

Unidad 3. Características de POO y excepciones

**MATERIA: Programación orientada a objetos I**

15 de mayo de 2020

Guadalupe García Vázquez

MATRICULA: ES1921008556

Desarrollo en Software

**Docente: Claudia Erika González Gómez**

**Actividad 3: Manejo de excepciones**

Contenido

[Introducción 3](#_Toc40652176)

[Indicaciones 5](#_Toc40652177)

[Desarrollo 7](#_Toc40652178)

[Conclusiones 17](#_Toc40652179)

[Bibliografía 18](#_Toc40652180)

Imágenes

[Ilustración 1 Try-Catch 4](#_Toc40652386)

[Ilustración 2 Ejemplo de Jerarquía de excepciones 5](#_Toc40652387)

[Ilustración 3 Programa Figuras 8](#_Toc40652388)

[Ilustración 4 Programa con Formato 8](#_Toc40652389)

[Ilustración 5 Exception 1 9](#_Toc40652390)

[Ilustración 6 Exception- java.lang.NumberFormatException: For input string: "Si" 10](#_Toc40652391)

[Ilustración 7 Exception java.lang.NumberFormatException: For input string: "2.3" 11](#_Toc40652392)

[Ilustración 8 Exception 2 11](#_Toc40652393)

[Ilustración 9 Exception 3 16](#_Toc40652394)

# Introducción

El siguiente tema veremos el concepto de excepciones, pero ¿qué son las excepciones?

Cuando un programa Java viola las restricciones semánticas del lenguaje (se produce un error), la máquina virtual Java comunica este hecho al programa mediante una excepción. Por tanto, la manera más simple de explicarlo es que una excepción es un error que ha ocurrido en un programa en ejecución.

Muchas clases de errores pueden provocar una excepción, desde un desbordamiento de memoria o un disco duro estropeado, un intento de dividir por cero o intentar acceder a un arreglo fuera de sus límites.

¿Qué pasa cuando hay un error?

Cuando algún error como los mencionados ocurre, la máquina virtual Java crea un objeto de la clase **exception**, se notifica el hecho al sistema de ejecución y se dice que se ha lanzado una excepción.

Lo que acontece a continuación es que el programa en ejecución es interrumpido por dicho error mostrándolo, como típicamente se suele decir, “el programa truena”.

Pero en java disponemos de un mecanismo consistente en el uso de bloques **try/catch/finally.**

Consiste en colocar las instrucciones que podrían provocar problemas dentro de un bloque try, y colocar a continuación uno o más bloques catch, de tal forma que si se provoca un error de un determinado tipo, lo que haremos será saltar al bloque catch capaz de gestionar ese tipo de error específico.

El bloque catch contendrá el código necesario para gestionar ese tipo específico de error. Suponiendo que no se hubiesen provocado errores en el bloque try, nunca se ejecutarían los bloques catch.

Ilustración 1 Try-Catch

Esencialmente cuando se refiere a los tipos de errores en el lenguaje Java, éstos se

Clasifican dentro de dos **tipos básicos:**

Runtime Exception

1. Aquellos generados por el lenguaje Java, éstos se generan cuando hay errores de ejecución, como al tratar de acceder a métodos de una referencia no asignada a un objeto, división por cero, etc. Es decir, que al programar no se codificó con la sintaxis correcta, y no se previeron estos posibles errores al crear el programa.

IOException

2. Aquellos no generados por el lenguaje, sino incluidos por el programador, como al tratar de leer un archivo o tratar de hacer conexión a una base de datos específica.

Jerarquía de excepciones (ejemplo)

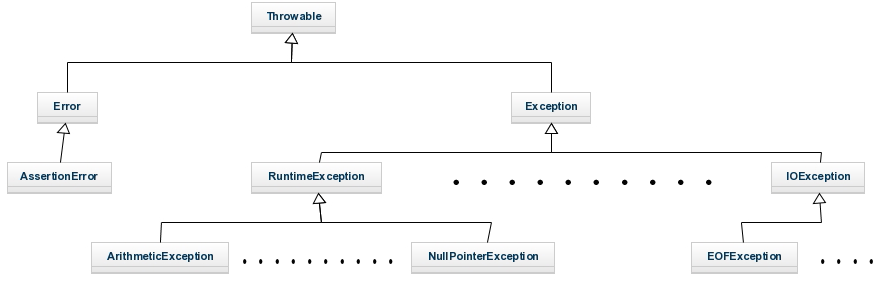
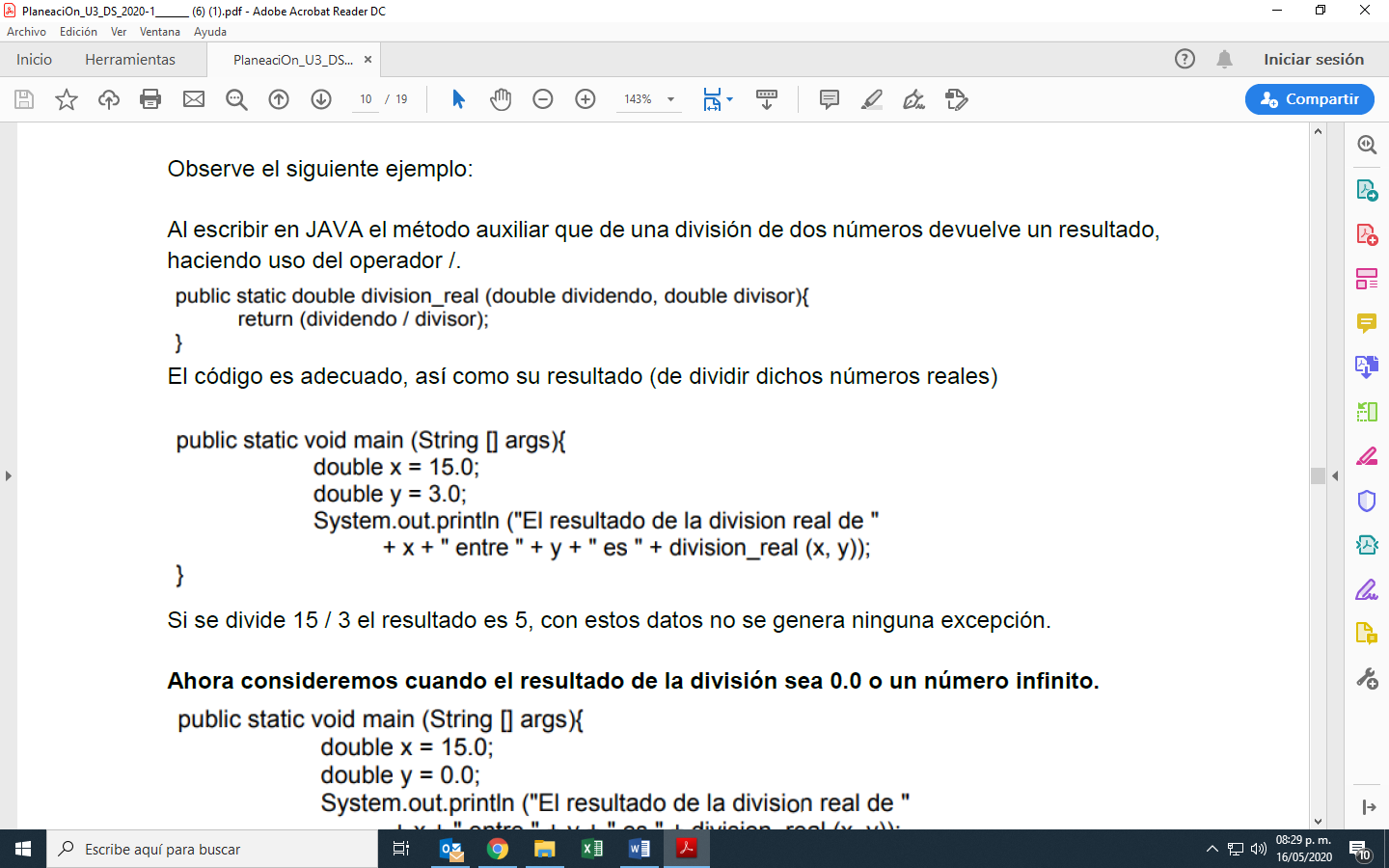


Ilustración 2 Ejemplo de Jerarquía de excepciones

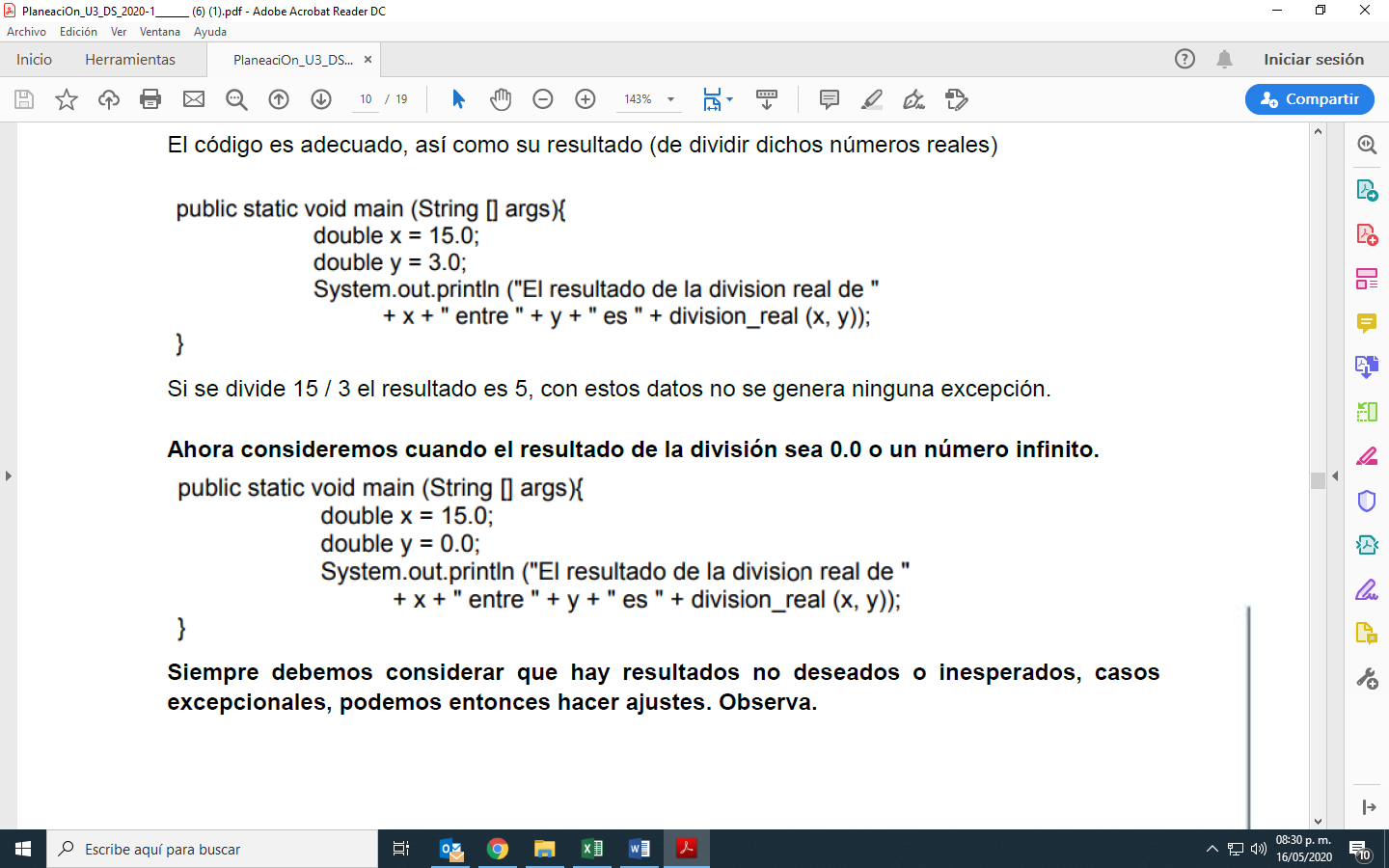
### Indicaciones

1. Lee detalladamente y analiza el siguiente planteamiento de problema:

Las excepciones permiten conseguir, que, si un código fuente no se ejecuta de la forma que se planeaba desde un inicio, el desarrollador de software pueda controlar esa situación e indicar como debe responder el programa ante tal eventualidad.









