**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

BUAP

** FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**

**PROGRAMACIÓN 2**

**“PRACTICA 12B. REPORTE PROGRAMA SISTEMA DE ECUACIONES”**

**PROFESOR:**

**DRA. CARMEN CERÓN GARNICA.**

**EQUIPO:  
BRENDA LIZETH COCA GARCIA**

**MATRICULA:201917154**

**BRENDA AILED RODRIGUEZ COLIS**

**MATRICULA:201910848**

**MIGUEL CARREON VAZQUEZ**

**MATRICULA: 201915389**

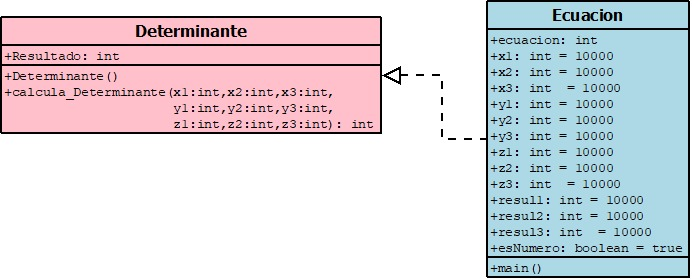
**IRVYN XICALE CARRERA**

**MATRICULA: 201963582**

**INTRODUCCIÓN**Una Excepción es una ocurrencia indeseable que puede ser detectada durante la ejecución de un programa. Por ejemplo, una división entre cero, leer datos no validos o de diferentes tipos, tratar de abrir un archivo que no existe, también acceder a un elemento de un arreglo que no existe, es una excepción. Para comprender mejor el término, una excepción es: una condición inesperada, una condición que altera el flujo de ejecución de un programa.

En esta practica realizamos un programa que lee un sistema de ecuaciones de 3 incógnitas y nos muestra el valor de cada incógnita, agregando diferentes excepciones.

**PROGRAMA SISTEMA DE ECUACIONES  
DIAGRAMA UML**



**CLASE ECUACION**

//Fecha:3-Mayo-2021

//Objetivo:Ecuacion sistema de tres incognitas

//Autor:Brenda Coca Garcia, Miguel Carreon, Irvyn Xicale, Brenda Rodriguez Colis

import java.io.\*;

public class Ecuacion {

public static void main(String[] args) {

int[][] ecuacion = new int[3][3];

int x1 = 10000, x2 = 10000, x3 = 10000, y1 = 10000, y2 = 10000, y3 = 10000, z1 = 10000, z2 = 10000, z3 = 10000, resul1 = 10000, resul2 = 10000, resul3 = 10000;

boolean esNumero = true;

try {

// lectura

InputStreamReader entrada = new InputStreamReader(System.in);

BufferedReader flujoEntrada = new BufferedReader(entrada);

String datoEntrada;

System.out.println("Ingresar variable 1:");

datoEntrada = flujoEntrada.readLine();

try {

x1 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (x1 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

// excepcion del String

System.out.println("Ingresar variable 2:");

datoEntrada = flujoEntrada.readLine();

try {

y1 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (y1 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 3:");

datoEntrada = flujoEntrada.readLine();

try {

z1 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (z1 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 1:");

datoEntrada = flujoEntrada.readLine();

try {

x2 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (x2 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 2:");

datoEntrada = flujoEntrada.readLine();

try {

y2 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (y2 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 3:");

datoEntrada = flujoEntrada.readLine();

try {

z2 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (z2 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 1:");

datoEntrada = flujoEntrada.readLine();

try {

x3 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (x3 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 2:");

datoEntrada = flujoEntrada.readLine();

try {

y3 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (y3 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar variable 3:");

datoEntrada = flujoEntrada.readLine();

try {

z3 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (z3 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar el resultado de la ecuacion 1: ");

datoEntrada = flujoEntrada.readLine();

try {

resul1 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (resul1 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar el resultado de la ecuacion 2: ");

datoEntrada = flujoEntrada.readLine();

try {

resul2 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (resul2 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

System.out.println("Ingresar el resultado de la ecuacion 3: ");

datoEntrada = flujoEntrada.readLine();

try {

resul3 = Integer.parseInt(datoEntrada);

} catch (Exception e) {

esNumero = false;

}

if (esNumero == false) {

throw new Exception("Ingreso letras, simbolos o decimales");

}

// excepcion de mayor

if (resul3 >= 100000) {

throw new Exception("Valores extremales para un coeficiente del Sistema");

} else {

System.out.println("dato guardado");

}

Determinante solucion = new Determinante();

int rd = solucion.calcula\_Determinante(x1, y1, z1, x2, y2, z2, x3, y3, z3);

int rdx = solucion.calcula\_Determinante(resul1, y1, z1, resul2, y2, z2, resul3, y3, z3);

int rdy = solucion.calcula\_Determinante(x1, resul1, z1, x2, resul2, z2, x3, resul3, z3);

int rdz = solucion.calcula\_Determinante(x1, y1, resul1, x2, y2, resul2, x3, y3, resul3);

System.out.println("Soluciones");

if (rd == 0) {

System.out.println("El sistema no tiene solucion");

} else {

System.out.println("El valor de x es igual a " + rdx / rd);

System.out.println("El valor de y es igual a " + rdy / rd);

System.out.println("El valor de z es igual a " + rdz / rd);

