

# TEMA 6: DEBUGGEAR



# ÍNDICE

<b>PROGRAMACIÓN (archivos adjuntos)</b>	<b>3</b>
1. Ejercicio DNI	3
2. Ejercicios NumerosInversos	3
3. Ejercicio PiedraPapelTijera	3
<b>ENTORNO DESARROLLO</b>	<b>3</b>
4. Depuración 1	3
5. Depuración 2	4
6. Depuración 3	5
7. Depuración 4	6
8. Depuración 5	8
9. Depuración 6	8
10. Depuración 7	9

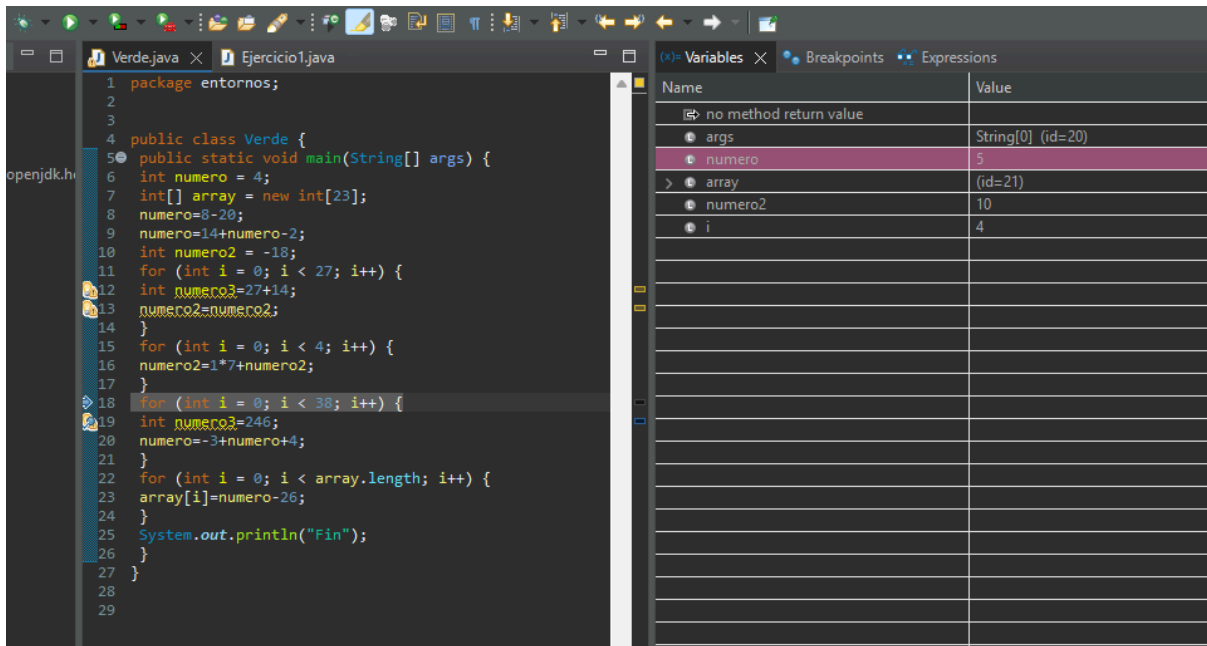
## **PROGRAMACIÓN (archivos adjuntos)**

- 1. En España cada persona está identificada con un Documento Nacional de Identidad (DNI) en el que figura un número y una letra, por ejemplo 56999545W. Realiza un programa donde le pidas al usuario SOLO el número del dni y el programa te devuelva la letra. Para calcular la letra solo tienes que dividir el número del DNI entre 23, el resto de esta división se corresponde con la posición de la letra en el abecedario. Utiliza un array para guardar CADA letra del abecedario. Tienes que tener en cuenta que el número de DNI está compuesto por 8 dígitos.**
- 2. Escribe un programa que lea 10 números por teclado y que luego los muestre en orden inverso, es decir, el primero que se introduce es el último en mostrarse y viceversa.**
- 3. Vamos a crear un juego sencillo de "Piedra, Papel o Tijeras". En este juego, el usuario jugará contra la computadora. En este juego, el usuario elige entre Piedra (1), Papel (2) o Tijeras (3). La computadora elige al azar una de las tres opciones. Luego, se determina el ganador según las reglas clásicas del juego. El juego continúa hasta que el usuario decide salir ingresando 0.**