2. Ahora que analizó el caso anterior, deberá *desarrollar un ejemplo* (de su autoría) donde se observe que implementa por lo menos tres excepciones diferentes, para ello puede utilizar cualquier programa desarrollado en esta materia. Debe argumentar porque es una excepción y de qué tipo es.

3. Documente su proyecto, en el desarrollo debe explicar a detalle y agregando capturas de pantalla de las corridas.

Integra la solución y guarda el documento de tu actividad que debe tener:

- Portada

- Índice

- Introducción

- desarrollo

a) Código

b) Excepciones, explicar de qué tipos son y cómo funcionan en Java

c) Capturas de pantalla donde se muestre como funciona cada excepción en las clases

- Conclusión

### Desarrollo

Trabajaré con un proyecto generado la unidad pasada, consistía en el cálculo del área de figuras geométricas.

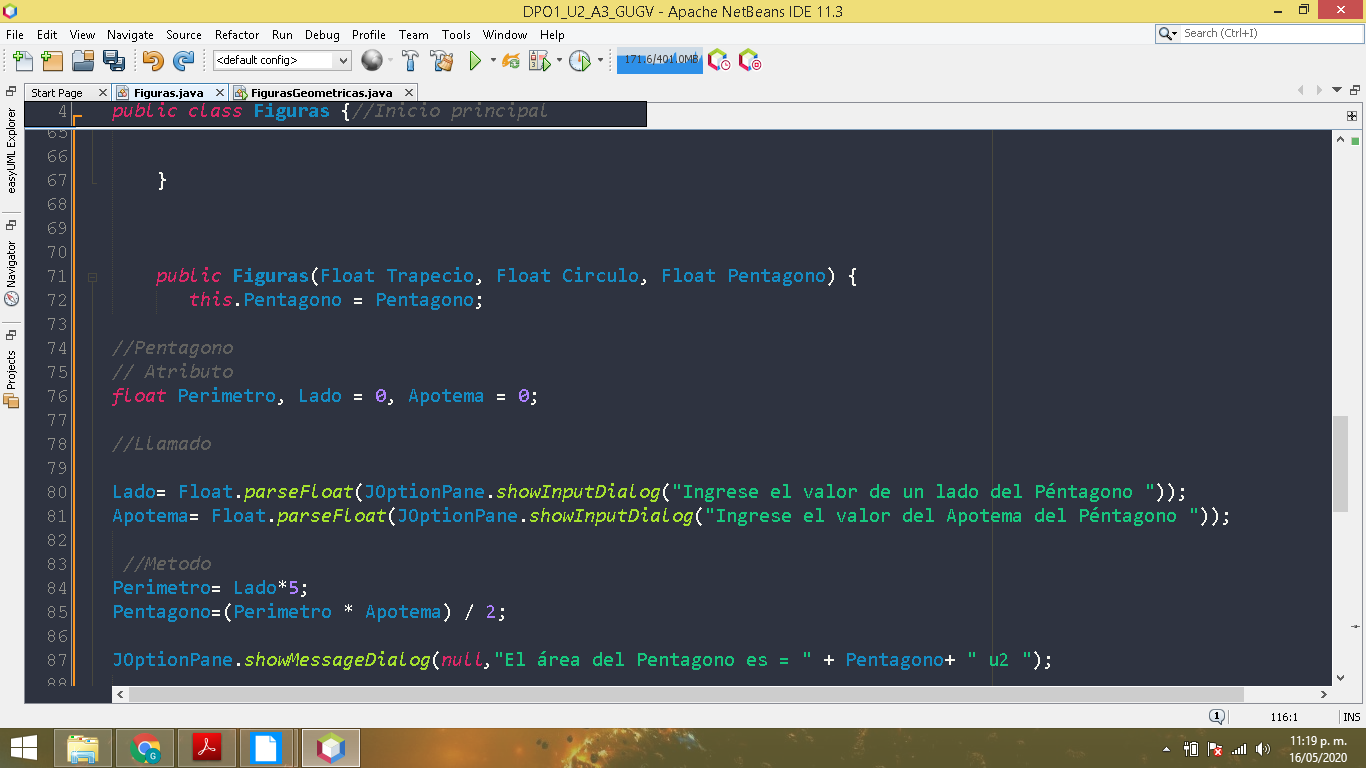


Ilustración 3 Programa Figuras

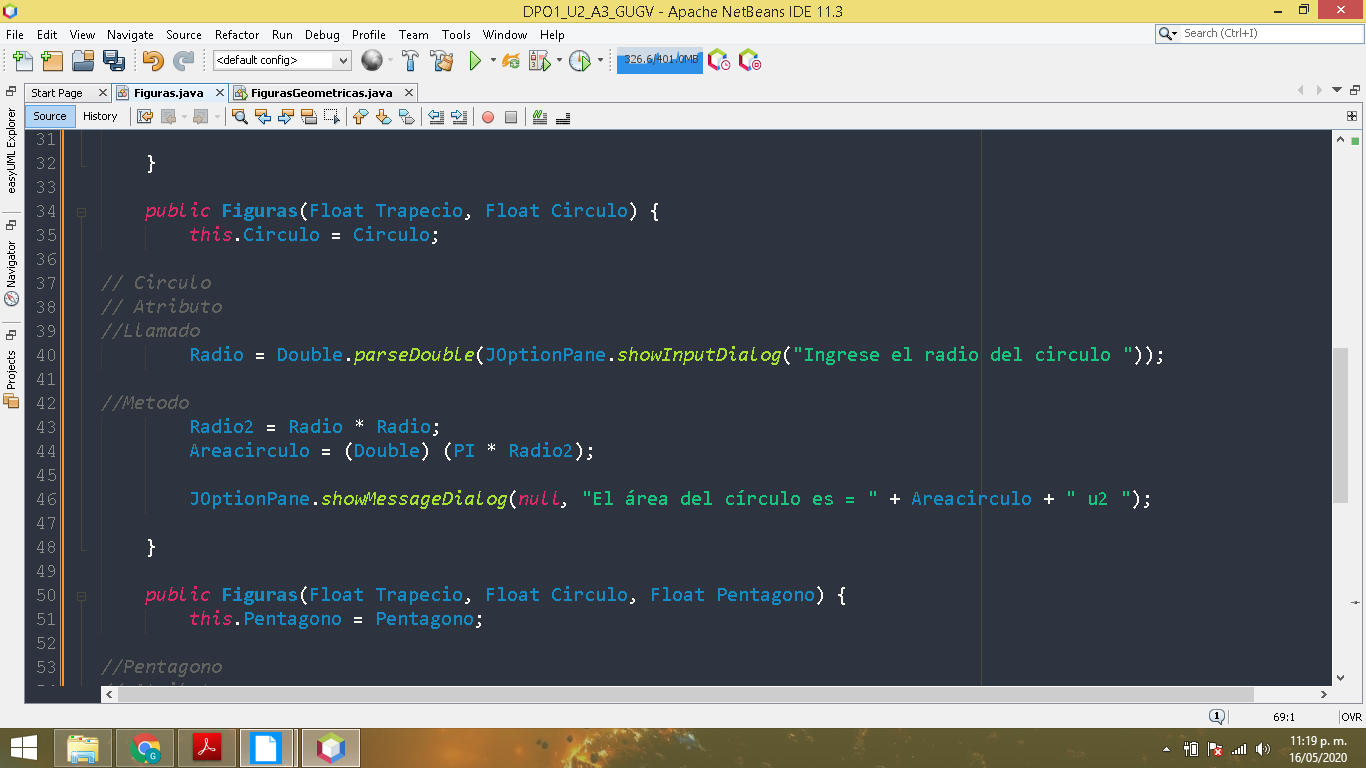


Ilustración 4 Programa con Formato

Error 1

Al inicio del programa se le pide al usuario digitar un número

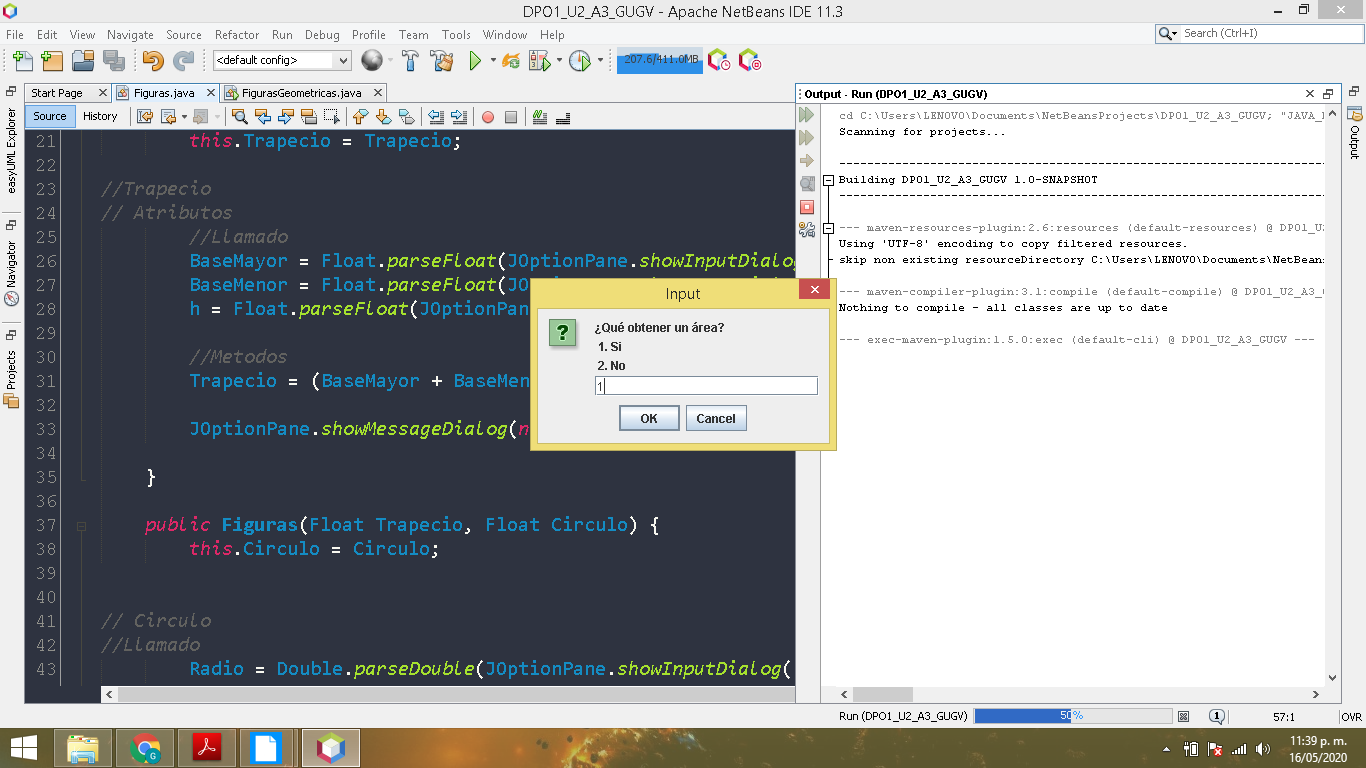
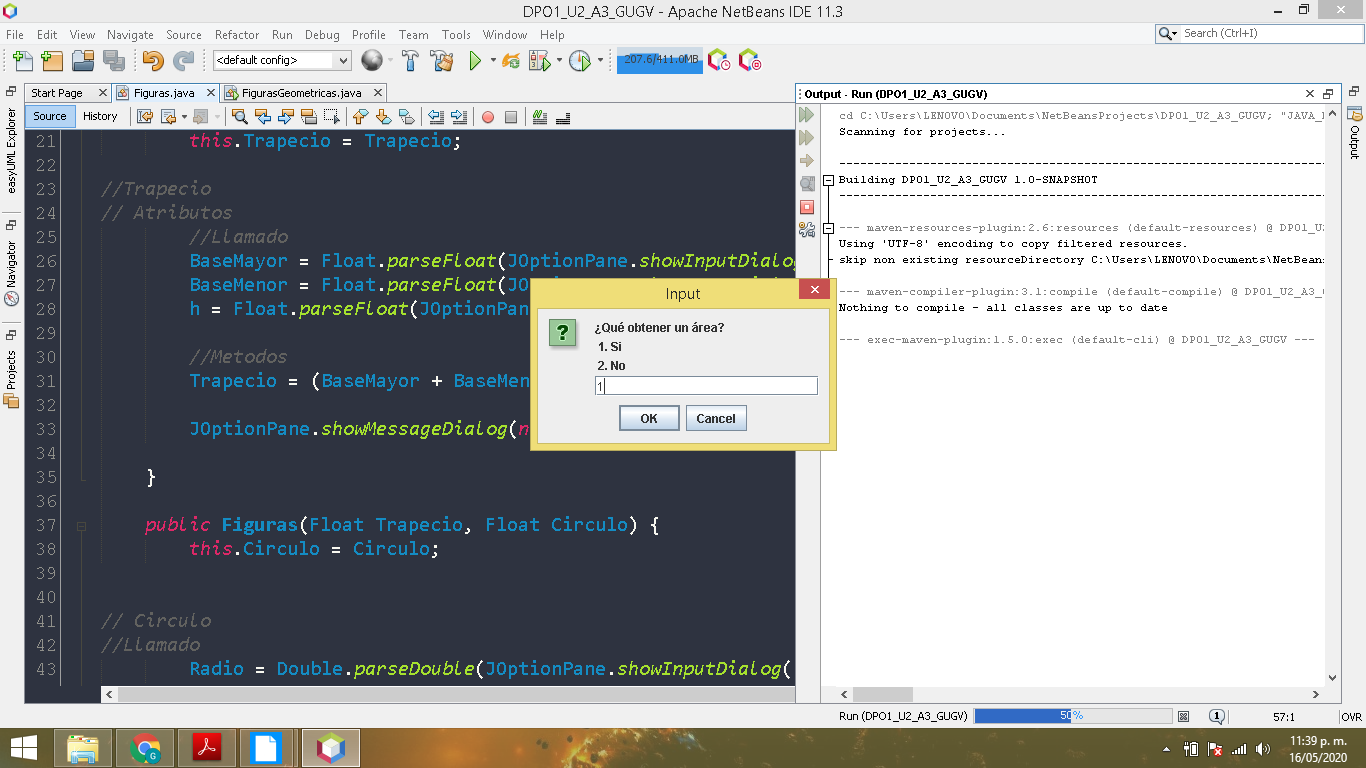
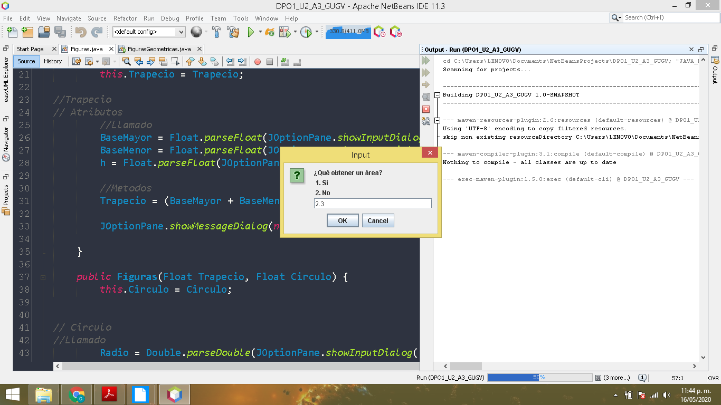
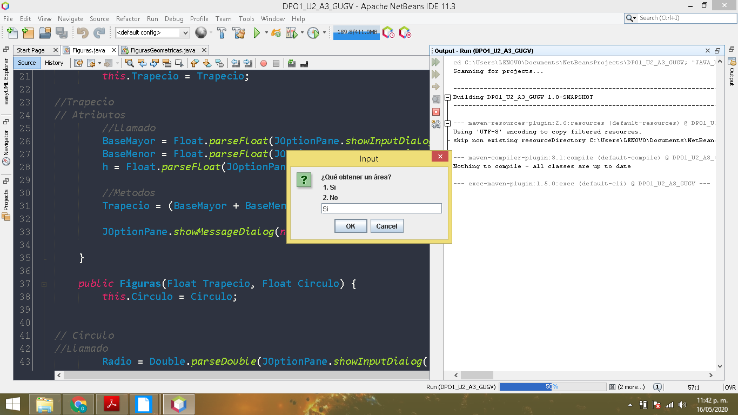


Ilustración 5 Exception 1

Si el usuario ingresa correctamente ya sea el número 1 o el dos, el programa sigue.

Pero ¿que pasa si ingresa otra opción, una cadena o un valor real?



El sistema espera recibir uno de esos dos valores, y si el usuario no los ingresa, el programa tiene un problema

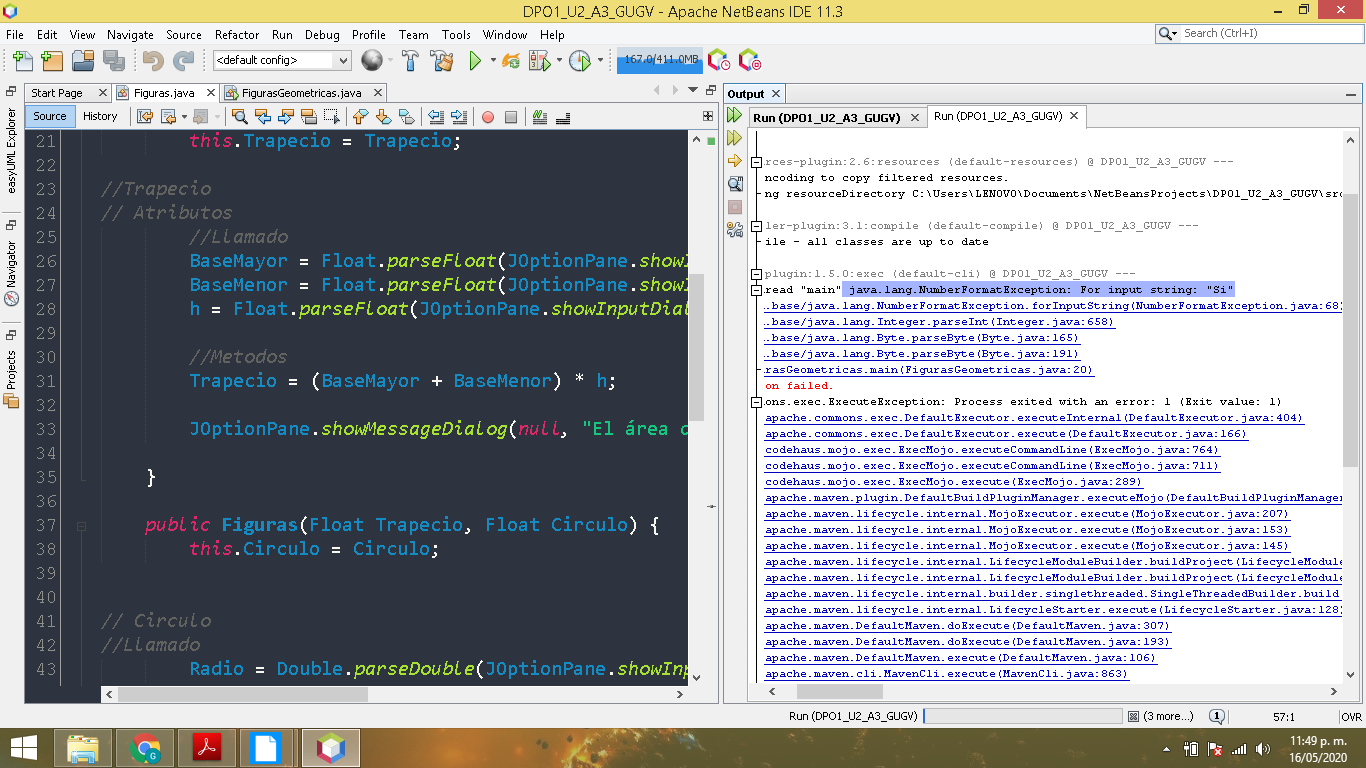


Ilustración 6 Exception- java.lang.NumberFormatException: For input string: "Si"

