

¿Que es la mediana?

En estadística, la mediana es el valor del medio de todos los datos ordenados de menor a mayor. Es decir, la mediana divide todo el conjunto de datos ordenados en dos partes iguales. La mediana es una medida de posición central que sirve para describir una distribución de probabilidad.



Como sacar la mediana en java

Para sacar la mediana en java primero tenemos que tener los datos a lo que queramos sacarle la mediana, para hacer esto tenemos dos formulas, una para cantidad par y otra para cantidad impar.

Para facilidad del usuario el conjunto de datos estará almacenado en un Array []

```
Scanner teclado = new Scanner(System.in);  
System.out.println("Ingrese una cantidad de numero a iterar(separados por ,)");  
String entrada = teclado.nextLine();  
String[] elementos = entrada.split(",");
```

Tratamiento de datos en los Arrays

El programa por comodidad del usuario va a solicitar un valor de tipo "String" dividido por (,) para separar los números a calcular.

Posterior a eso se crea un Array de tipo "String" con el método {.Split(",")}. el cual va a leer la cadena ingresada y cada que detecte la (,) va a registrar un índice nuevo en el Array.

Luego lo transformamos a valor numérico creando otro array de tipo numérico con la misma longitud que el array de texto; seguido se utiliza una línea "FOR" utilizando la variable de control como índice para migrar los datos ==

```
System.out.println("Ingrese una cantidad de numero a iterar(separados por ,)");
String entrada = teclado.nextLine();
String[] elementos = entrada.split(",");
double[] elementos2 = new double[elementos.length];

for (int i = 0; i < elementos.length; i++) {
    elementos2[i] = Double.parseDouble(elementos[i]);
}
```

Formula para caso Par

En el caso de que tengamos una cantidad de índices par tendremos que realizar una formula un poco mas compleja; tendremos que tomar los 2 valores centrales y promediarlos entre si, esta será nuestra mediana; para lograrlo trabajaremos con los índices y no con los valores.

El método(.sort) ordena los elemento de menor a mayor, la variable n contiene la longitud total(#indices).

El (IF) filtra cuando n es par y ejecuta la formula(tomando el promedio de los 2 valores centrales).

```
Arrays.sort(elementos2);
int n = elementos2.length;
if (n % 2 == 0) {
    double mediana = (elementos2[n / 2] + elementos2[n / 2 - 1])/2;
    System.out.println("La mediana es -->" + mediana);}

```

Formula para el caso de cantidad IMPAR

La formula para el caso impar es simple; debido a que buscamos el valor intermedio, solo tenemos que organizar de menor a mayor, y dividir la cantidad de índices (.length) entre 2 (Esto nos dará como resultado el índice central). Eso sería la mediana.

Después cerramos el Scanner.

```
else {  
    Arrays.sort(elementos2);  
    int n = elementos2.length;  
    double mediana = elementos2[n / 2];  
    System.out.println("La mediana es -->" + mediana);}  
teclado.close();
```

