



# **MANUAL TÉCNICO**

## **Práctica#1**

Ángel Andrés Godoy Valdéz

Laboratorio IPC 1 Sección "B"

Carnet: 202113539

INTRODUCCIÓN .....	3
REQUERIMIENTOS TÉCNICOS .....	3
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO .....	3
Java .....	3
Intelij IDEA .....	3
APLICACIÓN .....	4
Clases.....	4
CalculadoraJava .....	4
Operaciones .....	4
Menu .....	4
Menú Principal (Clase Operaciones) .....	4
OpAritmética .....	6
Suma, Resta, Multiplicación, División y Potencia.....	6
Trigonómicas .....	9
Seno.....	9
Coseno .....	10
Tangente .....	12
Estadísticas .....	12
Promedio y Media .....	13
Moda .....	14
Varianza y Desviación Estándar.....	14
Cálculo .....	16

## **INTRODUCCIÓN**

Este manual describe los procesos detrás de las operaciones realizadas en la aplicación “calculadora” en formato java realizada como practica para el Laboratorio del curso Introducción a la Programación de Computadoras 1, de igual de forma ilustra el pensamiento detrás del código e intenta desarmarlo de forma comprensiva para el entendimiento de otros desarrolladores.

## **REQUERIMIENTOS TÉCNICOS**

Sistema Operativo: Windows 7/Vista/8/8.1/10/11

Aplicaciones: Java Development Kit (JDK) 12.0-actual

## **HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

### **Java**

Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones. Los nuevos e innovadores productos y servicios digitales diseñados para el futuro también siguen basándose en Java.

Aunque la mayoría de aplicaciones Java modernas combinan el tiempo de ejecución y la aplicación de Java, todavía existen algunas aplicaciones e incluso sitios web que no funcionan sin instalar Java para escritorio. El sitio web Java.com está pensado para consumidores que todavía necesitan Java en sus aplicaciones de escritorio, sobre todo las aplicaciones que tienen como destino Java 8.

### **IntelliJ IDEA**

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: edición para la comunidad y edición comercial.

Cada aspecto de IntelliJ IDEA ha sido diseñado para maximizar la productividad del desarrollador. Juntos, la asistencia de codificación inteligente y el diseño ergonómico hacen que el desarrollo no solo sea productivo sino también agradable.

Después de que IntelliJ IDEA haya indexado su código fuente, ofrece una experiencia increíblemente rápida e inteligente al brindar sugerencias relevantes en cada contexto: finalización de código instantánea e inteligente, análisis de código sobre la marcha y herramientas de refactorización confiables. Las herramientas de misión crítica, como los

sistemas de control de versiones integrados y una amplia variedad de lenguajes y marcos compatibles, están todos a mano, sin complicaciones de complementos incluidos. Mientras que la finalización básica sugiere nombres de clases, métodos, campos y palabras clave dentro del ámbito de visibilidad, la finalización inteligente sugiere solo los tipos que se esperan en el contexto actual.

## **APLICACIÓN**

### **Clases**

Para la realización de esta práctica decidí separar el proyecto por 3 clases para que la manipulación del código fuera más conveniente.

Cada una de estas clases tiene la facultad de invocarse entre sí y los métodos que implemente en cada una de ellas ahorran varias líneas de código.

### **CalculadoraJava**

Dentro de esta clase tenemos el método main y es la clase ejecutable que llama al resto de clases para realizar los diferentes procesos, dentro de ella se puede apreciar únicamente una instancia de la clase auxiliar Operaciones, esta instancia llamada operador. MenuPrincipal invoca las funciones y métodos que acompañan al llamado menú principal el cual contiene el resto de atributos.

### **Operaciones**

De lejos la clase más robusta ya que la mayoría de procesos se llevan a cabo dentro de ella.

### **Menu**

Una clase auxiliar que tiene el propósito de que ciertas pequeñas operaciones como la realización de factoriales o la impresión de bloques de String sea más limpia y práctica.

### **Menú Principal (Clase Operaciones)**

Dentro de la aplicación se desplegará el siguiente menú principal en el que se muestran las opciones del usuario.

```

*****
* MENU                                     *
* 1. Aritmeticas                         *
* 2. Trigonometricas                   *
* 3. Estadisticas                       *
* 4. Calculo                            *
* 5. Salir                              *
*****

-> Ingrese una opcion:

```

Este bloque de String es invocado a partir de la clase auxiliar Menu, a partir de aquí se despliegan múltiples métodos con el prefijo "Salir", todos estos métodos tienen como propósito regresar la aplicación al menú de sus respectivas categorías luego de terminar una operación exitosamente.