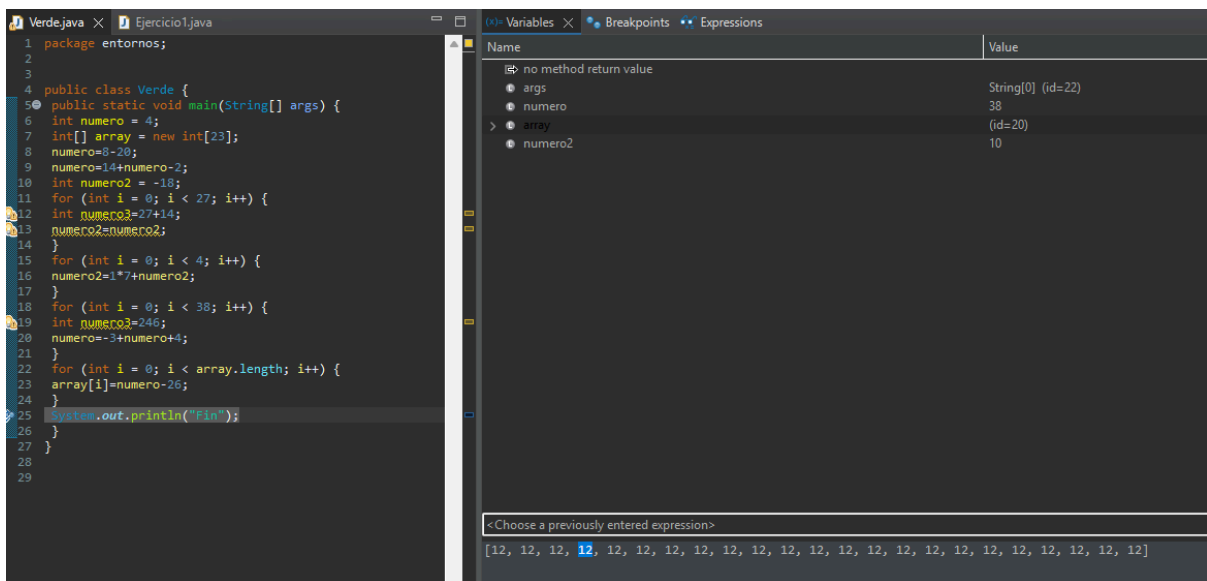
## **ENTORNO DESARROLLO**

### **4. Depuración:**

- El valor de “número” al comenzar la cuarta iteración del tercer bucle.

