

Depurar Ejercicio 3.4 (Arreglar ecuación)

El siguiente código nos ha fallado cuando a es igual a 0

```
public class ecuacion {  
    public static void main(String[] args) {  
        double a = 0, b = 0, c = 0, sol1 = 0, sol2 = 0;  
        Scanner scan = new Scanner(System.in);  
        System.out.println("Introduce el coeficiente de x²");//a=0 error  
        a = scan.nextDouble();  
        System.out.println("Introduce el coeficiente de x");  
        b = scan.nextDouble();  
        System.out.println("Introduce el termino independiente");  
        c = scan.nextDouble();  
        sol1 = (-1*b + Math.sqrt(b*b - 4*a*c))/(2*a);  
        sol2 = (-1*b - Math.sqrt(b*b - 4*a*c))/(2*a);  
        System.out.println("Las soluciones son: " + sol1 + "y" + sol2);  
    }  
}
```

Como podemos ver la solución nos da que no es un número o es infinito

```
Introduce el coeficiente de x²  
0  
Introduce el coeficiente de x  
7  
Introduce el termino independiente  
3  
Las soluciones son: NaNy-Infinity  
BUILD SUCCESSFUL (total time: 3 seconds)
```

Procederemos a depurar el código poniendo un breakpoint y presionando Ctrl + F5

```
10 public class ecuacion {  
11     public static void main(String[] args) {  
12         double a = 0, b = 0, c = 0, sol1 = 0, sol2 = 0;  
13         Scanner scan = new Scanner(System.in);  
14         System.out.println("Introduce el coeficiente de x²");//a=0 erro  
15         a = scan.nextDouble();  
16         System.out.println("Introduce el coeficiente de x");  
17         b = scan.nextDouble();  
18         System.out.println("Introduce el termino independiente");  
19         c = scan.nextDouble();  
20         sol1 = (-1*b + Math.sqrt(b*b - 4*a*c))/(2*a);  
21         sol2 = (-1*b - Math.sqrt(b*b - 4*a*c))/(2*a);  
22         System.out.println("Las soluciones son: " + sol1 + "y" + sol2);  
23     }  
24 }  
25 }
```

Introducimos los valores e iremos consultando las variables hasta que encontremos un error, metiendo como valor de a el número 0.

```
10 public class ecuacion {
11     public static void main(String[] args) {
12         double a = 0, b = 0, c = 0, sol1 = 0, sol2 = 0;
13         Scanner scan = new Scanner(System.in);
14         System.out.println("Introduce el coeficiente de x²"); //a=0 error
15         a = scan.nextDouble();
16         System.out.println("Introduce el coeficiente de x");
17         b = scan.nextDouble();
18         System.out.println("Introduce el termino independiente");
19         c = scan.nextDouble();
20         sol1 = (-1*b + Math.sqrt(b*b - 4*a*c))/(2*a);
21         sol2 = (-1*b - Math.sqrt(b*b - 4*a*c))/(2*a);
22         System.out.println("Las soluciones son: " + sol1 + "y" + sol2);
23     }
24 }
25 }
```

Name	Type	Value
<Enter new watd		
Static		
args	String[]	#295(length=0)
a	double	0.0
b	double	7.0
c	double	3.0
sol1	double	0.0
sol2	double	0.0
scan	Scanner	#296

Output X

ejercicioED (debug) X Debugger Console X

```
debug:
Introduce el coeficiente de x²
0
Introduce el coeficiente de x
7
Introduce el termino independiente
3
```

Encontramos que el error se encuentra al llegar a las líneas 20 y 21, así que modificaremos el código hasta solucionar el error

```
20 sol1 = (-1*b + Math.sqrt(b*b - 4*a*c))/(2*a);
21 sol2 = (-1*b - Math.sqrt(b*b - 4*a*c))/(2*a);
22 System.out.println("Las soluciones son: " + sol1 + "y" + sol2);
23 }
24 }
25 }
26 }
```

Name	Type	Value
<Enter new wa		
Static		
args	String[]	#295(length=0)
a	double	0.0
b	double	7.0
c	double	3.0
sol1	double	NaN
sol2	double	-Infinity
scan	Scanner	#296

Output X

ejercicioED (debug) X Debugger Console

```
debug:
Introduce el coeficiente de x²
0
Introduce el coeficiente de x
7
Introduce el termino independiente
3
```

Modificamos el código de forma que nos de un error si el valor de a igual 0 y de esta forma no daría ningún problema

```
10 public class ecuacion {
11     public static void main(String[] args) {
12         double a = 0, b = 0, c = 0, sol1 = 0, sol2 = 0;
13         Scanner scan = new Scanner(System.in);
14         System.out.println("Introduce el coeficiente de x²");//a=0 error
15         a = scan.nextDouble();
16         System.out.println("Introduce el coeficiente de x");
17         b = scan.nextDouble();
18         System.out.println("Introduce el termino independiente");
19         c = scan.nextDouble();
20         if(a!=0){
21             sol1 = (-1*b + Math.sqrt(b*b - 4*a*c))/(2*a);
22             sol2 = (-1*b - Math.sqrt(b*b - 4*a*c))/(2*a);
23             System.out.println("Las soluciones son: " + sol1 + "y" + sol2);
24         }else{
25             System.out.println("Esta fórmula no puede ser usada si el valor
26         }
27     }
28 }
```

