# GUIA DE EJERCICIOS ENTREGAR PARA TERCER PARCIAL EN EQUIPOS

**PROGRAMACIÓN II**

# PARTE I.- (VALOR DE 6 PUNTOS)

1. **Números Aleatorio, ordenaciones y archivos**. Realizar un programa que genere 100 números aleatorios y distintos, los cuales los muestre en pantalla, los ordene utilizando burbuja y los envié ya ordenados en un archivo.

**CLASE CREAR NUMERO**

import java.io.FileWriter;

import java.io.IOException;

public class CrearNumero {

int tamano = 10;

int[] nums = new int[tamano];

int[] bloqueo = new int[tamano];

boolean repite = false;

int numAleatorio;

public CrearNumero() {

for (int i = 0; i < tamano; i++) {

bloqueo[i] = -1;

}

}

public void numeroAleatorio() {

for (int i = 0; i < tamano; i++) {

do {

numAleatorio = (int) (Math.random() \* 200);

for (int j = 0; j < i; j++) {

if (numAleatorio == bloqueo[j]) {

repite = true;

}

}

} while (repite == true);

nums[i] = numAleatorio;

bloqueo[i] = numAleatorio;

}

}

public String verNumeros() {

String cadena = "";

for (int i = 0; i < tamano; i++) {

cadena = cadena + nums [i] + "\n";

}

return cadena;

}

public void burbuja() {

int auxiliar;

for (int i = 2; i < tamano; i++) {

for (int j = 0; j < tamano - i; j++) {

if (nums[j] > nums[j + 1]) {

auxiliar = nums[j];

nums[j] = nums[j + 1];

nums[j + 1] = auxiliar;

}

}

}

}

public void guardar(){

try {

FileWriter Registro = new FileWriter("resultado.txt");

Registro.write(verNumeros());

Registro.close();

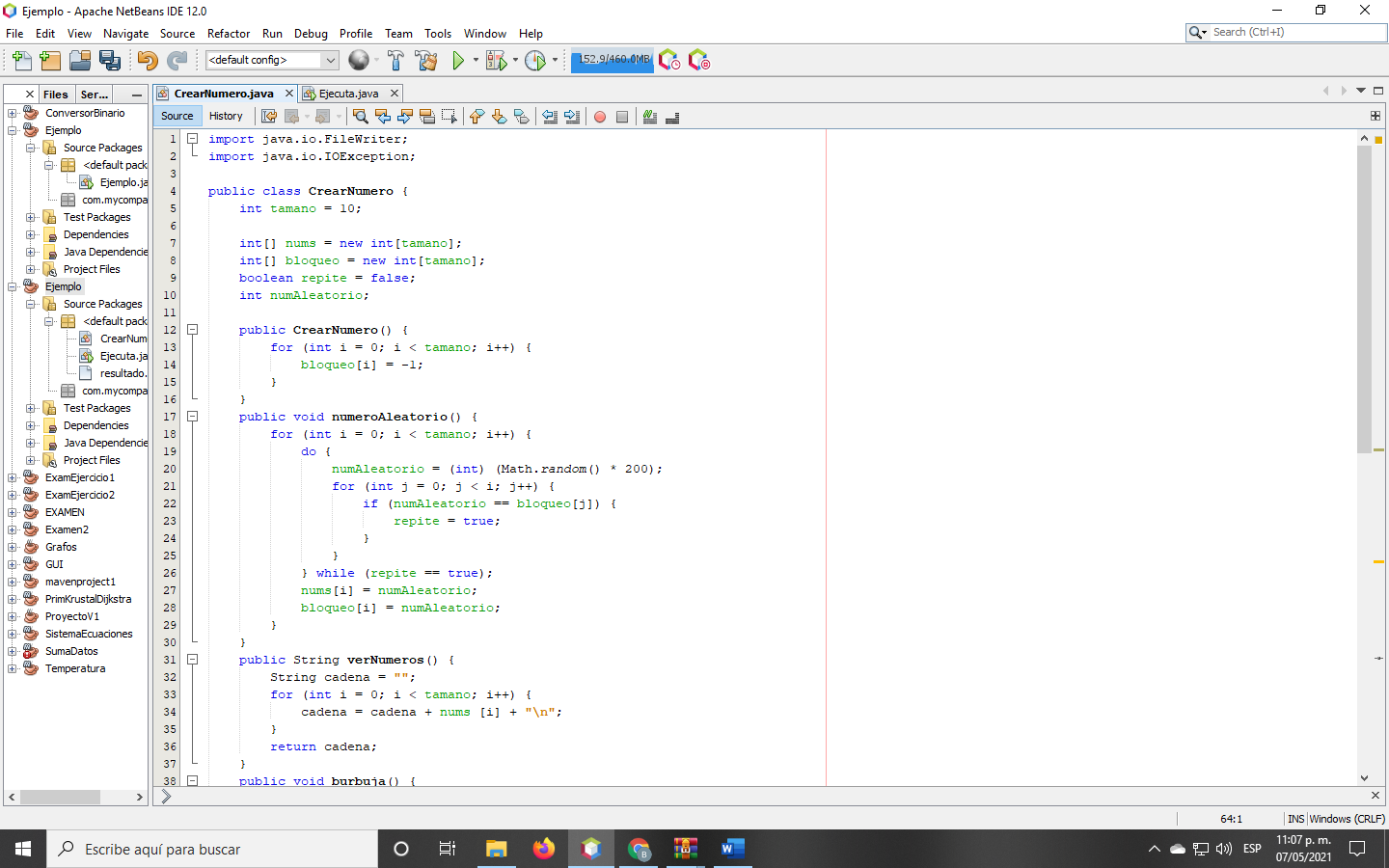
} catch (IOException er) {

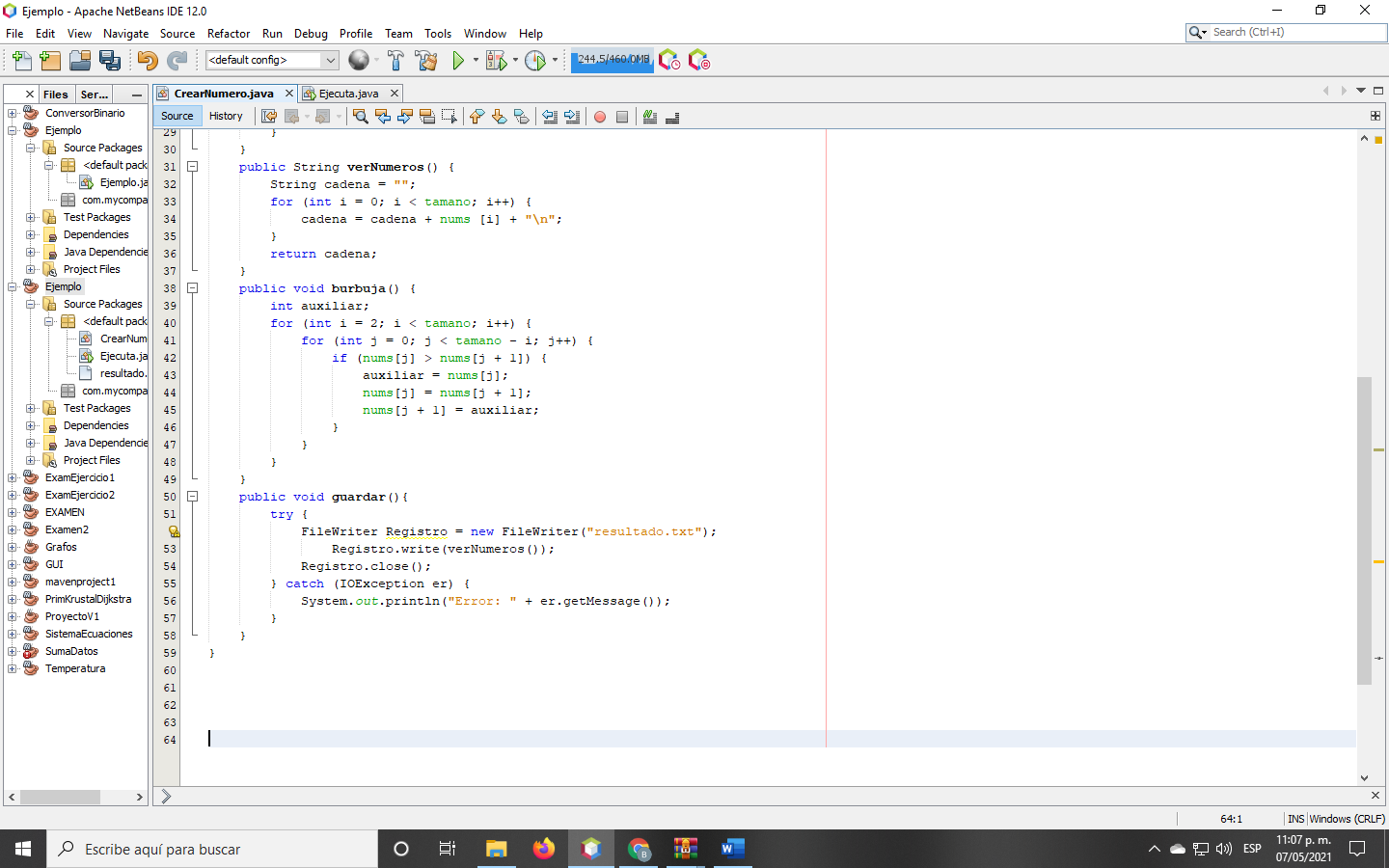
System.out.println("Error: " + er.getMessage());

}

}

}





**CLASE EJECUTA**

public class Ejecuta {

public static void main(String[] args){

CrearNumero numero = new CrearNumero();

numero.numeroAleatorio();

System.out.println("numeros generados:");

System.out.println(numero.verNumeros());

numero.burbuja();

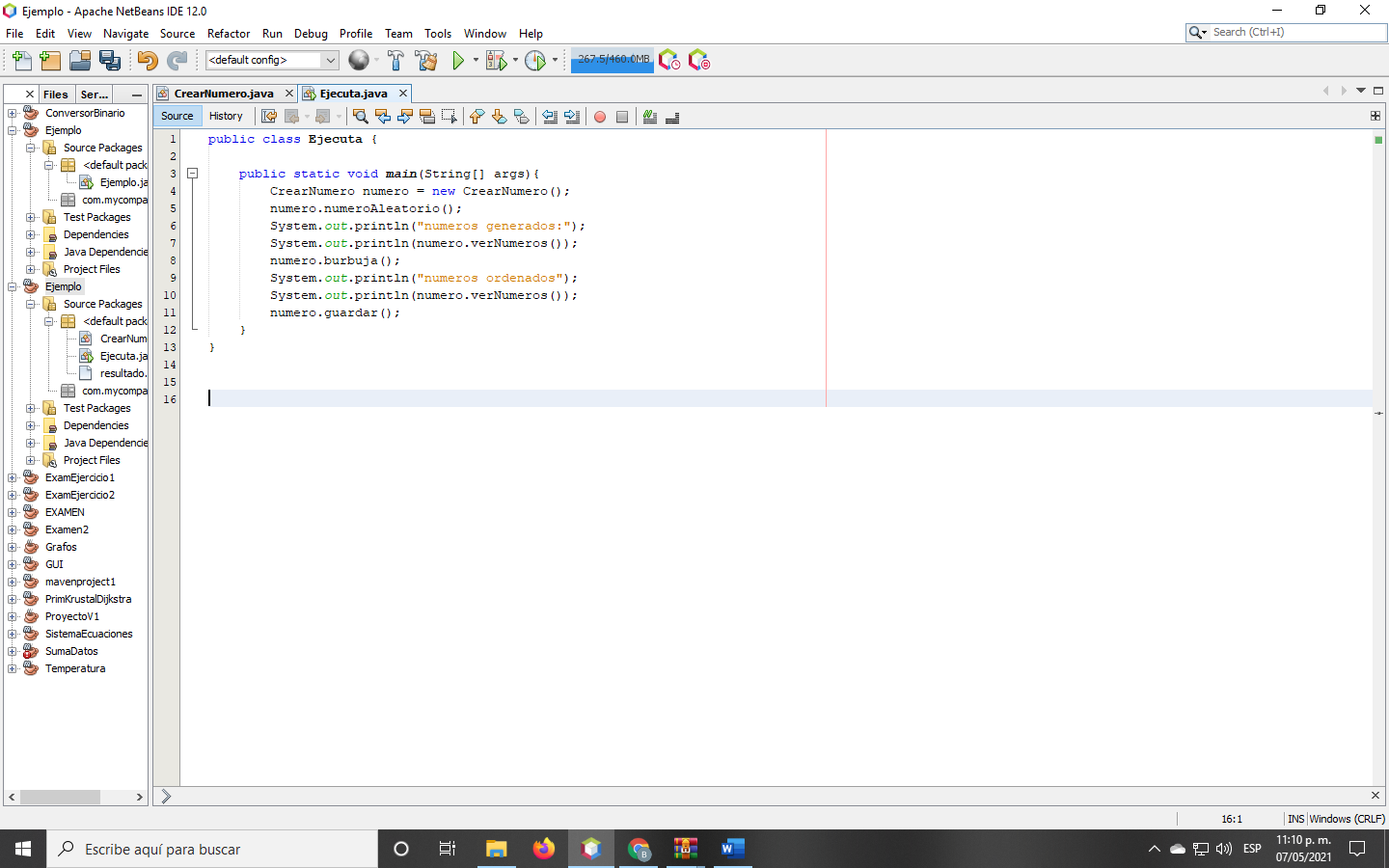
System.out.println("numeros ordenados");

System.out.println(numero.verNumeros());

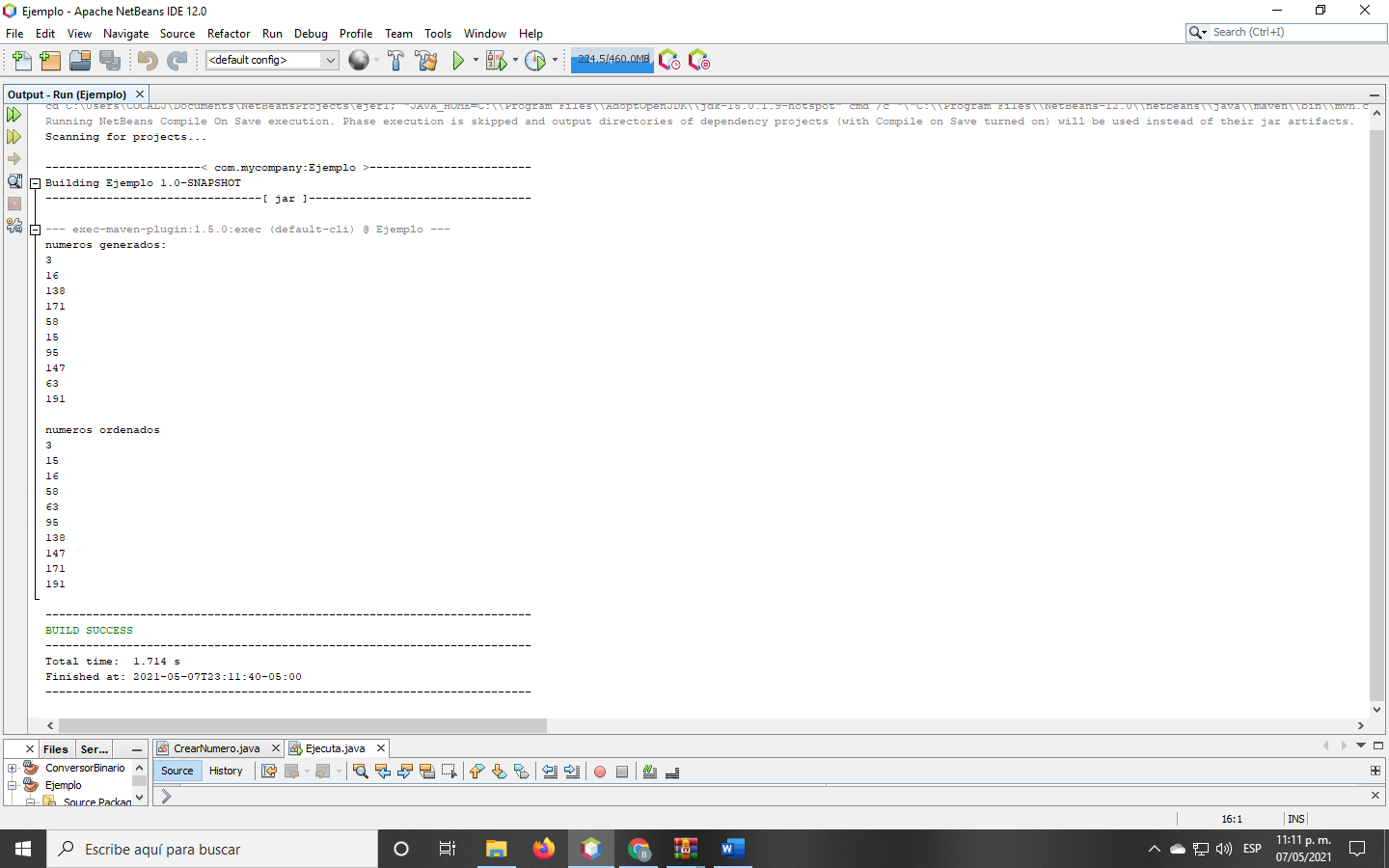
numero.guardar();