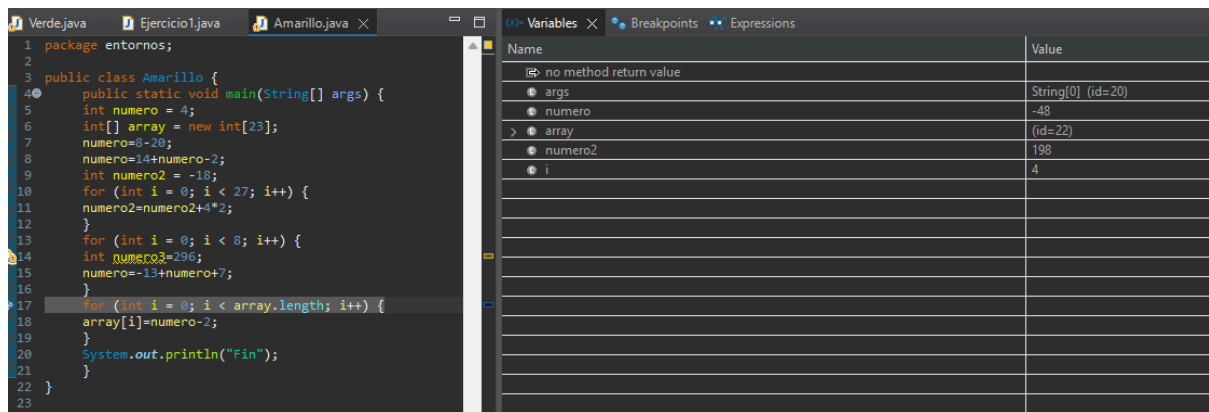


- La suma de “número” y “array[3]” al final del programa.

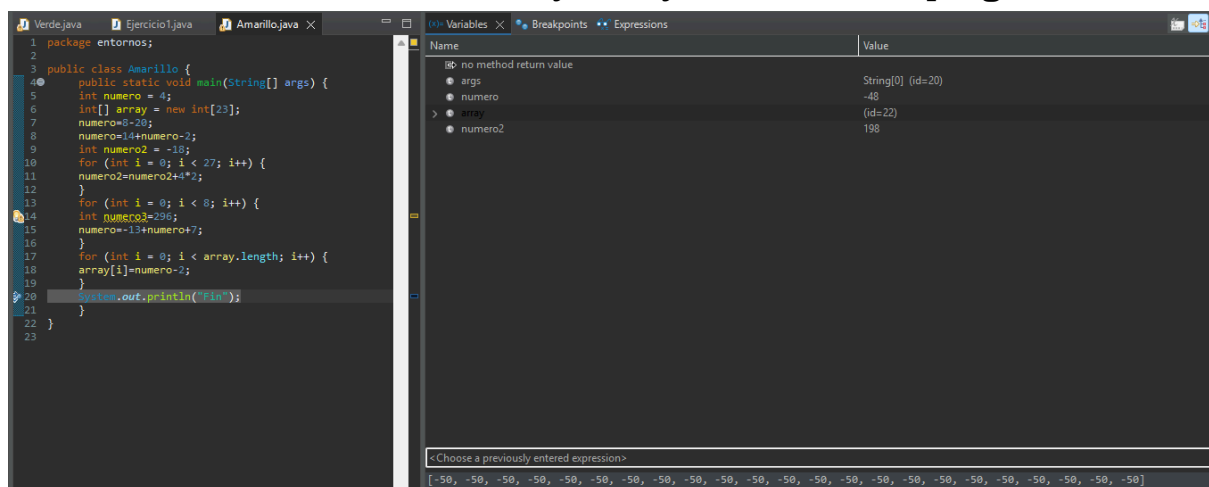


## 5. Dado el siguiente código

- El valor de “número” al comenzar la cuarta iteración del tercer bucle.

