

Unidad 2. Métodos y estructuras de control

**MATERIA: Programación orientada a objetos I**

2 de mayo de 2020

Guadalupe García Vázquez

MATRICULA: ES1921008556

Desarrollo en Software

**Docente: Claudia Erika González Gómez**

Actividad: Evidencia de aprendizaje. Programa de estructuras de repetición.

Contenido

[Introducción 3](#_Toc39358738)

[Indicaciones 4](#_Toc39358739)

[Desarrollo 6](#_Toc39358740)

[Conclusiones 22](#_Toc39358741)

[Bibliografía 23](#_Toc39358742)

# Introducción

Este último proyecto, tiene como finalidad aplicar los conocimientos de la Unidad 2. Métodos y estructuras de control, puliendo los conocimientos generados en las actividades anteriores.

El siguiente programa desarrollado tiene varias funciones para el usuario.

1. Le permite calcular el área de un rectángulo: pedir dos números enteros de entrada y dar como resultado el área del rectángulo.
2. Calcular el área de un triángulo: pedir la base y la altura de tipo número entero e imprimir el área con número flotante
3. Calcular el promedio: el usuario ingresa la cantidad de notas que quiere promediar el programa le imprime el resultado.
4. De dos números que del usuario el programa le dice la comparación entre ambos (el mayor, menor o igual).

Para el desarrollo del proyecto, se ha utilizado un formulario gráfico, facilitando la interacción usuario-programa. Con base a los botones utilizados se seleccionará la opción deseada, de acuerdo a la necesidad que tenga el usuario.

Se utilizó la estructura cíclica While, para obtener el promedio, ya que, el usuario decide cuántas calificaciones quiere ingresar y de acuerdo a esta información el programa va a ejecutar las instrucciones de ingresar las notas para posteriormente sacar el promedio.

Otra estructura condicional utilizada fue if- else, anidada. En el punto 4 de la comparación de dos números, es necesario representar los escenarios que surjan del ingreso de los números por parte del usuario. Por lo que la función mencionada, ayuda a mostrarle la comparativa entre los número ingresados.

## Indicaciones

*Después de leer todos y cada uno de los recursos y para aplicar los conocimientos adquiridos en la presente Unidad II, realizarás un programa donde implementes métodos con o sin parámetros, métodos que regresen o no valores, estructuras de selección y estructuras de repetición (iteración), además, realizarás en un documento una descripción de tu programa. Los detalles para realizarlo se describen a continuación.*

*1. Realizar un documento en procesador de textos cuyo contenido sea el siguiente:*

*1.1. Portada, que deberá tener: nombre de la universidad, carrera, curso, matrícula y nombre del estudiante, de la actividad, del docente, y fecha de entrega*

*1.2. Introducción: una breve descripción de lo que hace su programa*

*1.3. Diagrama de flujo general del programa*

*1.4. Explicación especifica de los métodos usados y de las variables*

*1.5. Explicar por qué es importante el usar de manera correcta las instrucciones repetitivas y selectivas*

*1.6. Conclusiones*

*1.7. Fuentes de información en formato APA*

*2. Realizar un proyecto en JAVA (modo texto o gráfico) en Netbeans que resuelva la siguiente situación.*

*2.1. El programa solicitará al usuario que seleccione una opción de cinco, por lo que se mostrará el siguiente menú:*

*1….. Calcular el área de un rectángulo*

*2….. Calcular el área de un triángulo*

*3….. Calcular el promedio*

*4….. Encontrar mayor, menor o igual de dos números*

*0….. Salir*

*2.2. Explicación de las opciones del menú:*

*1 Calcular el área de un rectángulo: pedir dos números enteros de entrada y dar como resultado el área del rectángulo*

*2 Calcular el área de un triángulo: pedir la base y la altura de tipo número entero e imprimir el área con número flotante*

*3 Calcular el promedio: pedir una cantidad “N” de calificaciones con decimal y al finalizar imprimir el promedio de las calificaciones*

*4 Encontrar mayor, menor o igual de dos números: leer dos números e indicar si el primero es mayor, menor o igual que el segundo.*

*0 Salir: con esta opción se dará por finalizado el programa*

*Nota: Si el usuario da una opción que no está en el menú, indicar que la opción es inválida*

*2.3. El programa deberá repetir el menú hasta que le den la opción salir*

*2.4. Deberá tener cuando menos 4 métodos (uno para cada opción del menú)*

*2.5. Al menos un método deberá llevar parámetros*

*2.6. Al menos un método deberá regresar un valor*

*2.7. El programa deberá usar la sentencia de selección switch para el menú*

*2.8. Usar cuando menos dos estructuras diferentes de repetición*

*2.9. Puede agregar métodos y/o mensajes que usted considere necesario*

*3. Guarda el archivo comprimido con tus dos documentos donde desarrollas tus evidencias (el proyecto generado en Netbeans y el documento generado en el procesador de textos) con el nombre DPO1\_U2\_EA\_XXYZ. Sustituye las XX por las dos primeras letras del primer nombre, la Y por la inicial del apellido paterno y la Z por la inicial del apellido materno*

*4. El programa no deberá marcar errores cuando se ejecute, debe funcionar*

*5. El programa puede correr en modo texto, es opcional en modo gráfico*

*6. Consulta la Escala de evaluación, para conocer los elementos de evaluación de la evidencia.*

*7. Revisar el calendario de entrega de actividades*

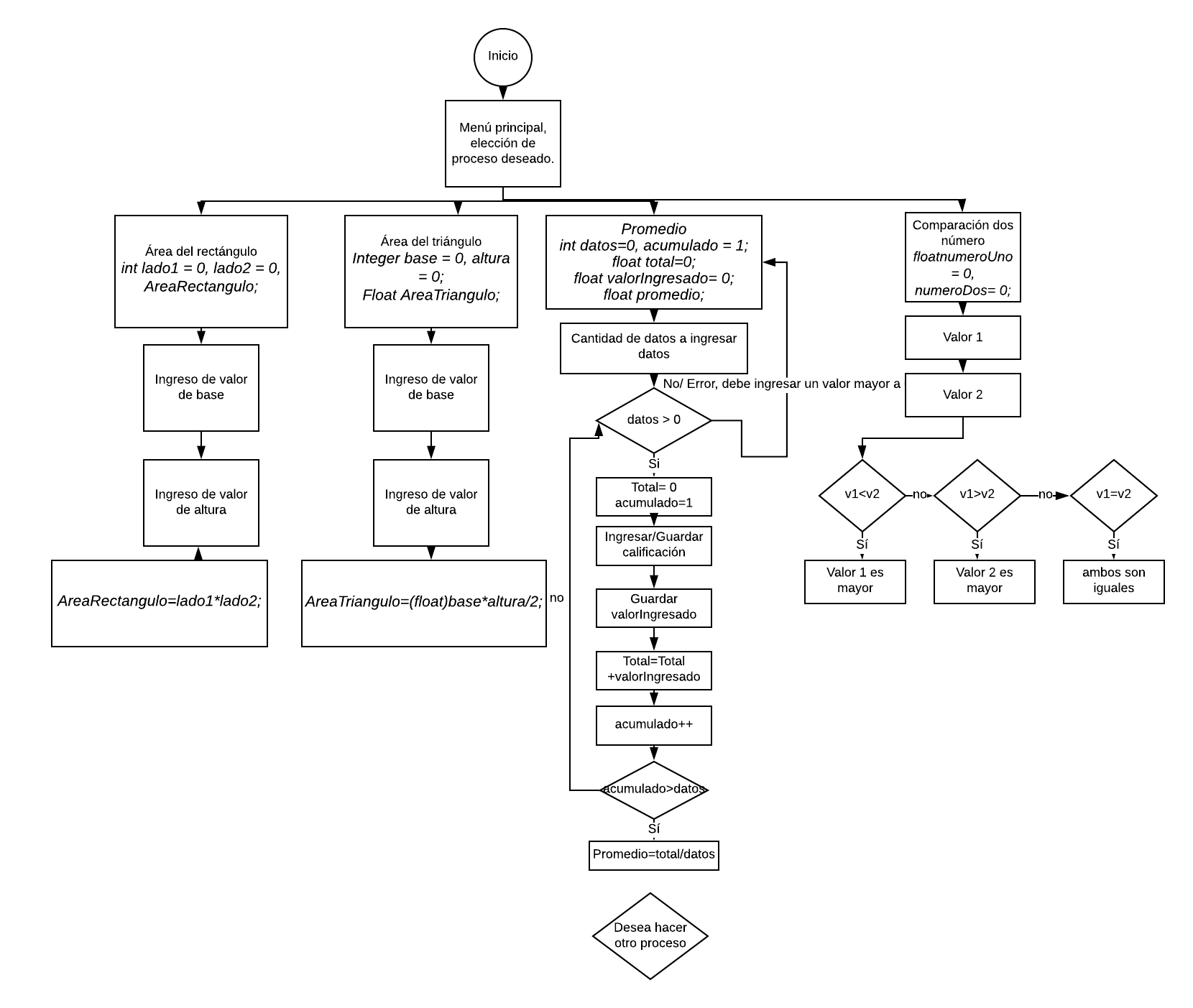
*8. Enviar un archivo comprimido con todas las evidencias a tu docente en línea para recibir la evaluación y retroalimentación correspondiente Producto por entregar: Un solo documento comprimido que contendrá*