El Método más importante es "MenuPrincipal" ya que por medio de la librería scanner es quien pide el número al usuario para luego ser interpretado por un "switch" el cual dependiendo del input despliega las diferentes instancias de la clase operaciones llamadas "operadores".

```

System.out.println(menu.interfaz());
int op = 0;
do {
    try {
        System.out.println("-> Ingrese una opcion:");
        op = scanner.nextInt();
    } catch (Exception e) {
        System.out.println("Debe ingresar una de las opciones!");
        break;
    }
} while (op != 1 && op != 2 && op != 3 && op != 4 && op != 5);
switch (op) {
    case 1:
        OpAritmetica();

    case 2:
        OpTrigonometria();

    case 3:
        OpEstadistica();

    case 4:
        OpCalculo();

    case 5:
        System.out.println("Saliendo de la aplicacion...");
        System.exit(status: 0);
        break;
}

```

## OpAritmética

A continuación se muestra el submenú que aparecerá luego de seleccionar la opción “1. Aritmeticas”.

```
*****
* Aritmetica                                *
* 1. Suma                                    *
* 2. Resta                                  *
* 3. Multiplicacion                         *
* 4. Division                               *
* 5. Potencia                               *
* 6. Salir                                  *
*****
-> Ingrese una opcion:
```

Dentro de este operador ocurren el resto de operaciones a nombrar en esta categoría:

### Suma, Resta, Multiplicación, División y Potencia

Un código bastante simple, toma el input ingresado en consola, guarda estos mismos en un array auxiliar y asigna valores a variables de tipo int, por medio de un ciclo for se realiza la sumatoria que en este caso es solo de dos enteros.

```
case 1:
    String sumT;
    System.out.println("Ingrese los numeros a sumar separados por una coma:");
    sumT = scanner.next();
    String[] sumTa = sumT.split(regex: ",");
    double sumTotal = 0;
    for (int i = 0; i < sumTa.length; i++) {
        sumTotal = sumTotal + Double.parseDouble(sumTa[i]);
    }
    System.out.println("El resultado de la suma es:" + sumTotal+"\n");
    SalidaAr();
    break;
```

De igual forma que con la suma la resta, multiplicación y división fungen de manera muy similar, esta vez sin el uso del ciclo for o un array auxiliar, simplemente asignando y ejecutando.

```
case 2:
    String resT;
    System.out.println("Ingrese los numeros a restar separados por una coma:");
    resT = scanner.next();
    String[] resTa = resT.split(regex: ",");
    double resTotal = Double.parseDouble(resTa[0]) - Double.parseDouble(resTa[1]);
    System.out.println("El resultado de la resta es:" + resTotal);
    SalidaAr();
```

```
case 3:
    String mulT;
    System.out.println("Ingrese los numeros a multiplicar separados por una coma:");
    mulT = scanner.next();
    String[] mulTa = mulT.split(regex: ",");
    double mulTotal = Double.parseDouble(mulTa[0]) * Double.parseDouble(mulTa[1]);
    System.out.println("El resultado de la multiplicacion es:" + mulTotal);
    SalidaAr();
    break;
```

```
case 4:
    String divT;
    String[] divTa;
    do {
        System.out.println("Ingrese los numeros a dividir separados por una coma:");
        divT = scanner.next();
        divTa = divT.split(regex: ",");
    } while (Double.parseDouble(divTa[0]) == 0);
    double divTotal = Double.parseDouble(divTa[0]) / Double.parseDouble(divTa[1]);
    System.out.println("El resultado de la division es:" + divTotal);
    SalidaAr();
    break;
```

```
case 5:
    int pot2;
    int pot1;
    do {
        System.out.println("Ingrese el numero base de la potencia:");
        pot1 = scanner.nextInt();
        System.out.println("Ingrese el numero exponente de la potencia:");
        pot2 = scanner.nextInt();
    } while (pot2 <= 0);
    int potTotal = pot1;
    for (int i = 0; i < pot2 - 1; i++) {
        potTotal = potTotal * pot1;
    }
    System.out.println("El resultado de la potencia es:" + potTotal);
    SalidaAr();
    break;
```



## Trigonómicas

A continuación se muestra el submenú que aparecerá luego de seleccionar la opción “2. Trigonometricas”.

```
*****
* Trigonometria                                *
* 1. Seno                                       *
* 2. Coseno                                    *
* 3. Tangente                                  *
* 4. Salir                                     *
*****
-> Ingrese una opcion:
```