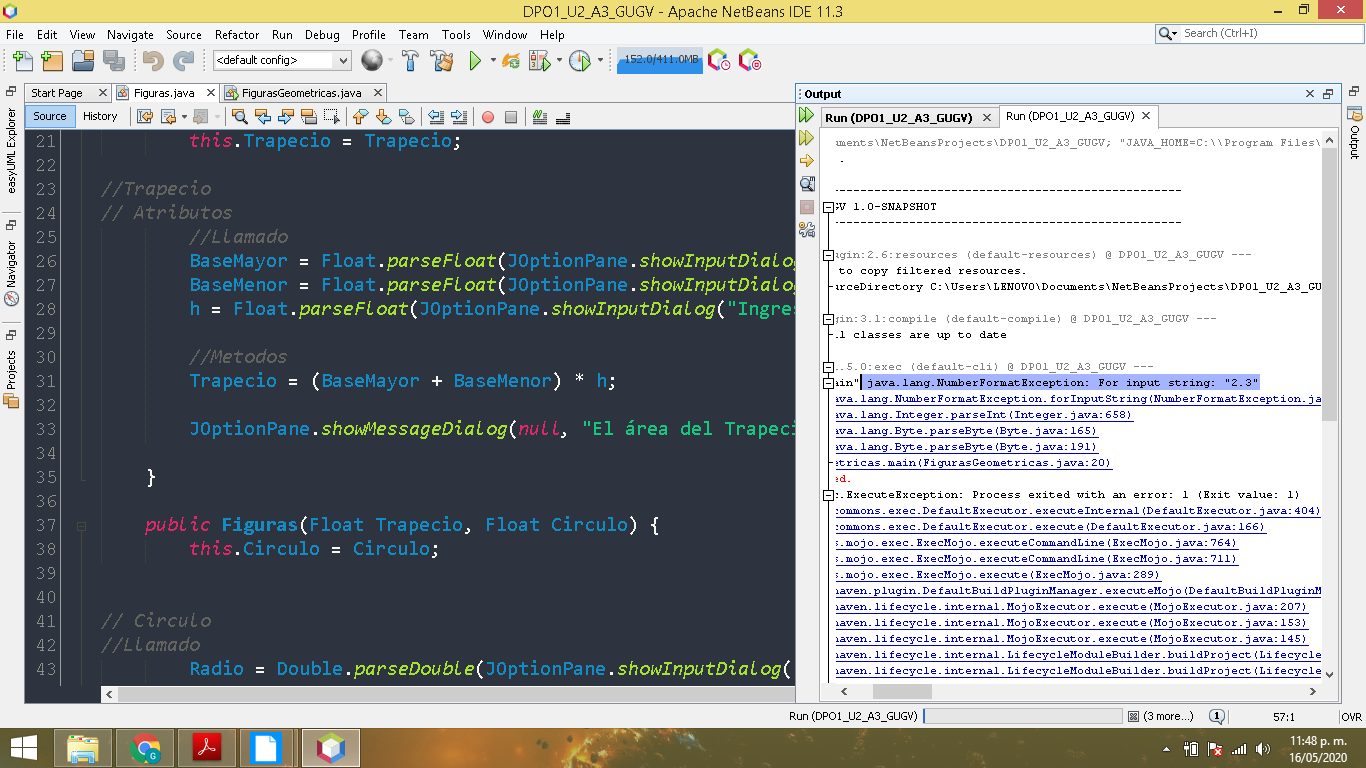


Ilustración 7 Exception java.lang.NumberFormatException: For input string: "2.3"

Exception- java.lang.NumberFormatException: For input string: "Si"

Exception java.lang.NumberFormatException: For input string: "2.3"

TIPO: RUN TIMEEXCEPTION

Vamos a comenzar, coloque el Try, desde el inicio de la pregunta al usuario, ya que de esta de detona el error. Además que quiero que el programa imprima “Error” y la excepción que esta surgiendo.

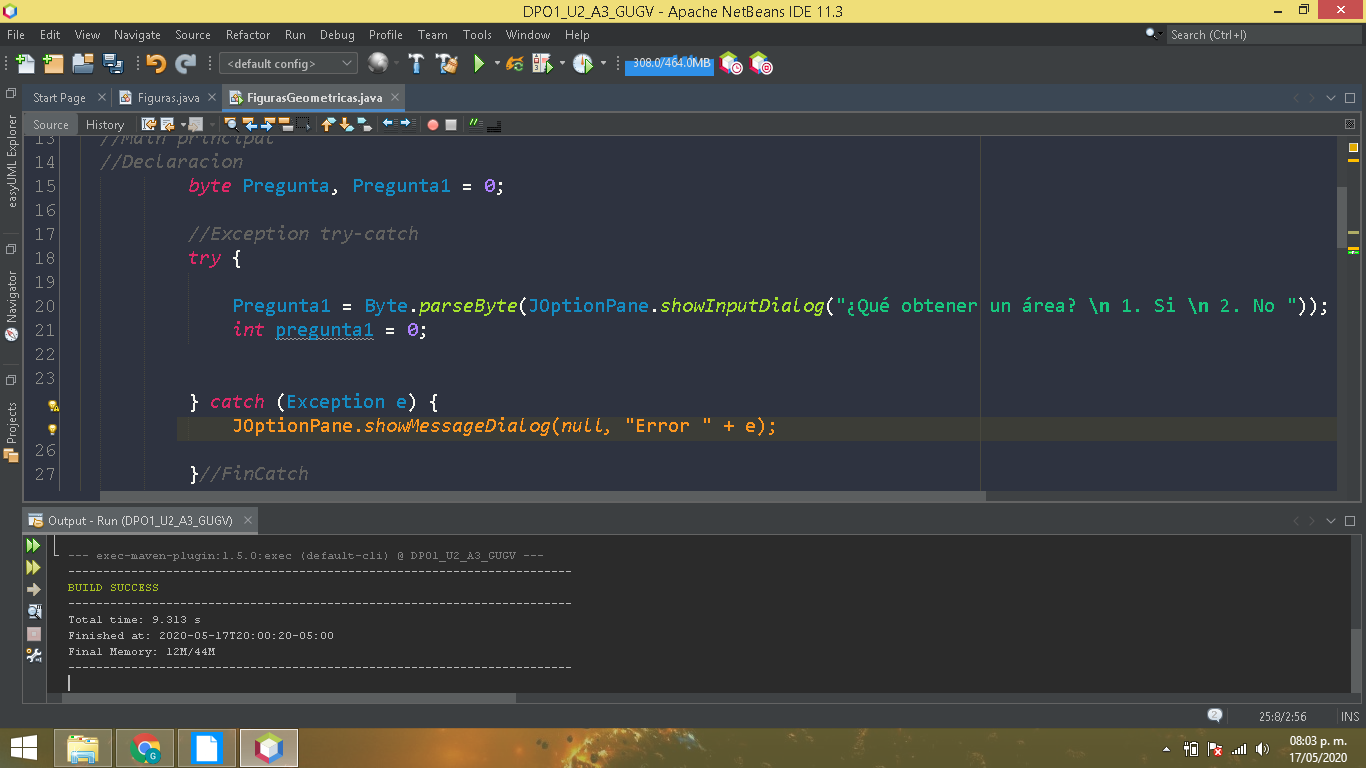
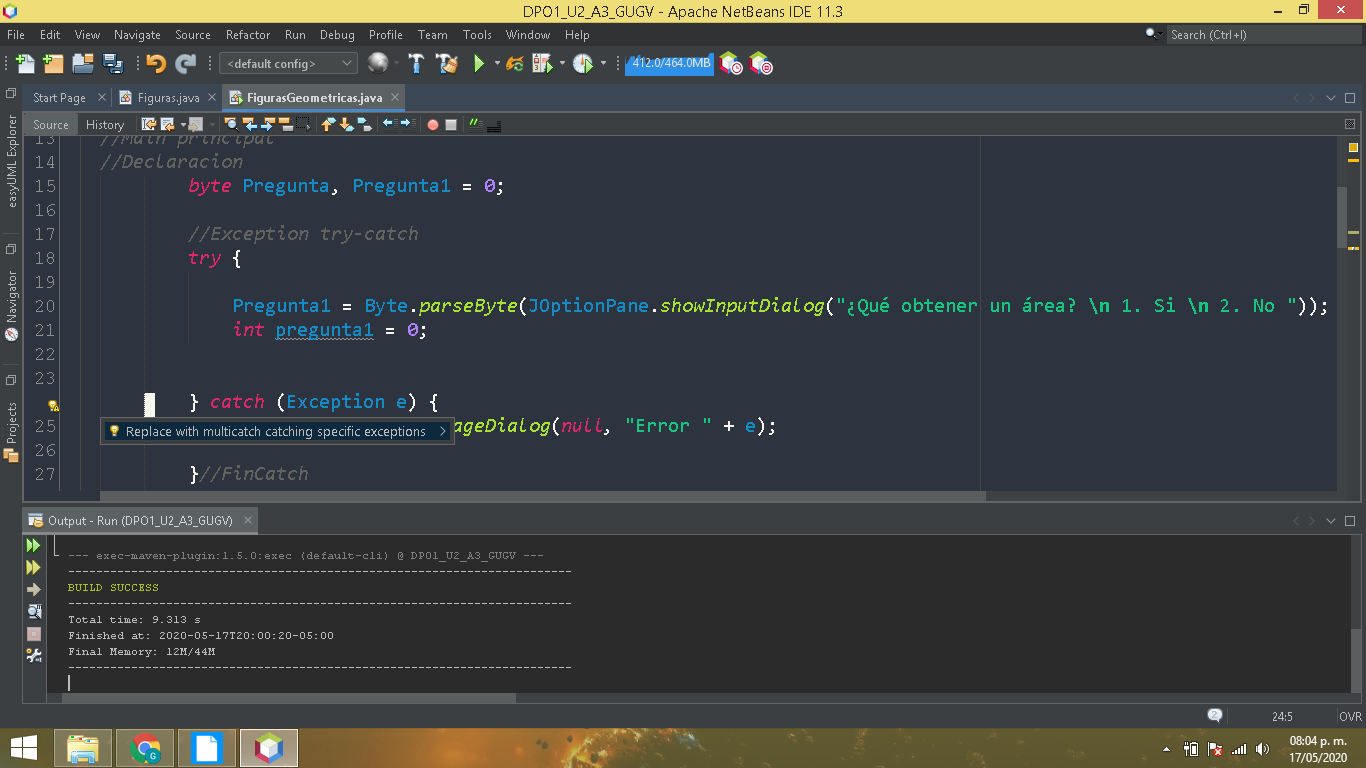


