor manera.

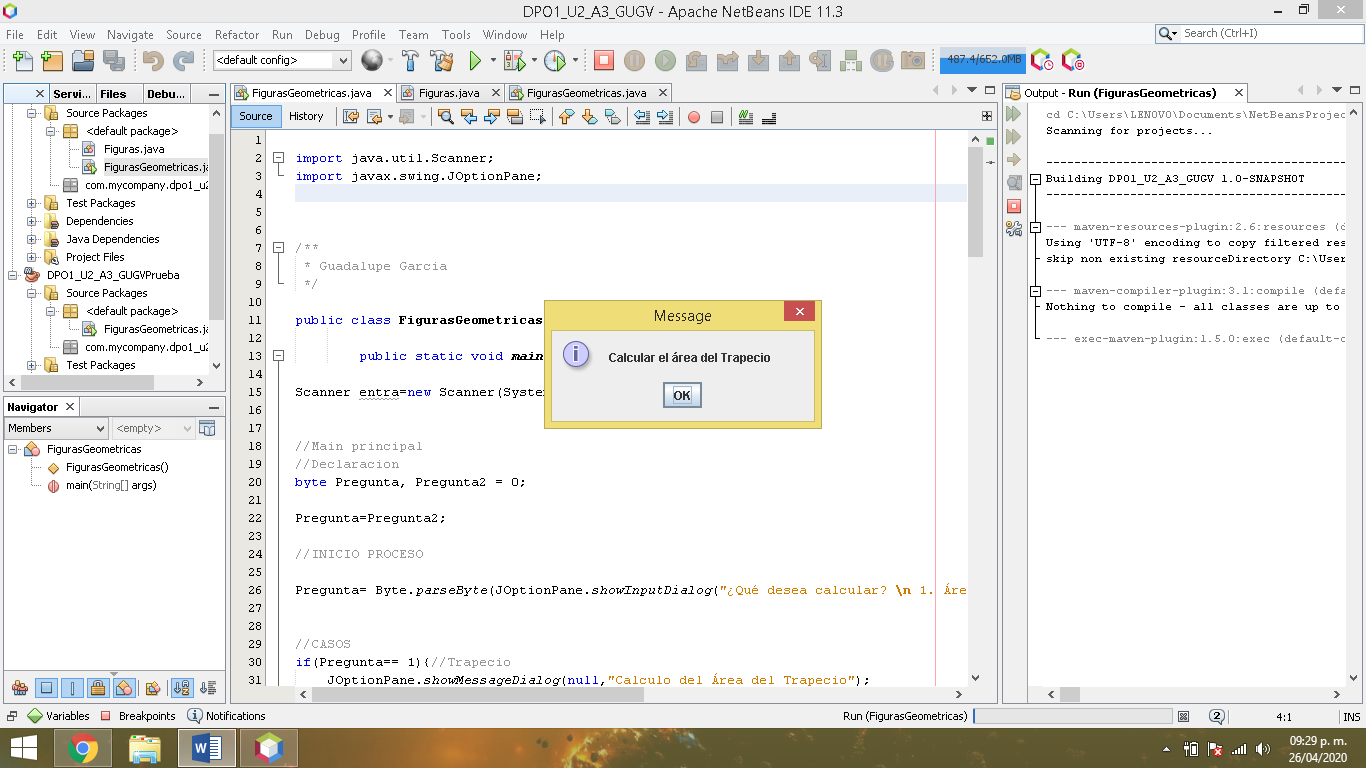
Así solo debía llamar a los métodos, cuándo el sistema requería de ellos, detectando de mejor manera si había errores y terminando con escribir las estructuras cíclicas.