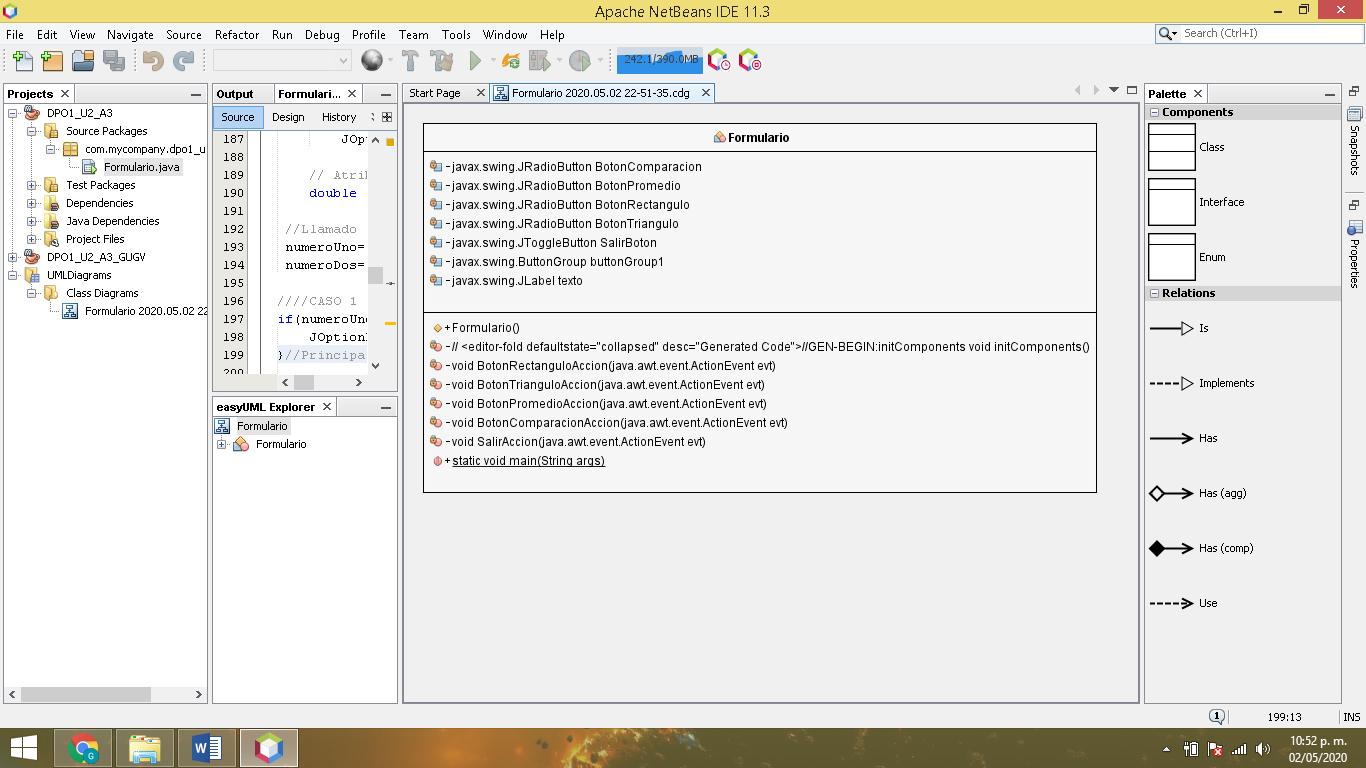
*• El proyecto comprimido generado en NetBeans y*

*• El documento en procesados de palabras*

### Desarrollo

Diagrama de flujo general del programa





**Explicación especifica de los métodos usados y de las variables**

1. Calcular el área de un rectángulo

**Rectángulo:** para el cálculo del área necesitamos dos valores, la medida de su base y la altura, por lo cual estas dos son variables.

//Rectángulo

// Atributos

*int lado1 = 0, lado2 = 0, AreaRectangulo;*

*El requerimiento es que se ingresen número enteros, por lo que el tipo de dato es int, para las variables de altura, base y el área.*

*//Metodo ingreso de datos, se requiere el usuario registre la información de entrada de las variables.*

*lado1= (int) Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de base"));*

*lado2= (int) Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de la altura"));*

*//Funciones matemáticas, para el cálculo del área se utiliza la siguiente formula, que da un valor de regreso.*

*AreaRectangulo=lado1\*lado2;*

*//Impresión de resultado*

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

**2. Calcular el área de un triángulo**

**Triángulo:** para el cálculo del área necesitamos dos valores, la medida de su base y la altura, por lo cual estas dos son variables.

*//Triángulo*

*// Atributos*

*Integer base = 0, altura = 0;*

*Float AreaTriangulo;*

El programa requiere que los datos de ingreso sean enteros, por lo que se ocupa Int, mientras que el resultado debe ser flotante.

// Método ingreso de datos, se requiere el usuario registre la información de entrada de las variables.

*base= (int) Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor de la base"));*

*altura= (int) Float.parseFloat(JOptionPane.showInputDialog("Ingrese el valor del la altura"));*

// Funciones matemáticas, para el cálculo del área se utiliza la siguiente formula, que da un valor de regreso.

*AreaTriangulo=(float)base\*altura/2;*

//Impresión de resultado

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

**3. Calcular el promedio**

*//Variables*

*int datos=0, acumulado = 1;*

*float total=0;*

*float valorIngresado= 0;*

*float promedio;*

Pensando en que las calificaciones suelen ser número reales, el tipo de datos son flotantes, excepto el de datos y acumulado que son la cantidad de calificaciones que el usuario desea ingresar, ya que no puede ingresar parciales de calificaciones.

//Método ingreso de datos de entrada, que dictamina cuántas notas esperar el sistema recibir.

*datos= (int) Float.parseFloat(JOptionPane.showInputDialog("¿Cuántas calificaciones desea ingresar?"));*

//Utilizamos el método While, para repetir el proceso hasta que se ingresen el total de calificaciones que se eligió al inicio.

*while(acumulado<=datos){*

*valorIngresado= Float.parseFloat(JOptionPane.showInputDialog("Ingrese la calificación número "+ acumulado));*

*total+=valorIngresado;*

*acumulado++;*

*}*

*//Operación para calcular el promedio de las calificaciones registradas.*

*promedio= total/datos;*

*JOptionPane.showMessageDialog(null,"El promedio de calificaciones ingresado es " + promedio);*

1. **Encontrar mayor, menor o igual de dos números**

*// Atributos*

*float numeroUno = 0, numeroDos= 0;*

El usuario tiene la posibilidad de ingresar datos con decimales. Por lo cuál el tipo de datos es flotante.

//Método ingreso de datos. Variables.

*numeroUno= Double.parseDouble(JOptionPane.showInputDialog("Ingrese el primer número"));*

*numeroDos= Double.parseDouble(JOptionPane.showInputDialog("Ingrese el segundo número"));*

Utilización de un proceso de estructuras de control selectivas, if- else, anidado. Con la finalidad de mostrar los posibles escenarios de acuerdo a los valores que el usuario puede ingresar en el sistema.

*////CASO 1*

*If (numeroUno<numeroDos){*

*JOptionPane.showMessageDialog(null,"El número " + numeroUno + " es < " + numeroDos);*

*else{*

*if (numeroUno > numeroDos){//CASO 2*

*JOptionPane.showMessageDialog(null,"El número " + numeroUno + " es > " + numeroDos);*

*} else {*

*if (numeroUno==numeroDos){ //CASO 3*

*JOptionPane.showMessageDialog(null,"El número " + numeroUno + " es = " + numeroDos);*

*} else {*

*JOptionPane.showMessageDialog(null,"Respuesta errónea");*

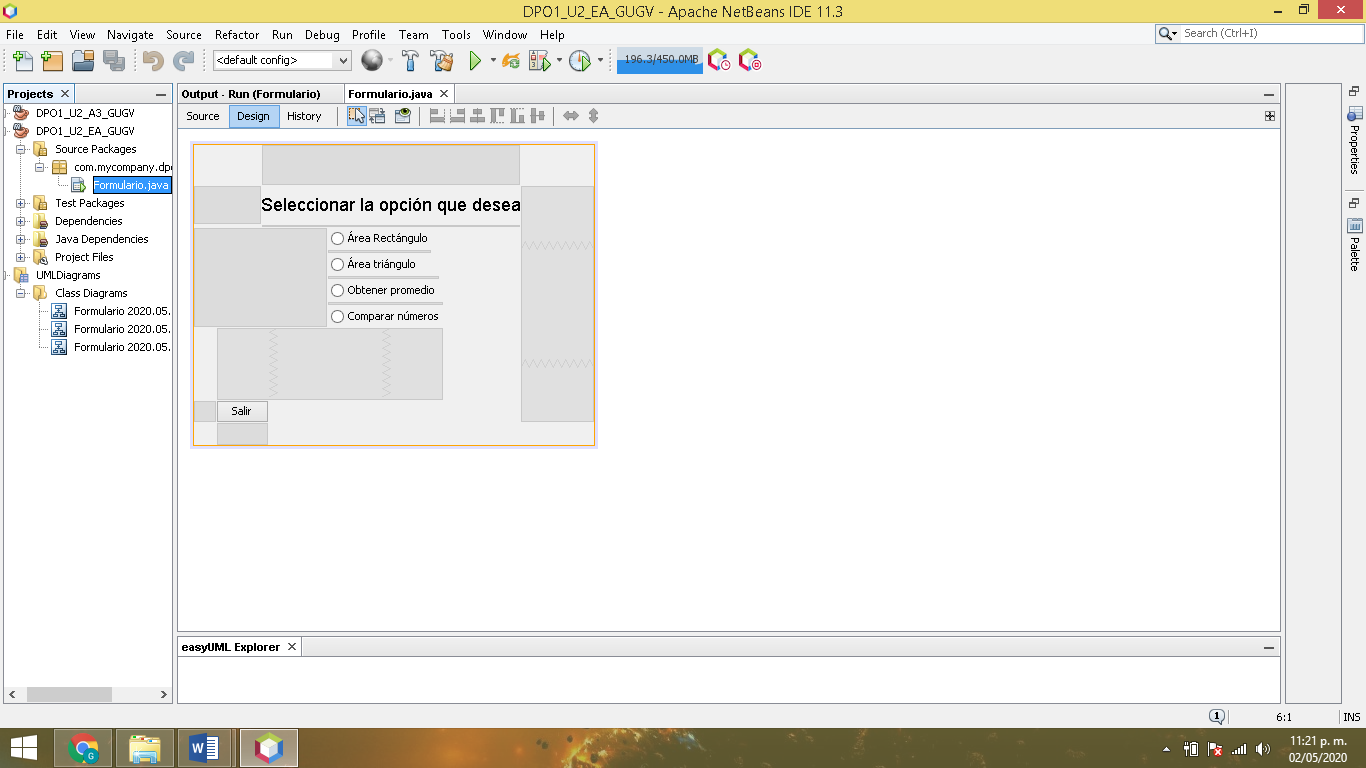
**Explicar por qué es importante el usar de manera correcta las instrucciones repetitivas y selectivas**

En la programación estructurada las indicaciones secuenciales nos permiten ahorrarnos tiempo en volver a escribir instrucciones que ocuparemos repetidamente. Esto facilita la programación, y así mismo permite detectar los errores y corregirlos de manera más eficaz. Debido a que el cambio solo se debe hacer en una parte del código y no en varios fragmentos.