}

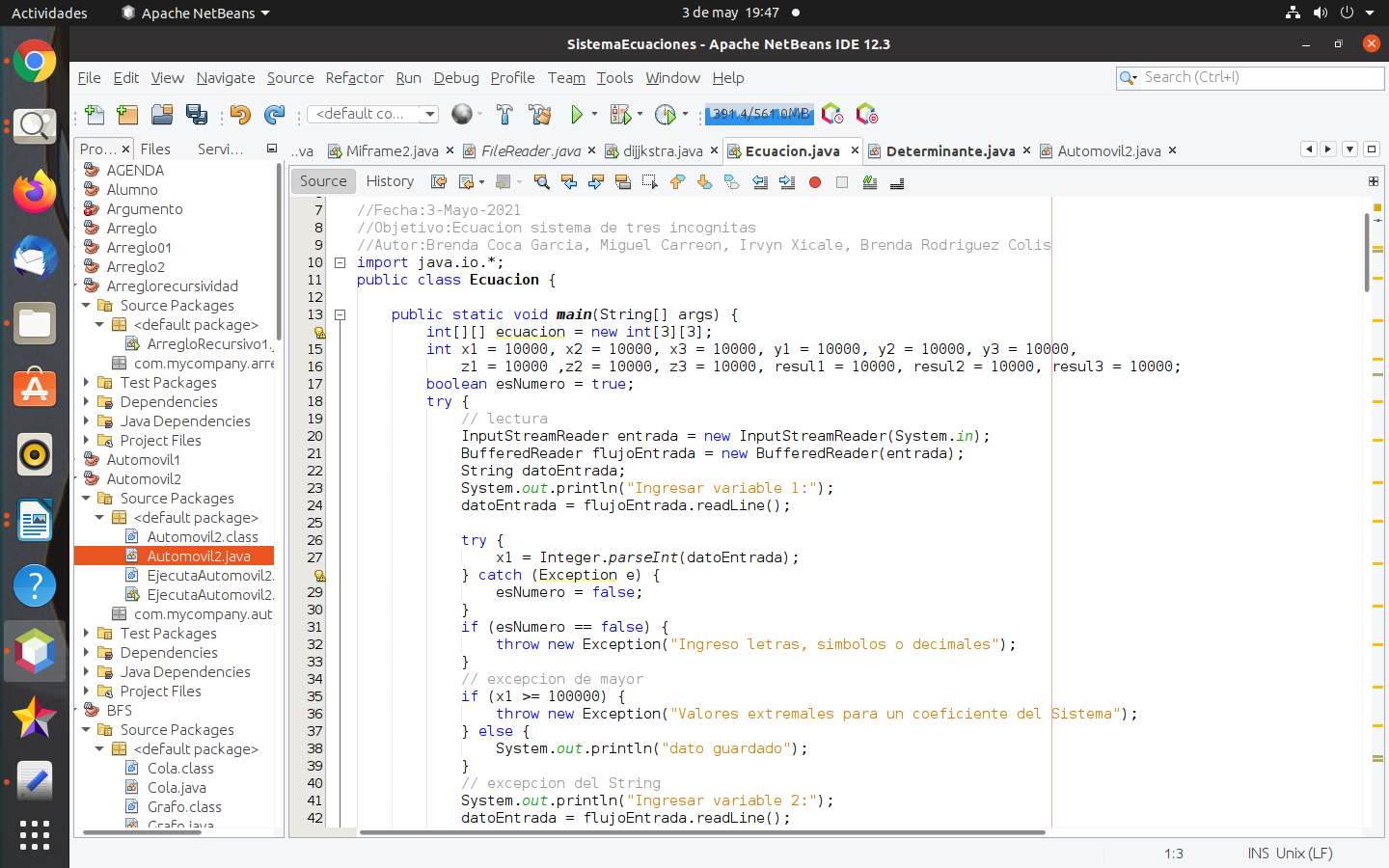
} catch (Exception e) {

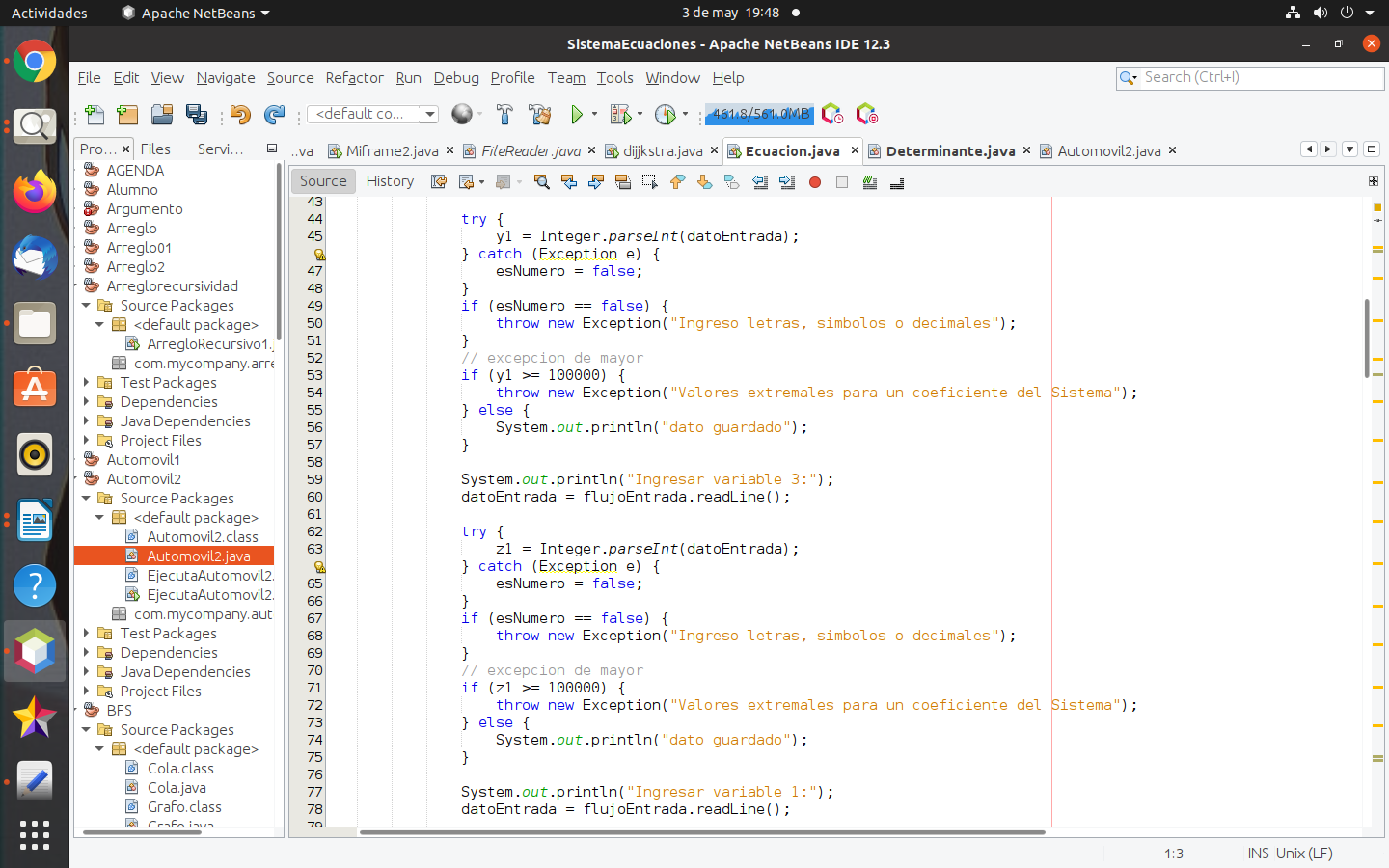
System.out.println("Error " + e.getMessage());