Dentro de este operador ocurren el resto de operaciones a nombrar en esta categoría:

### Seno

Dentro de esta función se hace uso de ambos métodos numéricos encontrados en menú llamados potencia y factorial, los cuales por medio de ciclos for realizan la operación que llevan por nombre.

```
4 usages Angel Andres Godoy Valdez
public long factorial(int n){
    long st = 1;
    int i =1;
    while(i <= n){
        st = st*i;
        i++;
    }
    return st;
}

8 usages Angel Andres Godoy Valdez
public double potencia(double b, int e){
    double pot=1;
    if(e==0){
        pot=1;
    }else{
        for (int i=1;i<=e;i++){
            pot=pot*b;
        }
    }

    return pot;
}
```

El siguiente código es una traducción de la suma de reimann:

$$\text{sen}(x) = \sum_{n=0}^i \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

Con i siendo un iterador que debe ser ingresado por consola y el resto de operaciones siendo separadas para un código más ordenado.

```
case 2:
    System.out.println("Ingrese el ángulo deseado (en grados) para calcular su Coseno:");
    double var1 = scanner.nextDouble()*(pi/180);
    System.out.println("Ingrese el iterador i (1-20):");
    int ite1 = scanner.nextInt();
    double arr1 = 0;
    double abb1 = 0;
    double cosT = 0;
    for (int i = 0; i < ite1; i++) {
        arr1 = menu.potencia(b: -1, i) * menu.potencia(var1, e: 2 * i);
        abb1 = menu.factorial(n: 2 * i);
        cosT = (cosT + arr1 / abb1);
    }
    System.out.println("El total es:" + cosT);
    SalidaTri();
    break;
```

## Coseno

Precisamente el mismo proceso es llevado acabo para esta función pero con una suma distinta

$$\text{cos}(x) = \sum_{n=0}^i \frac{(-1)^n x^{2n}}{(2n)!}$$

```
case 2:
    System.out.println("Ingrese el angulo deseado (en grados) para calcular su Coseno:");
    double var1 = scanner.nextDouble()*(pi/180);
    System.out.println("Ingrese el iterador i (1-20):");
    int ite1 = scanner.nextInt();
    double arr1 = 0;
    double abb1 = 0;
    double cosT = 0;
    for (int i = 0; i < ite1; i++) {
        arr1 = menu.potencia(b: -1, i) * menu.potencia(var1, e: 2 * i);
        abb1 = menu.factorial(n: 2 * i);
        cosT = (cosT + arr1 / abb1);
    }
    System.out.println("El total es:" + cosT);
    SalidaTri();
```

## Tangente

Para la Tangente simplemente se tomó ambas funciones para coseno y seno y al dividir su resultado se encuentra la tangente

## Estadísticas

A continuación se muestra el submenú que aparecerá luego de seleccionar la opción “3.Estadísticas”.

```
*****
* Estadística *
* 1. Promedio *
* 2. Media *
* 3. Moda *
* 4. Varianza *
* 5. Desviacion estandar *
* 6. Salir *
*****

-> Ingrese una opcion:
```

## Promedio y Media

Ya que ambos representan lo mismo, y el concepto en sí es bastante fácil, este es un caso en el que el código se limitó a asignar y operar, esto por medio de un array auxiliar para almacenar el input por consola

```
String promT;  
System.out.println("Ingrese los numeros a promediar separados por una coma:");  
promT = scanner.next();  
String[] promTa = promT.split(regex: ",");  
double promTotal = 0;  
for (int i = 0; i < promTa.length; i++) {  
    promTotal = promTotal + Double.parseDouble(promTa[i]);  
}  
System.out.println("El total del promedio es: " + promTotal / promTa.length);  
SalidaEst();  
break;
```

```
case 2:  
    String medT;  
    System.out.println("Ingrese los numeros para realizar la media separados por una coma:");  
    medT = scanner.next();  
    String[] medTa = medT.split(regex: ",");  
    double medTotal = 0;  
    for (int i = 0; i < medTa.length; i++) {  
        medTotal = medTotal + Double.parseDouble(medTa[i]);  
    }  
    System.out.println("La media es: " + medTotal / medTa.length);  
    SalidaEst();  
    break;
```