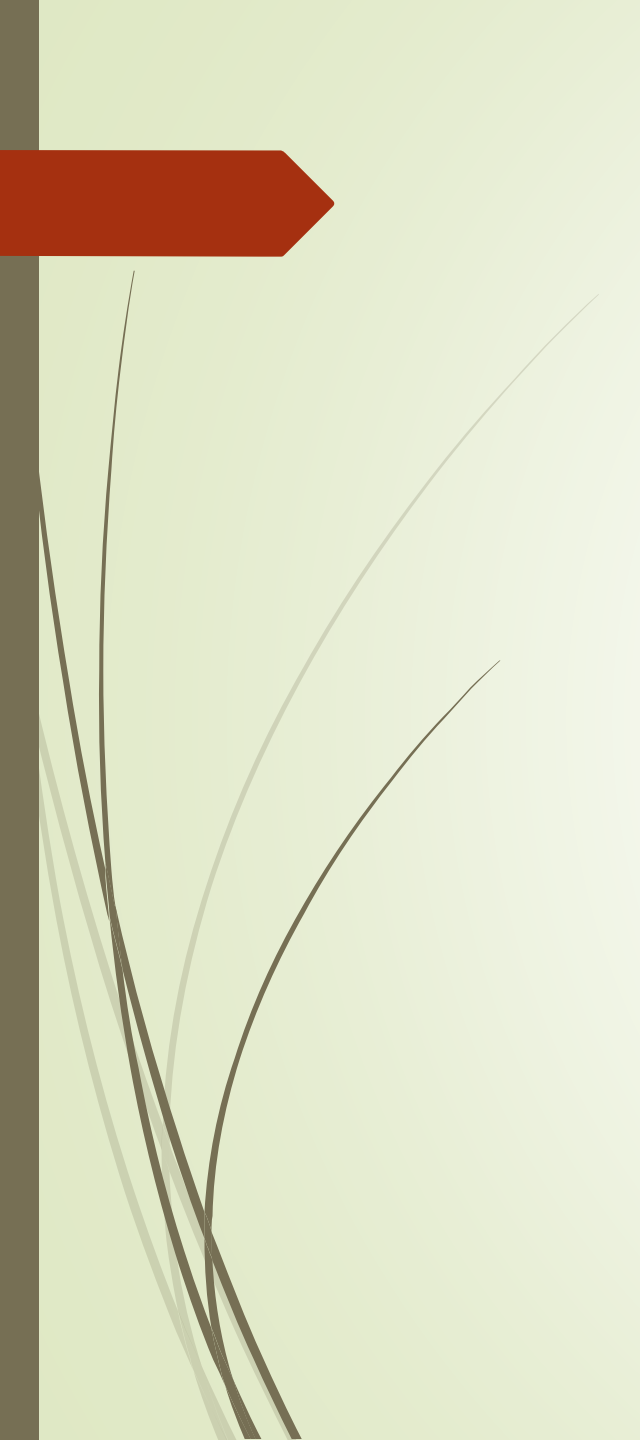
```
package com.example;

import java.util.Scanner;
import java.util.Arrays;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Ingrese una cantidad de numero a iterar(separados por ,)");
        String entrada = teclado.nextLine();
        String[] elementos = entrada.split(",");
        double[] elementos2 = new double[elementos.length];
        // double[] elementos3 = {1,2,3,4};

        for (int i = 0; i < elementos.length; i++) {
            elementos2[i] = Double.parseDouble(elementos[i]);
        }

        Arrays.sort(elementos2);
        int n = elementos2.length;
        if (n % 2 == 0) {
            double mediana = (elementos2[n / 2] + elementos2[n / 2 - 1]) / 2;
            System.out.println("La mediana es -->" + mediana);
        } else {
            double mediana = elementos2[n / 2];
            System.out.println("La mediana es -->" + mediana);
        }
        teclado.close();
    }
}
```



Creación de un Nuevo Ejercicio original

```

1 package com.example;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 public class Main {
7     Run | Debug
8     public static void main(String[] args) {
9         String tryUser, tryPassword, user, password;
10        Scanner teclado = new Scanner(System.in);
11        ArrayList<String> users = new ArrayList<>(); // contenedor iterable de la credencial "usuario"
12        ArrayList<String> passwords = new ArrayList<>(); // contenedor iterable de la credencial "contraseña"
13        boolean centinelaW = true;
14        while (centinelaW) {
15            System.out.println("Ingrese una opcion valida dentro del formulario \n");
16            System.out.println("1. REGISTRAR USUARIO\n");
17            System.out.println("2. INICIAR SESIÓN\n");
18            System.out.println("3. SALIR\n");
19            int centinela = teclado.nextInt();
20            switch (centinela) {
21                case 1:
22                    System.out.println("SELECCIONÓ --> REGISTRAR USUARIO");
23
24                    Scanner input = new Scanner(System.in);
25                    try {
26                        System.out.println("Ingrese el usuario a registrar");
27                        user = input.nextLine();
28                        users.add(user);
29                        System.out.println("Usuario valido");
30
31                        System.out.println("Ingrese la contraseña a registrar");
32                        password = input.nextLine();
33                        passwords.add(password);
34                    } catch (Exception e) {
35                        System.out.println("ERROR EN EL REGISTRO; CRENDECIALES INVALIDAS.\n vuelva a intentar");
36                        break;
37                    }
38                    System.out.println("Registro exitoso, puede registrar otro usuario repitiendo el formulario\n\n");
39
40                    break; // break "case 1"

```



```

45
46 case 2:
47     Scanner input2 = new Scanner(System.in);
48     System.out.println("SELECCIONÓ --> INICIO DE SESION\n");
49     System.out.println("INGRESE USUARIO");
50     tryUser = input2.nextLine();
51     System.out.println("INGRESE CONTRASEÑA");
52     tryPassword = input2.nextLine();
53     for (int i = 0; i <= passwords.size(); i++) {
54
55         try {
56             if (users.get(i).equals(tryUser) && passwords.get(i).equals(tryPassword)) {
57                 System.out.println("INICIO DE SESION EXITOSO\n");
58                 System.out.println("Bienvenido " + users.get(i) + "\n");
59                 break;
60             } else {
61                 continue;
62             }
63
64         } catch (Exception e) {
65             System.out.println("CREDENCIALES INVALIDAS\n");
66         }
67     }
68
69 }
70 break;// break "case 2"
71
72 case 3:
73     System.out.println("SALIENDO\nGRACIAS POR USAR LA INTERFAZ");
74     centinelaw = false;
75     break; // break "case 3"
76
77 default :
78     System.out.println("ingrese una opcion valida dentro del menu\n");
79
80 }
81
82 }
83 teclado.close();
84