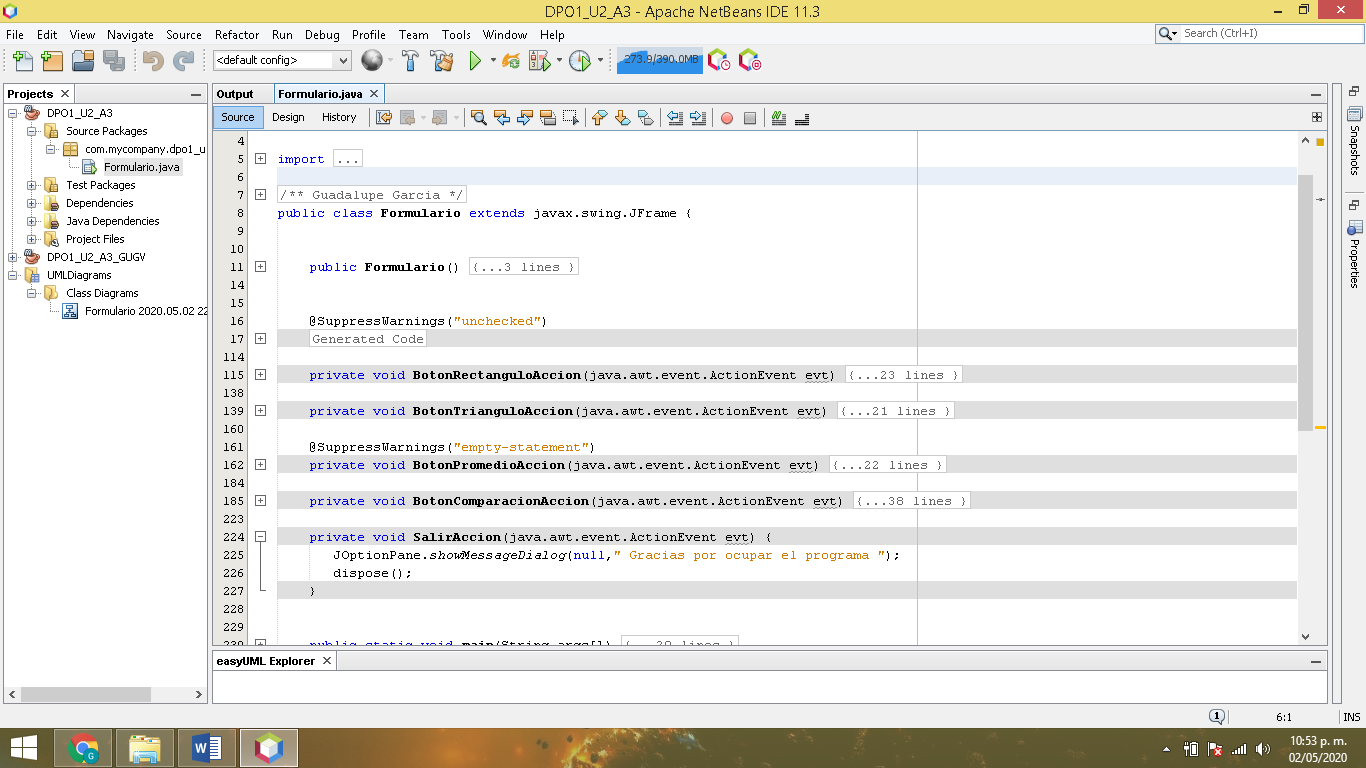
Así como las estructuras cíclicas, ya que de igual forma, facilitan el desarrollo de proyectos, ya que evitan el tener que volver a escribir el código una y otra vez. Solo basta con establecer las instrucciones y el sistema permitirá hacer uso de las instrucciones hasta que una cierta condición se cumpla.

**Realizar un proyecto en JAVA (modo texto o gráfico) en Netbeans que resuelva la siguiente situación.**

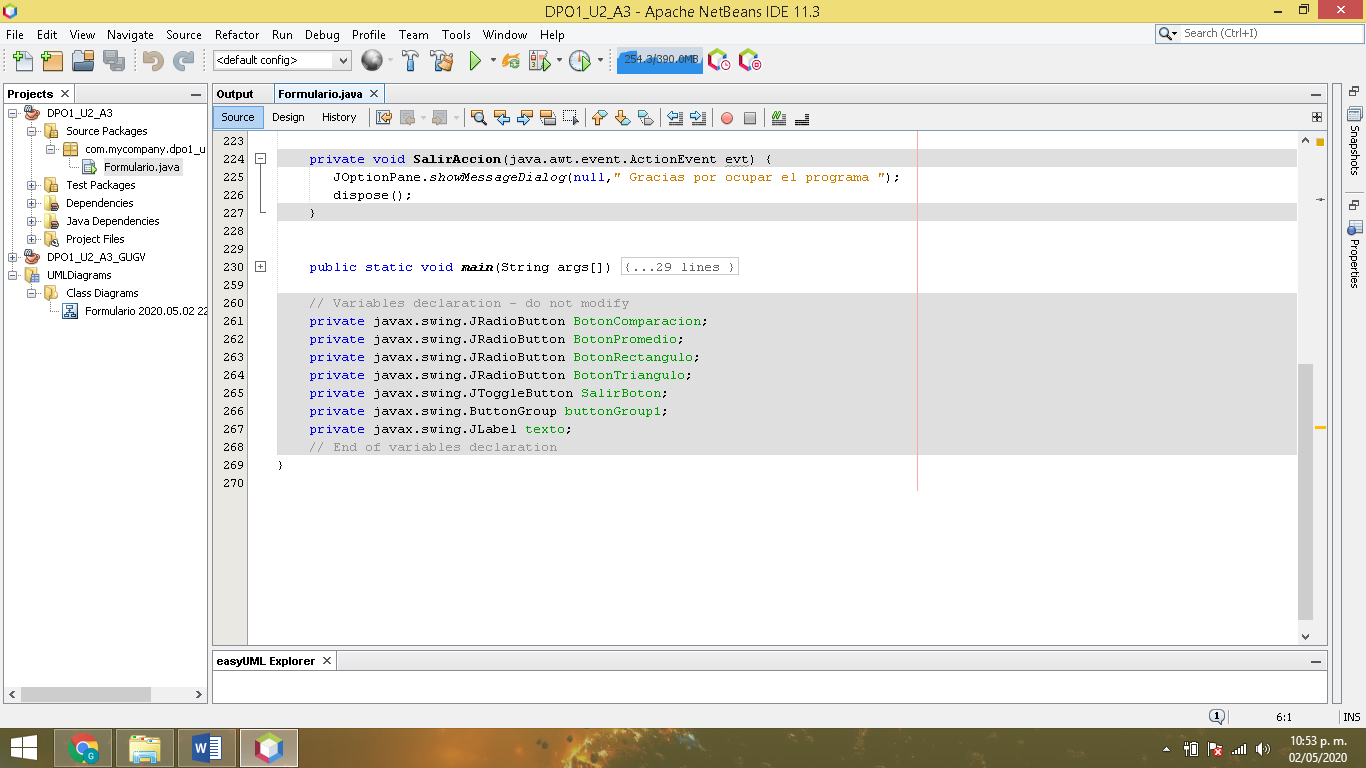
El programa desarrollado, lo hice con base en formulario gráfico, para ello, se debe de crear un nuevo formulario, en este se ingresan los botones necesarios para el proyecto. Para este, se pusieron las 4 opciones de procesos que el usuario puede acceder, así como la opción de salir, si es que ya no requiere ningún cálculo.



Netbeans basado en la configuración del formulario, codifica.

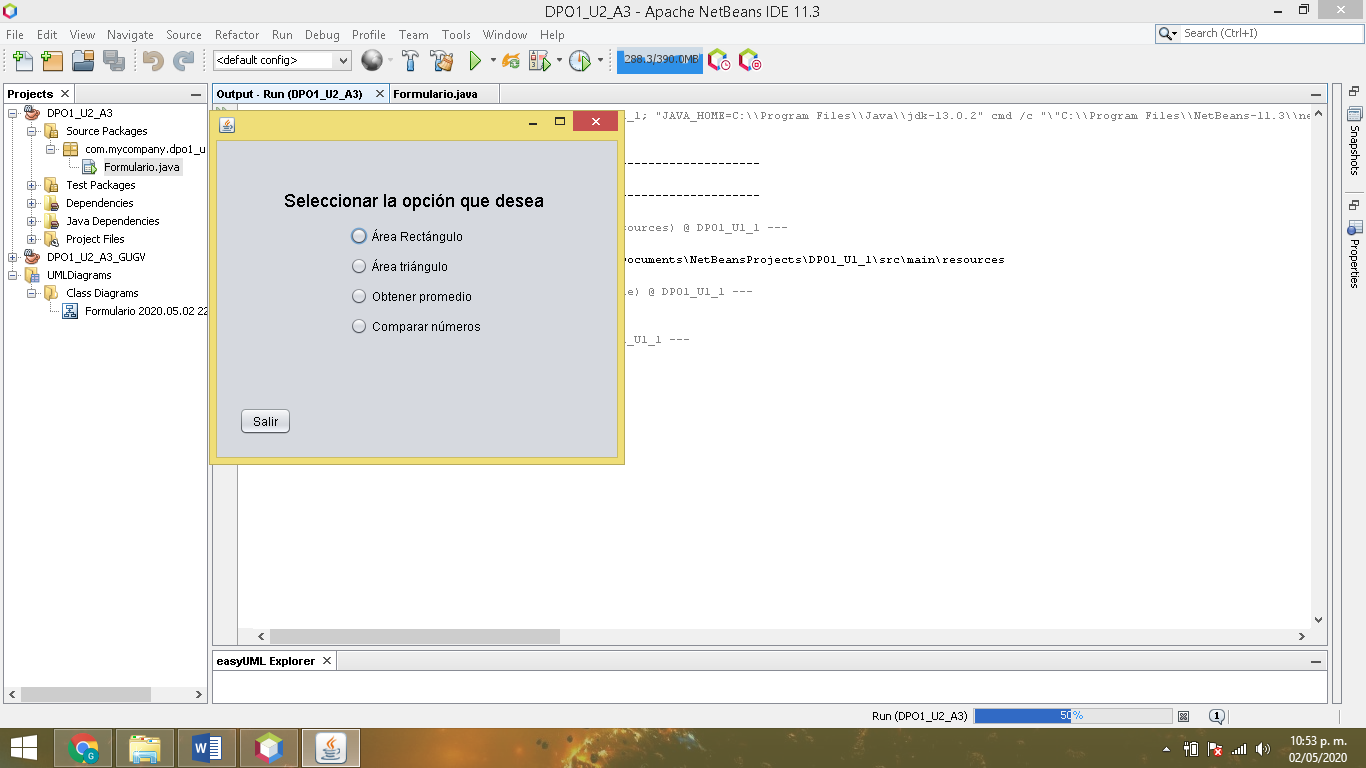


La actividad establecía utilizar Switch, pero quise realizar el formulario de forma gráfica, para una mejor interacción usuario-programa. Así que utilizando el código generado, ingrese los datos faltantes, para que cada botón realizada el método asignado.