Ilustración 8 Exception 2

Pero el programa sigue presentando un error, debemos especificar la excepción



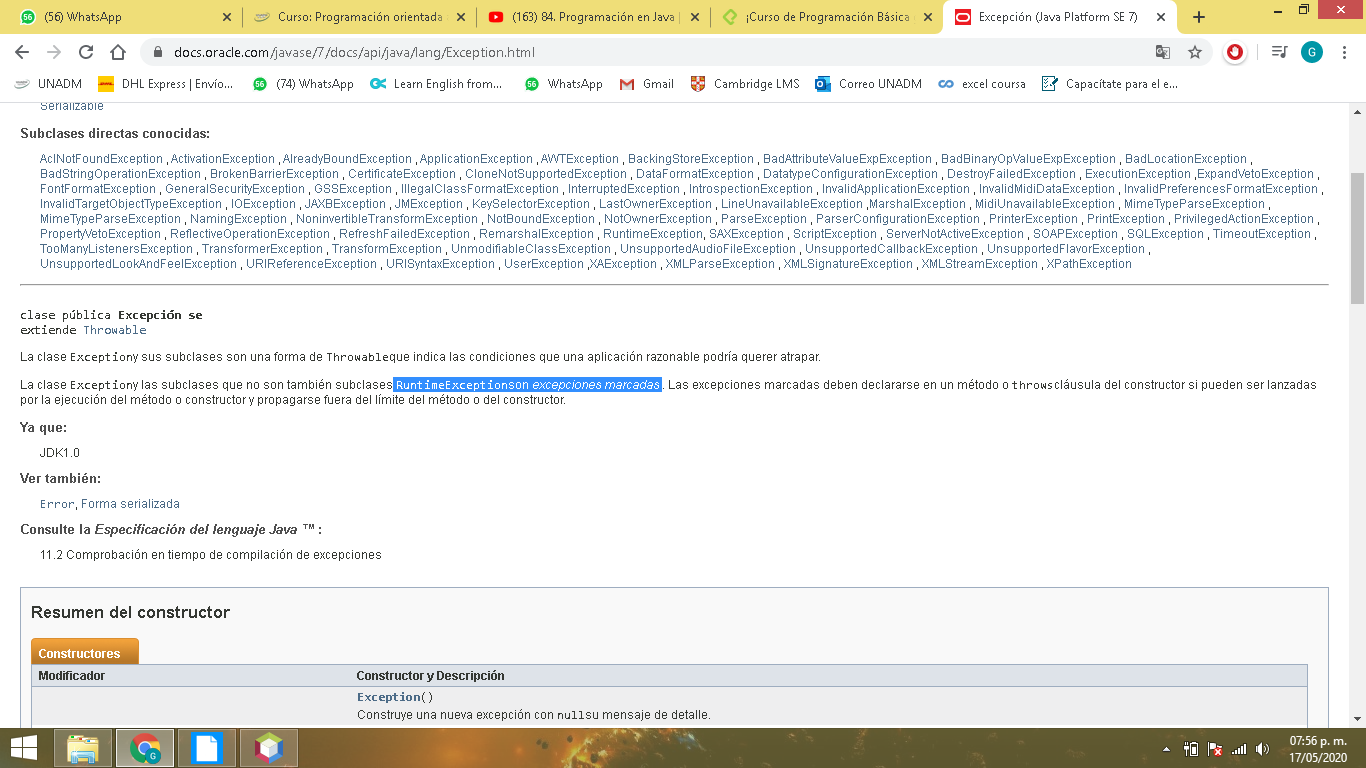
Vamos directamente a la página de Java para ver esas excepciones, java.lang.

Este tipo de excepciones son de tipo  [RuntimeException](https://docs.oracle.com/javase/7/docs/api/java/lang/RuntimeException.html) son excepciones marcadas.

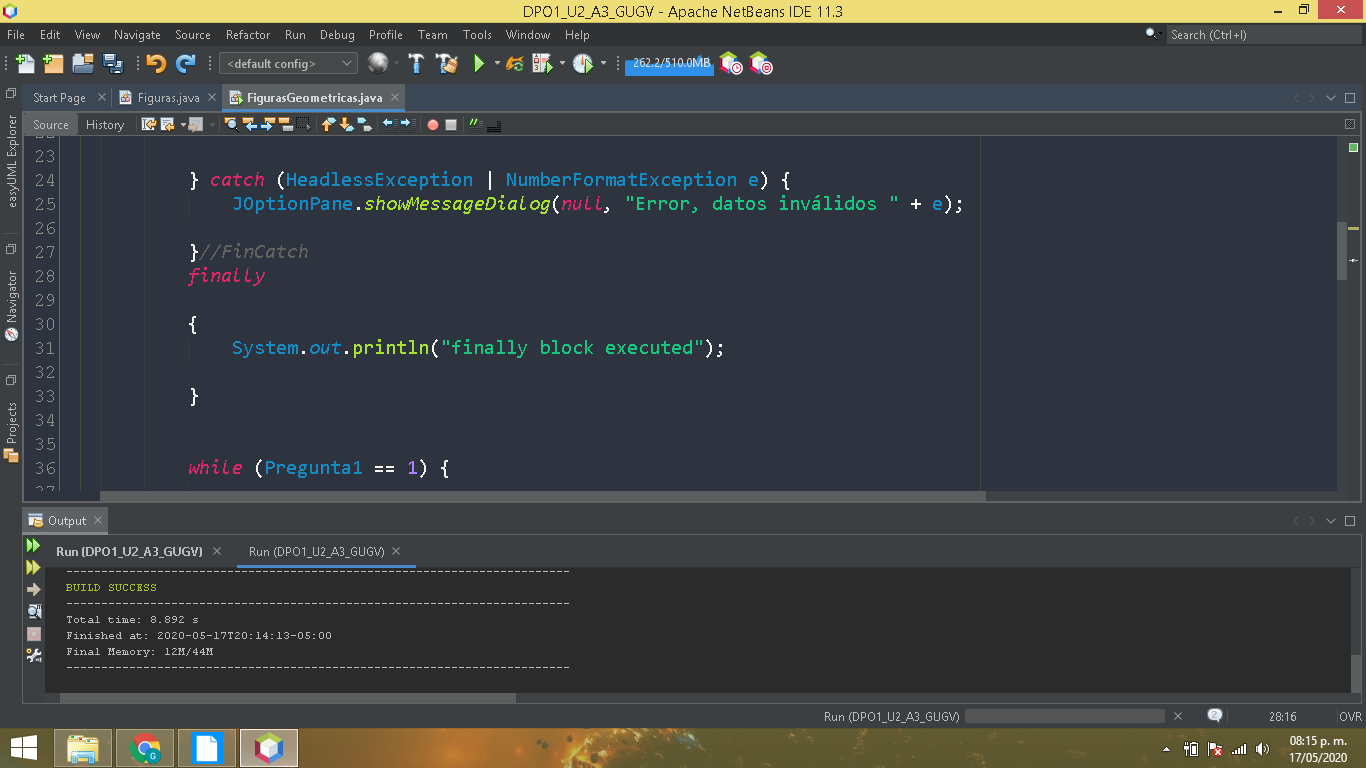
Ocupare las siguientes excepciones:

**HeadlessException**; Excepción que se lanza cuando un código dependiente de teclado, monitor o ratón es ejecutado en un entorno que no soporta teclado, monitor o ratón

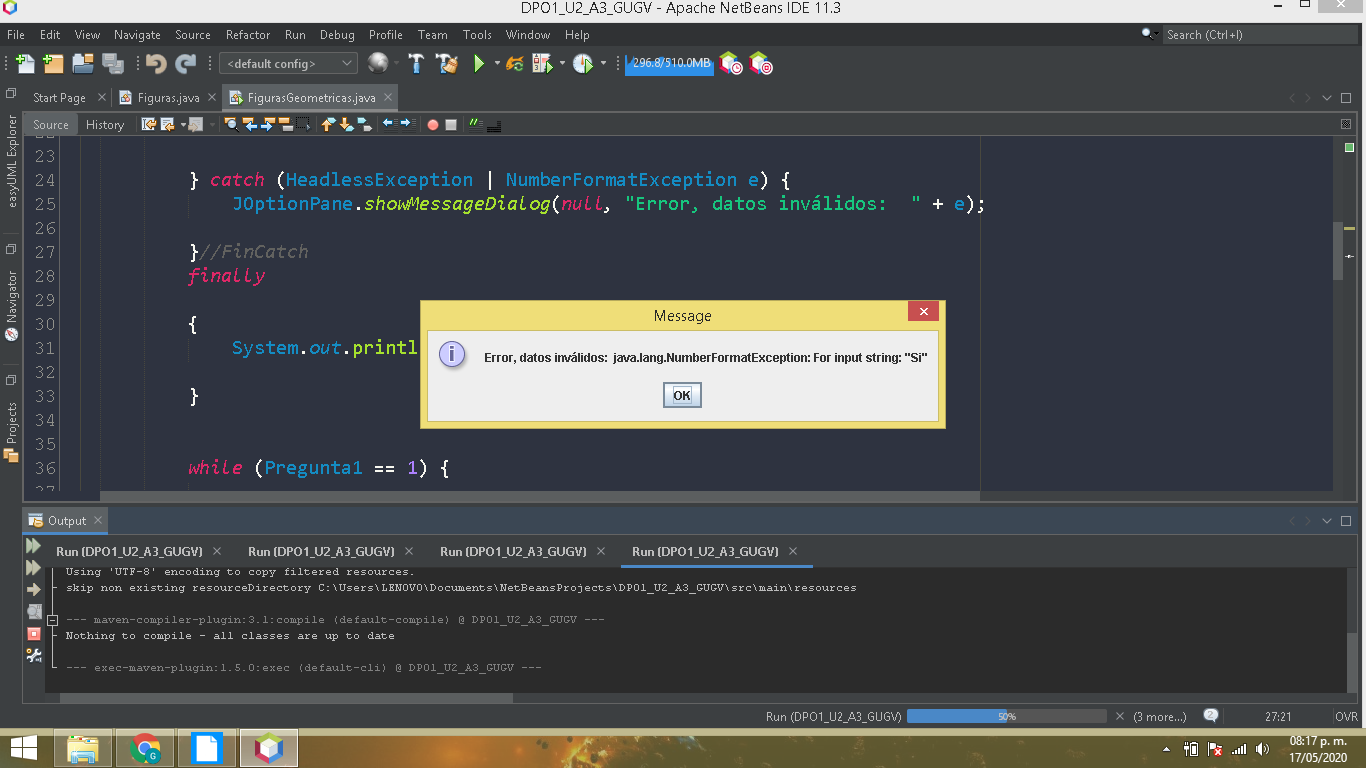
**NumberFormatException:** Se arroja para indicar que la aplicación ha intentado convertir una cadena a uno de los tipos numéricos, pero que la cadena no tiene el formato apropiado.