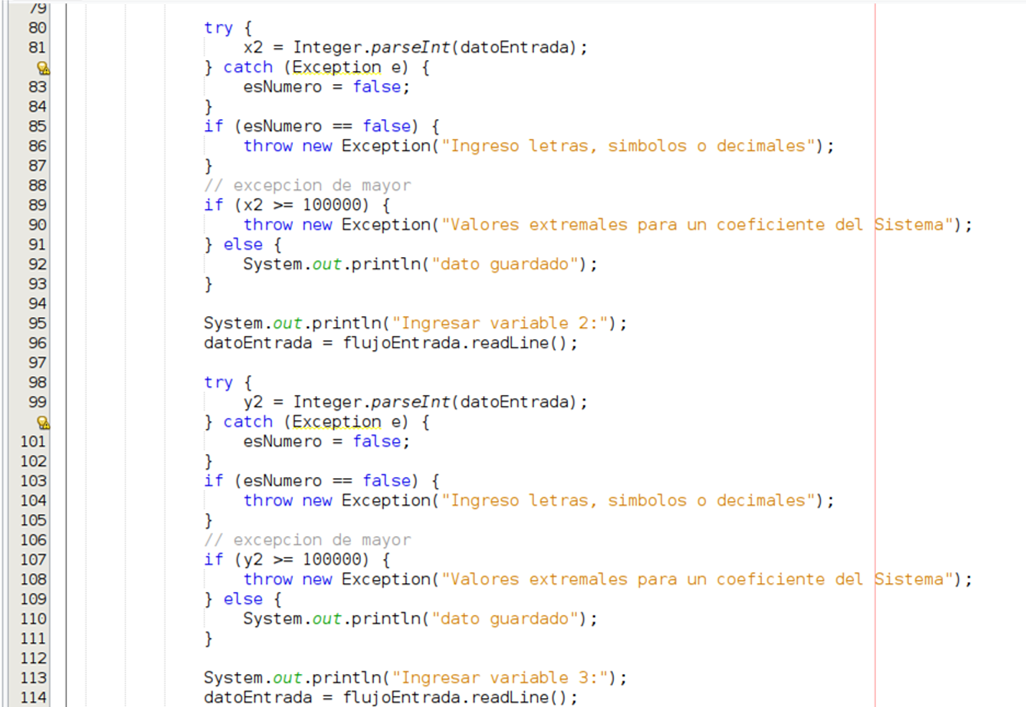
}

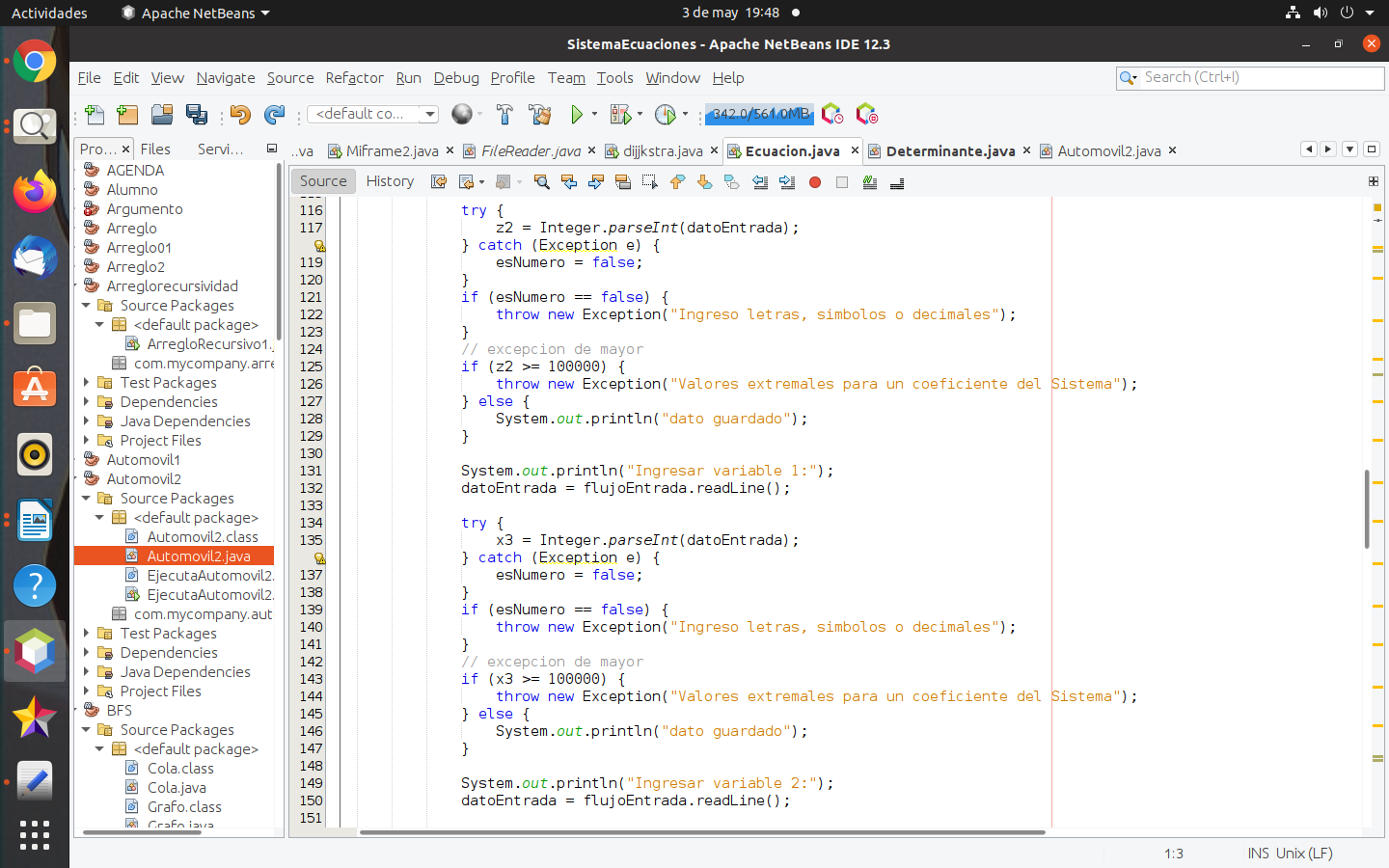
}

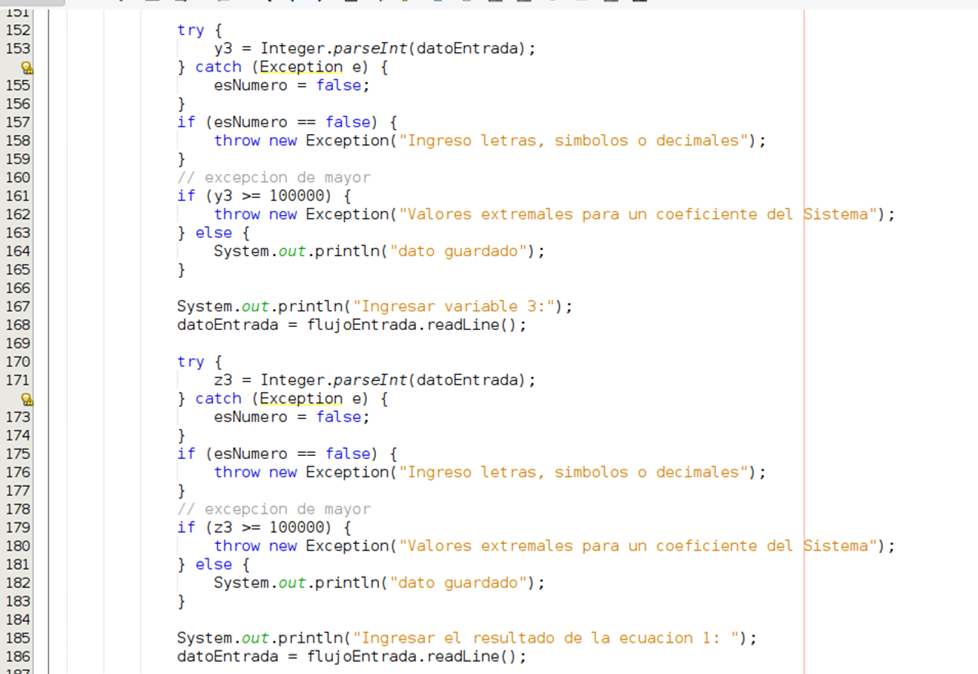
}

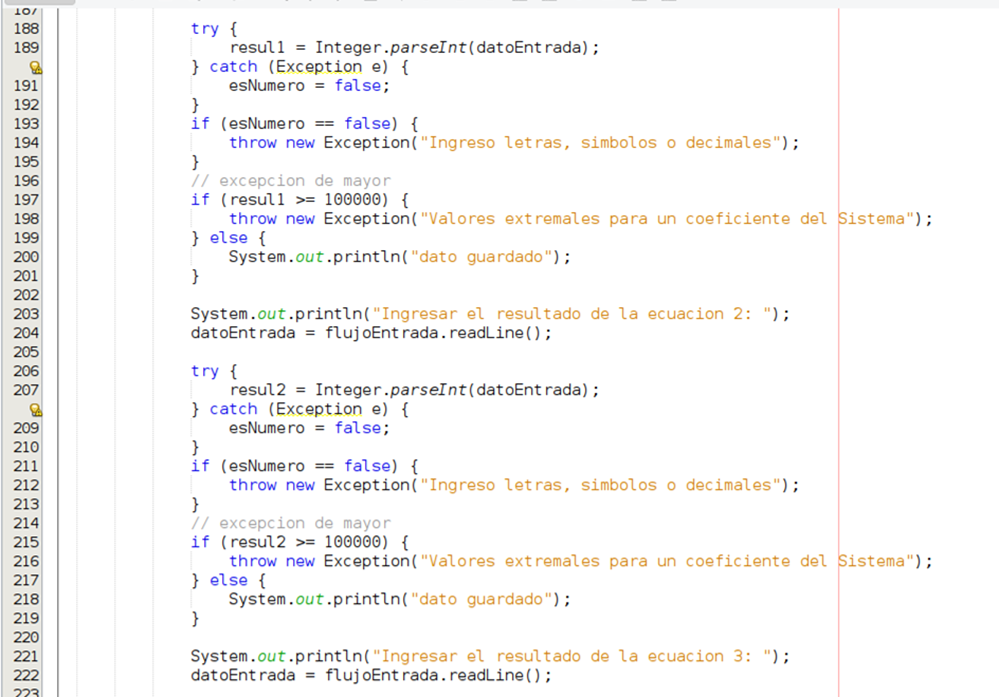


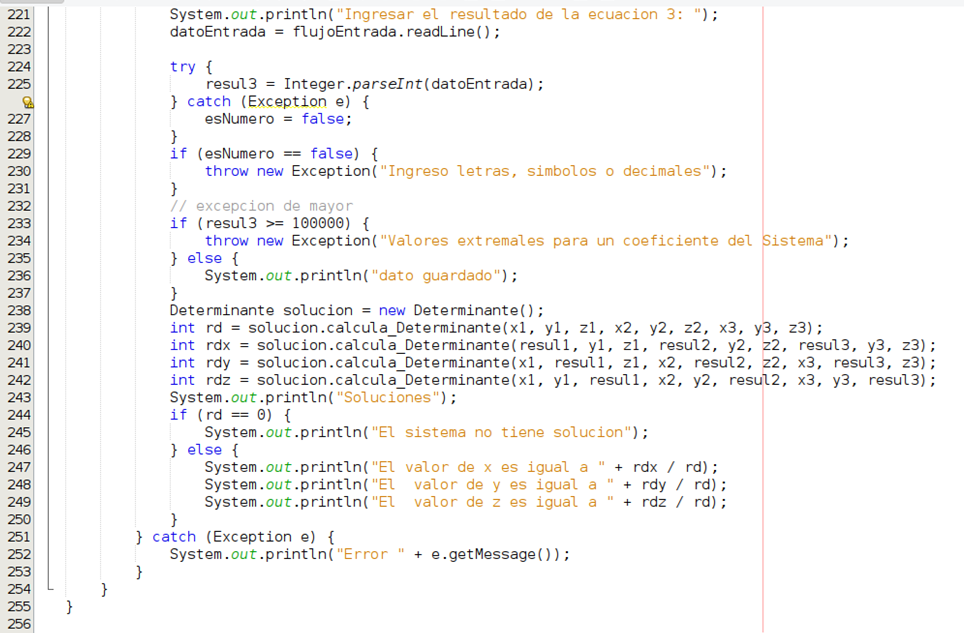


****



****

****

****

**CLASE DETERMINANTE**

//Fecha:3-Mayo-2021