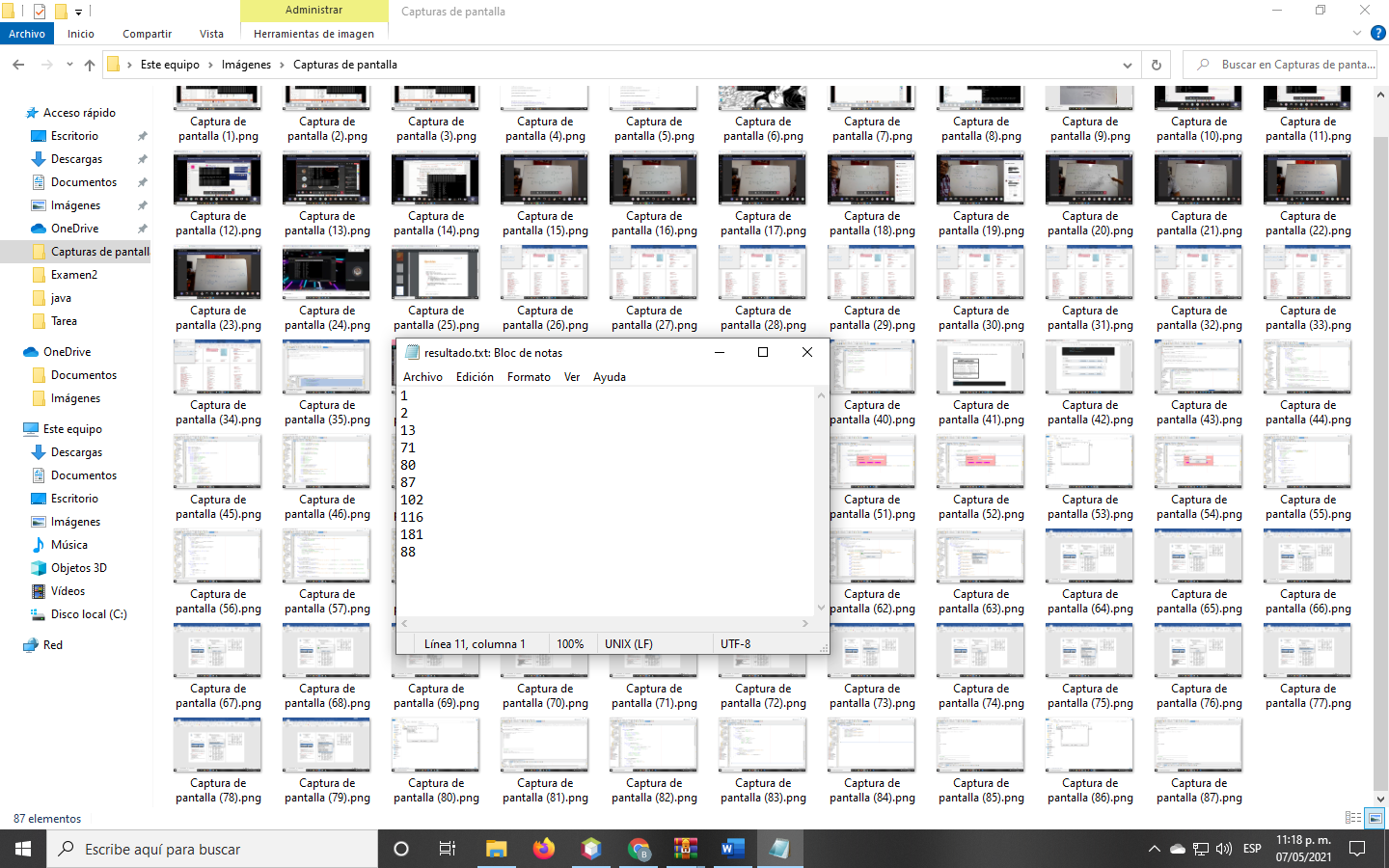
}

}



**CORRIDA**





1. **Arreglo y archivo**. Crear un programa que almacena en un arreglo n números pares y generar su promedio así como almacenarlos en un archivo del cual el usuario escriba el nombre del archivo.

**CLASE CREAR NUMEROS**

import java.io.\*;

public class CrearNumeros{

InputStreamReader entrada = new InputStreamReader (System.in);

BufferedReader flujoEntrada = new BufferedReader(entrada);

String datoEntrada;

int[] nums;

int n;

boolean cumple = true;

public CrearNumeros(){

System.out.println("ingresa el tamaño del arreglo");

do{

try{

datoEntrada = flujoEntrada.readLine();

n = Integer.parseInt(datoEntrada);

}catch(Exception e){

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

cumple = false;

}

}while (cumple == false);

nums = new int[n];

}

public void ingresarNumeros(){

int numAleatorio;

boolean repite = false;

for (int i = 0; i < n; i++) {

do {

numAleatorio = (int) (Math.random() \* 200);

if(numAleatorio%2 != 0){

repite = true;

}else{

repite = false;

}

} while (repite == true);

nums[i] = numAleatorio;

}

}

public String verNumeros() {

String cadena = "";

for (int i = 0; i < n; i++) {

cadena = cadena + nums [i] + "\n";

}

return cadena;

}

public double promedio(){

double promedio=0;

for(int i=0; i<n ; i++){

promedio = promedio + (double) nums[i];

}

promedio = promedio / (double) n;

return promedio;

}

public void guardar(){

String nombre = "";

do{

System.out.println("ingresa un nombre al archivo");

try{

datoEntrada = flujoEntrada.readLine();

nombre = datoEntrada;

}catch(Exception e){

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

cumple = false;

}

}while (cumple == false);

try {

FileWriter Registro = new FileWriter(nombre + ".txt");

Registro.write(verNumeros());

Registro.write("el promedio es: " + promedio());

Registro.close();

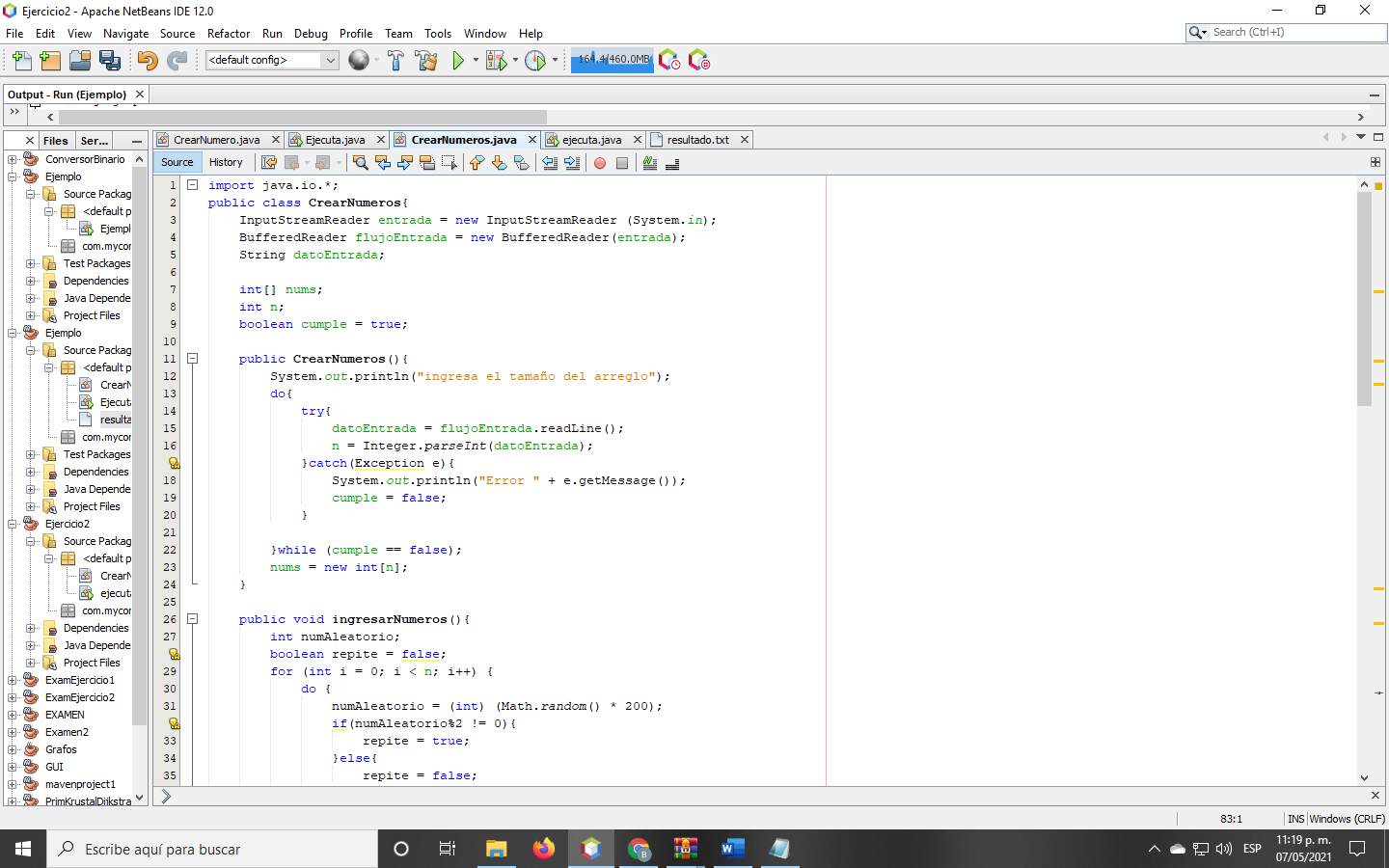
} catch (IOException er) {

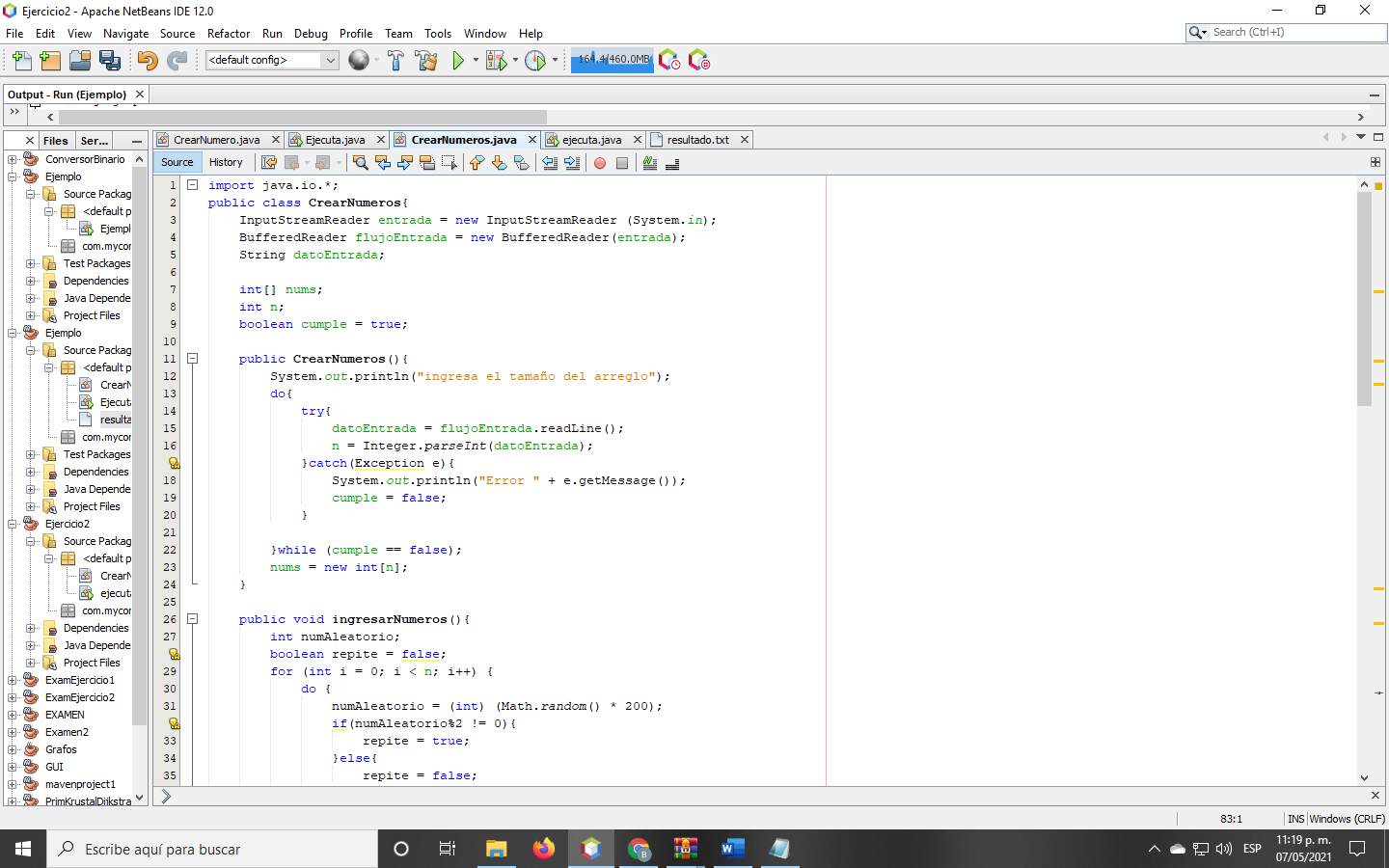
System.out.println("Error: " + er.getMessage());