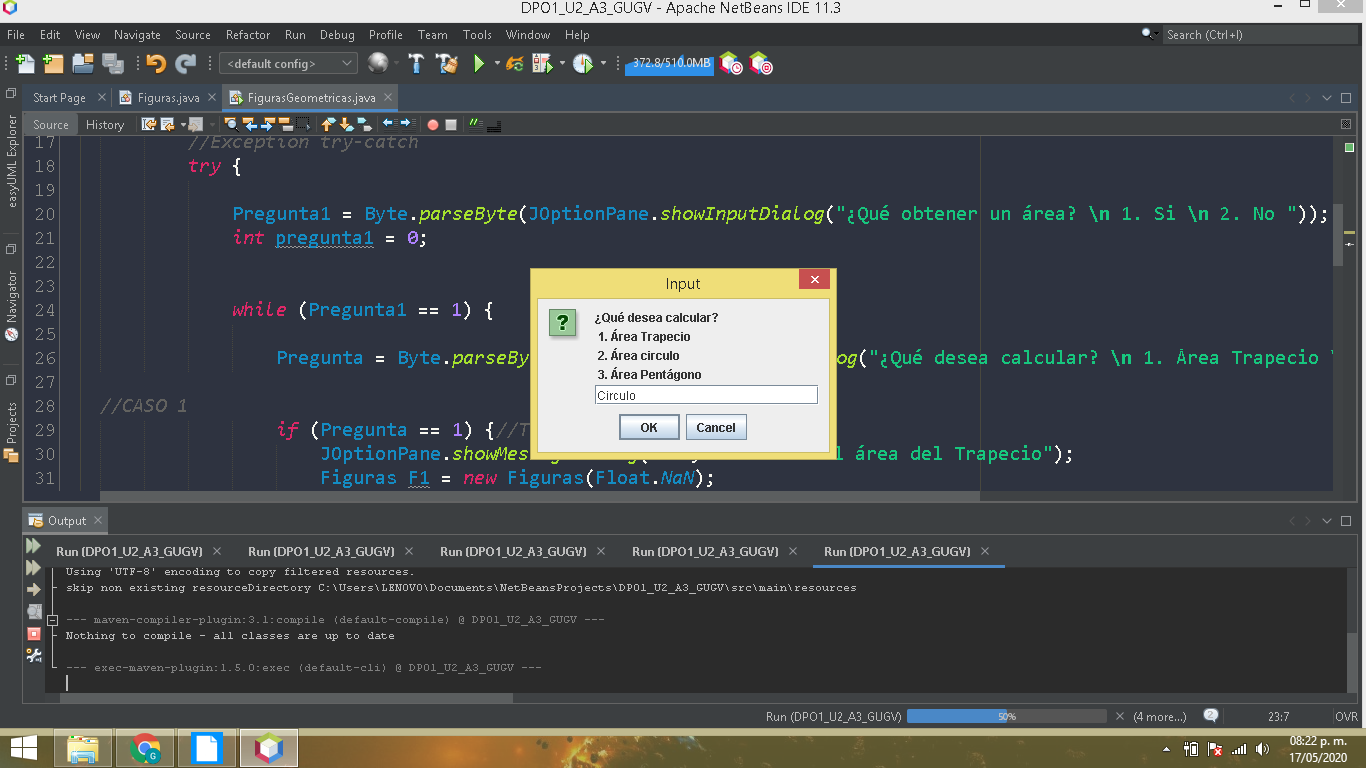
Las declaro en el programa.

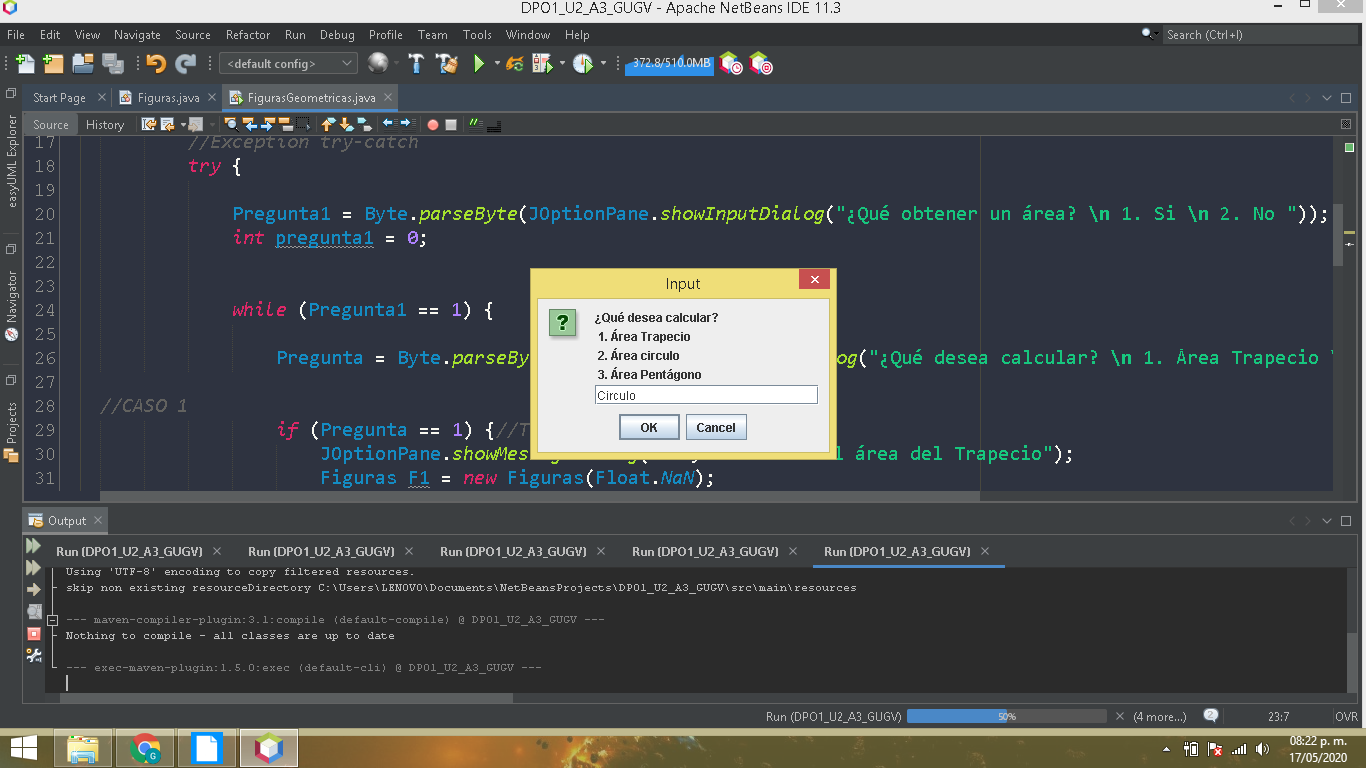


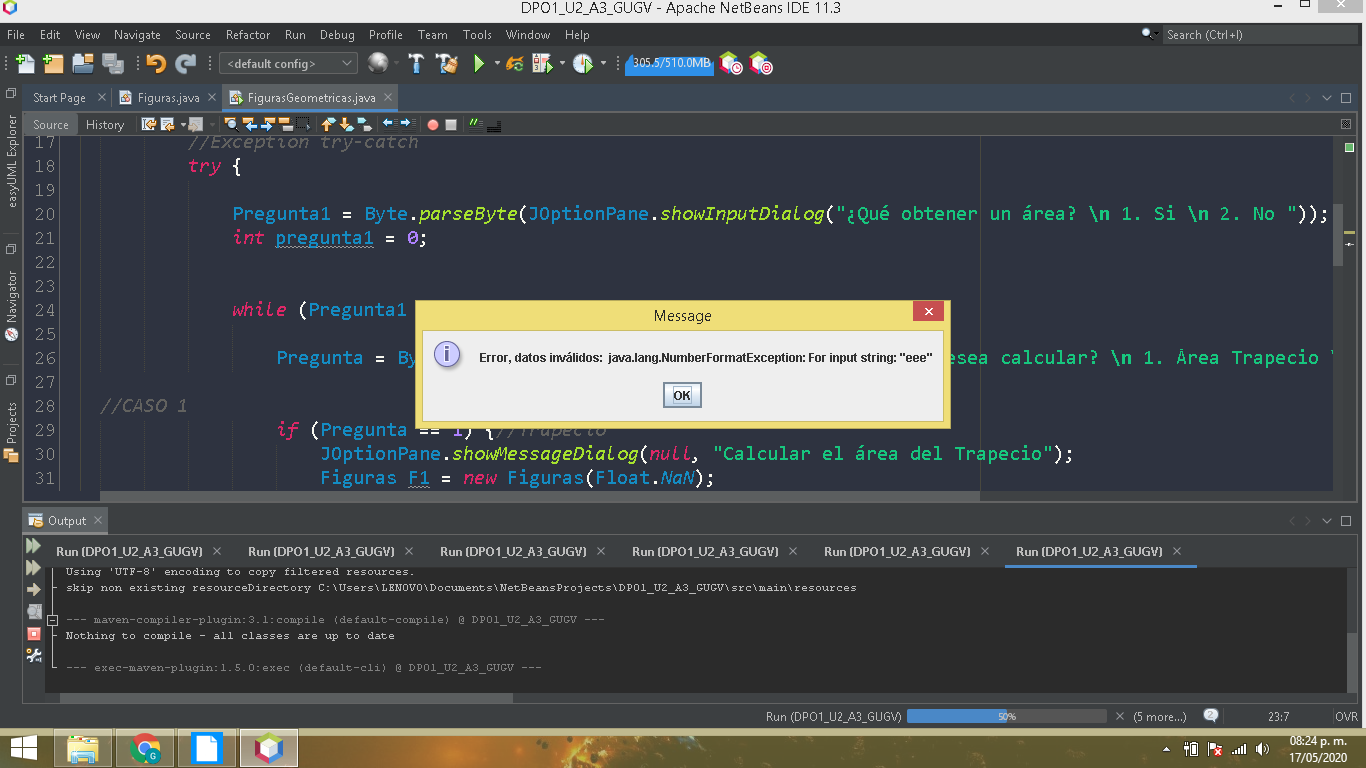
Ahora Correremos el programa, con los errores anteriores. Y Java nos muestra cómo funciona el Try-Catch