//Objetivo:Ecuacion sistema de tres incognitas

//Autor:Brenda Coca Garcia, Miguel Carreon, Irvyn Xicale, Brenda Rodriguez Colis

public class Determinante {

public int calcula\_Determinante(int x1,int x2, int x3,int y1,int y2,int y3,int z1,int z2,int z3)

throws Exception{

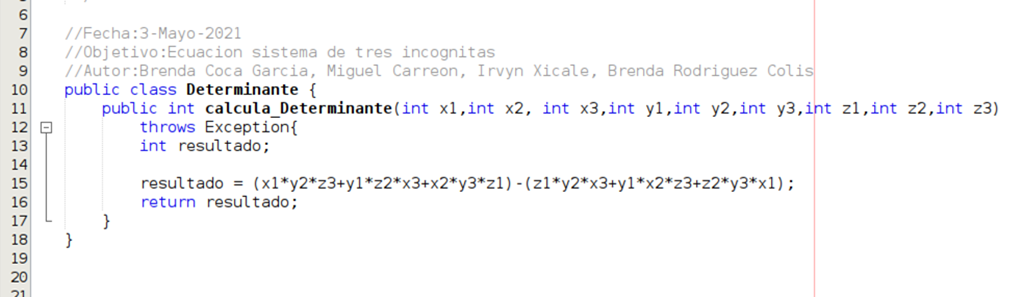
int resultado;