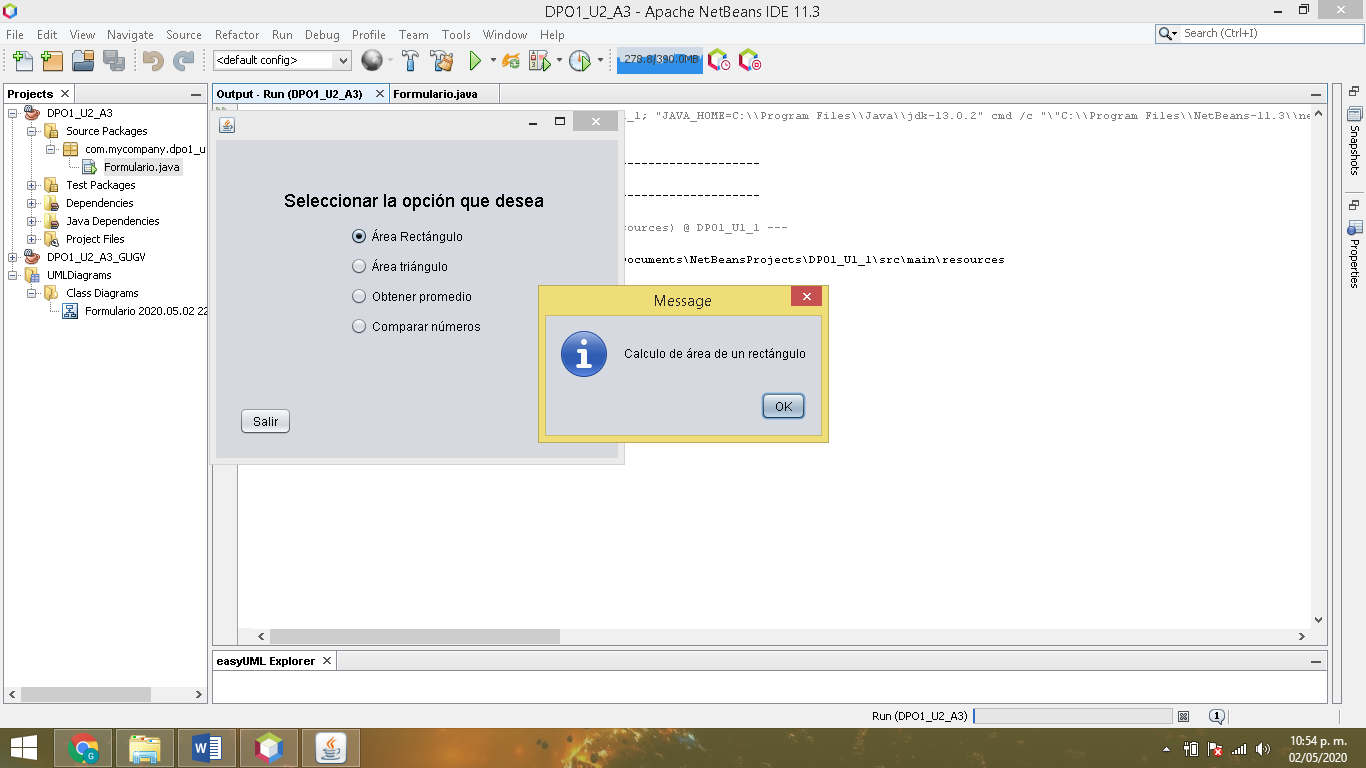


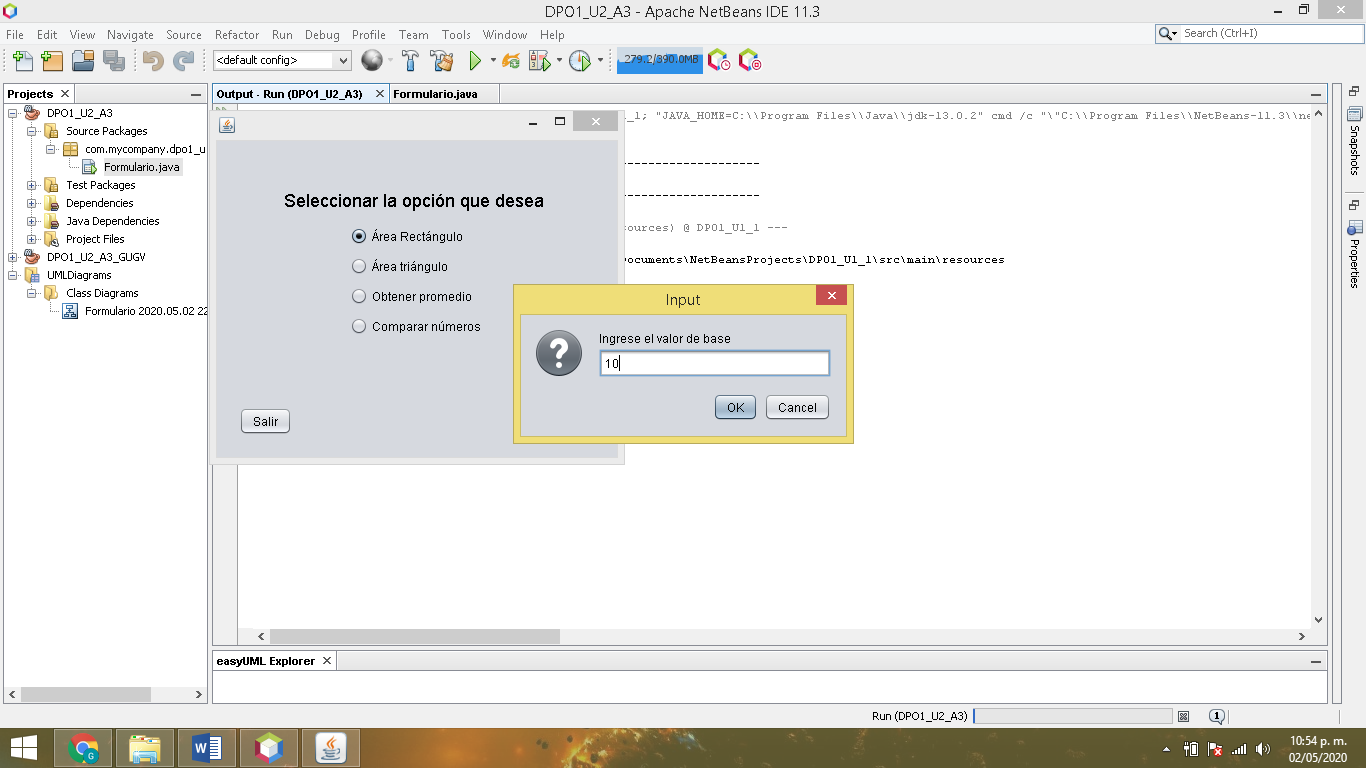
**Ejecución del programa**

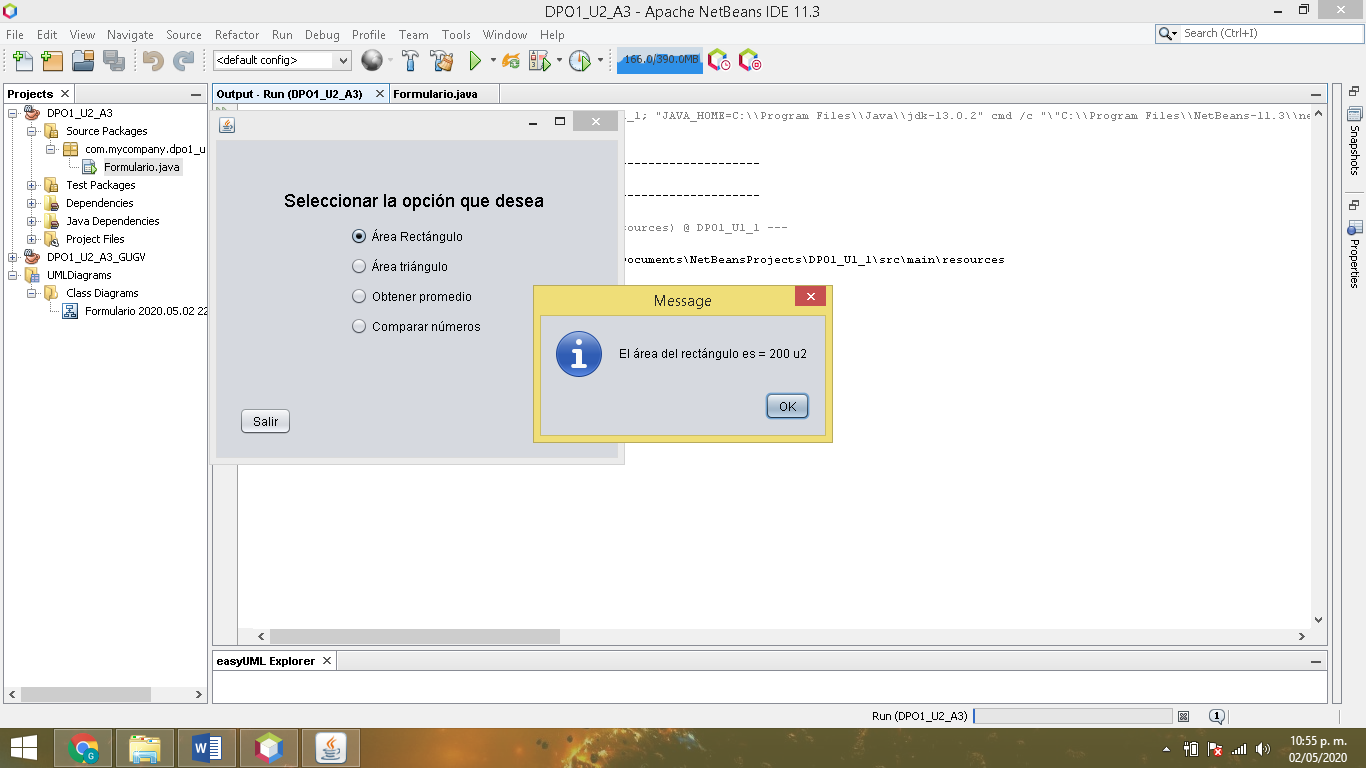
Formulario gráfico



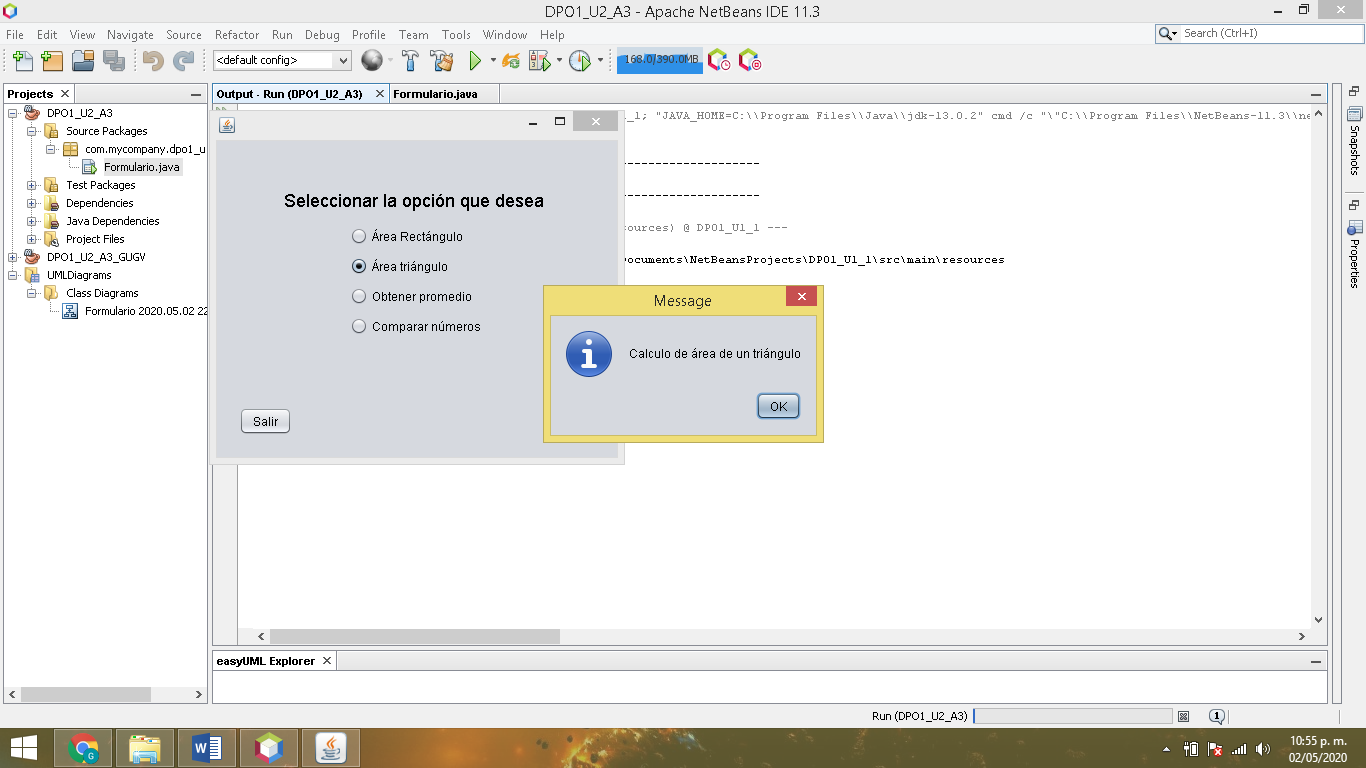
1 Calculo del área de un rectángulo.