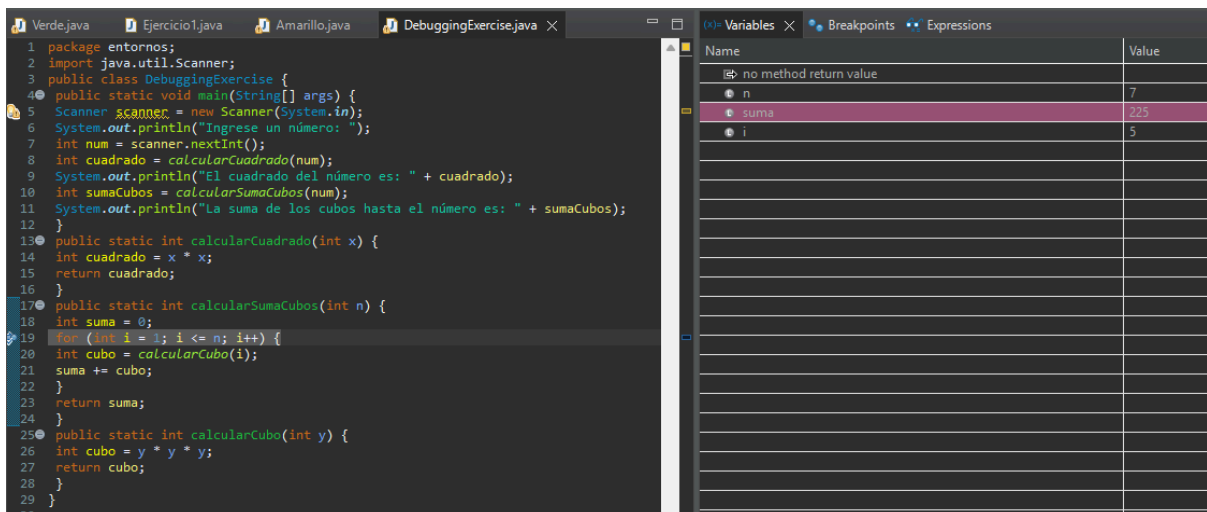


- La suma de “número” y “array[3]” al final del programa.

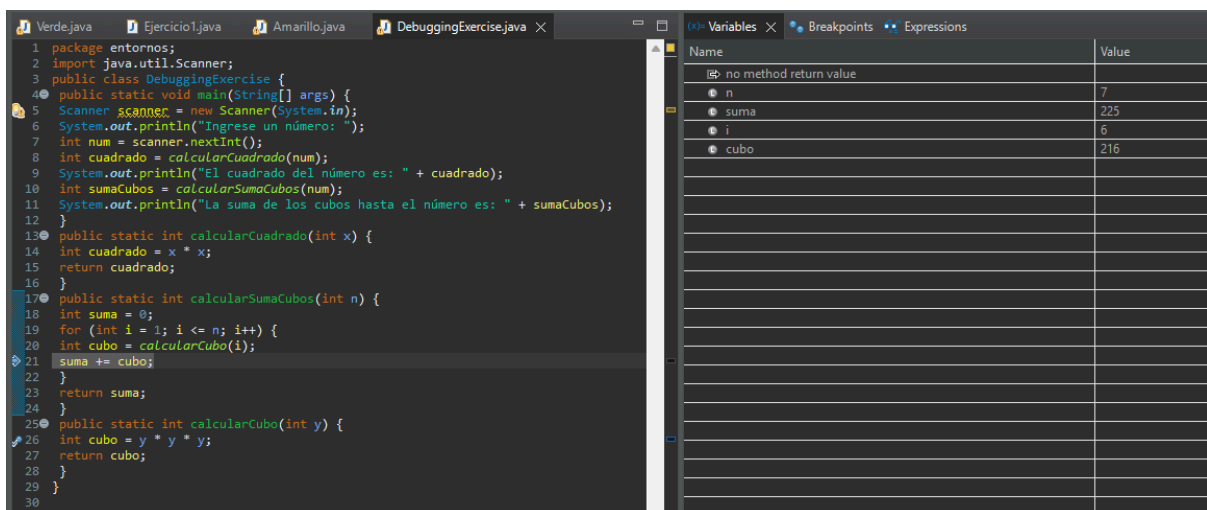


**6. Agrega puntos de interrupción en diferentes partes del código, especialmente antes y después de la ejecución de métodos. Observa cómo cambian los valores de las variables (num, cuadrado, suma, etc.) mientras ejecutas el programa en modo de depuración.**

- Queremos saber el valor de suma cuando se inserta 7, e  $i = 5$ .



b. Y la variable cubo cuando i = 6.