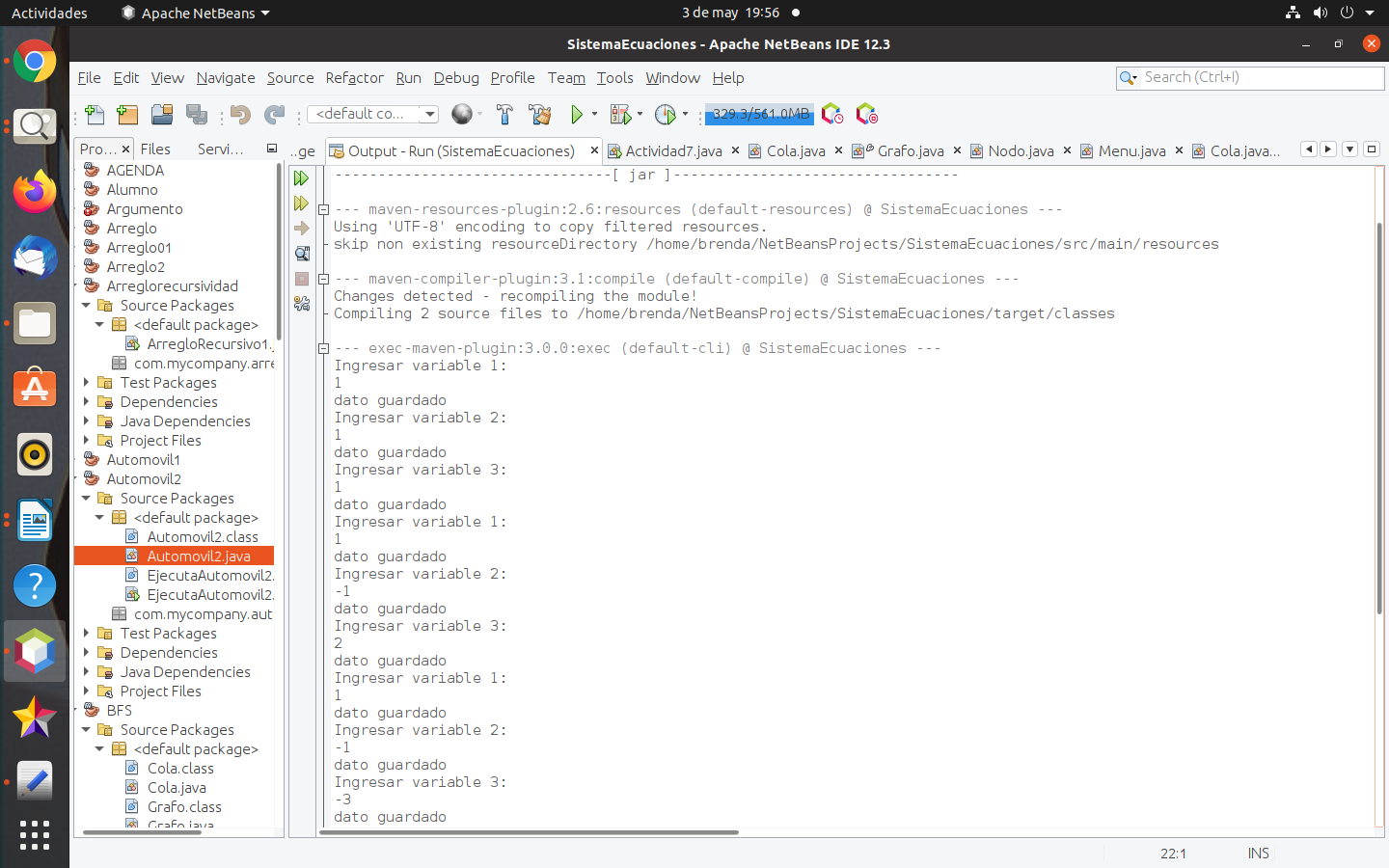
resultado = (x1\*y2\*z3+y1\*z2\*x3+x2\*y3\*z1)-(z1\*y2\*x3+y1\*x2\*z3+z2\*y3\*x1);