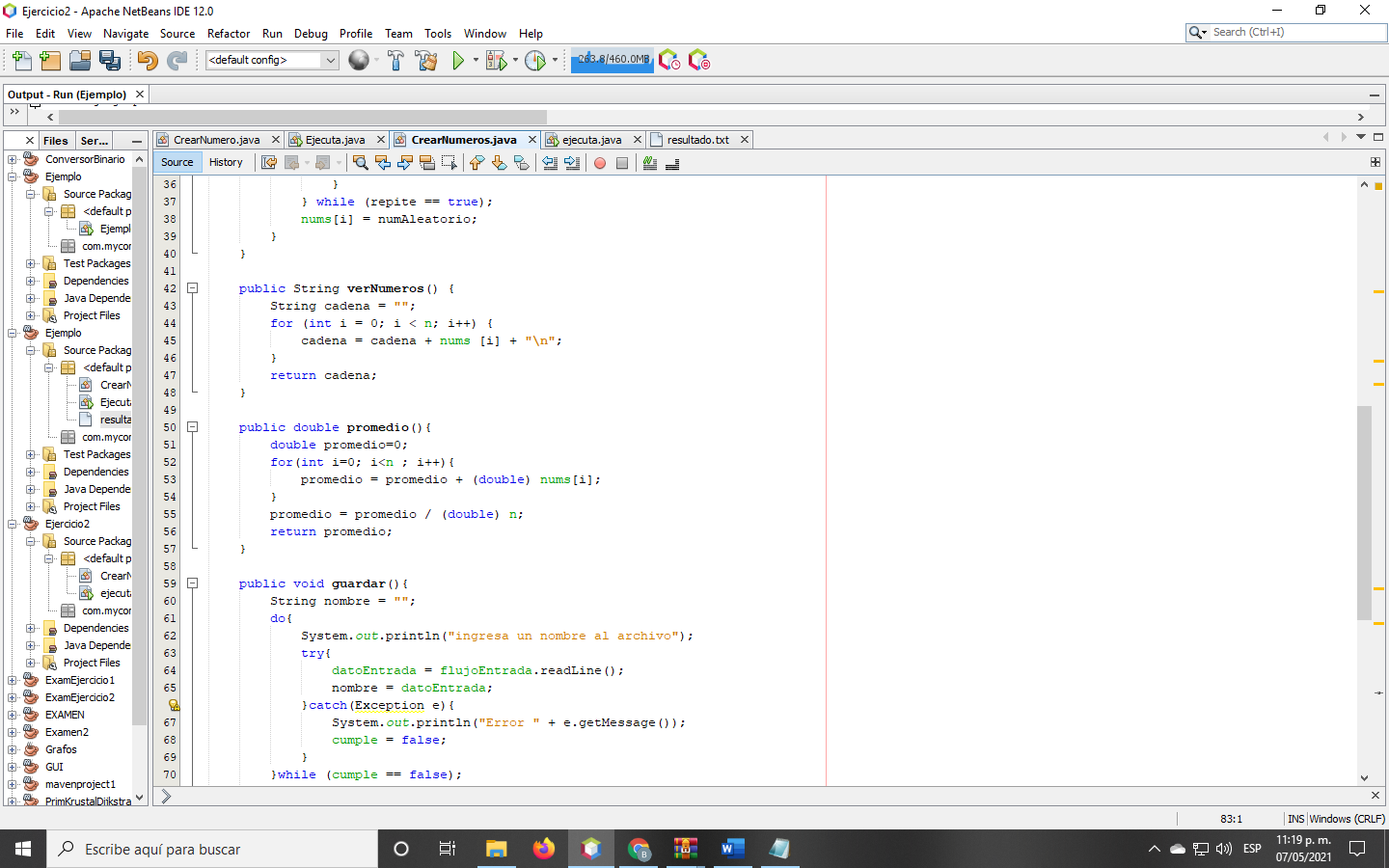
}

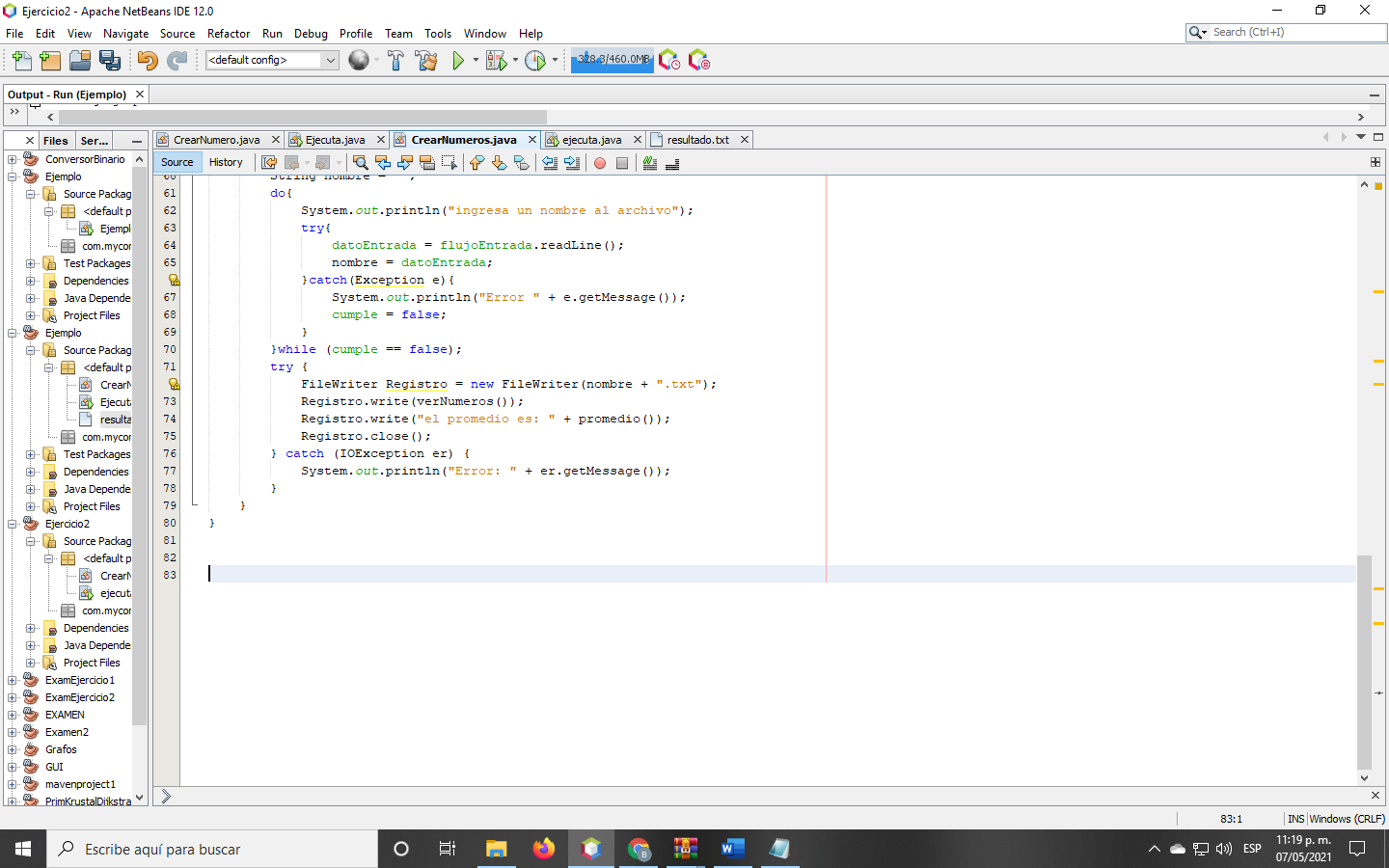
}

}









**CLASE EJECUTA**

public class ejecuta {

public static void main(String[] args){

CrearNumeros numeros = new CrearNumeros();

numeros.ingresarNumeros();

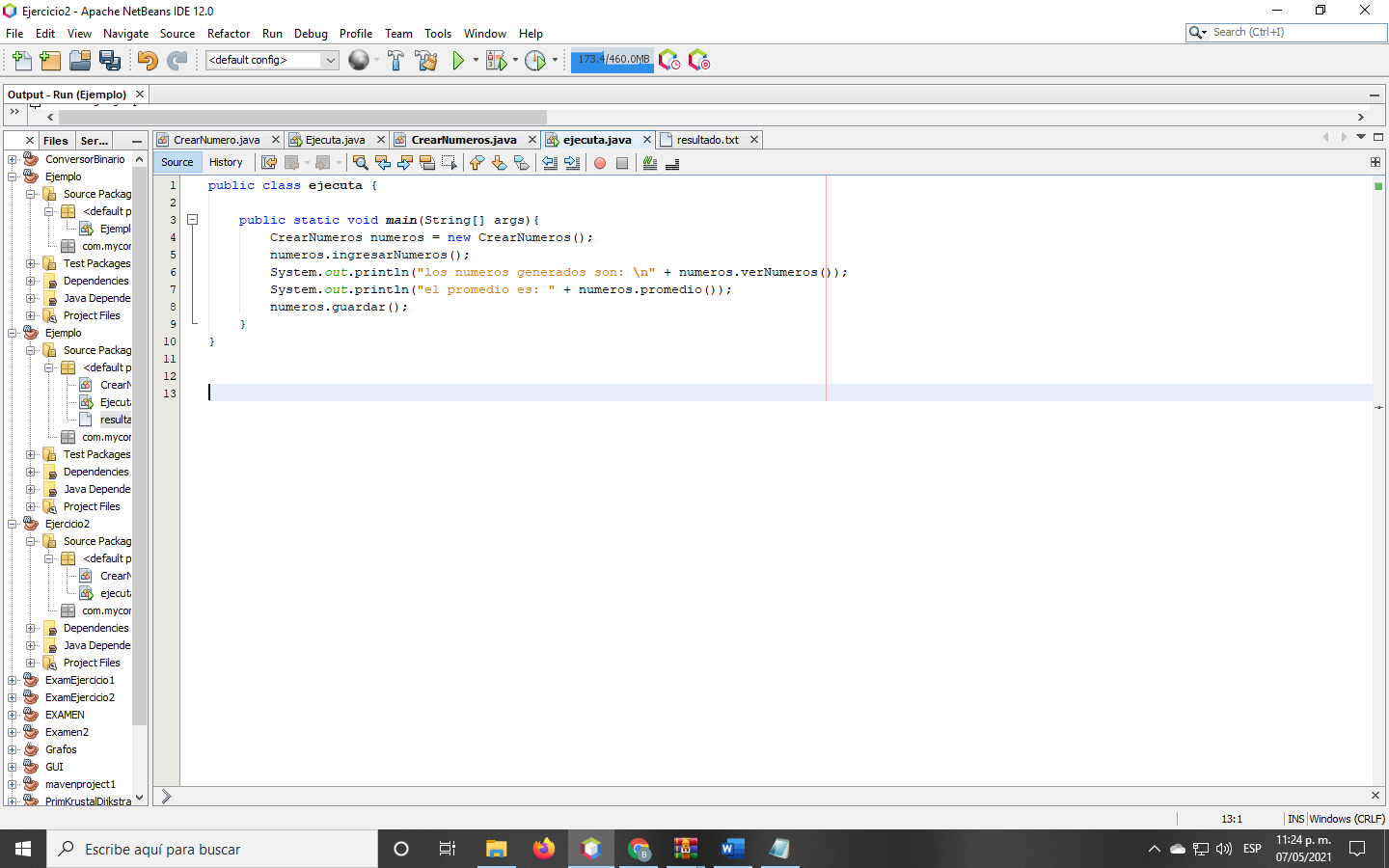
System.out.println("los numeros generados son: \n" + numeros.verNumeros());

System.out.println("el promedio es: " + numeros.promedio());

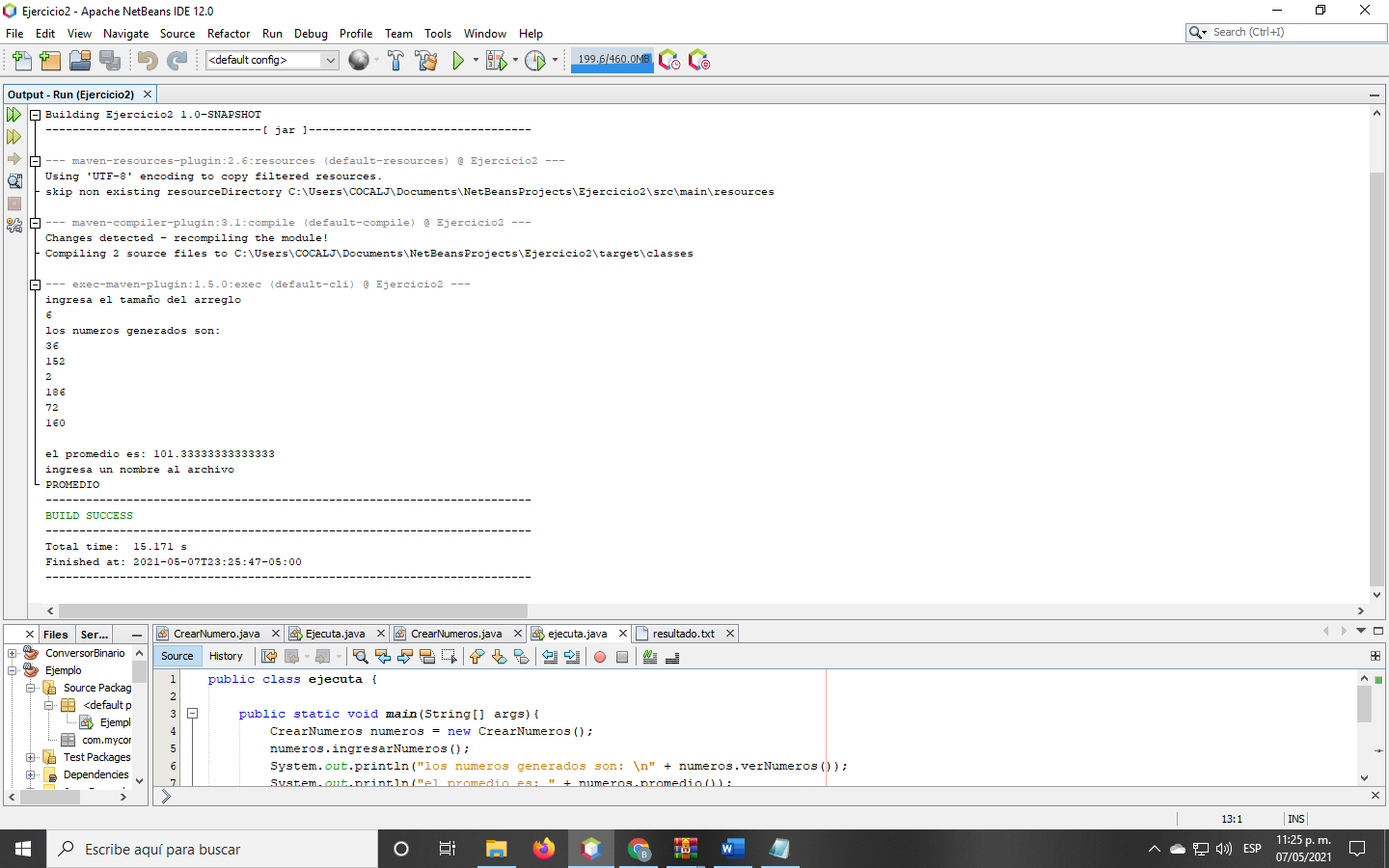
numeros.guardar();