Output ×

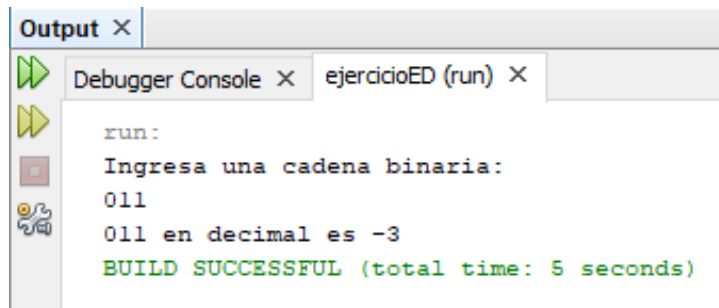
Debugger Console × ejercicioED (run) ×

run:
Introduce el coeficiente de x²
0
Introduce el coeficiente de x
7
Introduce el termino independiente
3
Esta fórmula no puede ser usada si el valor de a igual a 0
BUILD SUCCESSFUL (total time: 4 seconds)

Depurar Ejercicio 3.5 (Arreglar binario)

El siguiente código nos ha fallado cuando 011 en decimal es -3 en lugar de 3

```
public class binario {  
    public static void main(String[] args) {  
  
        String binStr;  
        int binStrLen;  
        int dec=0;  
        boolean invalid =false;  
  
        System.out.println("Ingresa una cadena binaria: ");  
        Scanner sc = new Scanner(System.in);  
  
        binStr = sc.nextLine();  
        binStrLen = binStr.length();  
  
        int number = 0;  
        int exp=binStrLen;  
  
        for (int pos=0; pos < binStrLen; ++pos){  
            char binChar = binStr.charAt(pos);  
            exp--;  
            if (binChar == '1') {  
                number -= Math.pow(2, exp);  
            }  
            else if (binChar == '0') {  
                continue;  
            } else {  
                invalid = true;  
                break;  
            }  
        }  
        if (!invalid){  
            System.out.println(binStr + " en decimal es " + number);  
        }else{  
            System.out.println("Entrada invlida");  
        }  
    }  
}
```



```
Output X  
Debugger Console X ejercicioED (run) X  
run:  
Ingresa una cadena binaria:  
011  
011 en decimal es -3  
BUILD SUCCESSFUL (total time: 5 seconds)
```

Procederemos a depurar el código poniendo un breakpoint y presionando Ctrl + F5

```
13 public class binario {
14     public static void main(String[] args) {
15
16         String binStr;
17         int binStrLen;
18         int dec=0;
19         boolean invalid =false;
20
21         System.out.println("Ingresa una cadena binaria: ");
22         Scanner sc = new Scanner(System.in);
23
24         binStr = sc.nextLine();
25         binStrLen = binStr.length();
26
27         int number = 0;
28         int exp=binStrLen;
29
30         for (int pos=0; pos < binStrLen; ++pos){
31             char binChar = binStr.charAt(pos);
32             exp--;
33             if (binChar == '1') {
34                 number -= Math.pow(2, exp);
35             }
36             else if (binChar == '0') {
37                 continue;
38             } else {
39                 invalid = true;
40                 break;
41             }
42         }
43         if (!invalid){
44             System.out.println(binStr + " en decimal es " + number);
45         }else{
46             System.out.println("Entrada invlida");
47         }
48     }
49 }
```

Introducimos el valor 011 e iremos consultando las variables hasta que encontremos un error.

```
for (int pos=0; pos < binStrLen; ++pos){
    char binChar = binStr.charAt(pos);
    exp--;
    if (binChar == '1') {
        number -= Math.pow(2, exp);
    }
    else if (binChar == '0') {
        continue;
    } else {
        invalid = true;
        break;
    }
}
```

Name	Type	Value
<Enter new watd		
Static		
args	String[]	#96(length=0)
dec	int	0
invalid	boolean	false
sc	Scanner	#297
binStr	String	"011"
binStrLen	int	3
number	int	-2
exp	int	1
pos	int	1

Output X

ejercicioED (debug) X Debugger Cor

debug:

Ingresar una cadena binaria:

011

Hemos encontrado que el error se encuentra en la línea 34, por lo tanto modificaremos el código para solucionar ese error. (number restaba en vez de sumar)

```
for (int pos=0; pos < binStrLen; ++pos){
    char binChar = binStr.charAt(pos);
    exp--;
    if (binChar == '1') {
        number += Math.pow(2, exp);
    }
    else if (binChar == '0') {
        continue;
    } else {
        invalid = true;
        break;
    }
}

if (!invalid){
    System.out.println(binStr + " en decimal es " + number);
}else{
    System.out.println("Entrada invlida");
}
}
```

Output X

Debugger Console X ejercicioED (run) X

run:

Ingresar una cadena binaria:

011

011 en decimal es 3

BUILD SUCCESSFUL (total time: 6 seconds)