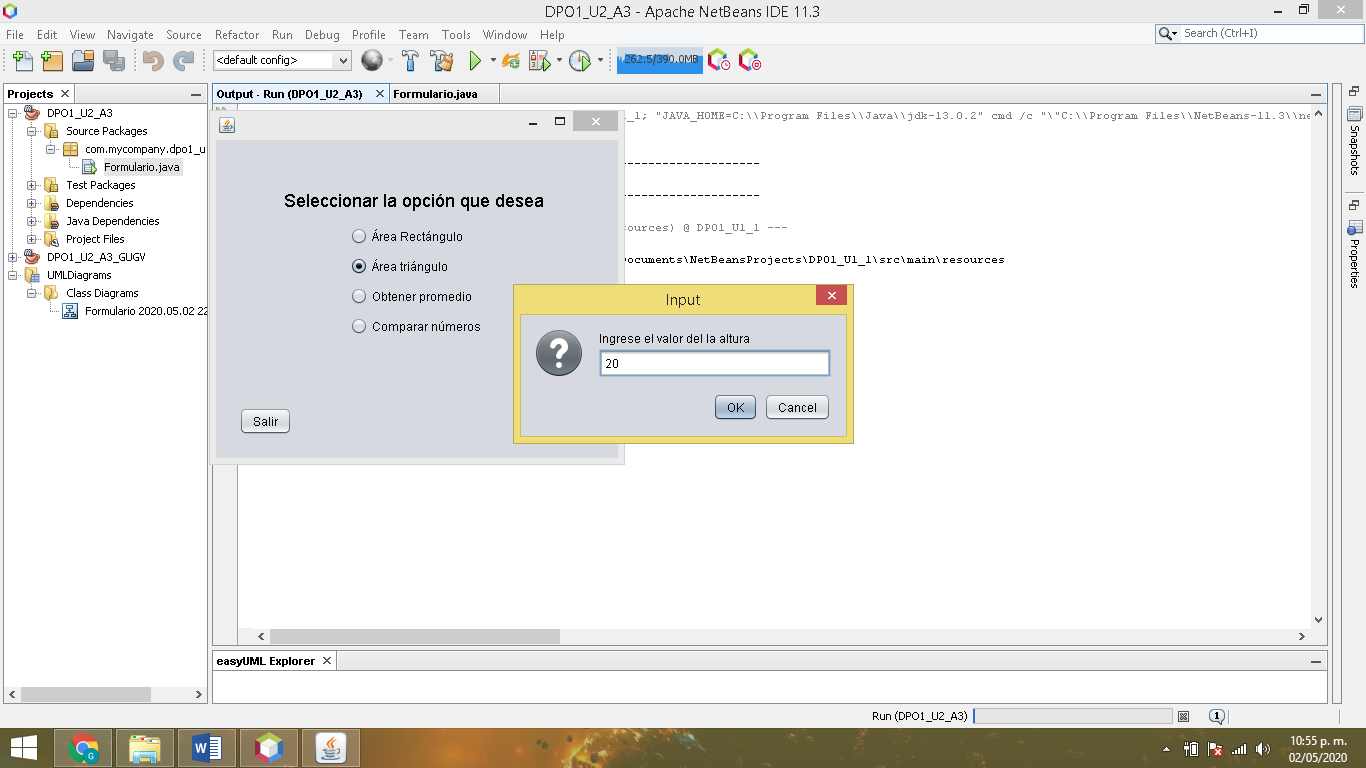


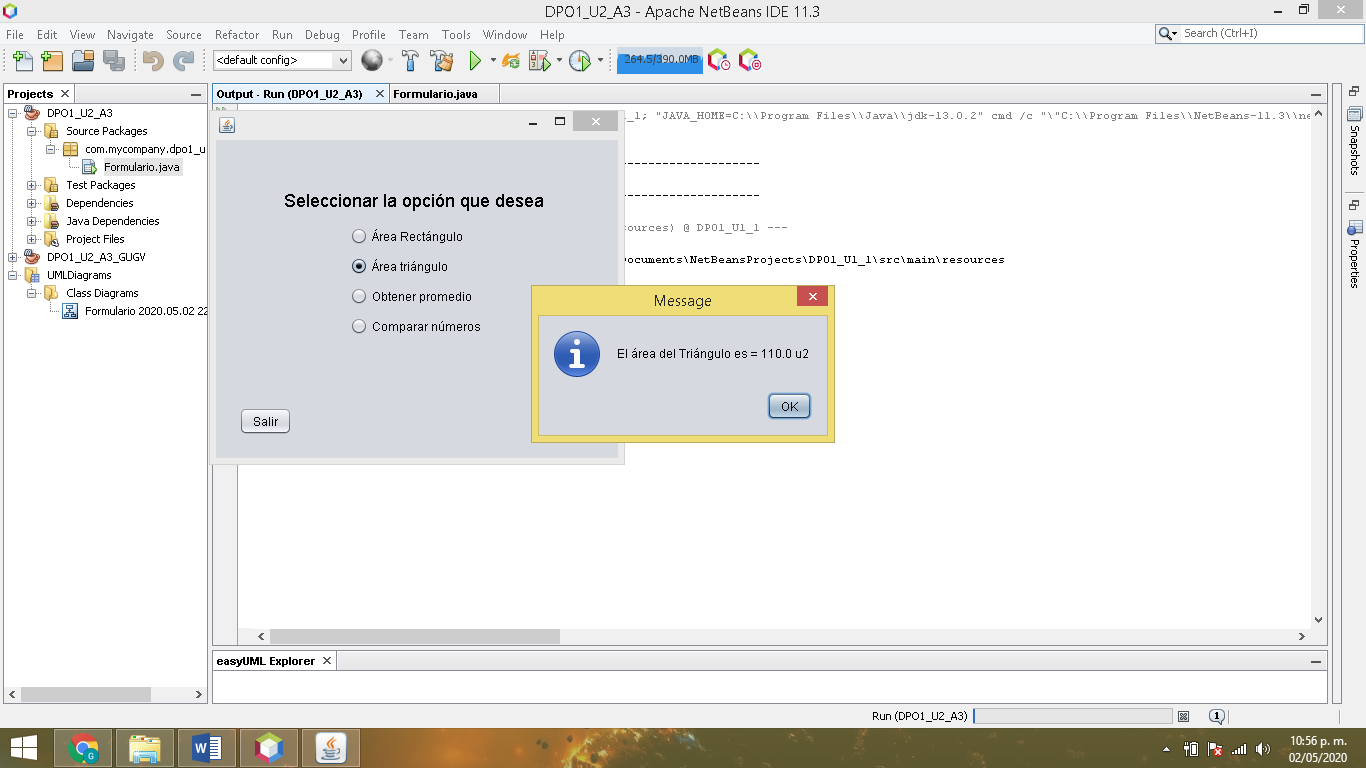




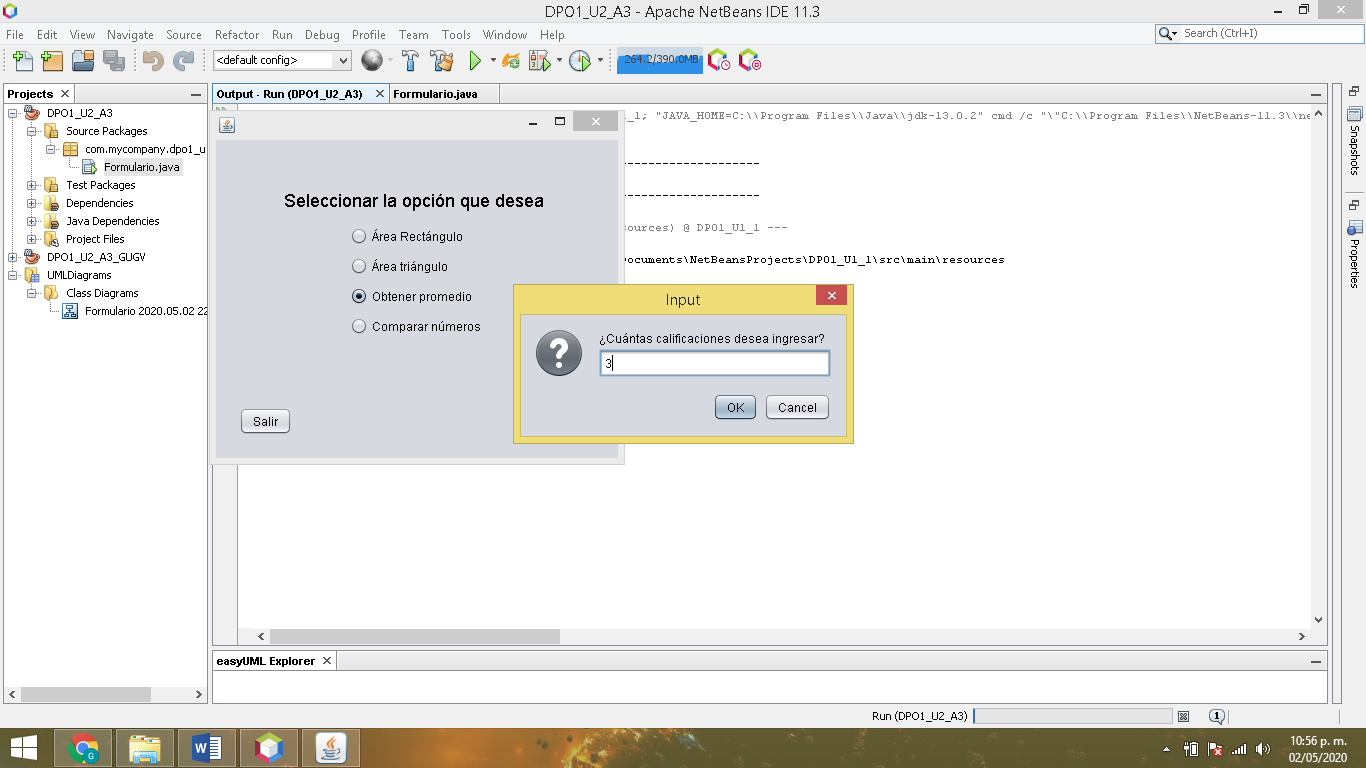
2.- Calculo de triángulo

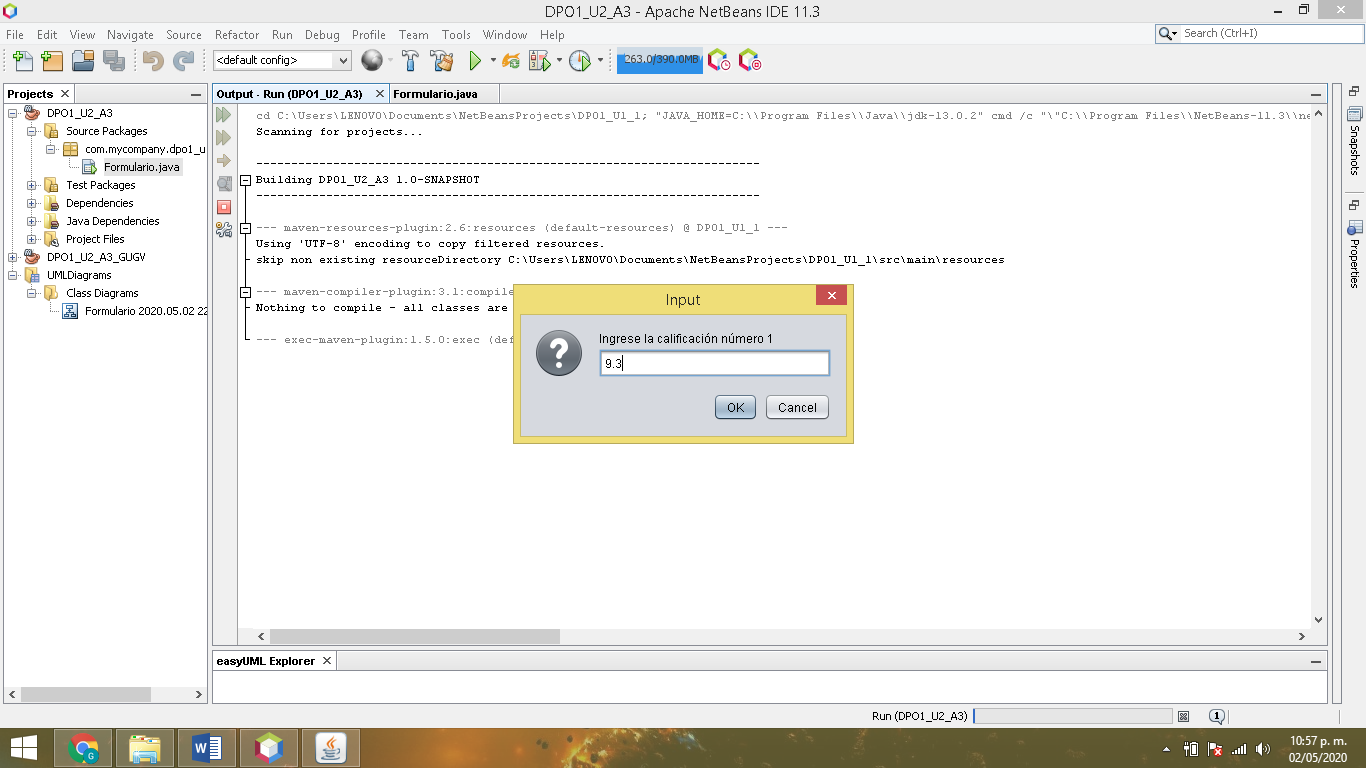


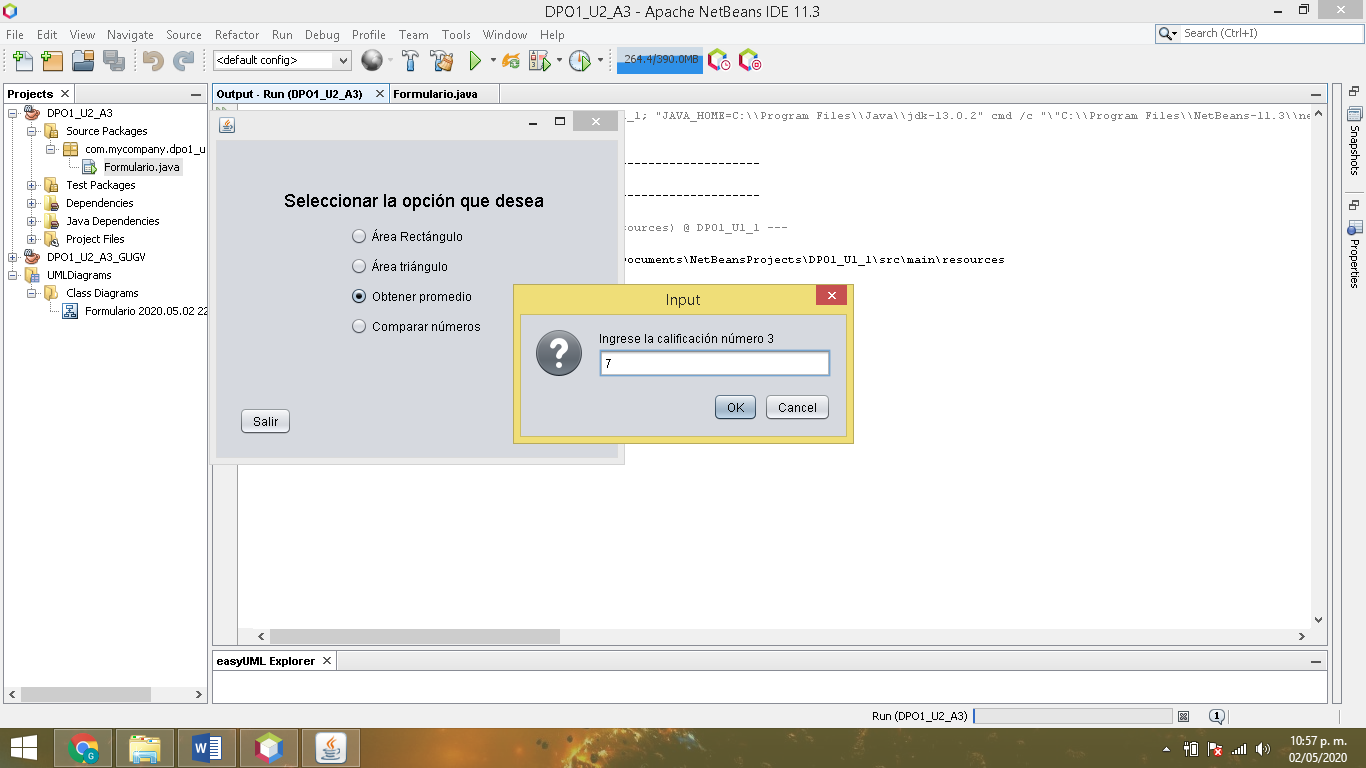


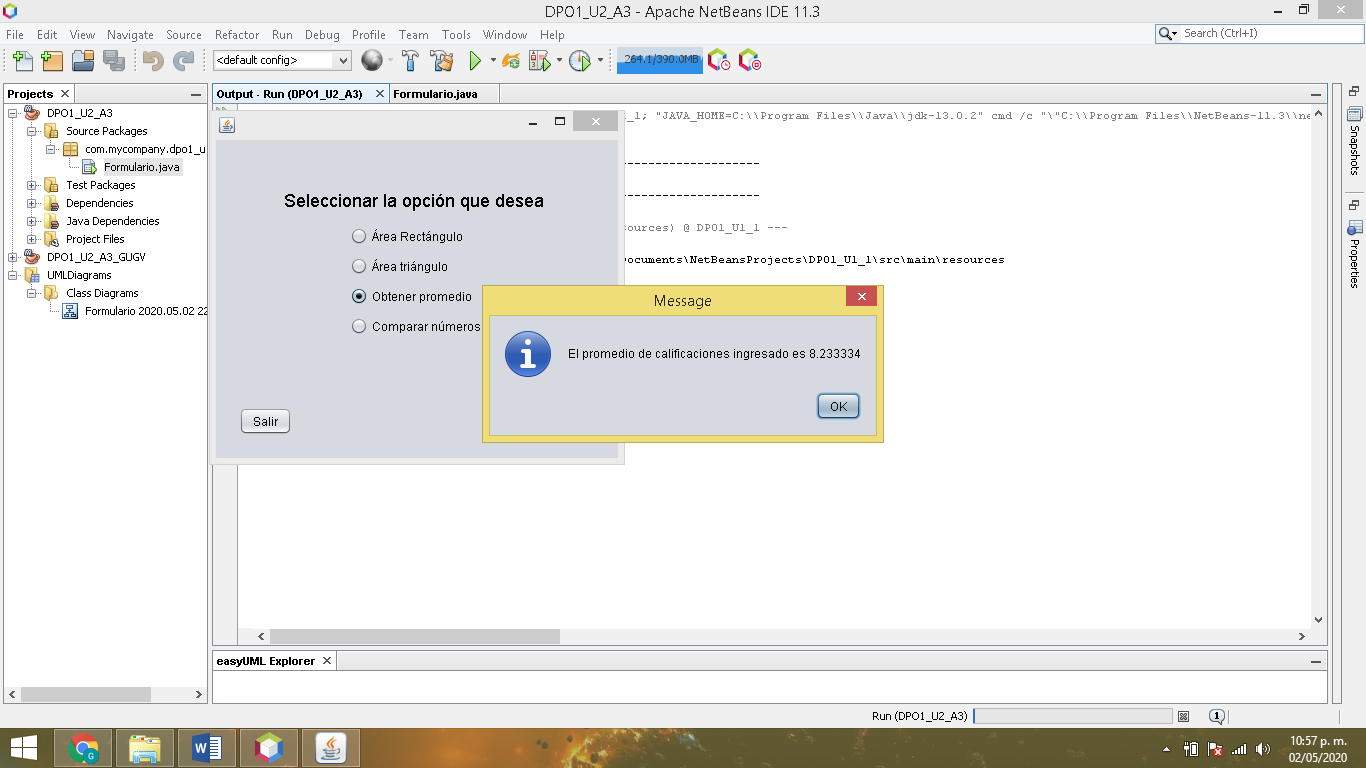


**3. Promedio**

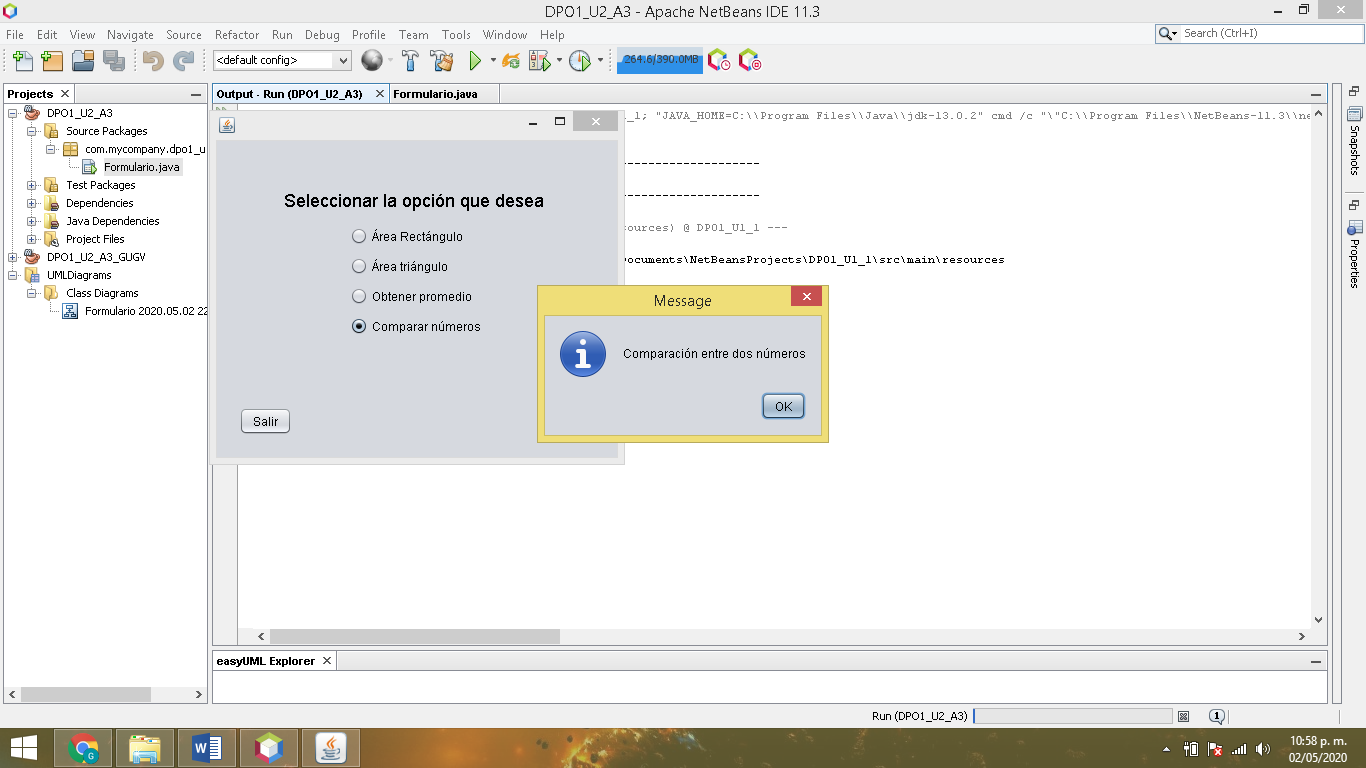


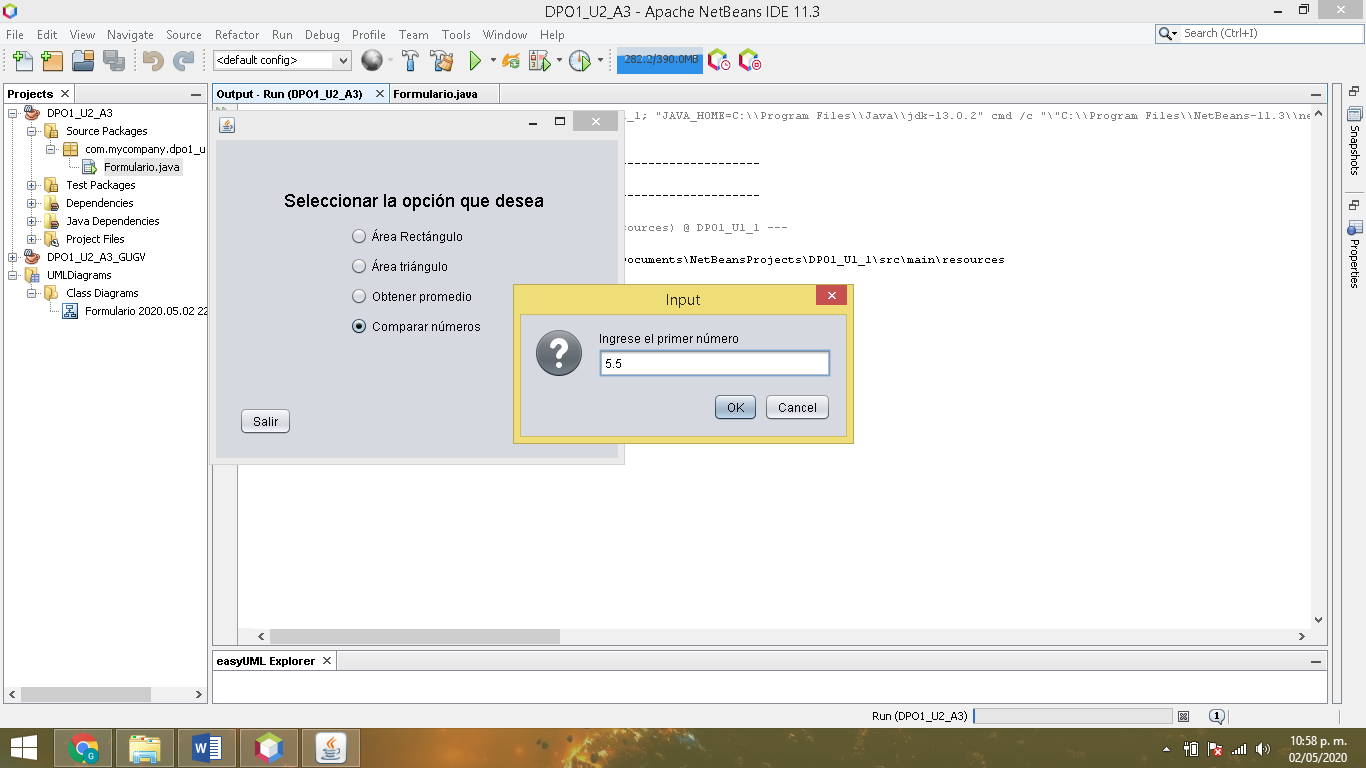


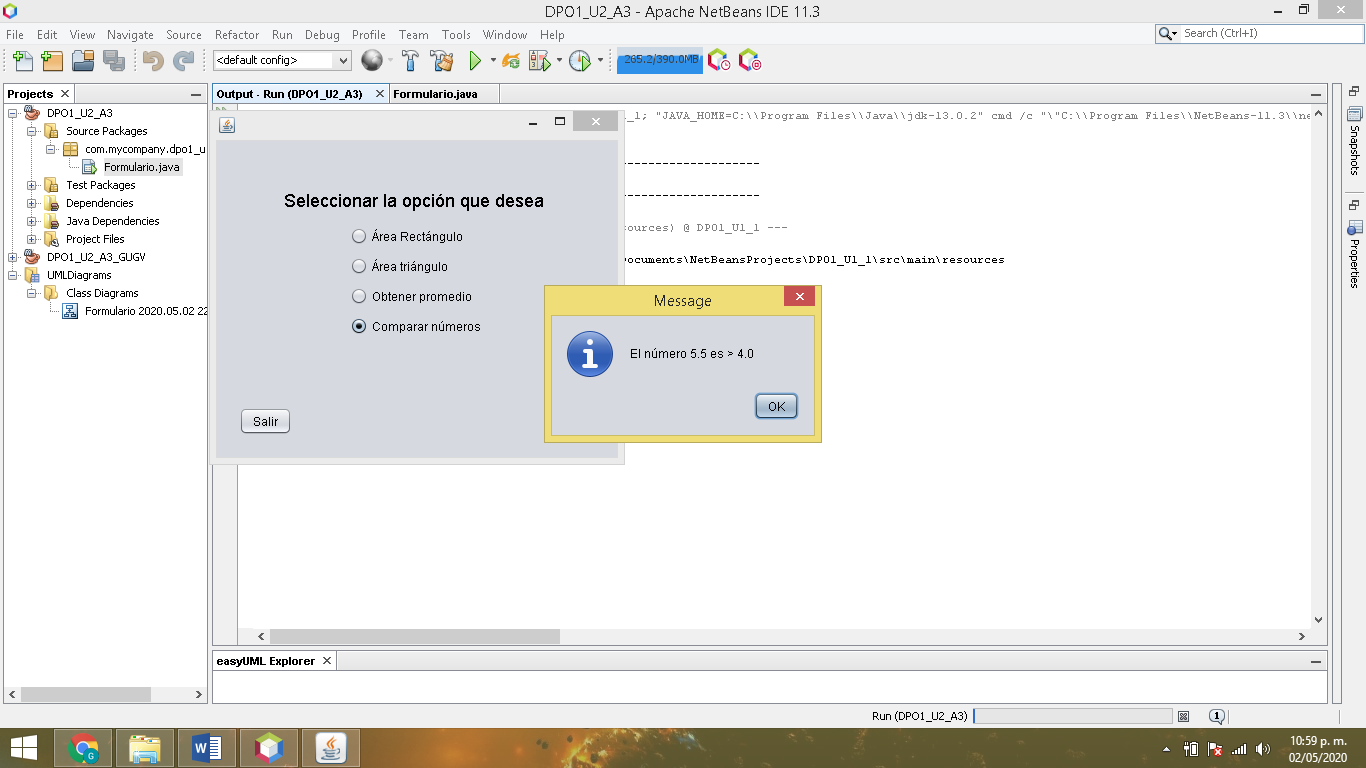


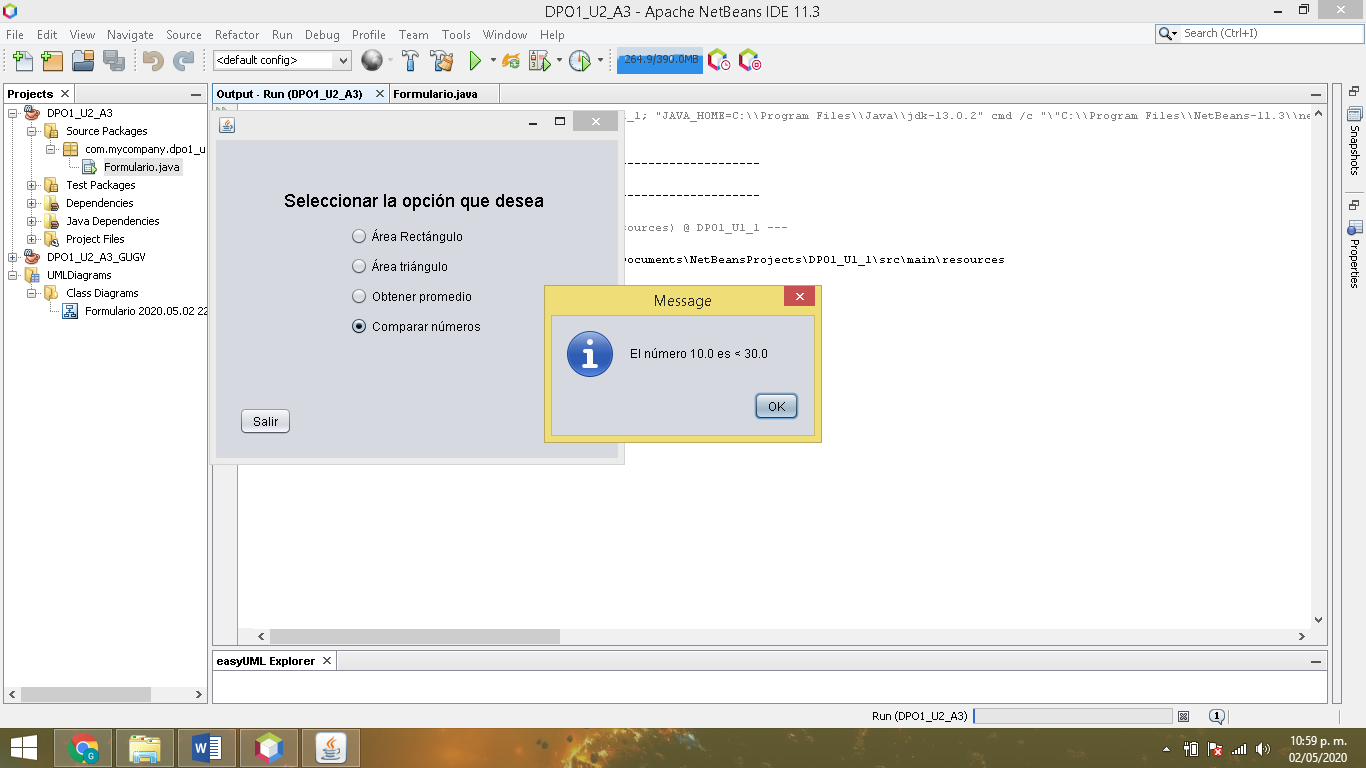


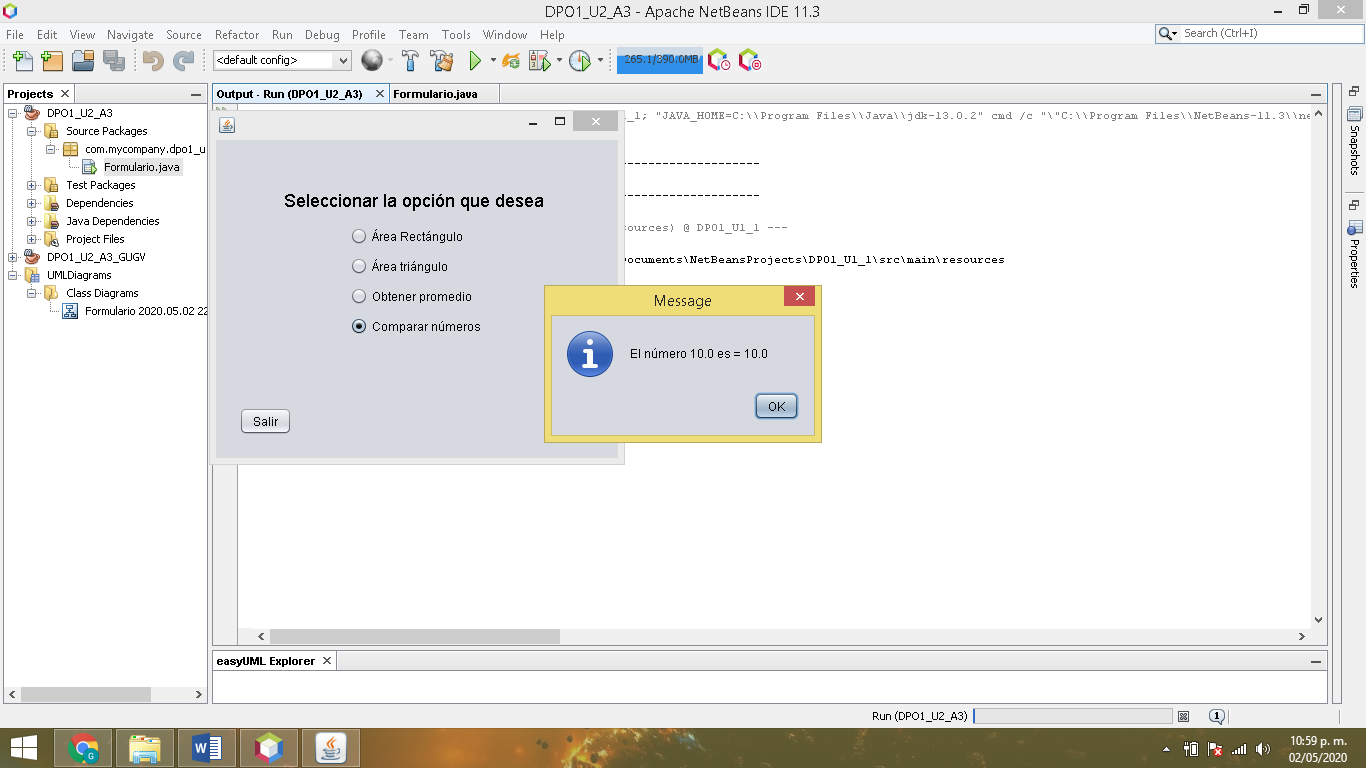