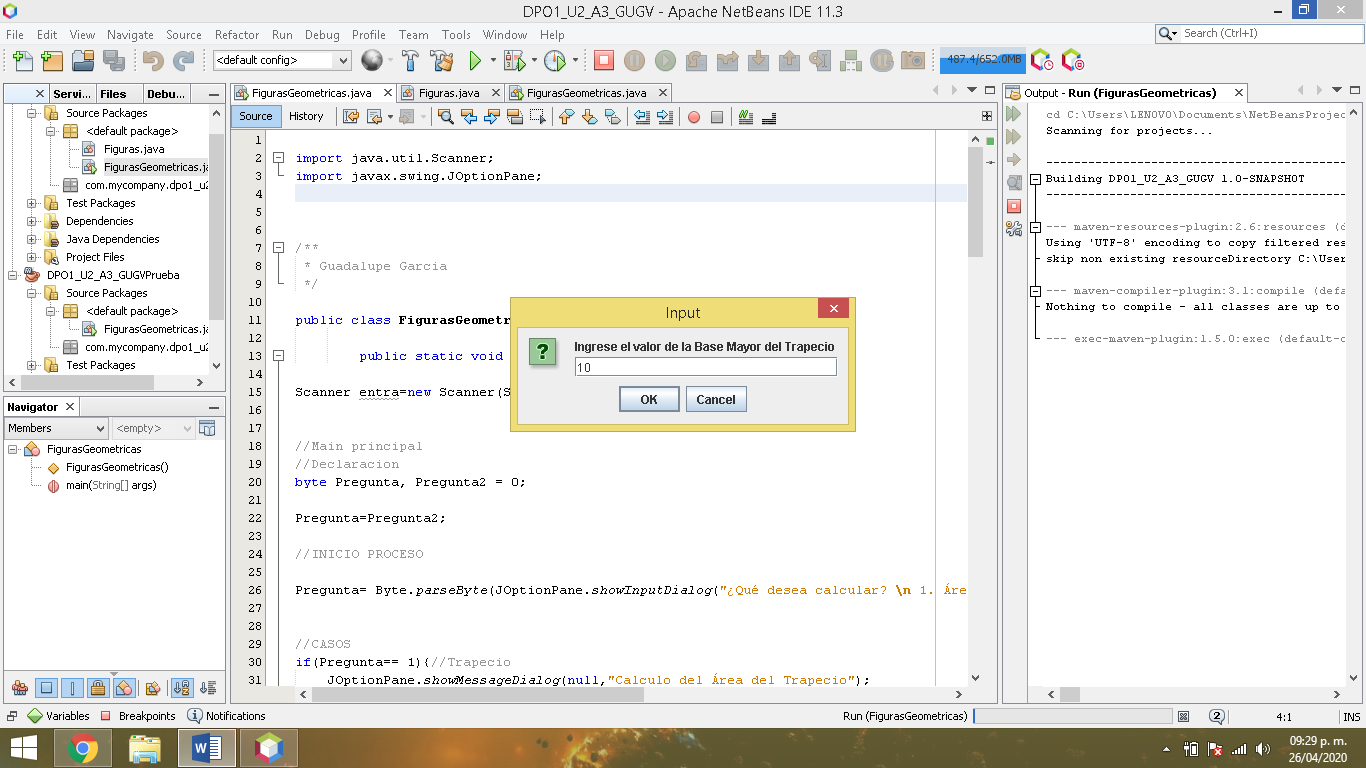


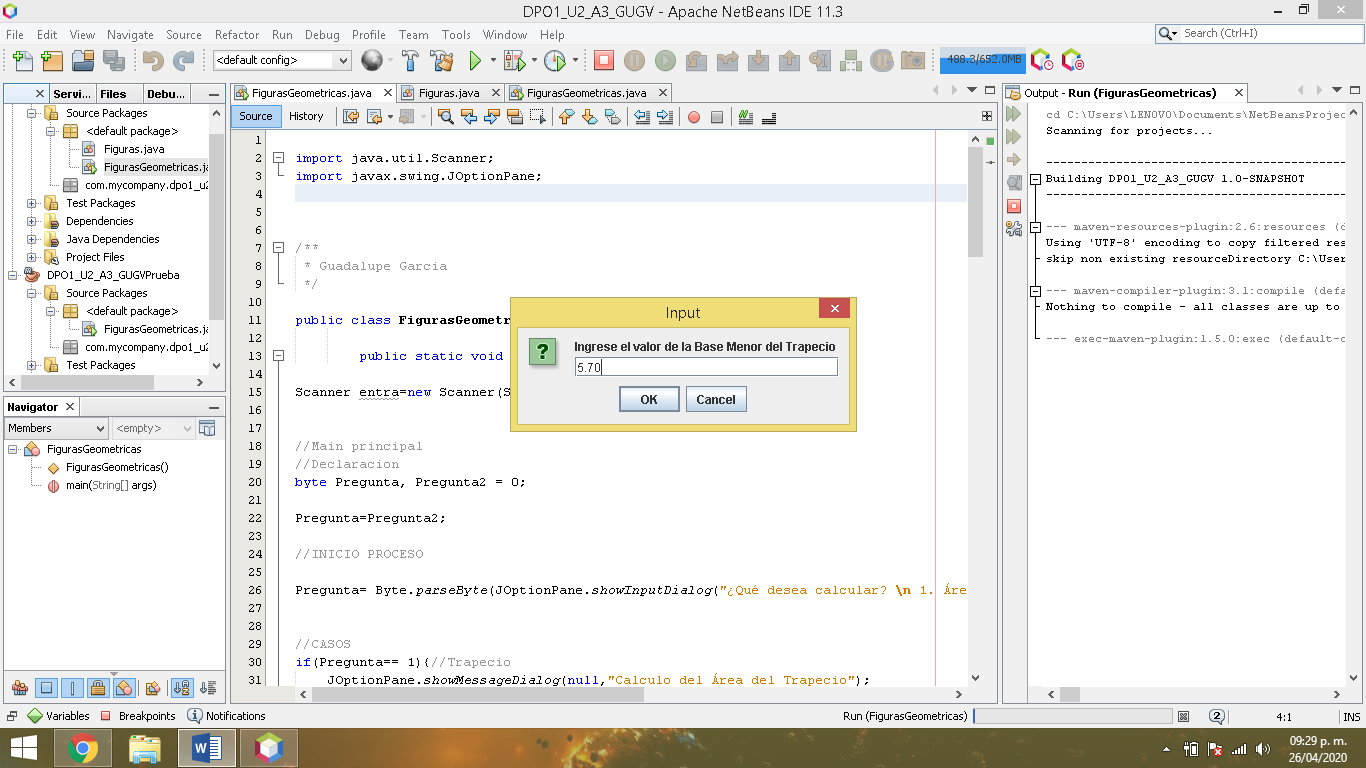
Ejecución del programa

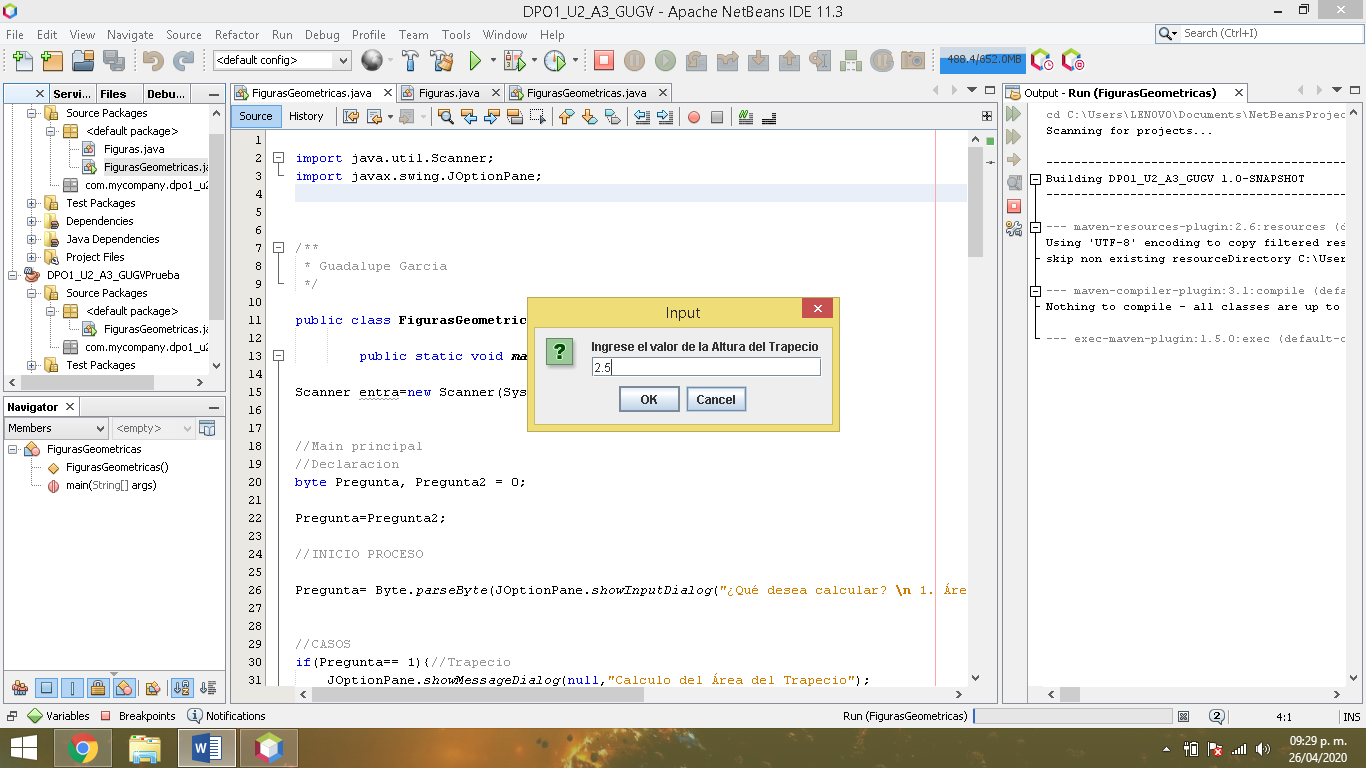


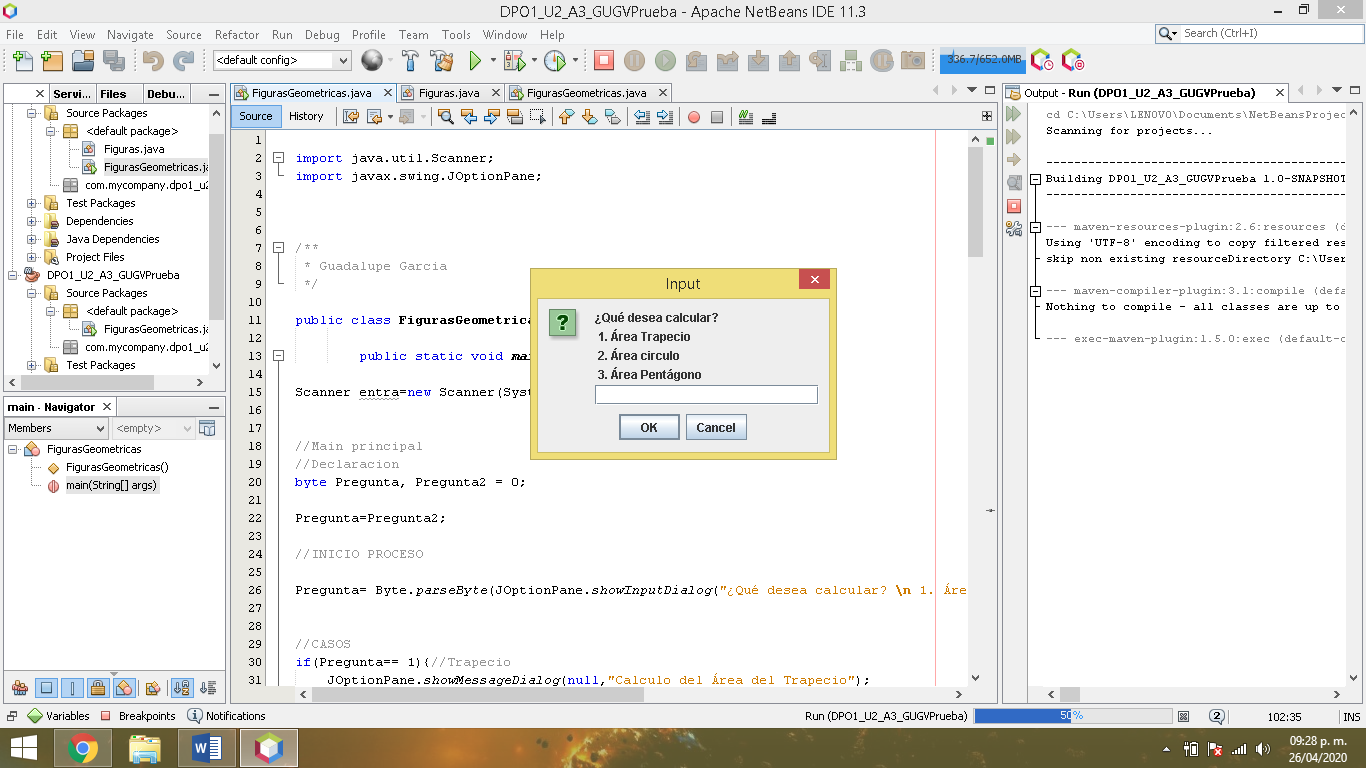


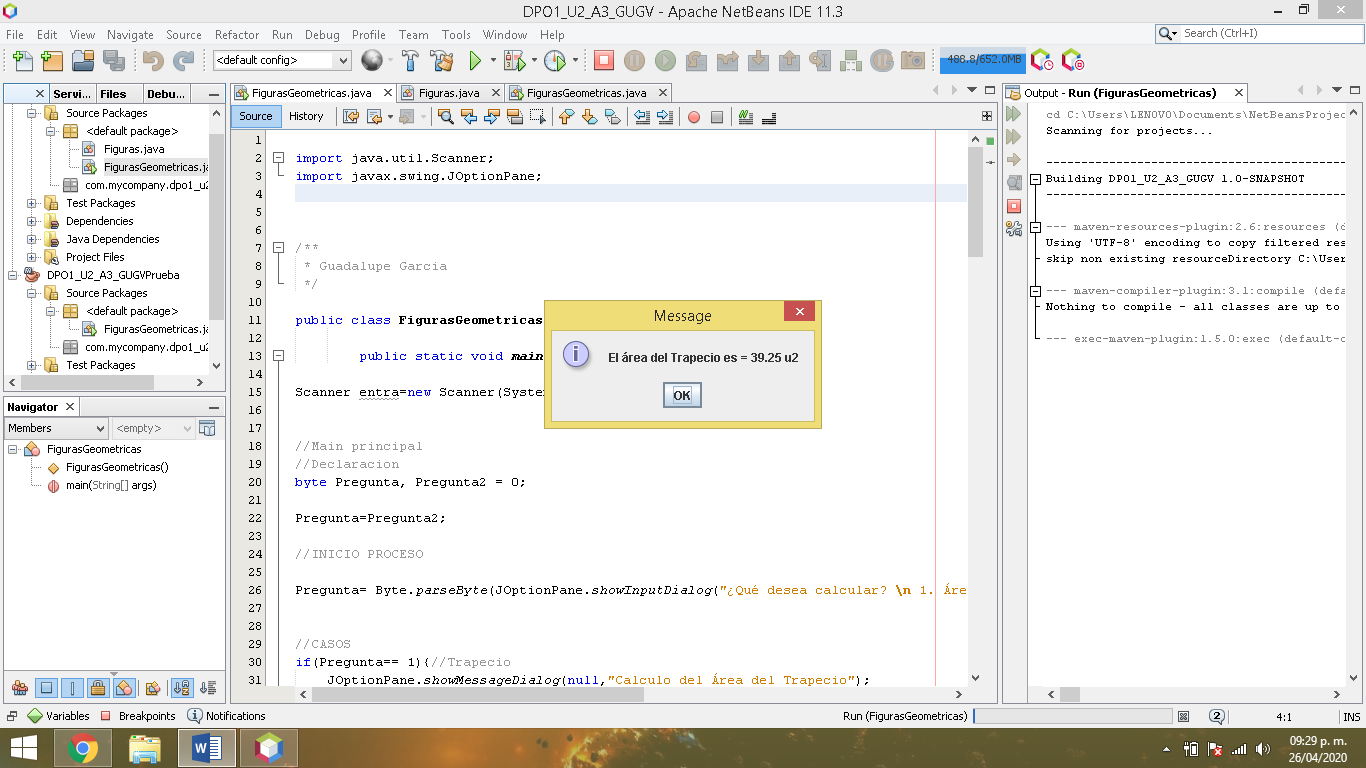


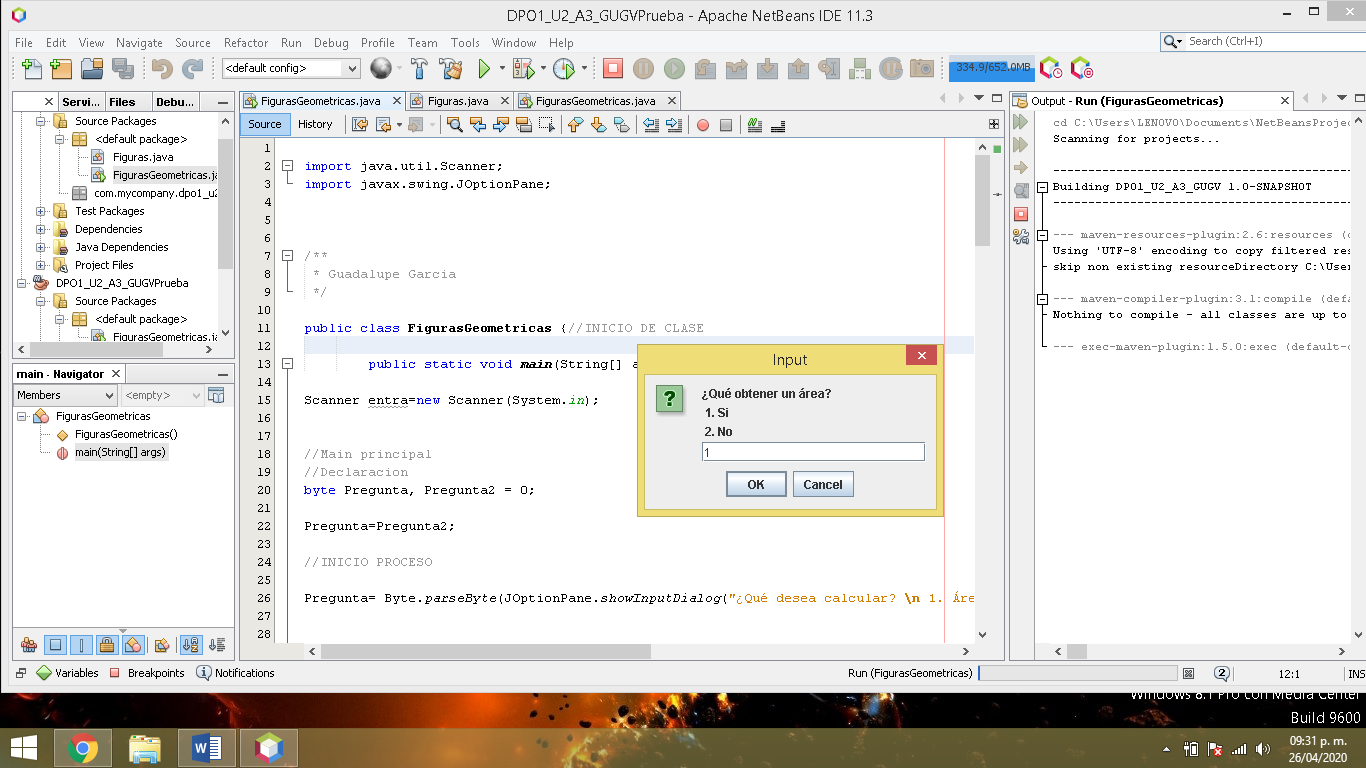


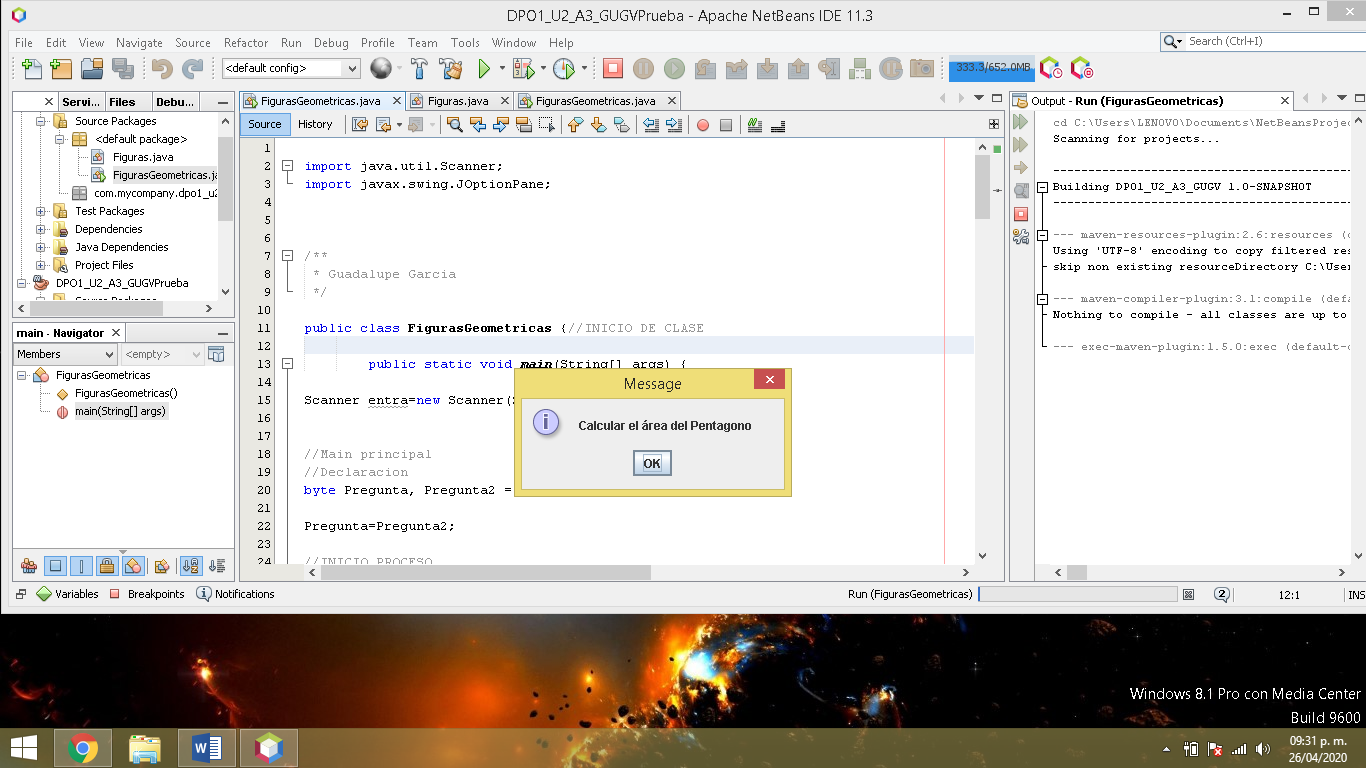