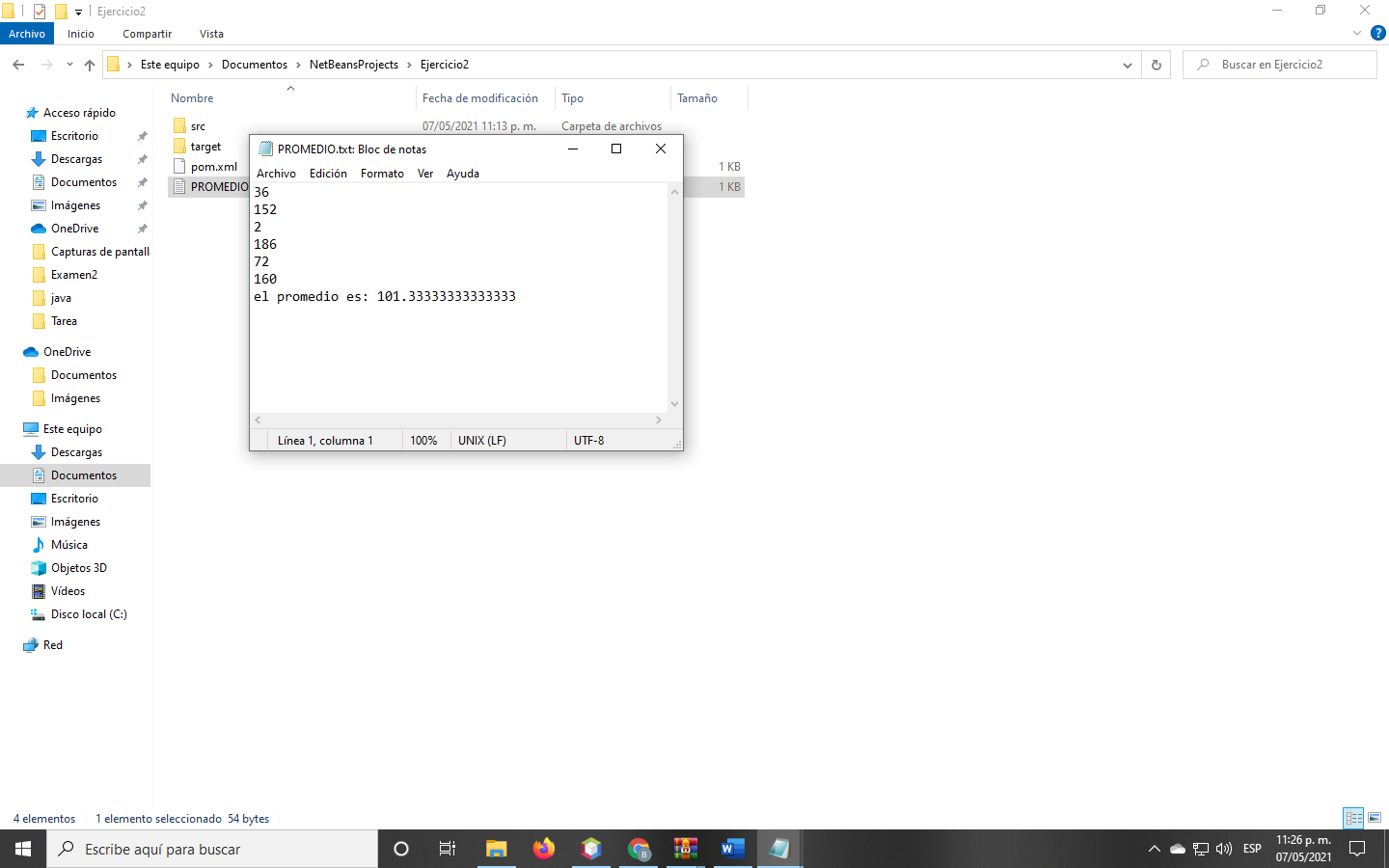
}

}



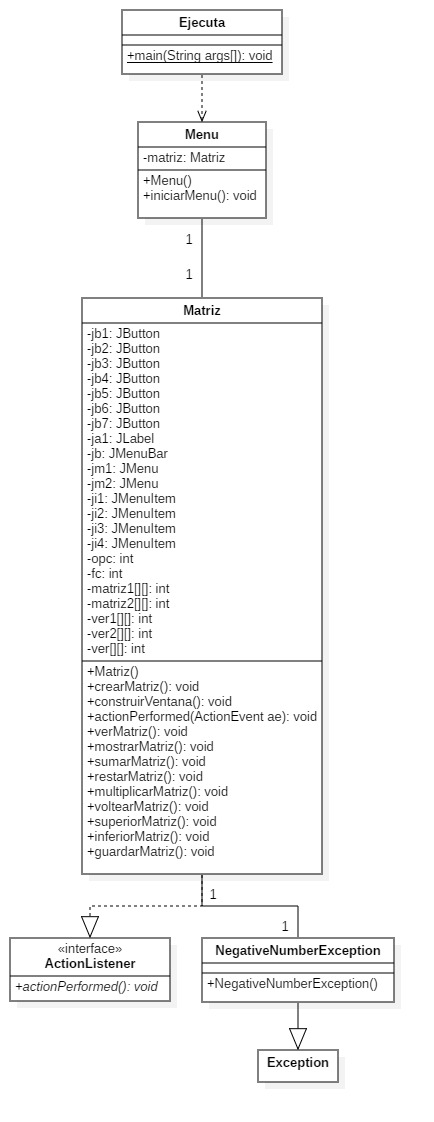
**CORRIDA**





1. **Lectura de Archivos**. Obtener la lectura de n personas de un archivo y pasarlos a un arreglo, mostrar las edades en pantalla y obtener el número de alturas mayores a la media y menores a ella. Mostrar resultados en pantalla y enviar cada uno de los resultados a dos archivos diferentes.
2. **Lectura y escritura de Archivos**. Realizar un programa que pueda leer en un archivo de 20 calificaciones flotantes y obtener :
   1. Desviación estándar
   2. Error absoluto
   3. Error relativo
   4. Media
   5. moda
3. **Quicksort**. Leer desde un archivo 10 nombres, almacenarlos en un arreglo ordenarlas mediante el método de Quicksort y enviarlos a un archivo que el usuario escriba su nombre.
4. **Matrices** Realizar un programa que sume, multiplique, reste matrices, transpuesta y matriz triangular superior e inferior y los resultados van a ser presentados en pantalla y enviados a un archivo según la opción del usuario.

**DIAGRAMA UML**



**CÓDIGO DEL PROGRAMA**

**Ejecuta.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Crear una matriz y realizar algunos ejercicios.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

public class Ejecuta {

public static void main(String args[]) {

//Variable de instancia

Menu menu = new Menu();

//Acceder a un método

menu.iniciarMenu();

}

}

//Fin de la clase

**Matriz.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Crear una matriz y realizar algunos ejercicios.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

import java.awt.event.\*;

import javax.swing.\*;

import java.awt.\*;

import java.io.\*;

public class Matriz implements ActionListener {

//Variables de instancia

private JButton jb1;

private JButton jb2;

private JButton jb3;

private JButton jb4;

private JButton jb5;

private JButton jb6;

private JButton jb7;

private JLabel ja1;

private JMenuBar jb;

private JMenu jm1;

private JMenu jm2;

private JMenuItem ji1;

private JMenuItem ji2;

private JMenuItem ji3;

private JMenuItem ji4;

//Atributos

private int opc = 0;

private int fc = 0;

//Matrices

private int matriz1[][] = new int[100][100];

private int matriz2[][] = new int[100][100];

private int ver1[][] = new int[100][100];

private int ver2[][] = new int[100][100];

private int ver[][] = new int[100][100];

//Método constructor

public Matriz() {}

//Metódo definido por el programador

public void crearMatriz() {

//Atributo

String panel;

int salir;

//Ciclo

do {

//Ingresar un dato

panel = JOptionPane.showInputDialog(null,"Ingrese el numero de filas y columnas de las matrices:","MATRIZ",JOptionPane.QUESTION\_MESSAGE);

//Iniciar excepcion

try {

//Asignar valores

fc = Integer.parseInt(panel);

salir = 1;

if(fc <= 0)

//Lanzar error

throw new NegativeNumberException();

} catch(NumberFormatException nfe) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+nfe.getMessage()

+"\nNo leiste dato numerico.","ERROR",JOptionPane.WARNING\_MESSAGE);

salir = 0;

} catch(NegativeNumberException nne) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+nne.getMessage()

+"\nEl numero de filas y columnas debe ser mayor que cero.","ERROR",JOptionPane.WARNING\_MESSAGE);

salir = 0;

}

} while(salir != 1);

//Ciclo

for(int i=0; i<fc; i++)

for(int j=0; j<fc; j++) {

//Ciclo

do {

//Iniciar excepcion

try {

//Ingresar un dato

panel = JOptionPane.showInputDialog(null,"Ingrese un valor para Elemento["+(i+1)+"]["+(j+1)+"]:","MATRIZ 1",JOptionPane.QUESTION\_MESSAGE);

matriz1[i][j] = Integer.parseInt(panel);

ver1[i][j] = Integer.parseInt(panel);

salir = 1;

} catch(NumberFormatException nfe) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+nfe.getMessage()

+"\nNo leiste dato numerico.","ERROR",JOptionPane.WARNING\_MESSAGE);

salir = 0;

}

} while(salir != 1);

}

//Ciclo

for(int i=0; i<fc; i++)

for(int j=0; j<fc; j++) {

//Ciclo

do {

//Iniciar excepcion

try {

//Ingresar un dato

panel = JOptionPane.showInputDialog(null,"Ingrese un valor para Elemento["+(i+1)+"]["+(j+1)+"]:","MATRIZ 2",JOptionPane.QUESTION\_MESSAGE);

matriz2[i][j] = Integer.parseInt(panel);

ver2[i][j] = Integer.parseInt(panel);

salir = 1;

} catch(NumberFormatException nfe) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+nfe.getMessage()

+"\nNo leiste dato numerico.","ERROR",JOptionPane.WARNING\_MESSAGE);

salir = 0;

}

} while(salir != 1);

}

//Acceder a un método

construirVentana();

}

//Metódo definido por el programador

public void construirVentana() {

//Variables de instancia

JPanel jp = new JPanel();

JFrame jf = new JFrame();

//Barra de herramientas

jb = new JMenuBar();

jb.setBounds(0,0,100,20);

jf.add(jb);

//Herramienta 1

jm1 = new JMenu("Matriz");

jb.add(jm1);

//Opcion 1

ji1 = new JMenuItem("Ver");

jm1.add(ji1);

ji1.addActionListener(this);

jm1.addSeparator();

//Opcion 2

ji2 = new JMenuItem("Guardar");

jm1.add(ji2);

ji2.addActionListener(this);

//Herramienta 2

jm2 = new JMenu("Ayuda");

jb.add(jm2);

//Opcion 1

ji3 = new JMenuItem("Tutorial");

jm2.add(ji3);

ji3.addActionListener(this);

jm2.addSeparator();

//Opcion 2

ji4 = new JMenuItem("Acerca de...");

jm2.add(ji4);

ji4.addActionListener(this);

//Etiqueta 1

ja1 = new JLabel("Ingrese el numero de la opcion a ejecutar:");

ja1.setBounds(20,30,240,30);

jf.add(ja1);

//Boton 1

jb1 = new JButton("Matrices originales");

jb1.setBounds(20,70,240,30);

jf.add(jb1);

//Presionar

jb1.addActionListener(this);

//Boton 2

jb2 = new JButton("Sumar");

jb2.setBounds(20,110,240,30);

jf.add(jb2);

//Presionar

jb2.addActionListener(this);

//Boton 3

jb3 = new JButton("Restar");

jb3.setBounds(20,150,240,30);

jf.add(jb3);

//Presionar

jb3.addActionListener(this);

//Boton 4

jb4 = new JButton("Multiplicar");

jb4.setBounds(20,190,240,30);

jf.add(jb4);

//Presionar

jb4.addActionListener(this);

//Boton 5

jb5 = new JButton("Transpuesta");

jb5.setBounds(20,230,240,30);

jf.add(jb5);

//Presionar

jb5.addActionListener(this);

//Boton 6

jb6 = new JButton("Triangular superior");

jb6.setBounds(20,270,240,30);

jf.add(jb6);

//Presionar

jb6.addActionListener(this);

//Boton 7

jb7 = new JButton("Triangular inferior");

jb7.setBounds(20,310,240,30);

jf.add(jb7);

//Presionar

jb7.addActionListener(this);

//Ventana

jp.setBounds(0,0,300,400);

jp.setLayout(null);

jf.setLayout(null);

jf.setSize(300,400);

jf.add(jp);

jf.setVisible(true);

jf.setTitle("MATRIZ");

jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

//Metódo definido por el programador

@Override

public void actionPerformed(ActionEvent ae) {

//Validar boton presionado

if(ae.getSource() == jb1)

//Acceder a un método

mostrarMatriz();

if(ae.getSource() == jb2)

//Acceder a un método

sumarMatriz();

if(ae.getSource() == jb3)

//Acceder a un método

restarMatriz();

if(ae.getSource() == jb4)

//Acceder a un método

multiplicarMatriz();

if(ae.getSource() == jb5)

//Acceder a un método

voltearMatriz();

if(ae.getSource() == jb6)

//Acceder a un método

superiorMatriz();

if(ae.getSource() == jb7)

//Acceder a un método

inferiorMatriz();

if(ae.getSource() == ji1)

//Acceder a un método

verMatriz();

if(ae.getSource() == ji2)

//Acceder a un método

guardarMatriz();

if(ae.getSource() == ji3)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Pasos a seguir:"

+"\n1) Lea cada una de las opciones en pantalla."

+"\n2) Presione el botón a ejecutar."

+"\n\nNota:"

+"\n - La herramienta Matriz mostrara y/o guardara la ultima matriz modificada.");

if(ae.getSource() == ji4)

//Acceder a un método

JOptionPane.showMessageDialog(null,"Ejercicio 6 - Matriz:"

+"\n - Equipo 4:"

+"\n + Autor 1: Flores Navarro Emmanuel.......201929142"

+"\n + Autor 2: Ramírez Morales Erick Brian...201955520"

+"\n - Fecha: 07 / Mayo / 2021"

+"\n - Derechos Reservados");

}

//Metódo definido por el programador

public void verMatriz() {

//Atributo

String texto = "";

if(opc == 0) {

//Ciclo

texto += "MATRIZ 1:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+ver1[i][j]+" |";

}

texto += "\n";

}

//Ciclo

texto += "\n\nMATRIZ 2:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+ver2[i][j]+" |";

}

texto += "\n";

}

} else {

//Ciclo

texto += "MATRIZ:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+ver[i][j]+" |";

}

texto += "\n";

}

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto,"MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void mostrarMatriz() {

//Atributos

String texto = "";

opc = 0;

//Ciclo

texto += "MATRIZ 1:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+matriz1[i][j]+" |";

ver1[i][j] = matriz1[i][j];

}

texto += "\n";

}

//Ciclo

texto += "\n\nMATRIZ 2:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+matriz2[i][j]+" |";

ver2[i][j] = matriz2[i][j];

}

texto += "\n";

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto,"MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void sumarMatriz() {

//Atributos

String texto = "";

opc = 1;

//Ciclo

texto += "SUMA:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

//Sumar valores

ver[i][j] = matriz1[i][j] + matriz2[i][j];

texto += " "+ver[i][j]+" |";

}

texto += "\n";

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto,"MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void restarMatriz() {

//Atributos

String texto = "";

opc = 1;

//Ciclo

texto += "RESTA:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

//Sumar valores

ver[i][j] = matriz1[i][j] - matriz2[i][j];

texto += " "+ver[i][j]+" |";

}

texto += "\n";

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto,"MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void multiplicarMatriz() {

//Arreglos

int fila[] = new int[fc\*fc];

int columna[] = new int[fc\*fc];

//Atributos

int e1,e2,k,y,z,mul;

String texto = "";

opc = 1;

//Ciclo

k = 0;

for(int j=0; j<fc; j++) {

for(int i=0; i<fc; i++) {

//Asignar valores

fila[k] = matriz2[i][j];

k++;

}

}

//Ciclo

k = 0;

for(int i=0; i<fc; i++) {

for(int j=0; j<fc; j++) {

//Asignar valores

columna[k] = matriz1[i][j];

k++;

}

}

//Asignar valores

e1 = 0;

z = 0;

for(int x=0; x<fc; x++) {

//Asignar valores

y = 0;

e2 = 0;

for(int i=0; i<fc; i++) {

//Asignar valor

mul = 0;

for(int j=0; j<fc; j++) {

mul = mul + (fila[j+z]\*columna[j+y]);

}

//Asignar valores

y = y+fc;

ver[e1][e2] = mul;

e2++;

}

//Asignar valores

z = z+fc;

e1++;

}

//Ciclo

texto += "MULTIPLICACION:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

//Asignar valor

texto += " "+ver[j][i]+" |";

}

texto += "\n";

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto,"MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void voltearMatriz() {

//Atributos

String texto = "";

opc = 0;

//Ciclo

texto += "MATRIZ 1:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+matriz1[j][i]+" |";

ver1[j][i] = matriz1[j][i];

}

texto += "\n";

}

//Ciclo

texto += "\n\nMATRIZ 2:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+matriz2[j][i]+" |";

ver2[j][i] = matriz2[j][i];

}

texto += "\n";

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto,"MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void superiorMatriz() {

//Atributos

int t1 = 0, t2 = 0;

//Ciclo

for(int i=1; i<fc; i++) {

for(int j=0; j<i; j++) {

//Validar valor

if(matriz1[i][j] != 0) {

t1++;

}

}

}

//Ciclo

for(int i=1; i<fc; i++) {

for(int j=0; j<i; j++) {

//Validar valor

if(matriz2[i][j] != 0) {

t2++;

}

}

}

//Validar valores

if(t1 == 0 && t2 == 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 es triangular superior."

+"\nMATRIZ 2 es triangular superior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

if(t1 > 0 && t2 == 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 no es triangular superior."

+"\nMATRIZ 2 es triangular superior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

if(t1 == 0 && t2 > 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 es triangular superior."

+"\nMATRIZ 2 no es triangular superior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

if(t1 > 0 && t2 > 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 no es triangular superior."