## Moda

Creí conveniente que para contar las veces que se repite un elemento y ver cual elemento es el que tiene más instancias crear un ciclo for que se actualiza a medida que i aumenta de valor.

```
String mod;
System.out.println("Ingrese los numeros separados por una coma para determinar su moda:");
mod = scanner.next();
String[] modArSt = mod.split(regex: ",");
Integer[] modAr = new Integer[modArSt.length];
for (int i = 0; i < modArSt.length; i++) {
    modAr[i] = Integer.parseInt(modArSt[i]);
}
int cont = 0;
int elm = 0;
for (int i = 0; i < modAr.length; i++) {
    int tmeIm = modAr[i];
    int tcont = 0;
    for (int j = 0; j < modAr.length; j++) {
        if (modAr[j] == tmeIm) {
            tcont++;
            if (tcont > cont) {
                elm = tmeIm;
                cont = tcont;
            }
        }
    }
}
```

## Varianza y Desviación Estándar

Ambas representan un reto ya que los datos deben ser almacenados en orden así como la cantidad de instancias que hay de los mismos, esto se logra por medio de iterar con un for al ser almacenados y luego ordenados para hacer las respectivas operaciones. Para el primer arreglo se van eliminando los elementos repetidos y simplemente se dejan adentro una instancia, para en la siguiente columna del arreglo colocar la cantidad de instancias que había en el arreglo original sin modificar, posteriormente para rellenar lo que queda del arreglo a forma de tabla estadística se llama a un método en menú llamado varianza para realizar esos cálculos, para la desviación estándar se repite el proceso con la diferencia que en el resultado final se encuentra la raíz cuadrada por medio de un for.

```

case 4:

    System.out.println("Ingrese los numeros separados por una coma para determinar su varianza:");
    String data = scanner.next();
    String[] dataArs = data.split(regex: ",");
    int[] dataAr = new int[dataArs.length];
    int[] aux0 = new int[dataArs.length];
    Integer[][] dataF = new Integer[dataArs.length][2];//array final1.0
    for (int i = 0; i < dataArs.length; i++) {
        dataAr[i] = Integer.parseInt(dataArs[i]);
    }
    for (int i = 0; i < dataArs.length; i++) {
        aux0[i] = Integer.parseInt(dataArs[i]);
    }

    Arrays.sort(dataAr);
    Arrays.sort(aux0);

    int cont1 = 0;
    int elm1 = 0;
    int pos1 = 0;

    for (int i = 0; i < dataAr.length; i++) {
        int tme1m1 = dataAr[i];
        int tcont1 = 0;
        for (int j = 0; j < dataAr.length; j++) {

```

```

case 5:

    System.out.println("Ingrese los numeros separados por una coma para determinar su desviacion estandar");
    String d1 = scanner.next();
    double varEj = menu.varianzaM(d1);
    double temp;
    double sr = varEj / 2;
    do {
        temp = sr;
        sr = (temp + (varEj / temp)) / 2;
    } while ((temp - sr) != 0);

    System.out.println("La desviacion estandar para los datos ingresados es: " + sr);
    SalidaEst();
    break;

case 6:

    MenuPrincipal();

```

## Cálculo

A continuación se muestra el submenú que aparecerá luego de seleccionar la opción “4.Calculo”.

Para esta función se ingresan los datos a una matriz por medio de un for y de arrays auxiliares, luego de tener llena la matriz de dos dimensiones se realizan operaciones guardando los datos en un nuevo array auxiliar para convertir en ceros todos aquellos datos que no lo sean y sean distintos a las casillas 0,0 1,1 2,2 etc.

```
case 1:
    int nf = 0;
    int nc = 0;
    System.out.println("Ingrese el numero de filas: ");
    nf = scanner.nextInt();
    System.out.println("Ingrese el numero de columnas: ");
    nc = scanner.nextInt();
    int[][] tablaGj = new int[nf][nc];
    int contf = 1;
    for (int i = 0; i < nf; i++) {
        System.out.print("Ingrese los elementos de la fila " + contf + ":");
        String aux = scanner.next();
        String[] auxAr = aux.split(regex: ",");
        for (int j = 0; j < nc; j++) {
            tablaGj[i][j] = Integer.parseInt(auxAr[j]);
        }
        contf++;
    }
}
```

```
System.out.println();
int x=(tablaGj[0][3])/(tablaGj[0][0]);
int y=(tablaGj[1][3])/(tablaGj[1][1]);
int z=(tablaGj[2][3])/(tablaGj[2][2]);
System.out.println("La solucion para la ecuacion es: x="+x+"\t y="+y+"\t z="+z);
SalidaCal();
break;
```