**4 Comparación entre dos número**



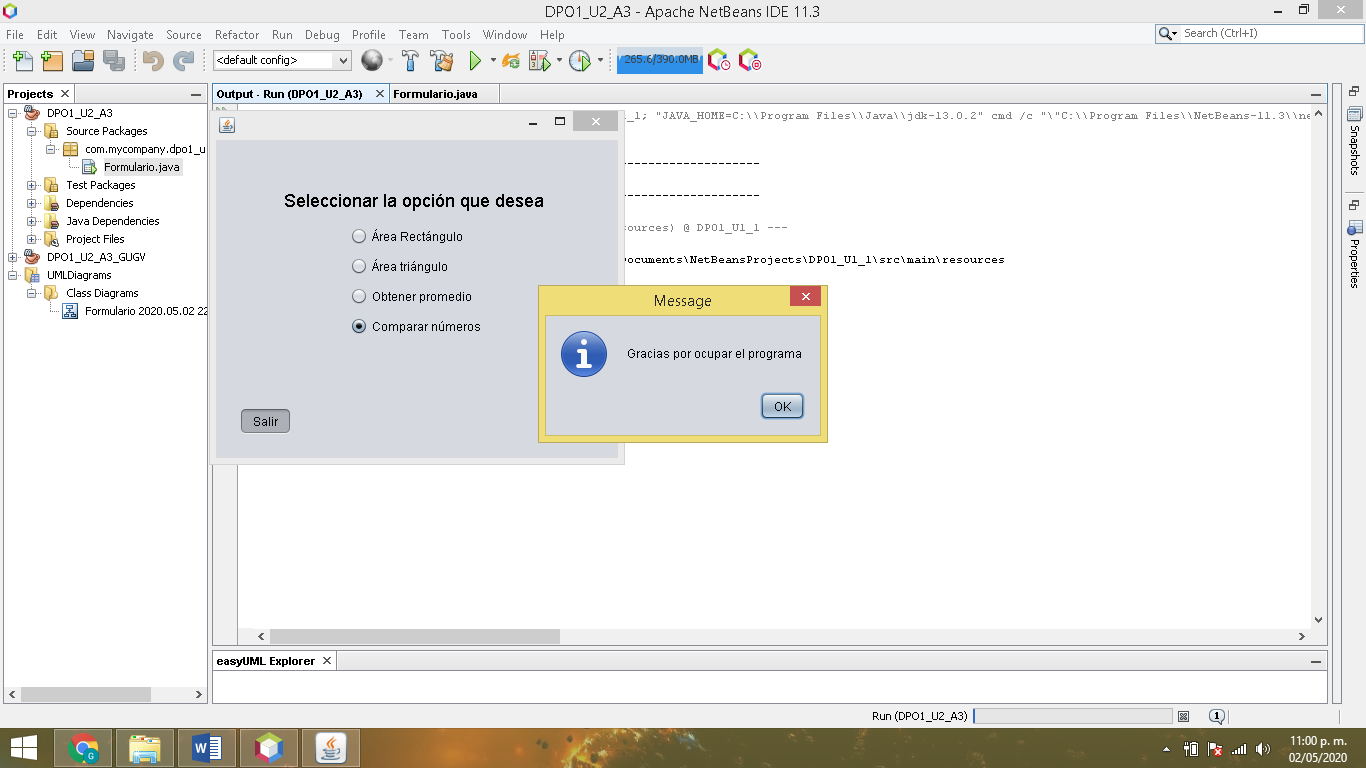








1. **Salir**



#### Conclusiones

A lo largo de esta unidad vimos el tema de las estructuras de control cíclicas, entendidas como bloques de instrucciones que se repiten un número de veces mientras se cumple, o hasta que se cumpla una condición, nos permiten que la programación sea más sencilla, ya que simplifica instrucciones.

También vimos las estructuras de control selectivas, 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, ayudan a el manejo de condicionales y que el programa tenga mayor fluidez.

En el programa desarrollado en las anteriores páginas se muestra el uso del formulario gráfico y del ciclo while y estructura if-else. Me permitió identificar como es su aplicación en un programa y las ventajas que se tiene al implementarlas. Además de seguir entendiendo como funcionan la programación orientada a objetos.

# Bibliografía

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)

González, J. D. (2012-2019). *programarya*. Obtenido de programarya: https://www.programarya.com/Cursos/C++/Ciclos

Netbeans. (s.f.). *Easy Plugin*. Obtenido de http://plugins.netbeans.org/plugin/55435/easyuml (Consultado 02/05/2020)

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].

Programador Novato (Dirección). (2019). *EasyUML Netbeans Tutorial [Crear nuestro primer UML y convertirlo a código de Java] Consultado en https://www.youtube.com/watch?v=mHDPLSs2d\_w* [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)