+"\nMATRIZ 2 no es triangular superior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void inferiorMatriz() {

//Atributos

int t1 = 0, t2 = 0;

//Ciclo

for(int i=0; i<fc; i++) {

for(int j=i+1; j<fc; j++) {

//Validar valor

if(matriz1[i][j] != 0) {

t1++;

}

}

}

//Ciclo

for(int i=0; i<fc; i++) {

for(int j=i+1; j<fc; j++) {

//Validar valor

if(matriz2[i][j] != 0) {

t1++;

}

}

}

//Validar valores

if(t1 == 0 && t2 == 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 es triangular inferior."

+"\nMATRIZ 2 es triangular inferior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

if(t1 > 0 && t2 == 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 no es triangular inferior."

+"\nMATRIZ 2 es triangular inferior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

if(t1 == 0 && t2 > 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 es triangular inferior."

+"\nMATRIZ 2 no es triangular inferior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

if(t1 > 0 && t2 > 0)

//Mostrar ventana

JOptionPane.showMessageDialog(null,"MATRIZ 1 no es triangular inferior."

+"\nMATRIZ 2 no es triangular inferior.","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

}

//Metódo definido por el programador

public void guardarMatriz() {

//Variables de instancia

FileWriter fw = null;

PrintWriter pw = null;

//Atributo

String texto = "";

if(opc == 0) {

//Ciclo

texto += "MATRIZ 1:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+ver1[i][j]+" |";

}

texto += "\n";

}

//Ciclo

texto += "\n\nMATRIZ 2:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+ver2[i][j]+" |";

}

texto += "\n";

}

} else {

//Ciclo

texto += "MATRIZ:\n";

for(int i=0; i<fc; i++) {

texto += "|";

for(int j=0; j<fc; j++) {

texto += " "+ver[i][j]+" |";

}

texto += "\n";

}

}

//Iniciar excepcion

try {

//Variables de instancia

fw = new FileWriter("Matriz.txt");

pw = new PrintWriter(fw);

//Guardar mensajes y resultado

pw.println("\n=============== MATRIZ ===============\n");

pw.println(texto);

pw.println("\n======================================\n");

} catch(Exception e1) {

//Capturar e imprimir error

JOptionPane.showMessageDialog(null,"ERROR..."+e1.getMessage()+"...","ERROR",JOptionPane.WARNING\_MESSAGE);

} finally {

//Iniciar excepcion

try {

//Validar dato

if (null != fw)

fw.close();

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Matriz guardada","MATRIZ",JOptionPane.INFORMATION\_MESSAGE);

} catch(Exception e2) {

//Capturar e imprimir error

JOptionPane.showMessageDialog(null,"ERROR..."+e2.getMessage()+"...","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}

}

//Fin de la clase

**Menu.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Crear una matriz y realizar algunos ejercicios.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

public class Menu {

//Variables de instancia

private Matriz matriz = new Matriz();

//Método constructor

public Menu() {}

//Metódo definido por el programador

public void iniciarMenu() {

//Atributos

String panel;

int opc;

//Inicio Menu

do{

panel = JOptionPane.showInputDialog(null,"Ingrese el numero de la opcion a ejecutar:"

+"\n 1) Inicio"

+"\n 2) Salir","MENU",JOptionPane.QUESTION\_MESSAGE);

//Iniciar excepcion

try {

//Ingresar un valor

opc = Integer.parseInt(panel);

} catch(Exception e) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+e.getMessage(),"ERROR",JOptionPane.WARNING\_MESSAGE);

opc = 0;

}

switch(opc) {

case 2 -> {

//Mostrar mensaje y cerrar panel

JOptionPane.showMessageDialog(null,"Gracias por usar este programa, hasta luego!!!");

System.exit(0);

}

case 1 -> {

//Acceder a un método

matriz.crearMatriz();

opc = 2;

}

default -> {

//Mostrar mensaje

JOptionPane.showMessageDialog(null,"Opcion no valida","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}while(opc != 2);

//Fin Menu

}

}

//Fin de la clase

**NegativeNumberException.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Crear una matriz y realizar algunos ejercicios.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

public class NegativeNumberException extends Exception {

//Método constructor

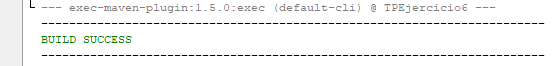
public NegativeNumberException() {}

}

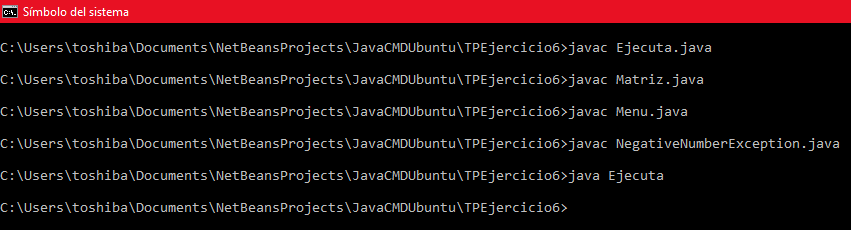
//Fin de la clase

**CAPTURAS DE PANTALLA**

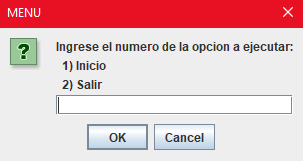
**NetBeans**

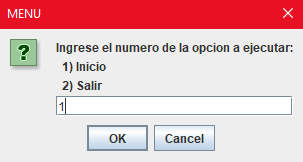


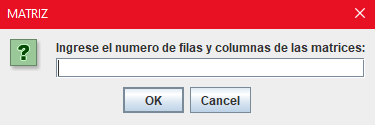
**CMD**

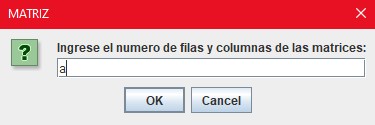


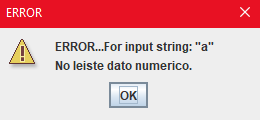
**Ventanas de ejecución**

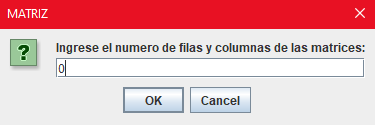


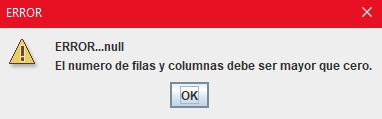


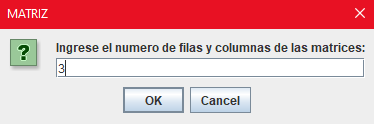


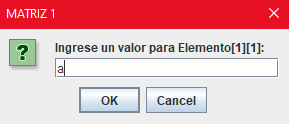


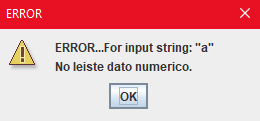


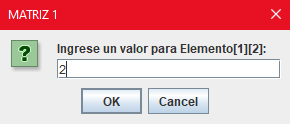
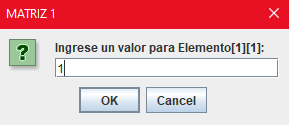


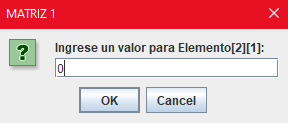
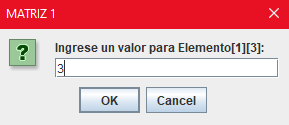


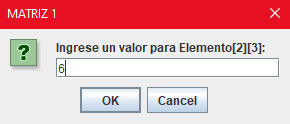
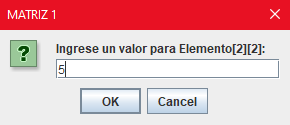


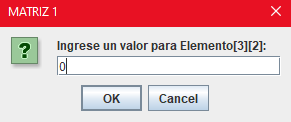
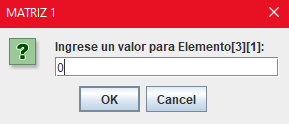


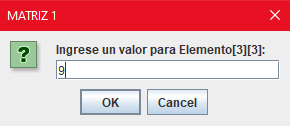


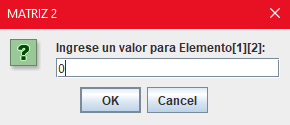
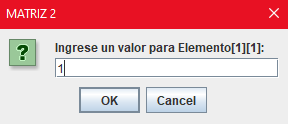


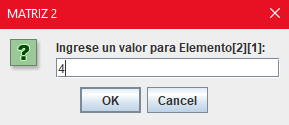
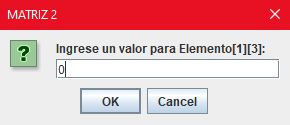


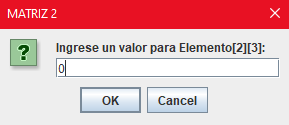
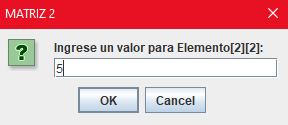


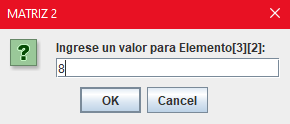
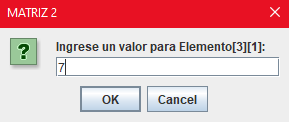


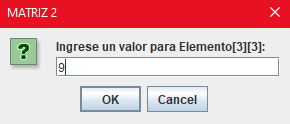


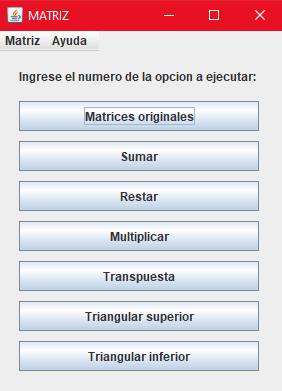


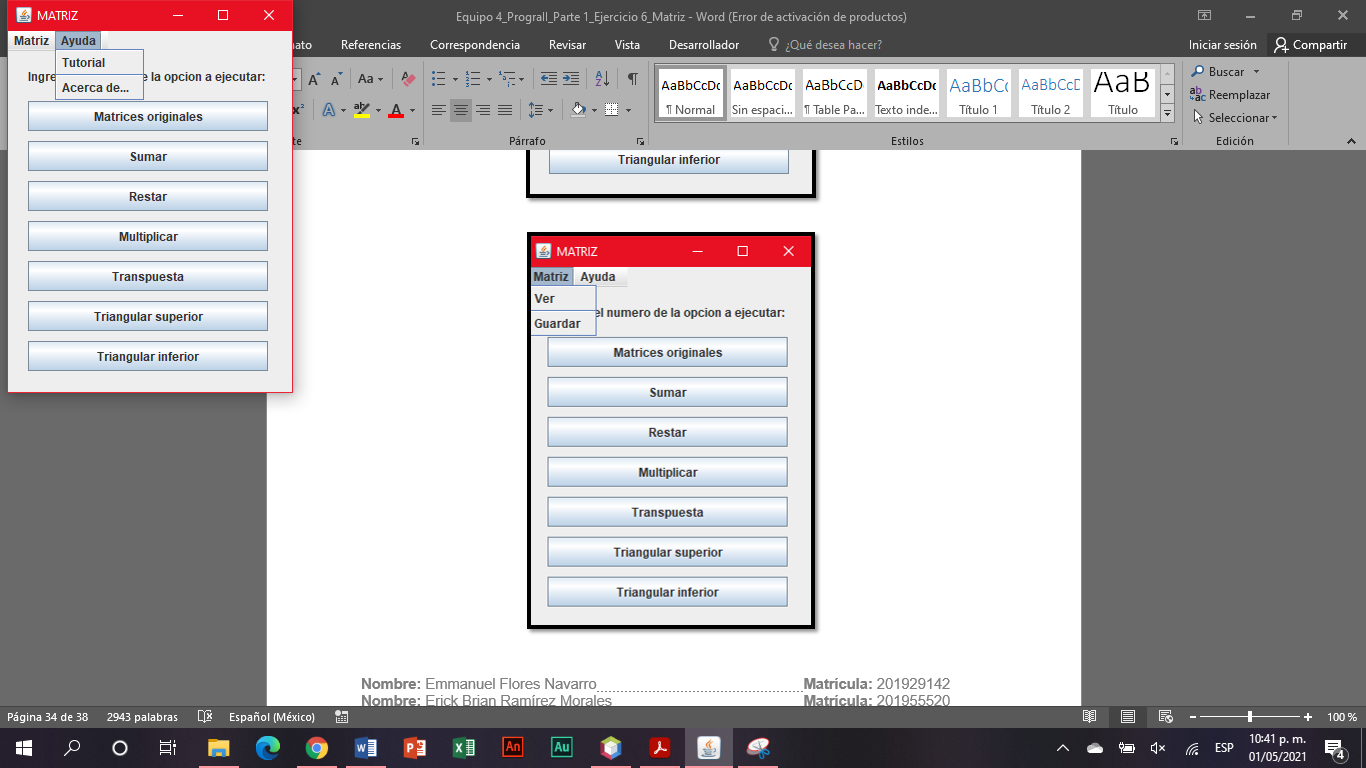
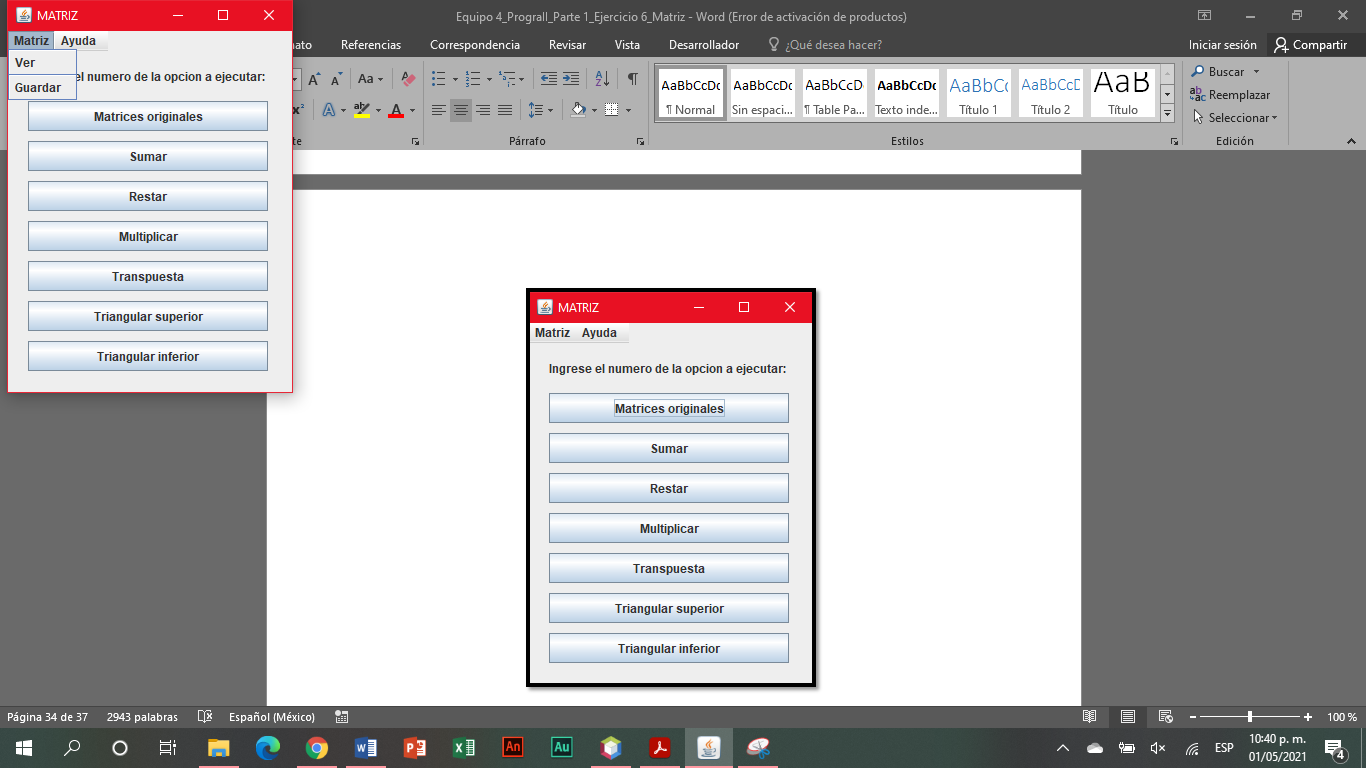