**7. Este programa solicita al usuario dos números, calcula la suma y el producto utilizando métodos separados, y luego imprime los resultados. Establece puntos de interrupción en lugares estratégicos, como antes y después de la entrada de datos, antes y después de llamar a los métodos, etc. Observa cómo cambian las variables (num1, num2, suma, multiplicación, etc.) mientras ejecutas el programa en modo de depuración.**

a. Necesito la trazabilidad de producto en cada una de las iteraciones de i, osea el valor producto para cada interacción de i.

Indicaremos como valor un número que deseemos (yo pondré el 3)

- Suma y multiplicación:

DebuggingExercise2.java

```

1 package entornos;
2
3 import java.util.Scanner;
4 public class DebuggingExercise2 {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         System.out.println("Ingrese un número: ");
8         int num1 = scanner.nextInt();
9         System.out.println("Ingrese otro número: ");
10        int num2 = scanner.nextInt();
11        int suma = sumarNumeros(num1, num2);
12        System.out.println("La suma de los dos números es: " + suma);
13        int multiplicacion = multiplicarNumeros(num1, num2);
14        System.out.println("El producto de los dos números es: " + multiplicacion);
15    }
16    public static int sumarNumeros(int a, int b) {
17        int suma = a + b;
18        return suma;
19    }
20    public static int multiplicarNumeros(int x, int y) {
21        int producto = 0;
22        for (int i = 0; i < y; i++) {
23            producto = sumarNumeros(producto, x);
24        }
25        return producto;
26    }
27 }

```

(x)= Variables

Name	Value
sumarNumeros() returned	3
x	3
y	3
producto	0
i	0

DebuggingExercise2.java

```

1 package entornos;
2
3 import java.util.Scanner;
4 public class DebuggingExercise2 {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         System.out.println("Ingrese un número: ");
8         int num1 = scanner.nextInt();
9         System.out.println("Ingrese otro número: ");
10        int num2 = scanner.nextInt();
11        int suma = sumarNumeros(num1, num2);
12        System.out.println("La suma de los dos números es: " + suma);
13        int multiplicacion = multiplicarNumeros(num1, num2);
14        System.out.println("El producto de los dos números es: " + multiplicacion);
15    }
16    public static int sumarNumeros(int a, int b) {
17        int suma = a + b;
18        return suma;
19    }
20    public static int multiplicarNumeros(int x, int y) {
21        int producto = 0;
22        for (int i = 0; i < y; i++) {
23            producto = sumarNumeros(producto, x);
24        }
25        return producto;
26    }
27 }

