



## TEST RESULTS

**Alumno:** Edgar Valentin Ruiz Padilla.

**Departamento:** Ciencias Computacionales.

**Línea de Investigación:** Ingeniería del Software.

Instrucciones:

1. Usando PSP0.1, escriba un programa para contar líneas de código (LOC).
  - a. Tamaño total del programa.
  - b. Tamaño total de cada una de las partes del programa (clases, funciones o procedimientos).
  - c. El número de elementos (o métodos) en cada parte.
2. Producir e imprimir:
  - a. Un solo recuento para todo el programa.
  - b. Tamaño y recuento de artículos para cada pieza junto con el nombre de la pieza.

El programa se dividió en 3 clases:

Clase Main: solicita al usuario la ubicación del archivo al cual serán contadas las líneas de código y será mostrado en consola el resultado

```

/*****
/* Program Assignment: PSP0.1
/* Name: Edgar Valentin Ruiz Padilla
/* Date: 24/09/2021
/* Description: Programa Contador de lineas
*****/

import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String Ubicacion=null;
        ArrayList<String> Codigo = new ArrayList();
        Scanner entrada=new Scanner(System.in);
        System.out.println("Programa para contar lineas de codigo, se contarán los siguientes aspectos:\r\n");
        System.out.println("Tipos de Metodos, Clases,\r\n");
        System.out.println("Teclear la ruta de ubicacion del archivo que contiene el codigo");
        Ubicacion=entrada.nextLine();
        //
        //Ubicacion="C:\\Users\\Edgar\\Desktop\\Texto prueba.txt";
        //
        FileToString ConverString = new FileToString();
        try {
            Codigo=ConverString.ConverString(Ubicacion);
            //Codigo = ConverString.ConverString(Ubicacion);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        System.out.println("El codigo leido es el siguiente:");
        for(int b=0;b<Codigo.size();b++) {

```

```
        System.out.println(Codigo.get(b));
    }
    entrada.close();
    System.out.println("Las lineas totales son:"+Codigo.size());

    Clasificador Clasificar= new Clasificador();

    int Conteo[]=Clasificar.ClasificadorT(Codigo);
    System.out.println("Se obtuvieron los siguientes resultados del conteo de lineas de codigo (LOC):");
    System.out.println("El programa tiene "+ Conteo[0]+" lineas de comentarios");
    System.out.println("El programa tiene "+ Conteo[1]+" clases");
    System.out.println("El programa tiene "+ Conteo[2]+" Metodos");
    System.out.println("El programa tiene "+ Conteo[3]+" Metodos get-set");
    System.out.println("El programa tiene "+ Conteo[4]+" Lineas de entrada o salida I/O");
    System.out.println("El programa tiene "+ Conteo[5]+" Lineas de Control");
    System.out.println("El programa tiene "+ Conteo[6]+" Lineas de Logica");
    System.out.println("El programa tiene "+ Conteo[7]+" lineas vacias");
}
}
```

La clase #2 es la llamada FileToString la cual se encarga de guardar en un array el texto del programa a contar las líneas:

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.ArrayList;

public class FileToString {
    public ArrayList<String> ConverString (String UbicacionT) throws FileNotFoundException {
        File archivo = null;
        FileReader fr = null;
        BufferedReader br = null;
        String Temps=null;
        ArrayList<String> ConvertedToString = new ArrayList();
        int Tempi=0;
        try {
            // Apertura del fichero y creacion de BufferedReader para poder
            // hacer una lectura comoda (disponer del metodo readLine()).
            archivo = new File (UbicacionT);
            fr = new FileReader (archivo);
            br = new BufferedReader(fr);
            // Lectura del fichero
            while((Temps=br.readLine())!=null)
                //Contador de lineas
                {
                    ConvertedToString.add(Temps);
                    //System.out.println("Se guarda el valor"+ ConvertedToString.get(Tempi));
                    Tempi++;
                }
        }
        catch(Exception e){
            e.printStackTrace();
        }finally{
            // En el finally cerramos el fichero, para asegurarnos
            // que se cierra tanto si todo va bien como si salta
            // una excepcion.
            try{
                if( null != fr ){
                    fr.close();
                }
            }catch (Exception e2){
                e2.printStackTrace();
            }
        }
    }
}
```



```
        return ConvertedToString;
    }
}
```

Por último se tiene la clase “Clasificador” la cual identificara que tipo de línea de código se encuentra en cada renglón

```
import java.util.ArrayList;

public class Clasificador {
    public static int[] ClasificadorT (ArrayList<String> codigo) {
        //ArrayList<String> ClasificadoRes = new ArrayList();
        int[] Clasificado = new int[] { 0,0,0,0,0,0,0};
        int comentarios=0,LineaVacía=0, CicloFor=0,CicloWhile=0, Cicloswitch=0, logica=0
            , CicloIf=0, variables=0,constructor=0,cierre=0, clases=0, GetSet=0, MetodoIO=0;