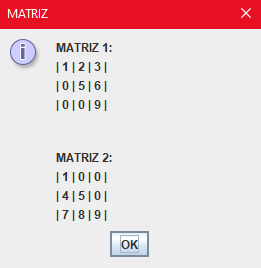


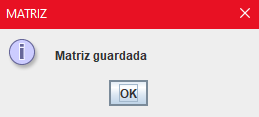


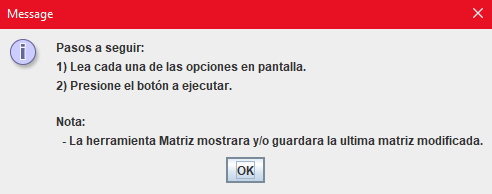


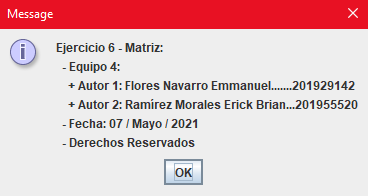


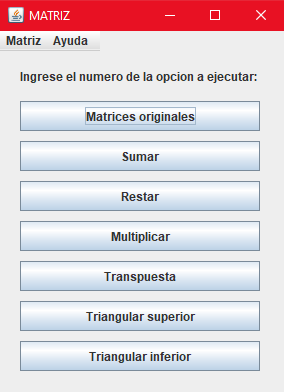


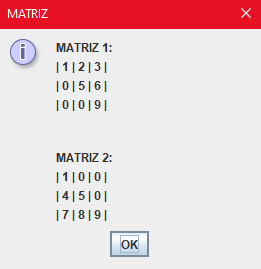


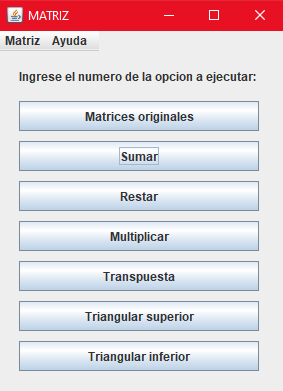


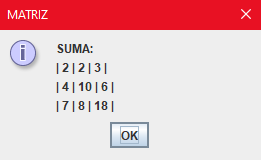


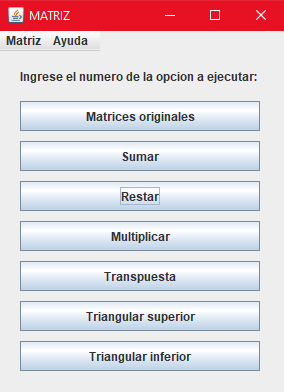




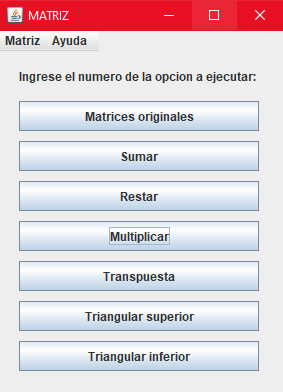




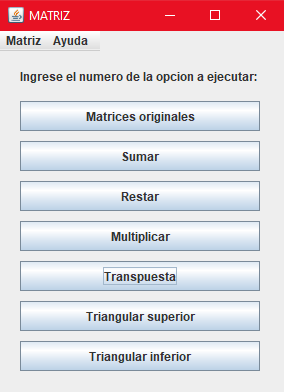


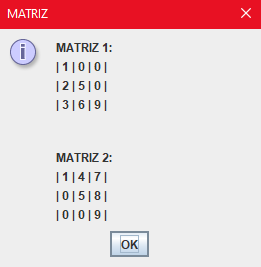


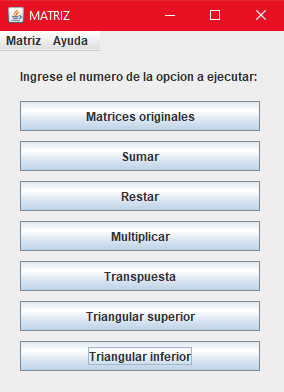
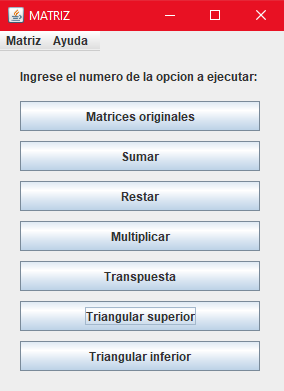


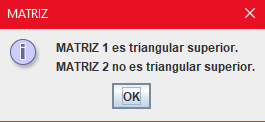
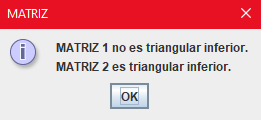


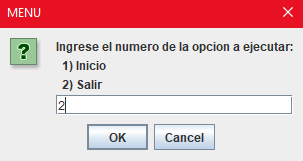


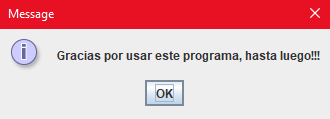








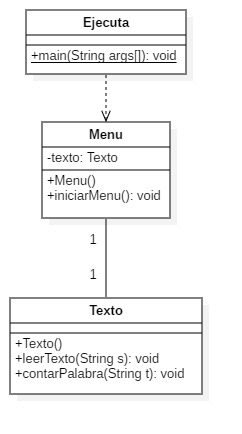


**Matriz.txt**



1. **Leer un Archivo.** Leer el contenido de un archivo y contar el número de caracteres, letras, caracteres especiales y verificar cuales son constantes, vocales y otros caracteres.

**DIAGRAMA UML**



**CÓDIGO DEL PROGRAMA**

**Ejecuta.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Contar letras, numeros, caracteres, vocales...de un archivo.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

public class Ejecuta {

public static void main(String args[]) {

//Variable de instancia

Menu menu = new Menu();

//Acceder a un método

menu.iniciarMenu();

}

}

//Fin de la clase

**Menu.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Contar letras, numeros, caracteres, vocales...de un archivo.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

public class Menu {

//Variable de instancia

private Texto texto = new Texto();

//Método constructor

public Menu() {}

//Metódo definido por el programador

public void iniciarMenu() {

//Atributos

String panel;

int opc;

//Inicio Menu

do{

panel = JOptionPane.showInputDialog(null,"Ingrese el numero de la opcion a ejecutar:"

+"\n 1) Inicio"

+"\n 2) Salir","MENU",JOptionPane.QUESTION\_MESSAGE);

//Iniciar excepcion

try {

//Ingresar un valor

opc = Integer.parseInt(panel);

} catch(Exception e) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+e.getMessage(),"ERROR",JOptionPane.WARNING\_MESSAGE);

opc = 0;

}

switch(opc) {

case 2 -> {

//Mostrar mensaje y cerrar panel

JOptionPane.showMessageDialog(null,"Gracias por usar este programa, hasta luego!!!");

System.exit(0);

}

case 1 -> {

//Acceder a un método

texto.leerTexto("Texto.txt");

opc = 2;

}

default -> {

//Mostrar mensaje

JOptionPane.showMessageDialog(null,"Opcion no valida","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}while(opc != 2);

//Fin Menu

}

}

//Fin de la clase

**Texto.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Contar letras, numeros, caracteres, vocales...de un archivo.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import java.io.FileNotFoundException;

import javax.swing.JOptionPane;

import java.io.BufferedReader;

import java.io.FileReader;

public class Texto {

//Método constructor

public Texto() {}

//Metódo definido por el programador

public void leerTexto(String s) {

//Iniciar excepcion

try {

//Variable de instancia

BufferedReader br = new BufferedReader(new FileReader(s));

//Atributos

String texto = "";

String lectura;

//Ciclo

while((lectura = br.readLine()) != null)

//Guardar texto

texto += lectura;

//Acceder a un método

contarPalabra(texto);

} catch(FileNotFoundException fnfe) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"Fichero no encontrado","ERROR",JOptionPane.WARNING\_MESSAGE);

} catch(Exception e) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,e.getMessage(),"ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

//Metódo definido por el programador

public void contarPalabra(String t) {

//Atributos

int mayusCont = 0, minusCont = 0, vocCont = 0, puntCont = 0, inexCont = 0, operCont = 0, espCont = 0, numCont = 0, otroCont = 0;

char mayuscula, minuscula, numero;

//Ciclo

for(int i=0; i<t.length(); i++) {

//Asignar datos

mayuscula = 'A';

minuscula = 'a';

numero = '0';

//Ciclo

for(int j=0; j<26; j++) {

//Validar dato

if(mayuscula == t.charAt(i))

mayusCont++;

if(minuscula == t.charAt(i))

minusCont++;

//Asignar valor

mayuscula++;

minuscula++;

}

//Ciclo

for(int j=0; j<10; j++) {

//Validar dato

if(numero == t.charAt(i))

numCont++;

//Asignar valor

numero++;

}

if(t.charAt(i)=='A' || t.charAt(i)=='E' || t.charAt(i)=='I' || t.charAt(i)=='O' || t.charAt(i)=='U' || t.charAt(i)=='a' || t.charAt(i)=='e' || t.charAt(i)=='i' || t.charAt(i)=='o' || t.charAt(i)=='u')

vocCont++;

if(t.charAt(i)==',' || t.charAt(i)==';' || t.charAt(i)==':' || t.charAt(i)=='.')

puntCont++;

if(t.charAt(i)=='?' || t.charAt(i)=='!')

inexCont++;

if(t.charAt(i)=='+' || t.charAt(i)=='-' || t.charAt(i)=='/' || t.charAt(i)=='\*' || t.charAt(i)=='(' || t.charAt(i)==')' || t.charAt(i)=='[' || t.charAt(i)==']' || t.charAt(i)=='{' || t.charAt(i)=='}')

operCont++;

if(t.charAt(i)==' ')

espCont++;

otroCont++;

}

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Texto: "+t

+"\n\nContenido total = "+otroCont

+"\nLetras mayusculas = "+mayusCont

+"\nLetras minusculas = "+minusCont

+"\nAbecedario = "+(mayusCont+minusCont)

+"\nVocales = "+vocCont

+"\nSignos de puntuacion = "+puntCont

+"\nSignos de interrogacion y exclamacion = "+inexCont

+"\nSignos de operacion = "+operCont

+"\nEspacios = "+espCont

+"\nNumeros = "+numCont

+"\nOtros = "+(otroCont-mayusCont-minusCont-puntCont-inexCont-operCont-espCont-numCont),"TEXTO",JOptionPane.INFORMATION\_MESSAGE);

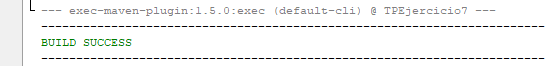
}

}

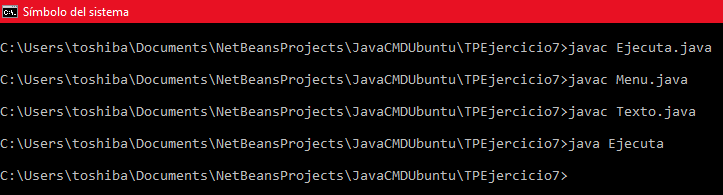
//Fin de la clase

**CAPTURAS DE PANTALLA**

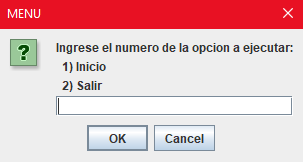
**NetBeans**

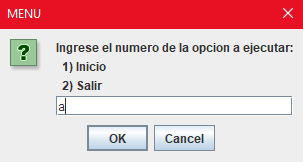


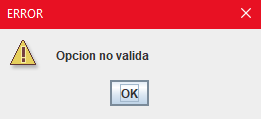
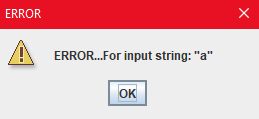
**CMD**

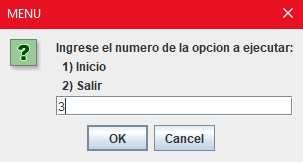


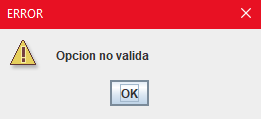
**Ventanas de ejecución**

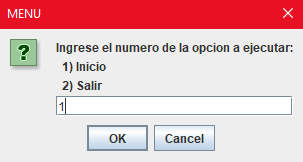


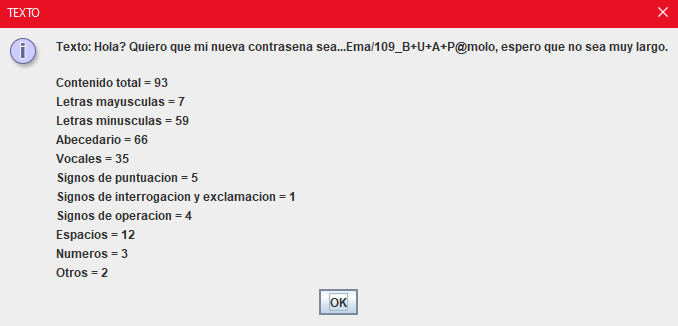


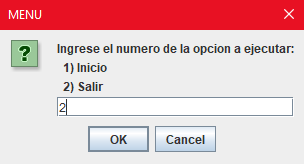


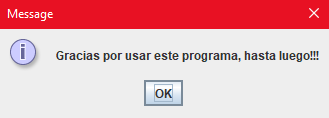




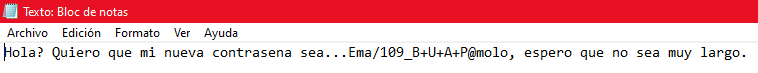






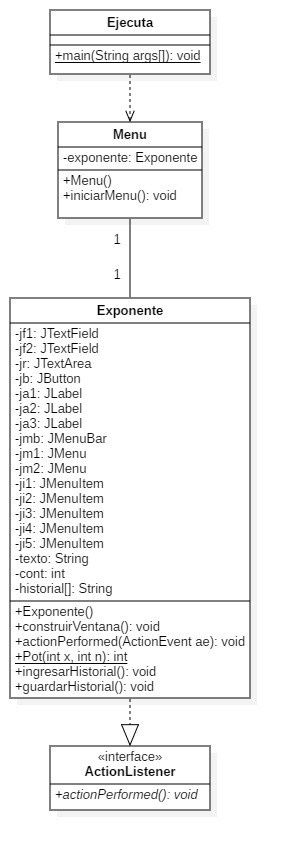


**Texto.txt**



1. **Recursividad.** Generar recursivamente el cálculo de un exponente y enviar a pantalla y archivo el resultado.