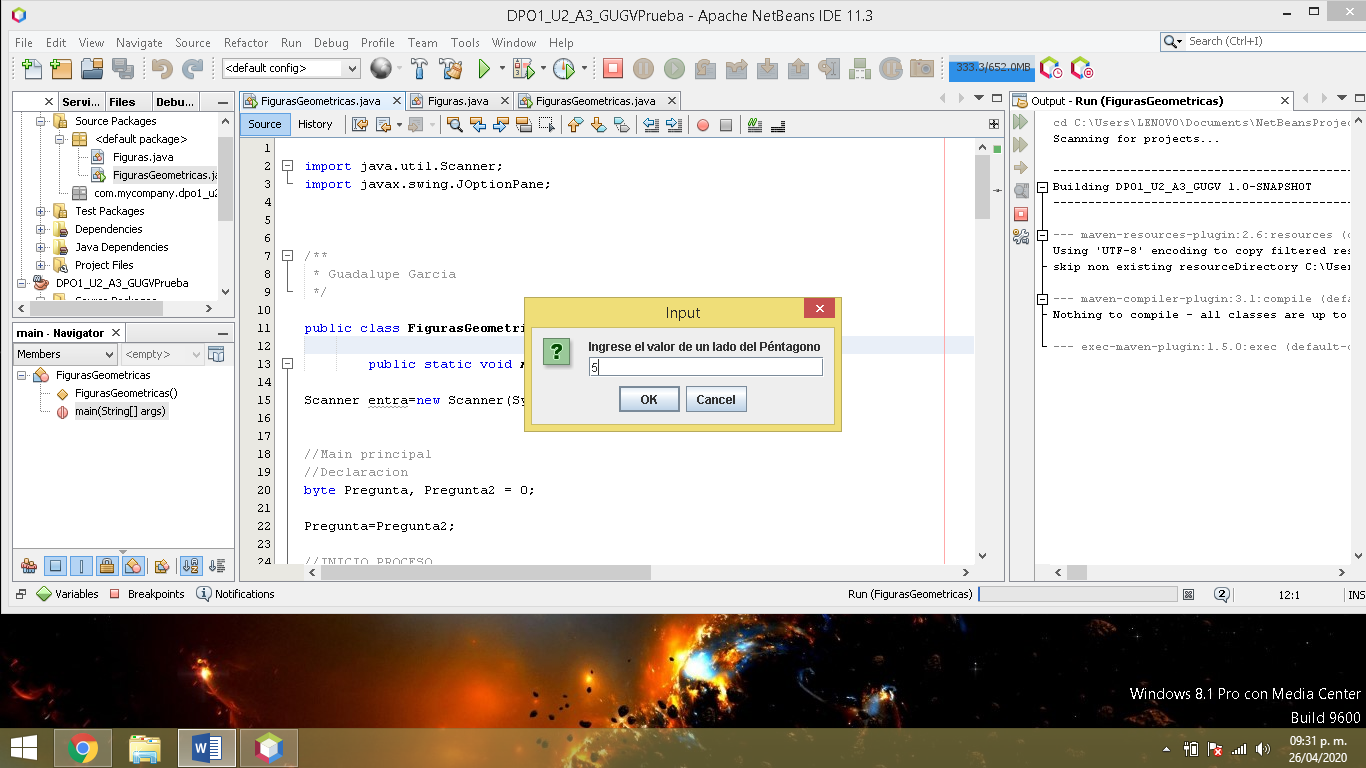


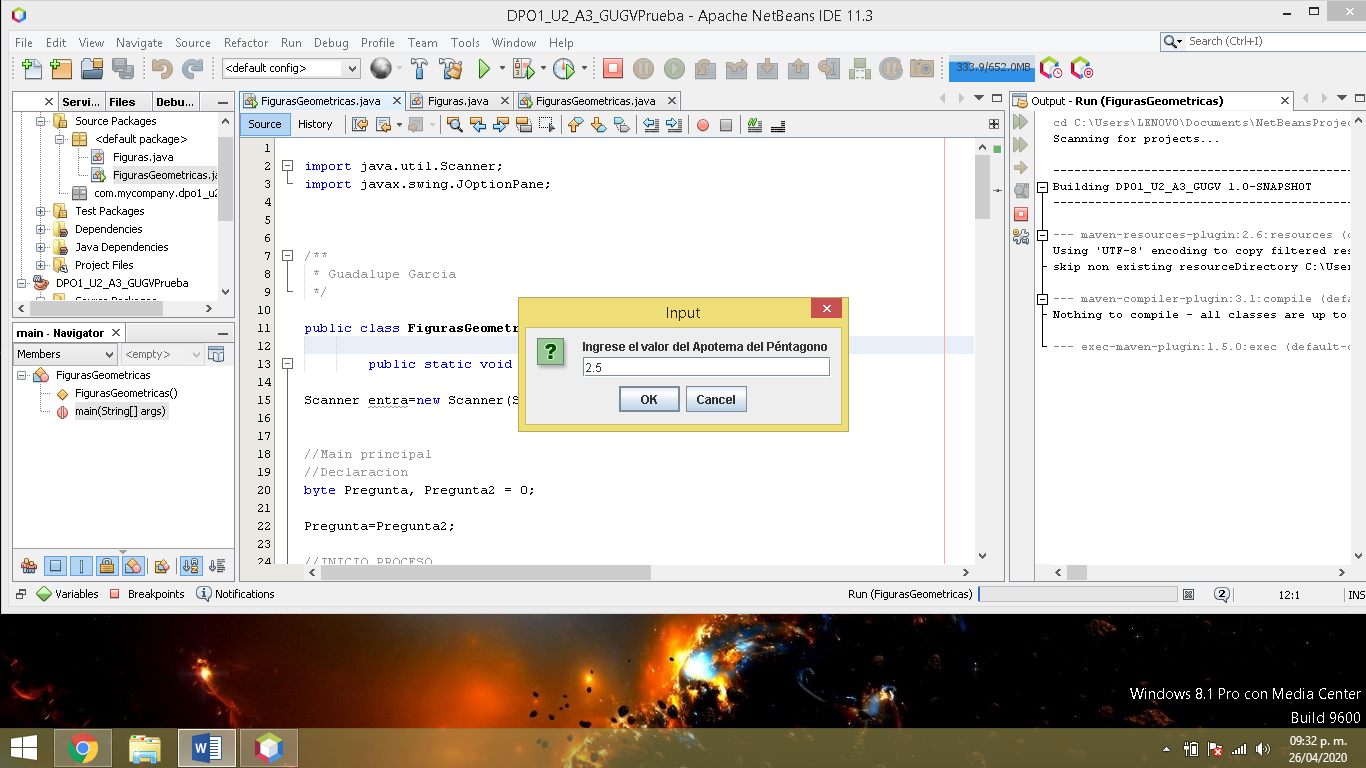


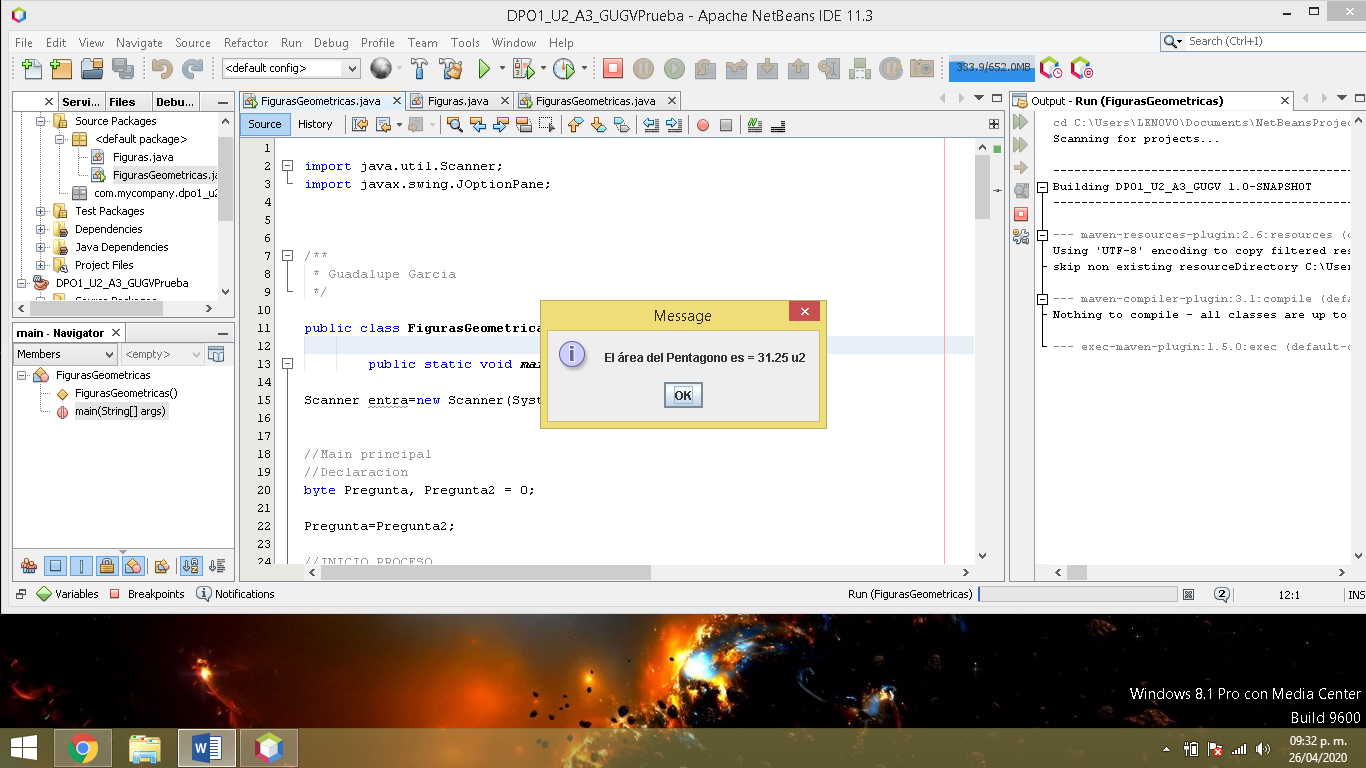


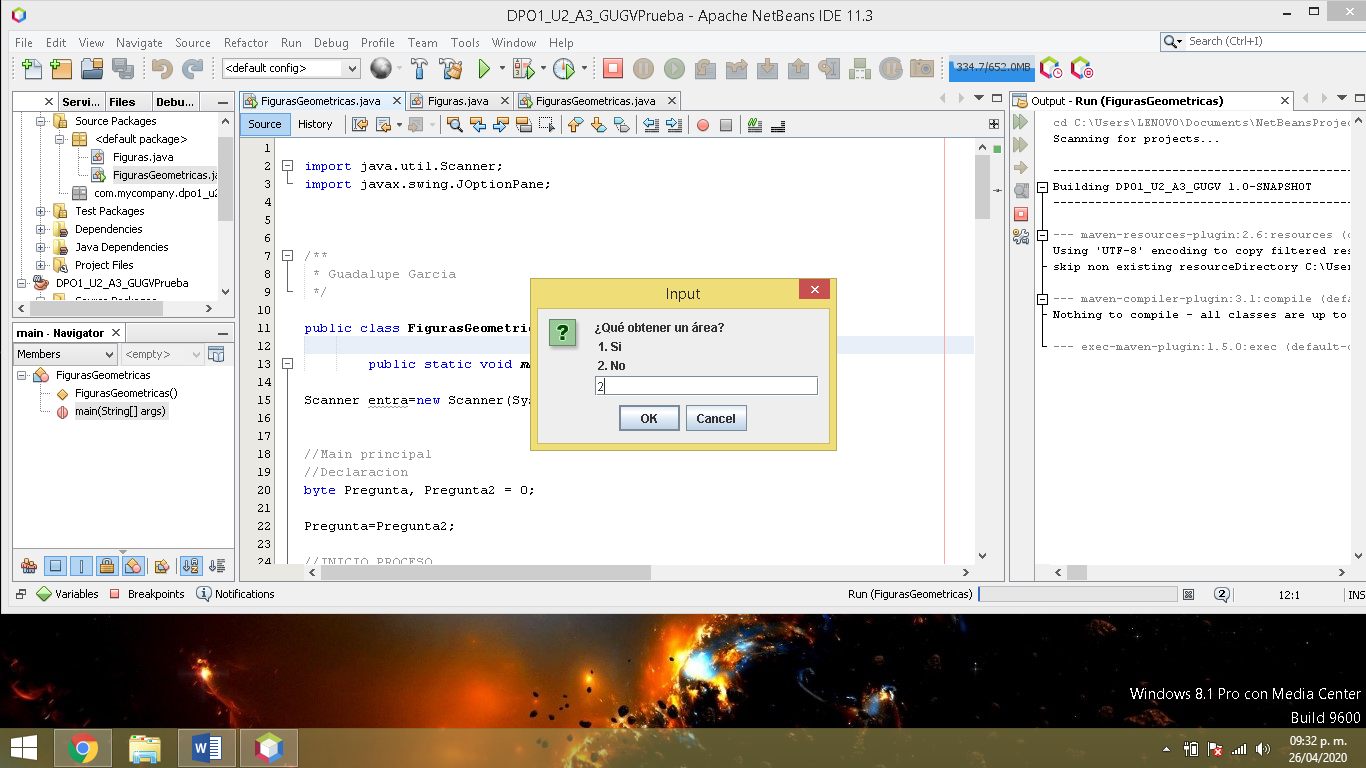


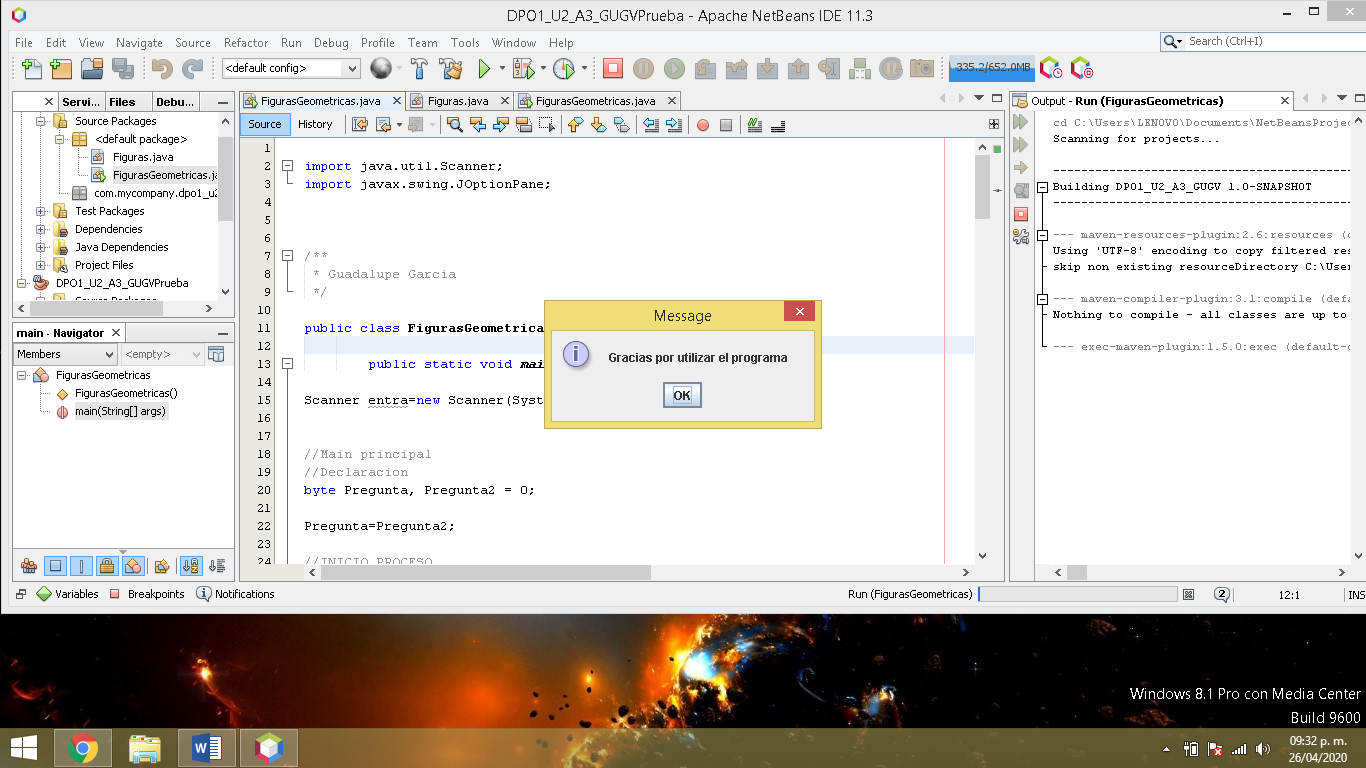












#### 

#### Conclusiones

Esta actividad me permitió entender realmente el uso de los métodos, dejando aún