```

(x)= Variables

Name	Value
sumarNumeros() returned	6
x	3
y	3
producto	3
i	1

DebuggingExercise2.java

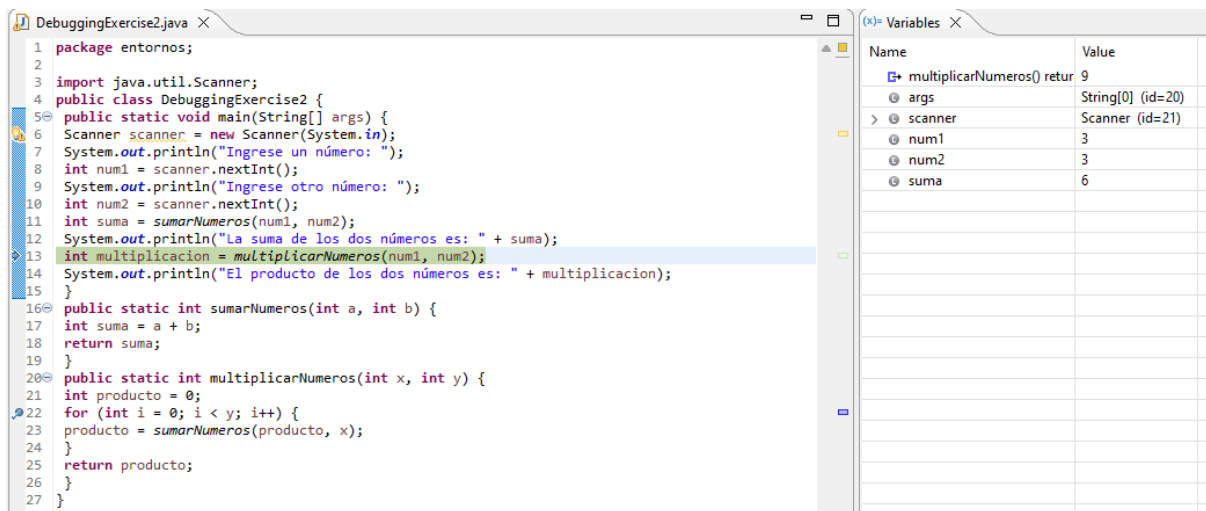
```

1 package entornos;
2
3 import java.util.Scanner;
4 public class DebuggingExercise2 {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         System.out.println("Ingrese un número: ");
8         int num1 = scanner.nextInt();
9         System.out.println("Ingrese otro número: ");
10        int num2 = scanner.nextInt();
11        int suma = sumarNumeros(num1, num2);
12        System.out.println("La suma de los dos números es: " + suma);
13        int multiplicacion = multiplicarNumeros(num1, num2);
14        System.out.println("El producto de los dos números es: " + multiplicacion);
15    }
16    public static int sumarNumeros(int a, int b) {
17        int suma = a + b;
18        return suma;
19    }
20    public static int multiplicarNumeros(int x, int y) {
21        int producto = 0;
22        for (int i = 0; i < y; i++) {
23            producto = sumarNumeros(producto, x);
24        }
25        return producto;
26    }
27 }