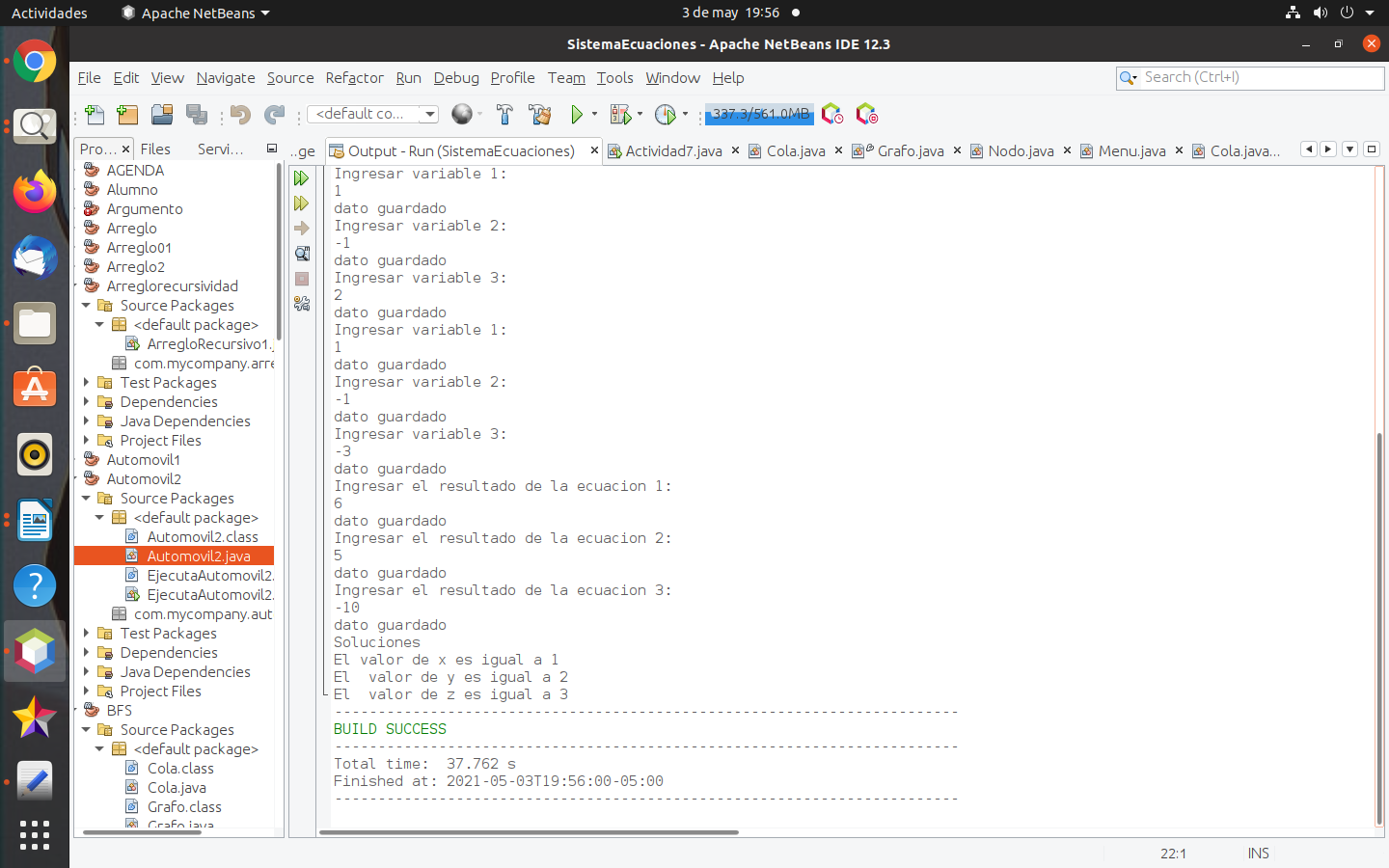
return resultado;