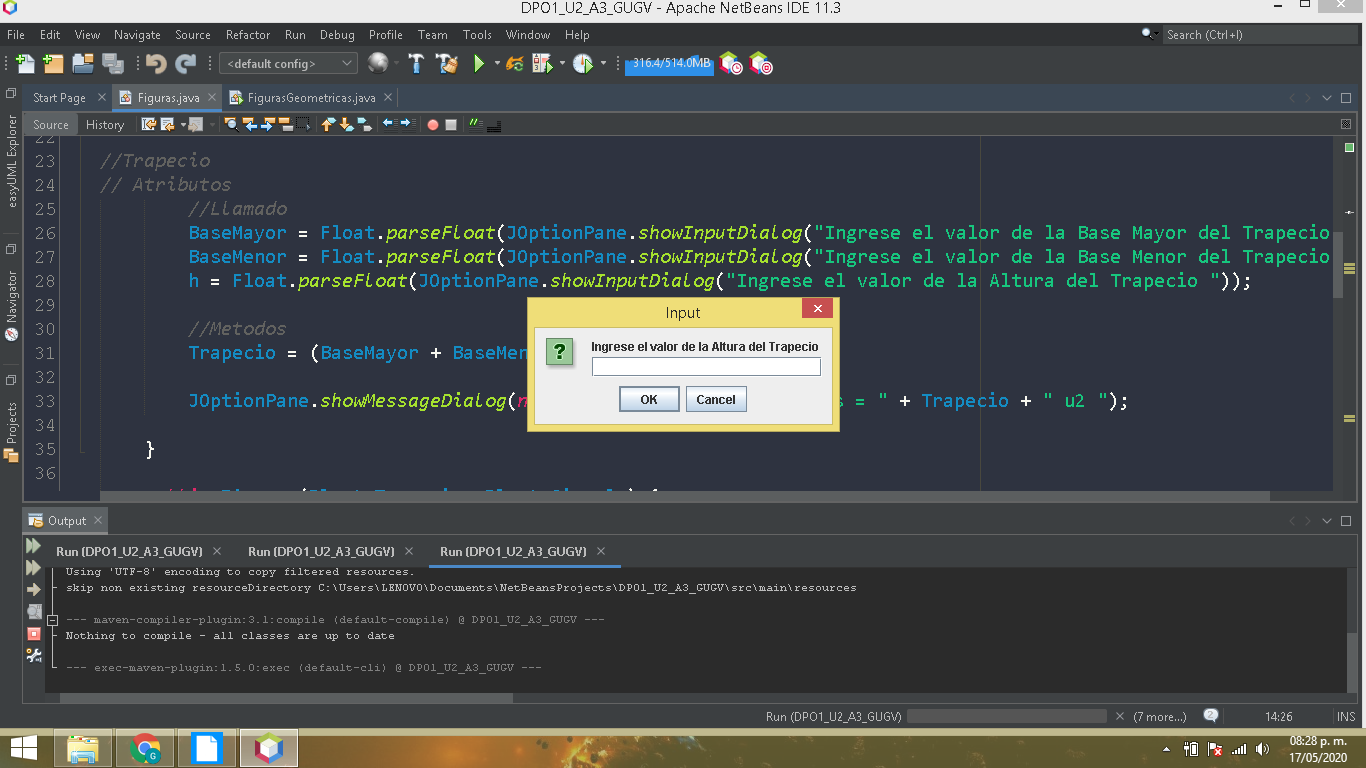
Pero me di cuenta que tenía más procesos con este tipo de errores, por lo que amplié la aplicación.





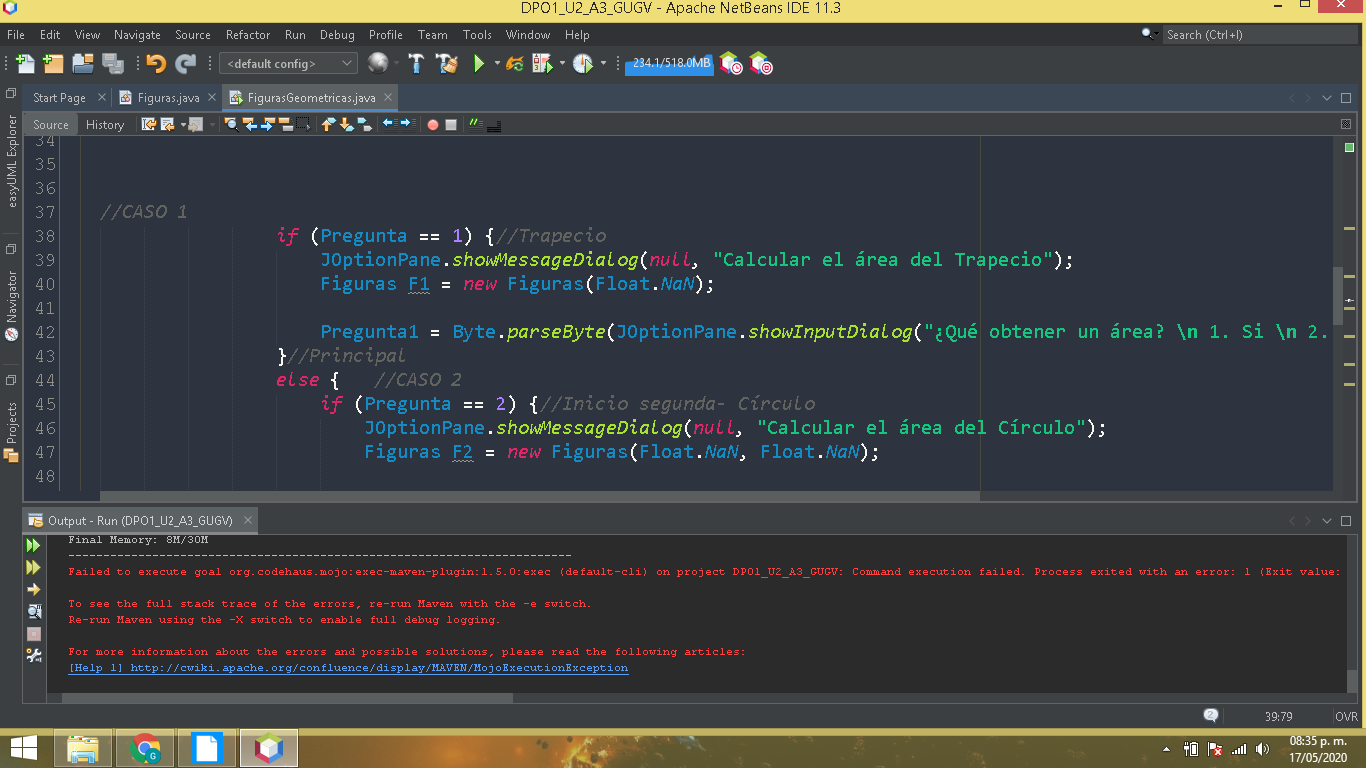


Otro error que me puedo encontrar es al pedir que el usuario ingrese los datos este, no lo haga

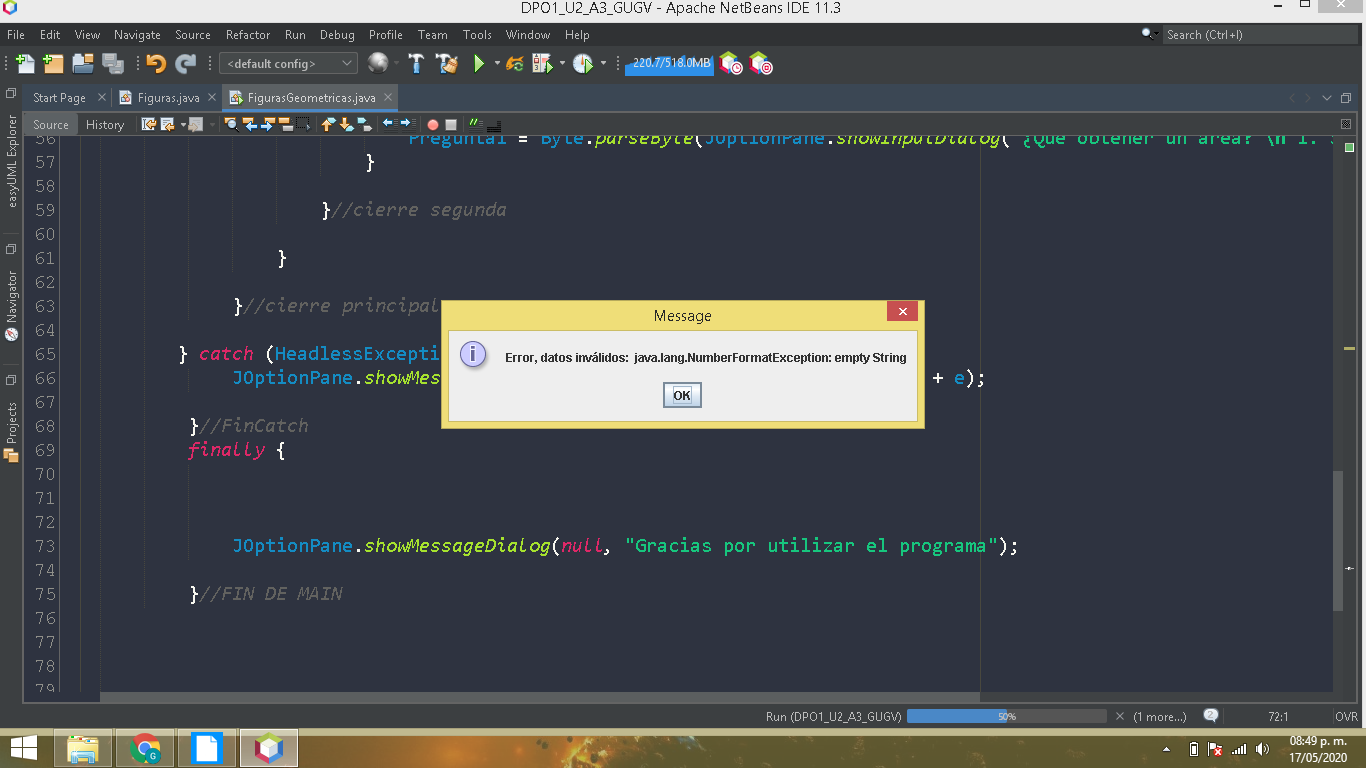


El programa entonces nos presenta un error.

Exception in thread "main" java.lang.NumberFormatException: empty String



De igual forma este vamos a utilizar Try-Catch.



Pero en este ejercicio me di cuenta que como es la misma excepción que los pasados, extiendo las excepciones a los largo del main. De esta manera se aplicará a todo el programa.

Las excepciones que detecte en el programa tienen el mismo tipo de [RuntimeException](https://docs.oracle.com/javase/7/docs/api/java/lang/RuntimeException.html).

Excepciones personalizadas

Para limitar la edad de los usuarios. Para comenzar creamos una clase que sea extensión de la clase Padre Exception, para heredar sus atributos y definimos un constructor.