```

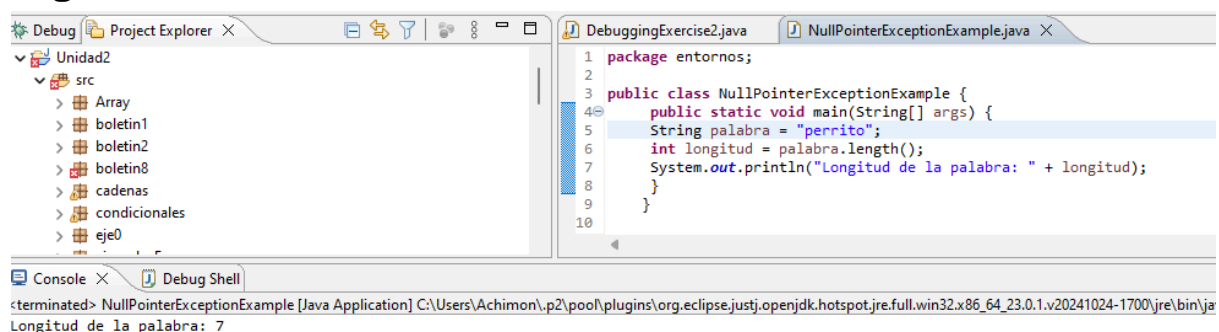
(x)= Variables

Name	Value
sumarNumeros() returned	9
x	3
y	3
producto	6
i	2



## 8. Corrige el código para evitar la excepción

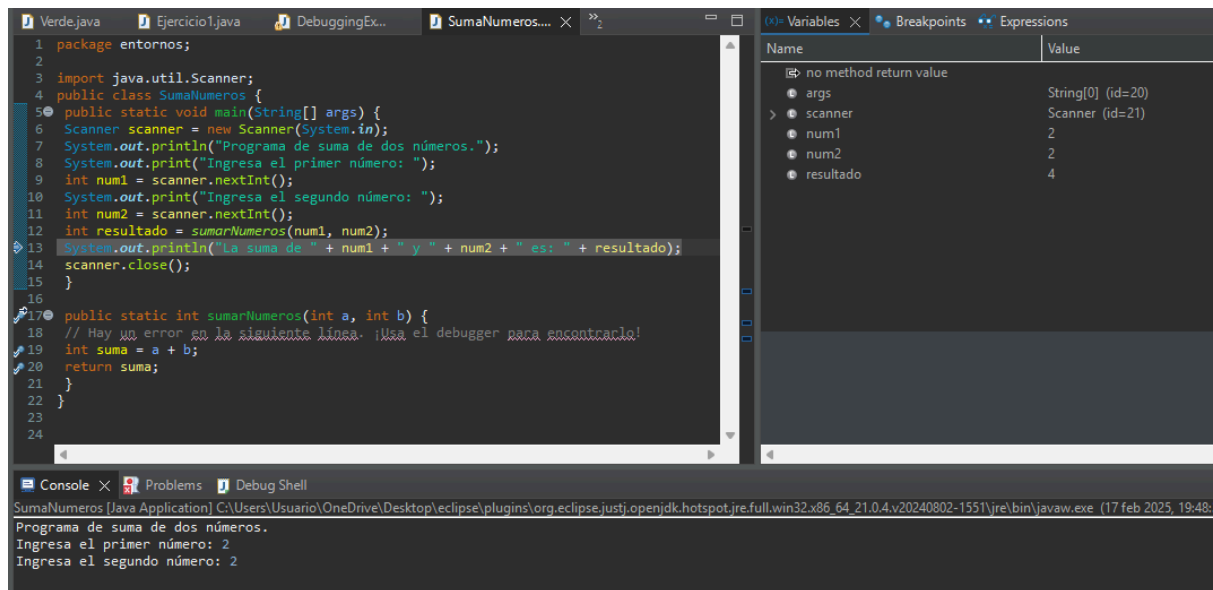
El error que obtenemos es debido a que la variable *palabra* es nula por lo tanto no se puede ejecutar el código ni mostrar el resultado, debemos de asignarle un valor:



## 9. Dando este código con un error y tu tarea será utilizar el debugger para identificar y corregir el problema.

El error que encontramos en la función de sumar números es debido a que no estamos sumando las variables y tenemos que cambiar el símbolo de la operación para que quede tal que así:





## 10. Creamos un programa que realiza la búsqueda binaria en un array ordenado. Sin embargo, hay un error en el código que deberás identificar y corregir utilizando el debugger.

El error que encontramos se debe a que cuando buscamos un número nos devuelve el anterior y no el mismo que indicamos, para solucionarlo he añadido un 0 dentro del array de los números y así al buscar entienda como índice 0 el propio cero y no utilice el uno.

```
BusquedaBinaria.java X NumerosdelReves.java Amarillo.java
1 package entornos;
2
3 public class BusquedaBinaria {
4     public static void main(String[] args) {
5         int[] array = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
6         int elementoBuscar = 5;
7         int indiceEncontrado = busquedaBinaria(array, elementoBuscar);
8
9         if (indiceEncontrado != -1) {
10             System.out.println("El elemento " + elementoBuscar + " se encuentra en el índice " + indiceEncontrado + ".");
11         }
12         else {
13             System.out.println("El elemento " + elementoBuscar + " no se encuentra en el array.");
14         }
15     }
16
17     public static int busquedaBinaria(int[] array, int elementoBuscar) {
18         int inicio = 0;
19         int fin = array.length;
20
21         while (inicio <= fin) {
22             int medio = inicio + (fin - inicio) / 2;
23
24             if (array[medio] == elementoBuscar) {
25                 return medio;
26             }
27
28             else if (array[medio] < elementoBuscar) {
29                 inicio = medio + 1;
30             }
31             else {
32                 fin = medio - 1;
33             }
34         }
35
36         return -1; // Elemento no encontrado
37     }
38 }
39
40
```

Console X Debug Shell

<terminated> BusquedaBinaria [Java Application] C:\Users\Achimon\.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.1.v20241024-1700\jre\bin\El elemento 5 se encuentra en el índice 5.