lado solamente la teoría. Los métodos son una parte indispensable en la programación porque son los que ayudan a realizar las acciones, ya sea que tenga datos de salida o no. Pero sin ellos el programa no podría llevar a cabo las instrucciones. En la programación estructurada las indicaciones secuenciales, estaban solo alineadas a ir paso por paso, pero con la aplicación del ejercicio noté cómo funcionan los objetos, los métodos no eran independientes al objeto, de hecho los métodos formaban parte de un objeto, como piezas establecidas que puedes utilizar y mover de una forma estratégica para que el desarrollo del código sea el óptimo, así solo es cuestión de llamarlas en el momento que se requieran.

Además otro tema importante son las estructuras cíclicas, ya que de igual forma, facilitan la laboran, ya que evitan el tener que volver a escribir el código una y otra vez. Solo basta con establecer las instrucciones y el sistema permitirá hacer uso de las instrucciones hasta que una cierta condición se cumpla.

# Bibliografía

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

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 (22/04/20)

Hernández, E. G. (s.f.). *Constructor en Java*. Obtenido de Blogger.: http://puntocomnoesunlenguaje.blogspot.com/2012/07/normal-0-21-false-false-false\_103.html Consultado 26/04/2020

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)

ATS, C. P. (Dirección). (2017). *Obtenido 26/04/2020*

https://www.youtube.com/watch?v=8f\_nVq7azoI

https://www.youtube.com/watch?v=t2R8ym8hjJ4

https://www.youtube.com/watch?v=jGfB6UzG-6A

https://www.youtube.com/watch?v=6qgOFZnhYEA