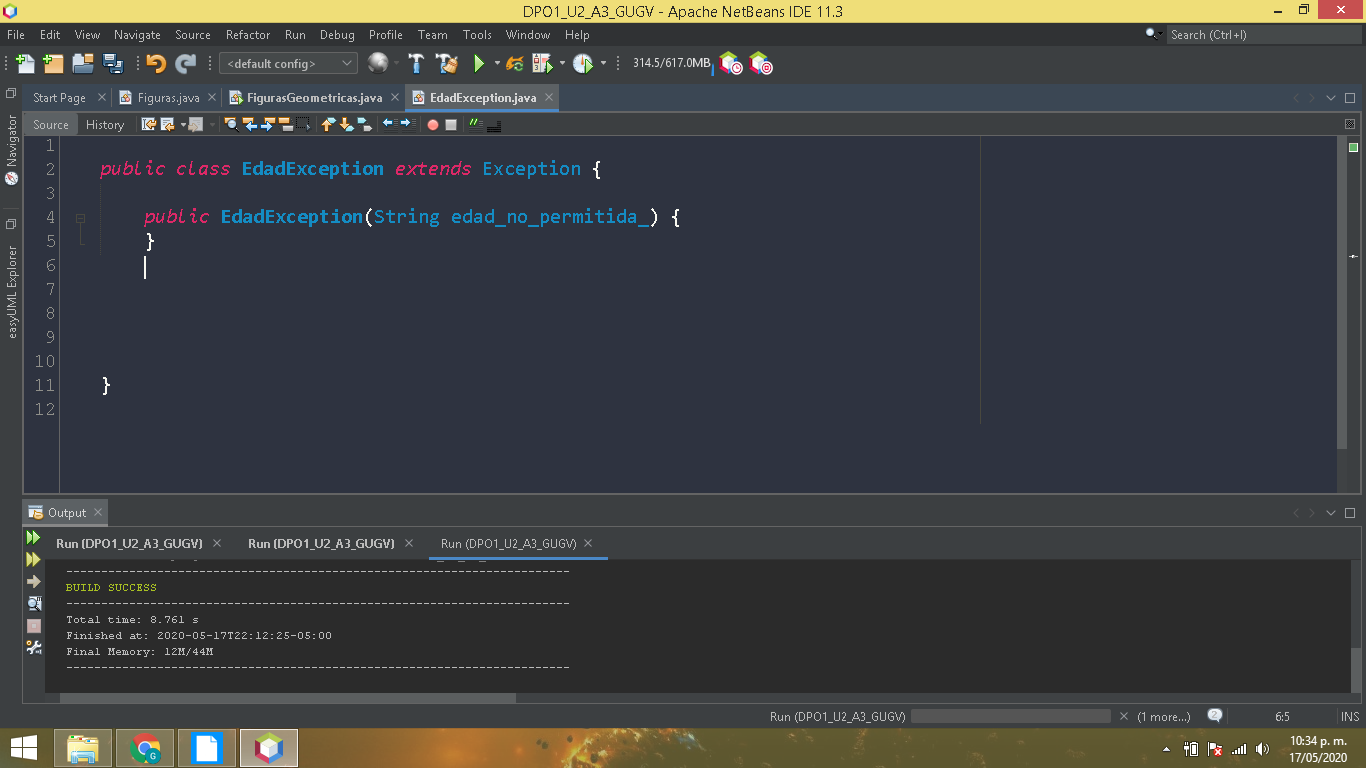
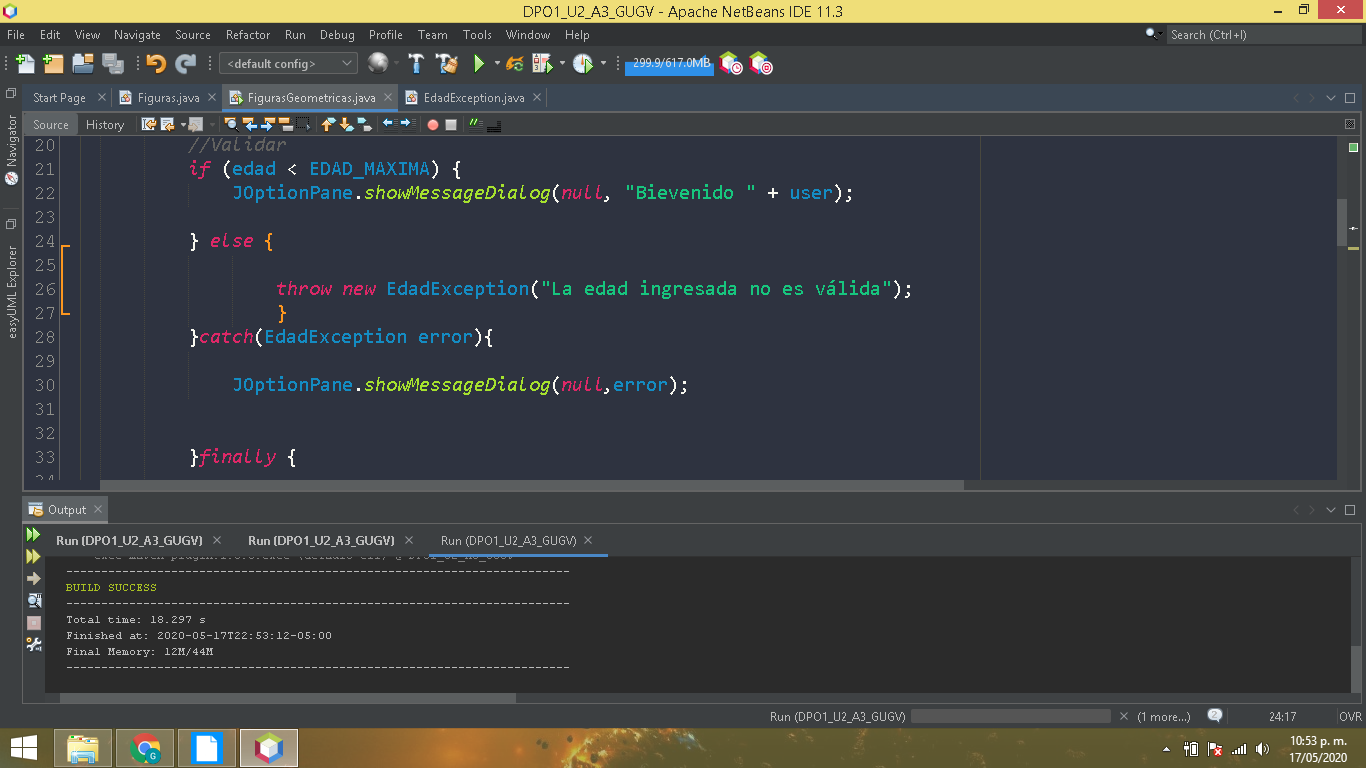
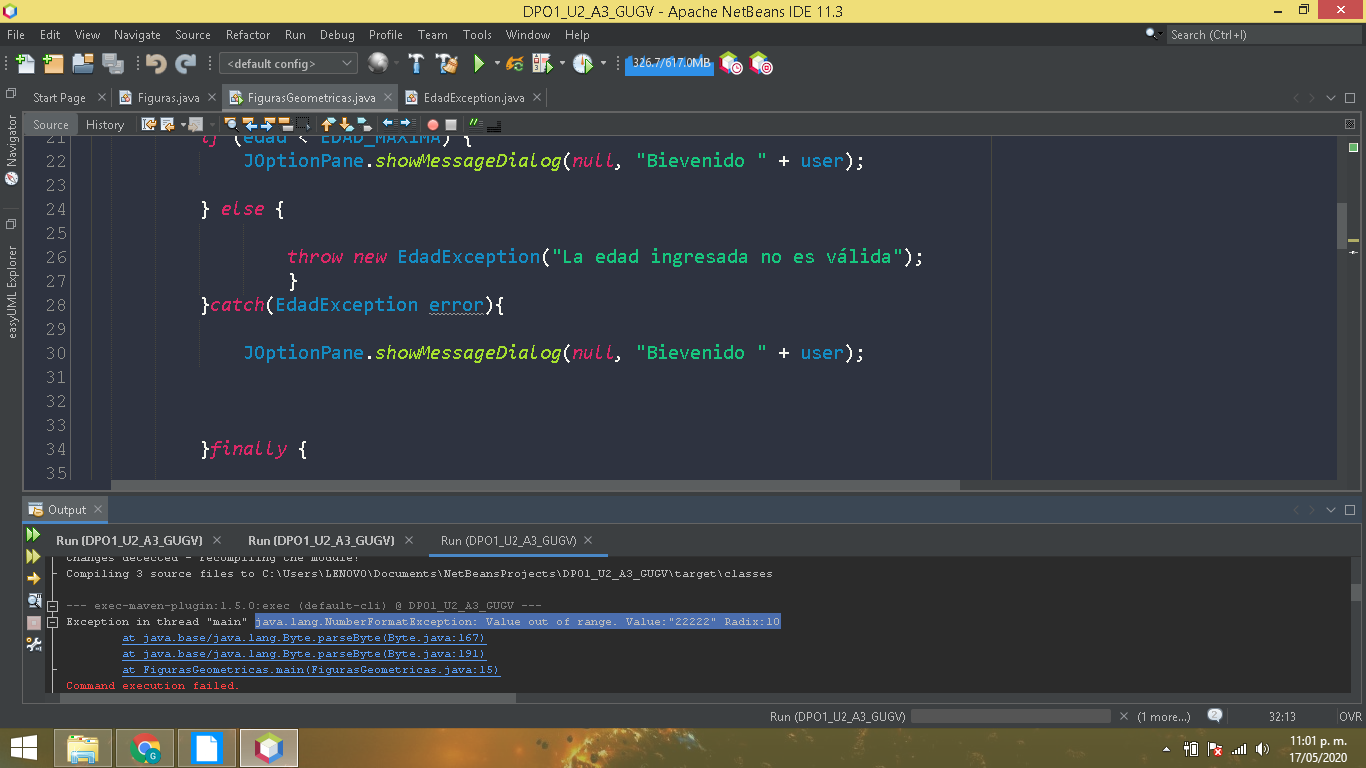


Ilustración 9 Exception 3

Posteriormente en el Main principal hacemos llamado de la Excepcion.



Me detecta el error.



#### Conclusiones

En esta actividad hicimos uso de las Exceptions, que son los errores que puede tener el programa que estemos diseñando. Existe una jerarquía entre los exceptions clasificándose en: Runtime Exception (generados por el lenguaje Java, al programar no se codificó con la sintaxis correcta, y no se previeron estos posibles errores al crear el programa) y los IOException (no generados por el lenguaje, sino incluidos por el programador, como al tratar de leer un archivo o tratar de hacer conexión a una base de datos específica).

Así que aplicando los Exception me di cuenta que en mis programas pasados, presentaba este tipo de errores, en su mayoría de la categoría RuntimeExceptions, ya que no previne los posibles errores al codificar. Exception es una herramienta que me permitirá identificar los problemas futuros y mejorar.

# Bibliografía

EXES - C/ Albasanz. (Consultado 16/05/2020). *Curso de introducción de Java*. Obtenido de Excepciones: Obtenido de : https://mundojava.net/excepciones.html?Pg=java\_inicial\_4\_6.html

Java™ Platform. (Consultado 17/05/20). *Java™ Platform, Standard Edition 7*. Obtenido de Consultado de :https://docs.oracle.com/javase/7/docs/api/

Programación ATS. (Consultado 14/05/20). *Curso de programación de Java*. Obtenido de Obtenido de: https://www.youtube.com/watch?v=2ZXiuh0rg3M&list=PLWtYZ2ejMVJkjOuTCzIk61j7XKfpIR74K

<https://www.youtube.com/watch?v=fDmuSDRSDLQ&t=196s>

<https://www.youtube.com/watch?v=5pMdEGfC2V8>

<https://www.youtube.com/watch?v=q9p1ZjpIifo>

Universidad Abierta y a Distancia de México. (Consultado 14/05/20). *Unidad 3. Características de POO y excepciones.* Obtenido de Obtenido de: file: /DPO1\_U3\_Contenido.pdf