**DIAGRAMA UML**



**CÓDIGO DEL PROGRAMA**

**Ejecuta.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Calcular un exponente.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

public class Ejecuta {

public static void main(String args[]) {

//Variable de instancia

Menu menu = new Menu();

//Acceder a un método

menu.iniciarMenu();

}

}

//Fin de la clase

**Exponente.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Calcular un exponente.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import java.awt.event.\*;

import javax.swing.\*;

import java.awt.\*;

import java.io.\*;

public class Exponente implements ActionListener {

//Atributos

private JTextField jf1;

private JTextField jf2;

private JTextArea jr;

private JButton jb;

private JLabel ja1;

private JLabel ja2;

private JLabel ja3;

private JMenuBar jmb;

private JMenu jm1;

private JMenu jm2;

private JMenuItem ji1;

private JMenuItem ji2;

private JMenuItem ji3;

private JMenuItem ji4;

private JMenuItem ji5;

private String texto;

private int cont = 0;

//Arreglo

private String historial[] = new String[100];

//Método constructor

public Exponente() {}

//Metódo definido por el programador

public void construirVentana() {

//Variables de instancia

JPanel jp = new JPanel();

JFrame jf = new JFrame();

//Barra de herramientas

jmb = new JMenuBar();

jmb.setBounds(0, 0, 110, 20);

jf.add(jmb);

//Herramienta 1

jm1 = new JMenu("Historial");

jmb.add(jm1);

//Opcion 1

ji1 = new JMenuItem("Ver");

jm1.add(ji1);

ji1.addActionListener(this);

//Opcion 2

ji2 = new JMenuItem("Borrar");

jm1.add(ji2);

ji2.addActionListener(this);

jm1.addSeparator();

//Opcion 3

ji3 = new JMenuItem("Guardar");

jm1.add(ji3);

ji3.addActionListener(this);

//Herramienta 2

jm2 = new JMenu("Ayuda");

jmb.add(jm2);

//Opcion 1

ji4 = new JMenuItem("Tutorial");

jm2.add(ji4);

ji4.addActionListener(this);

jm2.addSeparator();

//Opcion 2

ji5 = new JMenuItem("Acerca de...");

jm2.add(ji5);

ji5.addActionListener(this);

//Etiqueta 1

ja1 = new JLabel("Ingrese la base:");

ja1.setBounds(10,30,150,30);

jf.add(ja1);

//Etiqueta 2

ja2 = new JLabel("Ingrese el exponente:");

ja2.setBounds(10,70,150,30);

jf.add(ja2);

//Etiqueta 3

ja3 = new JLabel("Resultado:");

ja3.setBounds(10,110,150,30);

jf.add(ja3);

//Campo de texto 1

jf1 = new JTextField();

jf1.setBounds(160,30,160,30);

jf.add(jf1);

//Campo de texto 2

jf2 = new JTextField();

jf2.setBounds(160,70,160,30);

jf.add(jf2);

//Area de texto

jr = new JTextArea();

jr.setBounds(160,110,160,30);

jf.add(jr);

jr.setEditable(false);

//Boton

jb = new JButton("Resultado");

jb.setBounds(120,160,100,30);

jf.add(jb);

//Presionar

jb.addActionListener(this);

//Ventana

jp.setBounds(20,20,350,250);

jp.setLayout(null);

jf.setLayout(null);

jf.setSize(350,250);

jf.add(jp);

jf.setVisible(true);

jf.setTitle("EXPONENTE");

jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

//Metódo definido por el programador

@Override

public void actionPerformed(ActionEvent ae) {

//Atributos

int x, y, r;

//Iniciar excepcion

try {

//Validar boton presionado

if(ae.getSource() == jb) {

//Ingresar valores

x = Integer.parseInt(jf1.getText());

y = Integer.parseInt(jf2.getText());

//Acceder a los métodos

r = Pot(x,y);

jr.setText(r+"");

ingresarHistorial();

}

if(ae.getSource() == ji1) {

//Ciclo

texto = "Historial:";

for(int i=0; i<cont; i++)

texto += "\n"+historial[i];

//Mostrar ventana

JOptionPane.showMessageDialog(null,texto);

}

if(ae.getSource() == ji2) {

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Historial borrado");

cont = 0;

}

if(ae.getSource() == ji3)

guardarHistorial();

if(ae.getSource() == ji4) {

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Pasos a seguir:"

+"\n1) Ingrese un valor en los primeros dos campos de texto."

+"\n2) Presione el boton Resultado."

+"\n\nNota:"

+"\n - Si no hay un valor numerico en los primeros dos campos de texto, el resultado sera un aviso de error.");

}

if(ae.getSource() == ji5) {

//Acceder a un método

JOptionPane.showMessageDialog(null,"Ejercicio 8 - Exponente:"

+"\n - Equipo 4:"

+"\n + Autor 1: Flores Navarro Emmanuel.......201929142"

+"\n + Autor 2: Ramírez Morales Erick Brian...201955520"

+"\n - Fecha: 07 / Mayo / 2021"

+"\n - Derechos Reservados");

}

} catch(NumberFormatException nfe) {

//Capturar e imprimir error

jr.setText("Faltan valores numericos");

} catch(Exception e) {

//Capturar e imprimir error

jr.setText("Error");

}

}

//Metódo definido por el programador

public static int Pot(int x, int n) {

//Validar valor

if(n==0)

//Retorno

return 1;

else

//Retorno

return(x\*Pot(x,n-1));

}

//Metódo definido por el programador

public void ingresarHistorial() {

historial[cont] = jf1.getText()+"^"+jf2.getText()+" = "+jr.getText();

cont++;

}

//Metódo definido por el programador

public void guardarHistorial() {

//Variables de instancia

FileWriter fw = null;

PrintWriter pw = null;

//Atributo

texto = "";

//Iniciar excepcion

try {

//Variables de instancia

fw = new FileWriter("Exponente.txt");

pw = new PrintWriter(fw);

//Ciclo

for(int i=0; i<cont; i++)

texto += "\n "+historial[i];

//Guardar mensajes y resultado

pw.println("\n ========== EXPONENTE ==========");

pw.println(texto);

pw.println("\n ===============================\n");

} catch(Exception e1) {

//Capturar e imprimir error

JOptionPane.showMessageDialog(null,"ERROR..."+e1.getMessage()+"...","ERROR",JOptionPane.WARNING\_MESSAGE);

} finally {

//Iniciar excepcion

try {

//Validar dato

if (null != fw)

fw.close();

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Historial guardado");

} catch(Exception e2) {

//Capturar e imprimir error

JOptionPane.showMessageDialog(null,"ERROR..."+e2.getMessage()+"...","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}

}

//Fin de la clase

**Menu.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Calcular un exponente.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

public class Menu {

//Variable de instancia

private Exponente exponente = new Exponente();

//Método constructor

public Menu() {}

//Metódo definido por el programador

public void iniciarMenu() {

//Atributos

String panel;

int opc;

//Inicio Menu

do{

panel = JOptionPane.showInputDialog(null,"Ingrese el numero de la opcion a ejecutar:"

+"\n 1) Inicio"

+"\n 2) Salir","MENU",JOptionPane.QUESTION\_MESSAGE);

//Iniciar excepcion

try {

//Ingresar un valor

opc = Integer.parseInt(panel);

} catch(Exception e) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+e.getMessage(),"ERROR",JOptionPane.WARNING\_MESSAGE);

opc = 0;

}

switch(opc) {

case 2 -> {

//Mostrar mensaje y cerrar panel

JOptionPane.showMessageDialog(null,"Gracias por usar este programa, hasta luego!!!");

System.exit(0);

}

case 1 -> {

//Acceder a un método

exponente.construirVentana();

opc = 2;

}

default -> {

//Mostrar mensaje

JOptionPane.showMessageDialog(null,"Opcion no valida","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}while(opc != 2);

//Fin Menu

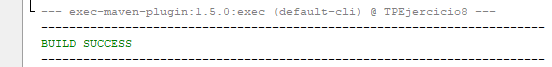
}

}

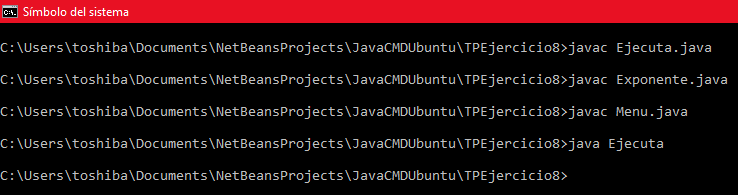
//Fin de la clase

**CAPTURAS DE PANTALLA**

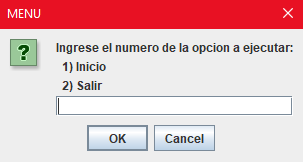
**NetBeans**

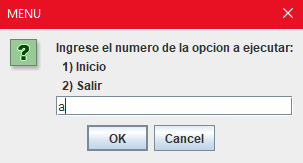


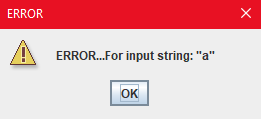
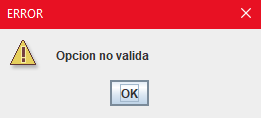
**CMD**