```

Estructura →

➤ Estructura Inicial

```
8 String tryUser, tryPassword, user, password;
9 Scanner teclado = new Scanner(System.in);
10 ArrayList<String> users = new ArrayList<>(); // contenedor iterable de la credencial "usuario"
11 ArrayList<String> passwords = new ArrayList<>(); // contenedor iterable de la credencial "contraseña"
12 boolean centinelaW = true;
13 while (centinelaW) {
14     System.out.println("Ingrese una opcion valida dentro del formulario \n");
15     System.out.println("1. REGISTRAR USUARIO\n");
16     System.out.println("2. INICIAR SESIÓN\n");
17     System.out.println("3. SALIR\n");
18     int centinela = teclado.nextInt();
19     switch (centinela) {
```

- Declaramos las variables a utilizar, estas siendo las contenedoras de “usuarios” y “contraseñas” (registros e ingresos)
- Las variables try... Contendrán las credenciales del ingreso, mientras que las otras se usarán en los registros
- Tendremos 2 conjuntos de datos para almacenar las credenciales; usaremos el índice como intento de ID
- Establecemos un bucle “while” con una variable “centinelaW” la cual controla su ejecución
- Dentro del while creamos un “switch case” controlado por otra variable “centinela”

Case 1 → REGISTRO

```
19 switch (centinela) {  
20     case 1:  
21         System.out.println("SELECCIONÓ --> REGISTRAR USUARIO");  
22  
23         Scanner input = new Scanner(System.in);  
24         try {  
25             System.out.println("Ingrese el usuario a registrar");  
26             user = input.nextLine();  
27             users.add(user);  
28             System.out.println("Usuario valido");  
29  
30             System.out.println("Ingrese la contraseña a registrar");  
31             password = input.nextLine();  
32             passwords.add(password);  
33  
34         } catch (Exception e) {  
35             System.out.println("ERROR EN EL REGISTRO; CRENECIALES INVALIDAS.\n vuelva a intentar");  
36             break;  
37         }  
38         System.out.println("Registro exitoso, puede registrar otro usuario repitiendo el formulario\n\n");  
39  
40         break; // break "case 1"  
41     }
```

- El caso 1 tiene como función registrar usuarios dentro de nuestra "BD" a través del objeto "input" (Scanner consola)
- Sí todo tiene una ejecución correcta el programa almacenara cada credencial en la última posición del ArrayList; esto mantiene un orden. De lo contrario se ejecuta un mensaje de error, sacando al usuario del "case 1" en el proceso

CASE 2 → INGRESO

```
case 2:
Scanner input2 = new Scanner(System.in);
System.out.println("SELECCIONÓ --> INICIO DE SESION\n");
System.out.println("INGRESE USUARIO");
tryUser = input2.nextLine();
System.out.println("INGRESE CONTRASEÑA");
tryPassword = input2.nextLine();
for (int i = 0; i <= passwords.size(); i++) {

    try {
        if (users.get(i).equals(tryUser) && passwords.get(i).equals(tryPassword)) {
            System.out.println("INICIO DE SESION EXITOSO\n");
            System.out.println("Bienvenido " + users.get(i) + "\n");
            break;
        } else {
            continue;
        }
    } catch (Exception e) {
        System.out.println("CREDENCIALES INVALIDAS\n");
    }
}
break; // break "case 2"
```

- Posterior al registro tendremos las credenciales de nuestros usuarios almacenadas en las estructuras de datos (ArrayLists)
- El programa solicitará al usuario las credenciales de acceso, las cuales comparará con las almacenadas en el registro
- Usando un bucle "for" iteraremos cada posición de nuestros 2 contenedores ("users" y "passwords"); recordemos que las credenciales de cada usuario comparten el índice, por ende, tenemos una forma de relacionarlas
- Sí la credencial "usuario" almacenada en el índice "i" concuerda con la credencial suministrada, y además la credencial "contraseña" en el mismo índice "i" concuerda con la suministrada, entonces el condicional dará "true" permitiendo la ejecución del mensaje de ingreso. De lo contrario, al salir del rango en el bucle "for" (no encontró coincidencias) arrojará un "out of bounds" ejecutando así el mensaje de error "CREDENCIALES INVALIDAS"

CASE 3 → SALIDA & DEFAULT

```
case 3:
    System.out.println("SALIENDO\nGRACIAS POR USAR LA INTERFAZ");
    centinelaW = false;
    break; // break "case 3"

default :
    System.out.println("ingrese una opcion valida dentro del menu\n");
```

- Para culminar el bucle el programa cambiara el valor del "centinelaW" a "false"
- Default especifica al usuario que la opción es invalida, redirigiéndolo de nuevo al menú