        boolean Comment= false;
        boolean LineaContada=false;
        for(int b=0;b<codigo.size();b++) {
            //System.out.println(codigo.get(b));
            if(codigo.get(b).contains("*/")&&LineaContada==false) {
                comentarios++;
                Comment=false;
                LineaContada=true;
            }
            if(codigo.get(b).contains("/*")&&LineaContada==false) {
                comentarios++;
                LineaContada=true;
            }
            if(codigo.get(b).contains("/")&&LineaContada==false) {
                comentarios++;
                Comment=true;
                LineaContada=true;
            }
            if(LineaContada==false&&Comment==true) {
                comentarios++;
                LineaContada=true;
            }
            if(codigo.get(b).contains("for")&&LineaContada==false) {
                CicloFor++;
                LineaContada=true;
            }
            if(codigo.get(b).contains("while")&&LineaContada==false) {
                CicloWhile++;
                LineaContada=true;
            }
            if(codigo.get(b).contains("switch")&&LineaContada==false) {
                Cicloswitch++;
                LineaContada=true;
            }
            if(codigo.get(b).contains("if")&&LineaContada==false) {
                CicloIf++;
                LineaContada=true;
            }
            if(codigo.get(b).contains("class")&&LineaContada==false){
                clases++;
                LineaContada=true;
            }
            if((codigo.get(b).contains("get")||codigo.get(b).contains("set"))&&LineaContada==false) {
                GetSet++;
                LineaContada=true;
            }
            if((codigo.get(b).contains("public")||codigo.get(b).contains("private")||codigo.get(b).contains("protected"))&&LineaContada==false){
                constructor++;
                LineaContada=true;
            }
        }
    }
}
```

```
if((codigo.get(b).contains("Scanner"))||codigo.get(b).contains("iterate"))||codigo.get(b).contains("new"))&&LineaContada==false){
    MetodoIO++; //Metodos entrada salida I/O---Scanner--iterate-----new
    LineaContada=true;
}
if((codigo.get(b).contains("int "))||codigo.get(b).contains("double "))||codigo.get(b).contains("float ")||codigo.get(b).contains("char ")||codigo.get(b).contains("String ")||codigo.get(b).contains("boolean ")||codigo.get(b).contains("char "))&&LineaContada==false) {
    variables++;
    LineaContada=true;
}
if((codigo.get(b).contains(";")||codigo.get(b).contains("try"))&&LineaContada==false){
    logica++;
    LineaContada=true;
}
if(codigo.get(b).contains("}")&&LineaContada==false){
    cierre++;
    LineaContada=true;
}
if(codigo.get(b).contains("")&&LineaContada==false){
    LineaVacía++;
    LineaContada=true;
}
LineaContada=false;
}
Clasificado[0]=comentarios; //Lineas de comentarios
Clasificado[1]=clases; //Cantidad de clases
Clasificado[2]=constructor; //Metodos
Clasificado[3]=GetSet; //Metodos get-set
Clasificado[4]=MetodoIO; //Metodos entrada salida I/O---Scanner--iterate-----new
Clasificado[5]=CicloFor+CicloWhile+Cicloswitch+CicloIf; //Lineas de Control
Clasificado[6]=variables+cierre+logica; //Lineas de Logica
Clasificado[7]=LineaVacía; //Lineas Vacías

return Clasificado;
}
}
```

Se realiza la prueba realizando conteo del programa 1:

```
Programa para contar líneas de código, se contarán los siguientes aspectos:

Tipos de Metodos, Clases,

Teclear la ruta de ubicación del archivo que contiene el código
C:\\Users\\Edgar\\Desktop\\Texto prueba.txt
El código leído es el siguiente:
import java.util.Scanner;

//Programa 1 Cálculo de Media y desviación estándar de conjuntos de números
//Lectura de datos

public class Main {

    public static void main(String[] args) {
        //Declaración de Variables
        int Ntemp=0;
        String Entrada="";
        Boolean VEntrada=false;
        Scanner entrada=new Scanner(System.in);
        System.out.println("Programa para calcular la media y la desviación estándar de números reales.");

        while(VEntrada!=true) {
            System.out.println("Ingresa la cantidad de números a considerar");
            Entrada=entrada.nextLine();
            VEntrada=Entrada.matches("[+]?\\d*(\\.\\d+)?");
        }
        Ntemp=Integer.parseInt(Entrada);
        double [] DB= new double [Ntemp];
        VEntrada=false;
        for(int b=0;b<Ntemp;b++) {
            while(VEntrada!=true) {
                int bT=b+1;
                System.out.println("Ingresa el número " + bT);
                Entrada=entrada.nextLine();
                VEntrada=Entrada.matches("[+]?\\d*(\\.\\d+)?");
            }
            DB[b]=Double.parseDouble(Entrada);
            VEntrada=false;
        }
    }
}
```

```
        entrada.close();
        Operaciones operacion = new Operaciones();
        double Media =operacion.MediaAritmetica(DB);
        System.out.println("La Media Aritmetica es: "+ Media);
        double Desviacion =operacion.DesviacionEstandar(DB, Media);
        System.out.println("La Desviación Estandar es: "+ Desviacion);
    }

}

public class Operaciones {

    public double MediaAritmetica (double Numeros[]) {
        double Media=0;
        double Num[]=Numeros;
        int Total=Num.length;
        for(int M=0;M<Total; M++) {
            Media=Num[M]+Media;
        }
        Media=Media/Total;
        return Media;
    }
    public double DesviacionEstandar (double Numeros[],double Media) {
        double MediaT=Media,Desviacion=0, Temp=0;
        double Num[]=Numeros;
        int Total=Num.length;
        for(int M=0;M<Total; M++) {
            Temp=Temp+Math.pow((Num[M]-MediaT),2);
        }
        Desviacion=Temp/(Total-1);
        Desviacion=Math.sqrt(Desviacion);
        return Desviacion;
    }
}
```

Las líneas totales son:70

Se obtuvieron los siguientes resultados del conteo de líneas de código (LOC):

El programa tiene 3 líneas de comentarios

El programa tiene 2 clases

El programa tiene 3 Metodos

El programa tiene 0 Metodos get-set

El programa tiene 4 Líneas de entrada o salida I/O

El programa tiene 5 Líneas de Control

El programa tiene 43 Líneas de Logica

El programa tiene 10 líneas vacías