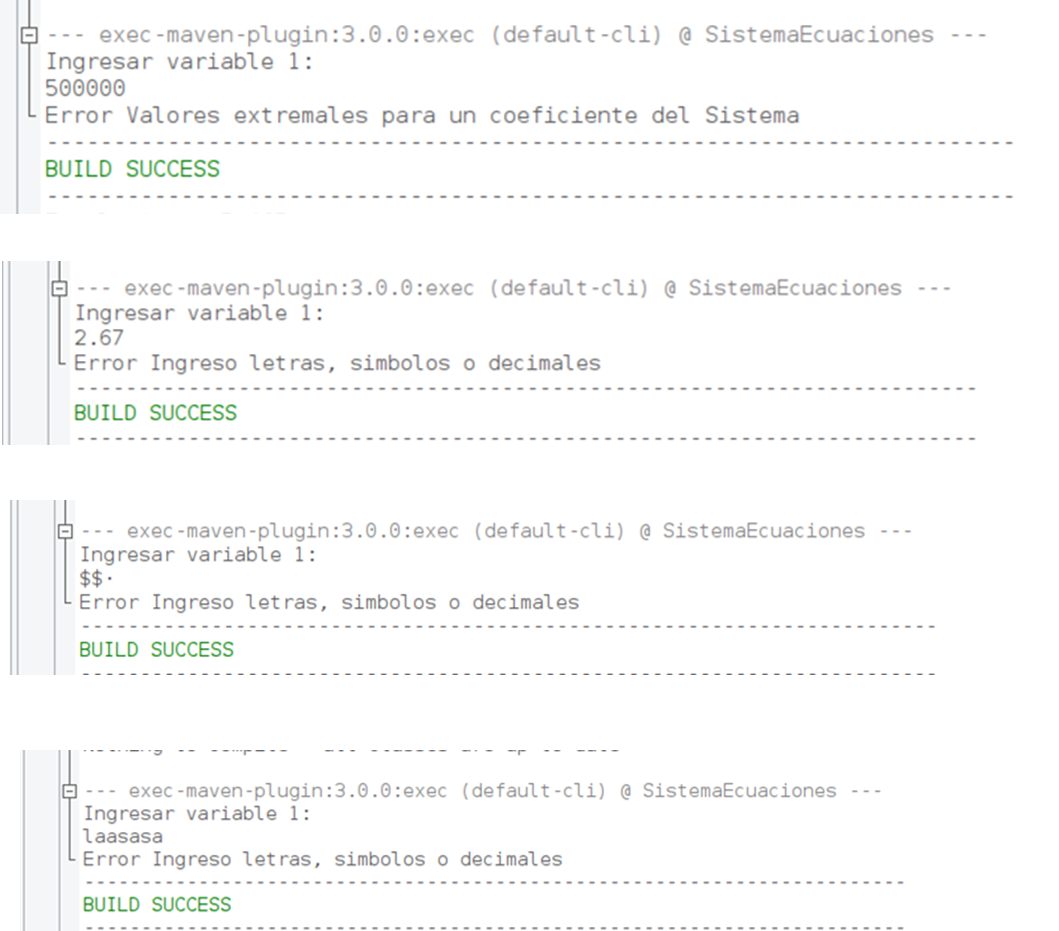
}

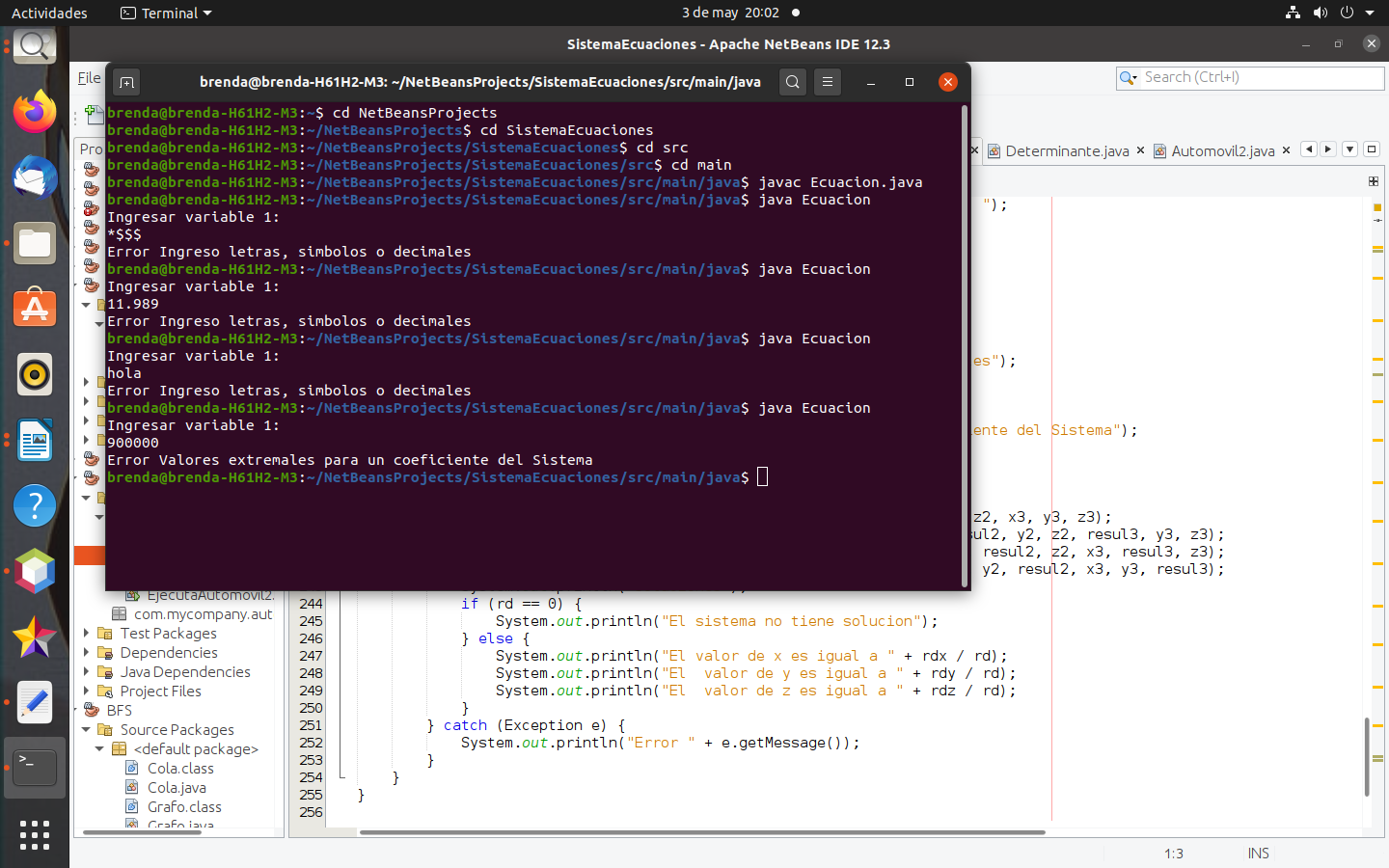
}

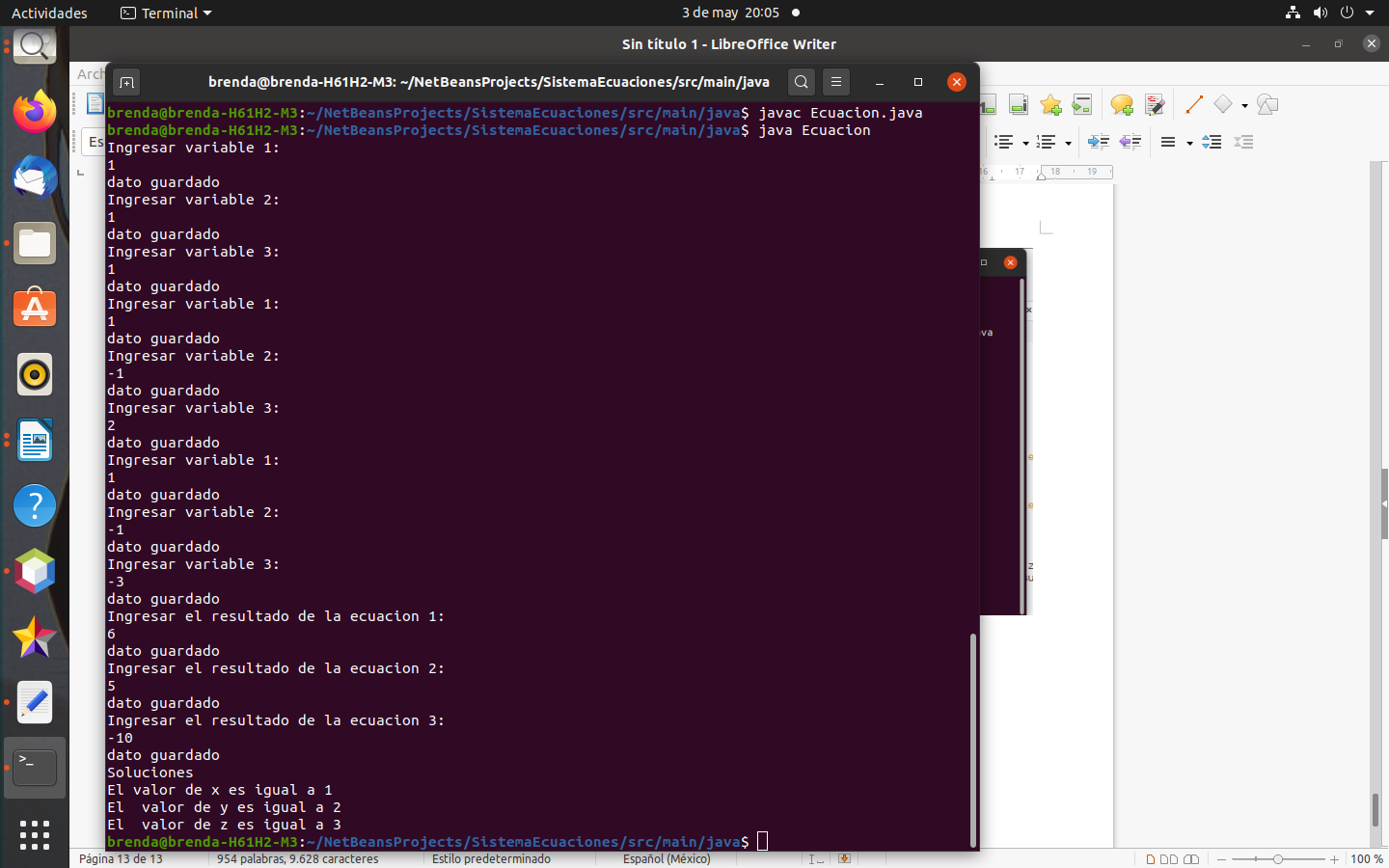


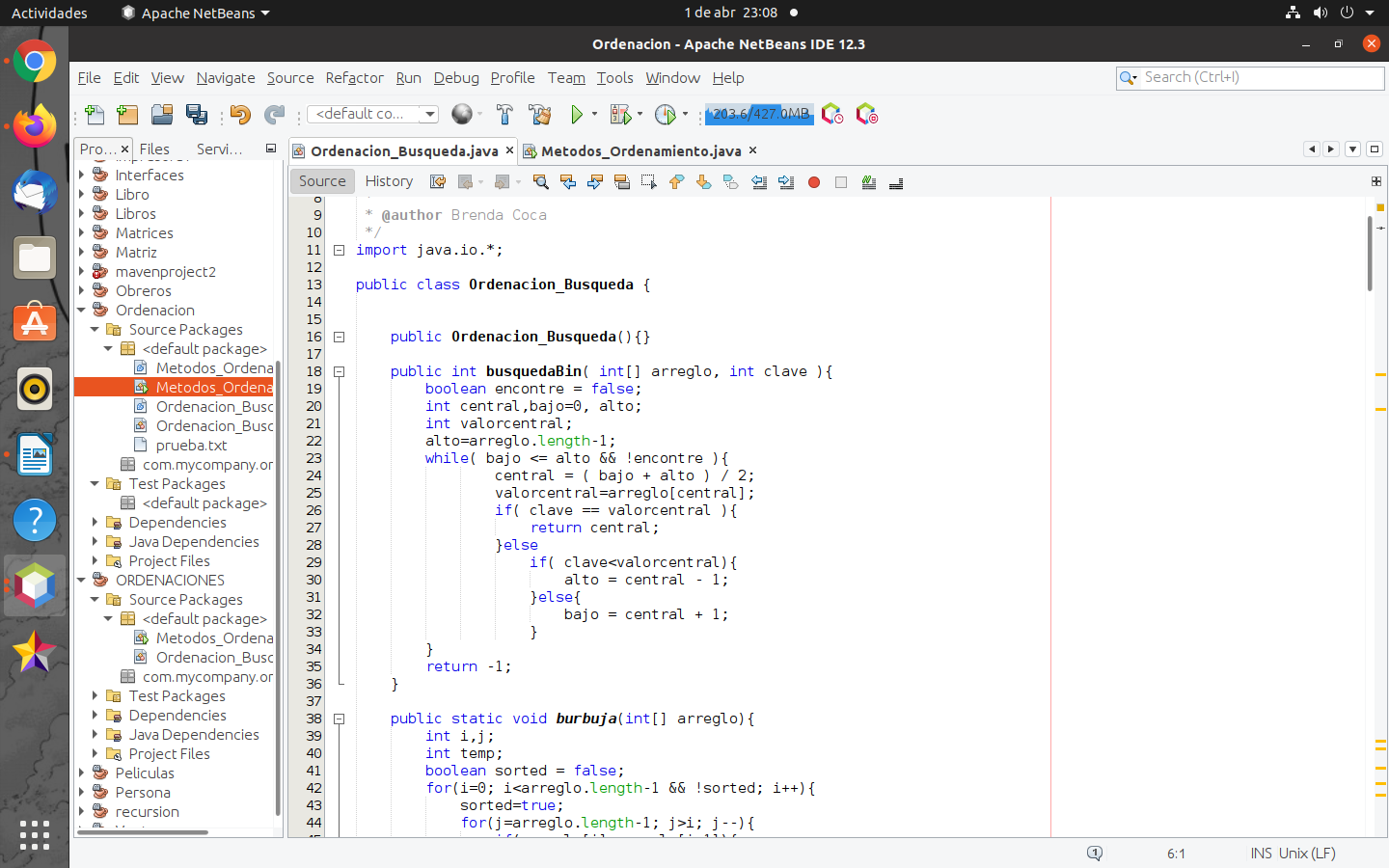
**CAPTURA CORRIDA PROGRAMA**

****

****

**CORRIDA CDM**

****



**CONCLUSIÓN**

Java provee un número importante de clases de excepción para tratar con varios tipos de situaciones, sin embargo, no provee todas las clases de excepción que quizá puedan necesitarse. Java provee un mecanismo al programador para crear sus propias excepciones. Tus propias excepciones deben ser lanzadas con la sentencia.

Para crear tus propias excepciones debes derivar tu clase de tipo excepción de la clase Exception de Java o de alguna de sus subclases. Una vez creada tu propia clase de excepción, puedes definir otra clase de tipo excepción extendiendo la definición de tu clase propia de excepción

**REFERENCIAS BIBLIOGRÁFICAS**

* Deitel, P. (2016). Como programar en Java. España: Pearson.
* Joyanes Aguilar, L., & Zahonero Martínez, I. (2011). Programación en Java 6
* https://www.softwaretestinghelp.com/java/java-exceptions/ .Consultado el 2 de agosto
* del 2020
* IndiaBIX Technologies. (Última edición: 2009-2020). Recuperado de: https://indiabix.com/java-programming/exceptions/ .Consultado el 2 de agosto del 2020
* Malik D. S. (2012). Java Programming: From Problem Analysis to Program Design, Fifth
* Edition, Course Technology, Cengage Learning.
* Liang D. Y. (2015). Intro to Java Programming, Comprehensive Version, 10th Edition, Georgia Southern University, Pearson.

**PROGRAMACION II**

**RUBRICA HOJA DE EVALUACIÓN DE PRACTICAS**

**MATRICULA: 201963582 FECHA: 3-MAYO-2021**

**NOMBRE: Irvyn Xicale Cabrera NO. PRACTICA: 12B INDIVIDUAL ( ) COLABORATIVA( X )**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Deficiente** | **Suficiente** | **Bueno** | **Excelente** | **Calificación Obtenida** | |
| **CRITERIOS** | **1-5.9** | **6‐7.9** | **8‐9** | **9.1‐10** |
| **%** | **Puntos** |
| **CONOCIMIENTO TEORICO**  **20%** | Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases. | Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto. | Conocimiento claro de los fundamentos teóricos POO, pero requiere mejorar el modelado de las clases, sus métodos  y sus relaciones entre clases. | Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase. |  |  |
| **EJECUCCIÓN DE LA PRACTICA**  **30%** | No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java. | Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java | Realiza la práctica, pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación (sintaxis y semántica) | Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica). |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SOLUCIÓN DE LA PRACTICA**  **40%** | No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación. | Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación | Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje. | Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación. |  |  |
| **ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO**  **10%** | No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo. | Posee una actitud propositiva para un aprendizaje autónomo, participa, pero no le gusta trabajar en equipo. | Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo, pero requiere mejorar su participación y portaciones de forma profunda. | Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo |  |  |
| **Total** | | | | |  |  |  |



Brenda Lizeth Coca Garcia   
 Matricula:201917154