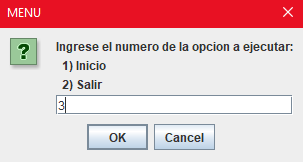


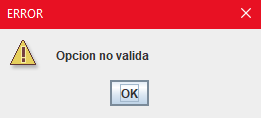
**Ventanas de ejecución**

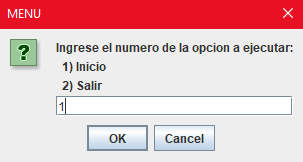


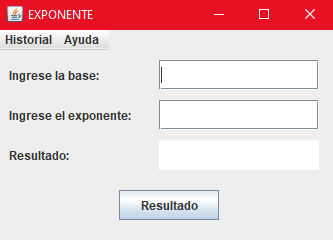


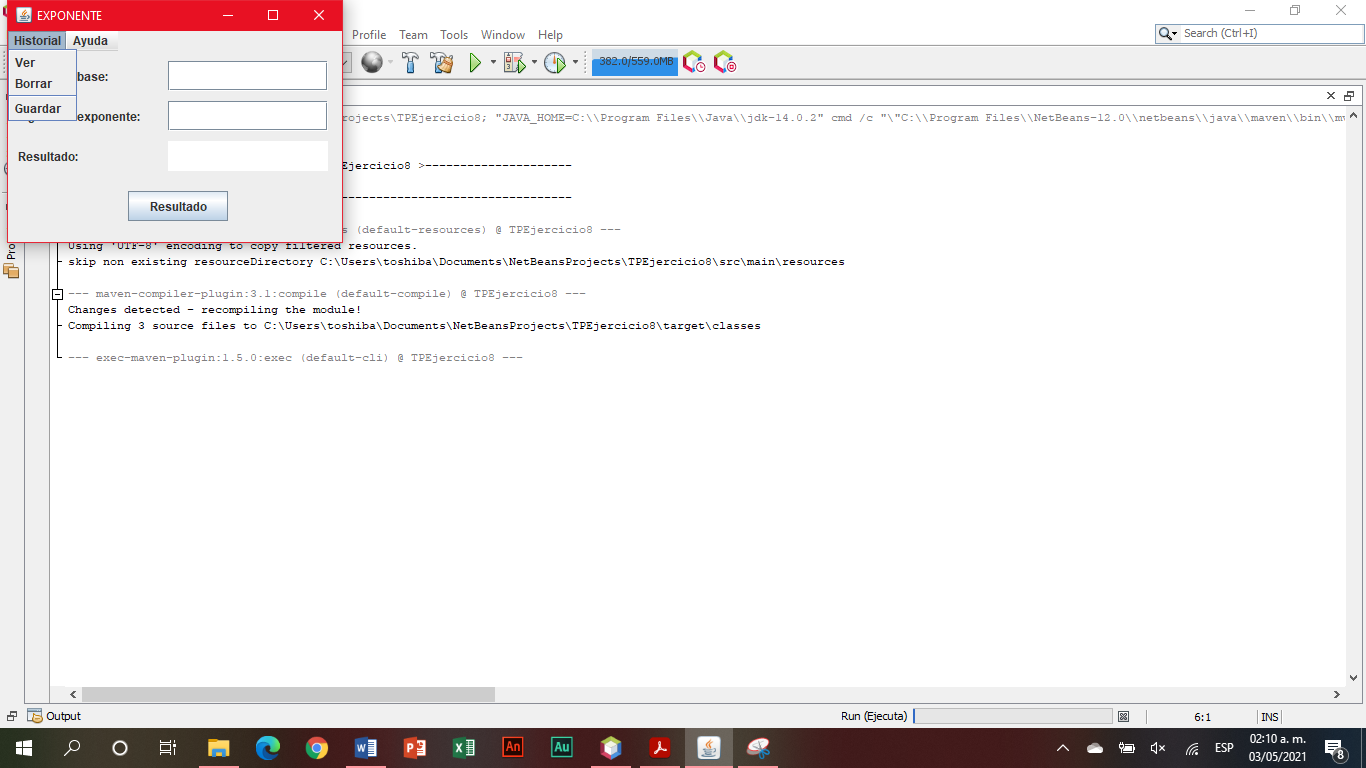
 

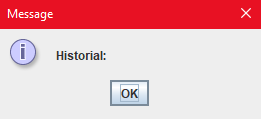


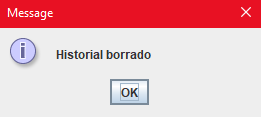


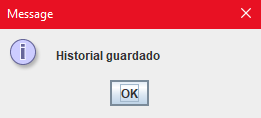


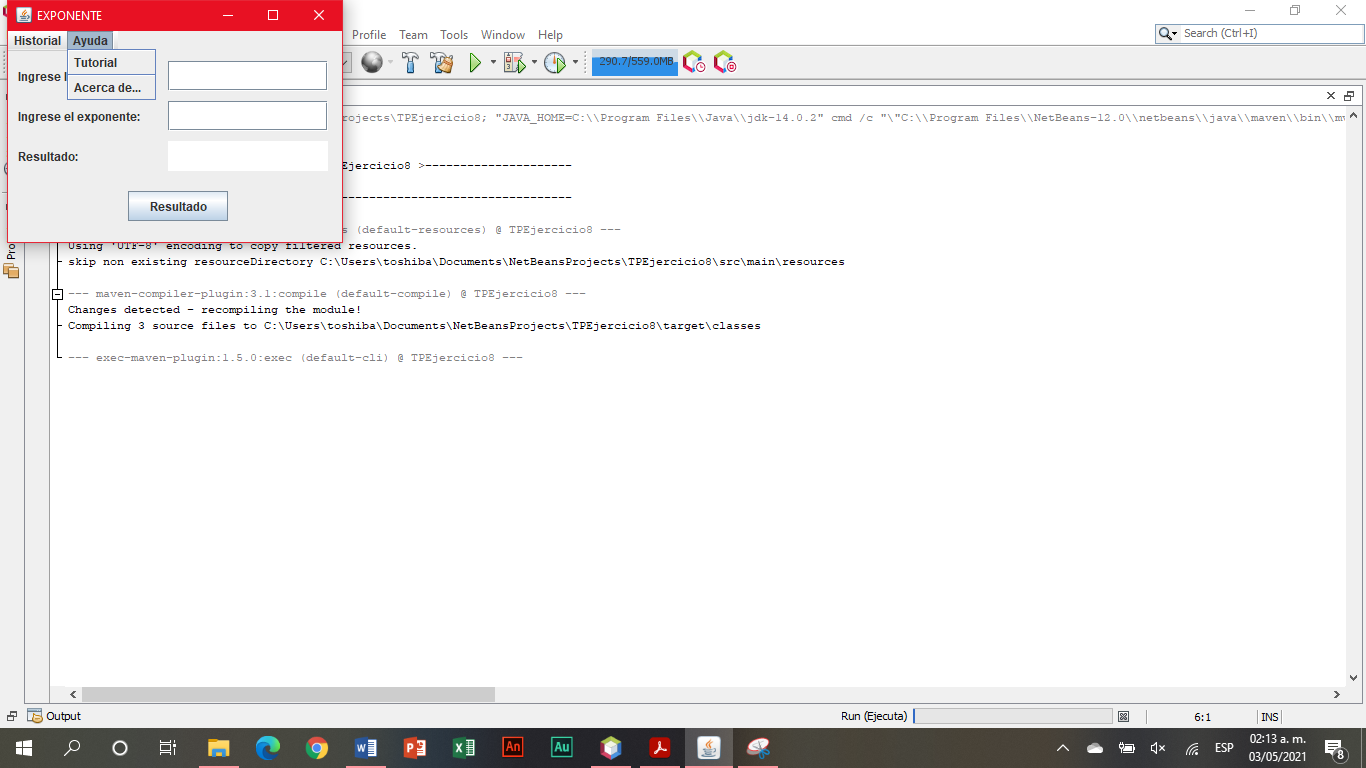


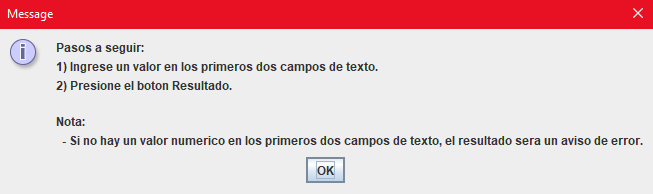


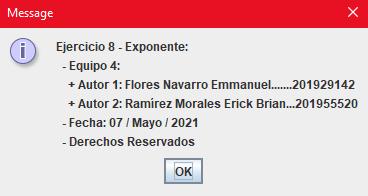


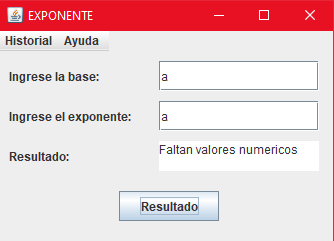


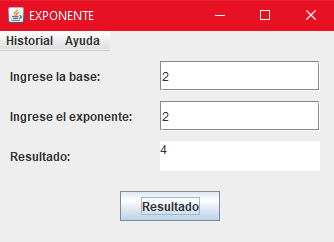


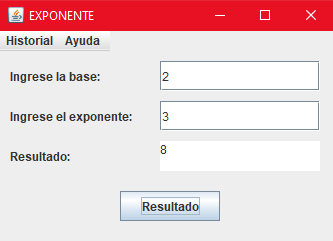


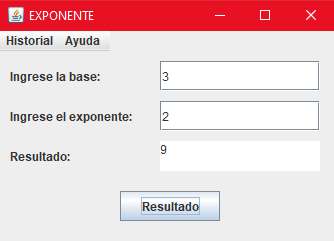


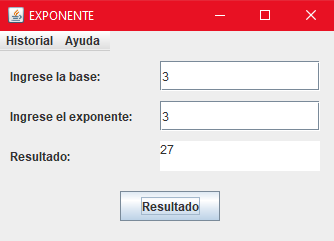


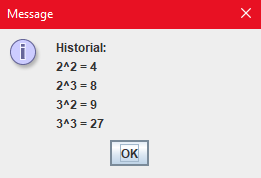


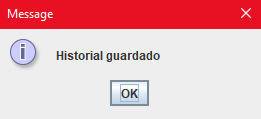


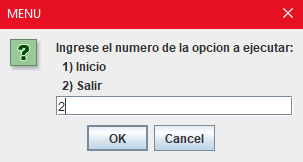


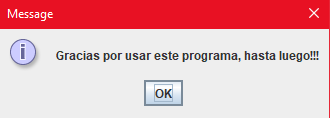




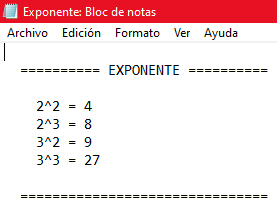






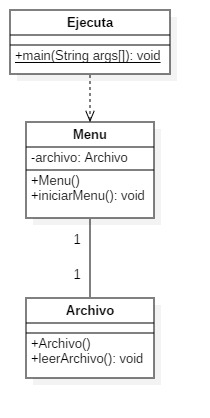


**Exponente.txt**



1. **Atributos de un Archivo.** Programa que lee una extensión de archivo y muestre el contenido con nombre del archivo, fecha, tamaño y ruta.

**DIAGRAMA UML**



**CÓDIGO DEL PROGRAMA**

**Archivo.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Mostrar la extension de un archivo.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import java.util.GregorianCalendar;

import javax.swing.JOptionPane;

import java.util.Calendar;

import java.util.Date;

import java.io.\*;

public class Archivo {

//Método constructor

public Archivo() {}

//Metódo definido por el programador

public void leerArchivo() {

//Variables de instancia

File f = new File("Archivo.txt");

Date d = new Date(f.lastModified());

Calendar c = new GregorianCalendar();

//Acceder a un método

c.setTime(d);

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Nombre: "+f.getName()

+"\nFecha: "+Integer.toString(c.get(Calendar.DATE))+"/"

+Integer.toString(c.get(Calendar.MONTH))+"/"

+Integer.toString(c.get(Calendar.YEAR))+" - "

+Integer.toString(c.get(Calendar.HOUR\_OF\_DAY))+":"

+Integer.toString(c.get(Calendar.MINUTE))

+"\nLongitud: "+f.length()

+"\nRuta: "+f.getAbsolutePath(),"ARCHIVO",JOptionPane.INFORMATION\_MESSAGE);

}

}

//Fin de la clase

**Ejecuta.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Mostrar la extension de un archivo.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

public class Ejecuta {

public static void main(String args[]) {

//Variable de instancia

Menu menu = new Menu();

//Acceder a un método

menu.iniciarMenu();

}

}

//Fin de la clase

**Menu.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Mostrar la extension de un archivo.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

public class Menu {

//Variable de instancia

private Archivo archivo = new Archivo();

//Método constructor

public Menu() {}

//Metódo definido por el programador

public void iniciarMenu() {

//Atributos

String panel;

int opc;

//Inicio Menu

do{

panel = JOptionPane.showInputDialog(null,"Ingrese el numero de la opcion a ejecutar:"

+"\n 1) Inicio"

+"\n 2) Salir","MENU",JOptionPane.QUESTION\_MESSAGE);

//Iniciar excepcion

try {

//Ingresar un valor

opc = Integer.parseInt(panel);

} catch(Exception e) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+e.getMessage(),"ERROR",JOptionPane.WARNING\_MESSAGE);

opc = 0;

}

switch(opc) {

case 2 -> {

//Mostrar mensaje y cerrar panel

JOptionPane.showMessageDialog(null,"Gracias por usar este programa, hasta luego!!!");

System.exit(0);

}

case 1 -> {

//Acceder a un método

archivo.leerArchivo();

opc = 2;

}

default -> {

//Mostrar mensaje

JOptionPane.showMessageDialog(null,"Opcion no valida","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}while(opc != 2);

//Fin Menu

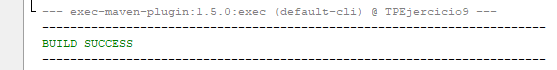
}

}

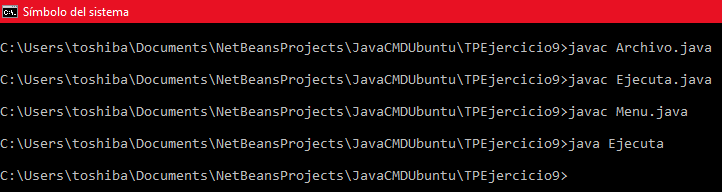
//Fin de la clase

**CAPTURAS DE PANTALLA**

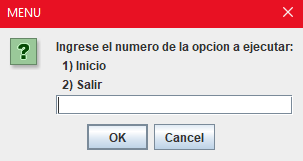
**NetBeans**

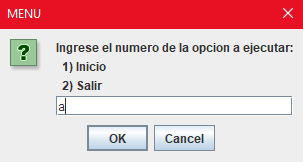


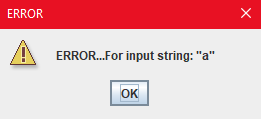
**CMD**

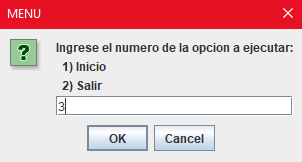


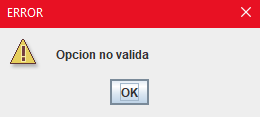
**Ventanas de ejecución**

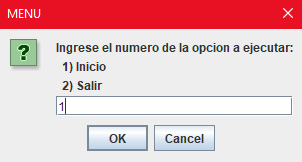


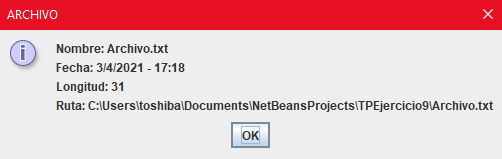


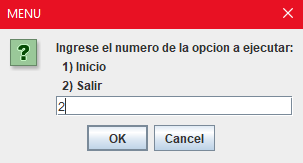


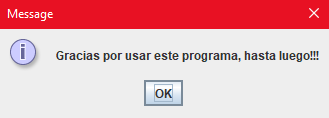




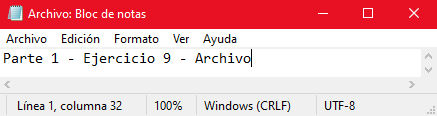






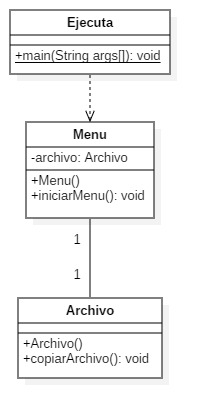


**Archivo.txt**



1. **Copiar Archivos.** Crear un programa que Lea el contenido de un archivo y lo copie a otro archivo solicitando el nombre del archivo.

**DIAGRAMA UML**



**CÓDIGO DEL PROGRAMA**

**Archivo.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Copiar un archivo en otro.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

import java.io.\*;

public class Archivo {

//Método constructor

public Archivo() {}

//Metódo definido por el programador

public void copiarArchivo() {

//Variables de instancia

InputStream is = null;

OutputStream os = null;

//Atributos

String nombre1, nombre2;

int c;

//Iniciar excepcion

try {

//Mostrar ventana

nombre1 = JOptionPane.showInputDialog(null,"Ingrese el nombre del archivo a copiar:","ARCHIVO",JOptionPane.QUESTION\_MESSAGE);

nombre1 += ".txt";

//Variable de instancia

is = new FileInputStream(nombre1);

//Mostrar ventana

nombre2 = JOptionPane.showInputDialog(null,"Ingrese el nombre del nuevo archivo:","ARCHIVO",JOptionPane.QUESTION\_MESSAGE);

nombre2 += ".txt";

//Variable de instancia

os = new FileOutputStream(nombre2);

//Mostrar ventana

JOptionPane.showMessageDialog(null,"Archivo copiado y guardado");

//Ciclo

while((c = is.read()) != -1) {

os.write(c);

}

} catch(FileNotFoundException fnfe) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"Archivo no encontrado","ERROR",JOptionPane.WARNING\_MESSAGE);

} catch(IOException ioe) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"Error de E/S","ERROR",JOptionPane.WARNING\_MESSAGE);

}

//Validar dato

if(is != null)

//Iniciar excepcion

try {

//Acceder a un método

is.close();

} catch(IOException ioe) {}

if(os != null)

//Iniciar excepcion

try {

//Acceder a un método

os.close();

} catch(IOException ioe) {}

}

}

//Fin de la clase

**Ejecuta.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Copiar un archivo en otro.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

public class Ejecuta {

public static void main(String args[]) {

//Variable de instancia

Menu menu = new Menu();

//Acceder a un método

menu.iniciarMenu();

}

}

//Fin de la clase

**Menu.java**

// Fecha: 07 / Mayo / 2021

// Objetivo: Copiar un archivo en otro.

// Elaboro Equipo 4:

// 1) Flores Navarro Emmanuel

// 2) Ramírez Morales Erick Brian

//Inicio de la clase

import javax.swing.JOptionPane;

public class Menu {

//Variable de instancia

private Archivo archivo = new Archivo();

//Método constructor

public Menu() {}

//Metódo definido por el programador

public void iniciarMenu() {

//Atributos

String panel;

int opc;

//Inicio Menu

do{

panel = JOptionPane.showInputDialog(null,"Ingrese el numero de la opcion a ejecutar:"

+"\n 1) Inicio"

+"\n 2) Salir","MENU",JOptionPane.QUESTION\_MESSAGE);

//Iniciar excepcion

try {

//Ingresar un valor

opc = Integer.parseInt(panel);

} catch(Exception e) {

//Capturar y mostrar error

JOptionPane.showMessageDialog(null,"ERROR..."+e.getMessage(),"ERROR",JOptionPane.WARNING\_MESSAGE);

opc = 0;

}

switch(opc) {

case 2 -> {

//Mostrar mensaje y cerrar panel

JOptionPane.showMessageDialog(null,"Gracias por usar este programa, hasta luego!!!");

System.exit(0);

}

case 1 -> {

//Acceder a un método

archivo.copiarArchivo();

opc = 2;

}

default -> {

//Mostrar mensaje

JOptionPane.showMessageDialog(null,"Opcion no valida","ERROR",JOptionPane.WARNING\_MESSAGE);

}

}

}while(opc != 2);

//Fin Menu

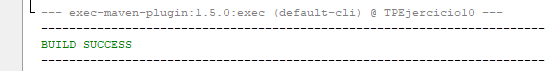
}

}

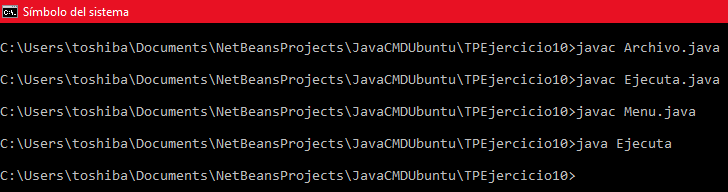
//Fin de la clase

**CAPTURAS DE PANTALLA**

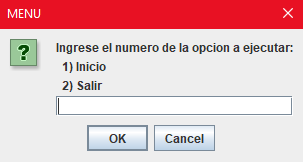
**NetBeans**

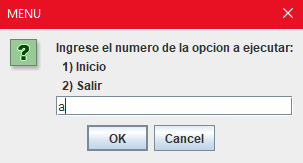


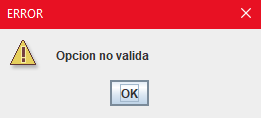
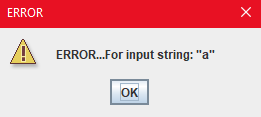
**CMD**

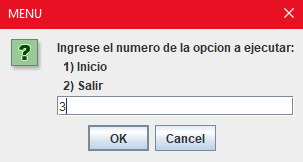


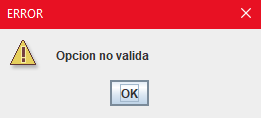
**Ventanas de ejecución**

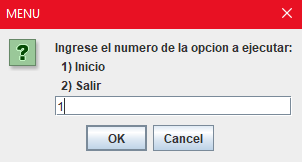




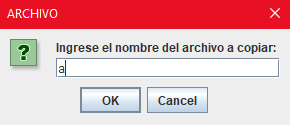


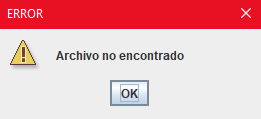


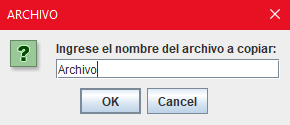


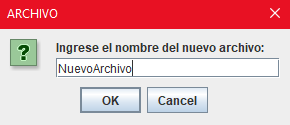


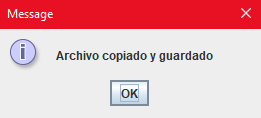


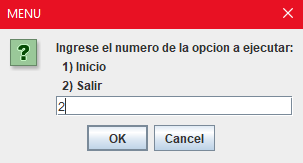


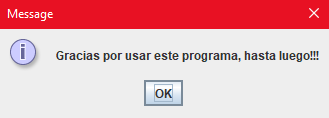




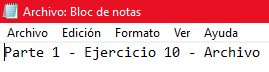




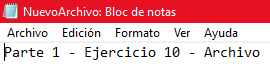




**Archivo.txt**



**NuevoArchivo.java**



# PARTE II- Gráficos (VALOR 4 PUNTOS)

1. Generar el factorial en modo gráfico con frame o jframe
2. Realizar un programa que genere números aleatorios y distintos, los cuales los muestre en pantalla en un Frame, los ordene utilizando un método de ordenación y los envié ya ordenados en un archivo.
3. Realizar en modo grafico con jframe, utilizando componentes gráficos: menús, cajas de texto, dialogo, etc. para el siguiente problema

Una institución requiere almacenar información de los tipos de usuario que son alumnos y docentes, genere las clases necesarias utilice clases abstractas ya que todos los usuario tiene nombre, dirección, teléfono y todos pueden mostrar sus datos.

Los docentes tienen un id-profesor que los identifica a cada uno

Los alumnos tienen una matrícula, que los identifica a cada uno y además se requiere datos de su promedio, nombre de la licenciatura, fecha de ingreso

Se requiere el siguiente menú

Usuario Ordenar Buscar Salir Alumno Ascendente Buscar Alumno por promedio Docente Descendente Buscar por matricula

Guardar

* Los docentes y alumnos serán almacenados en una lista ligada respectivamente de acuerdo a su matricula o id-docente y después almacenados en archivo cuando soliciten guardar información
* Las operaciones de ordenación será para ambos por lo cual se especifica docente o alumno.