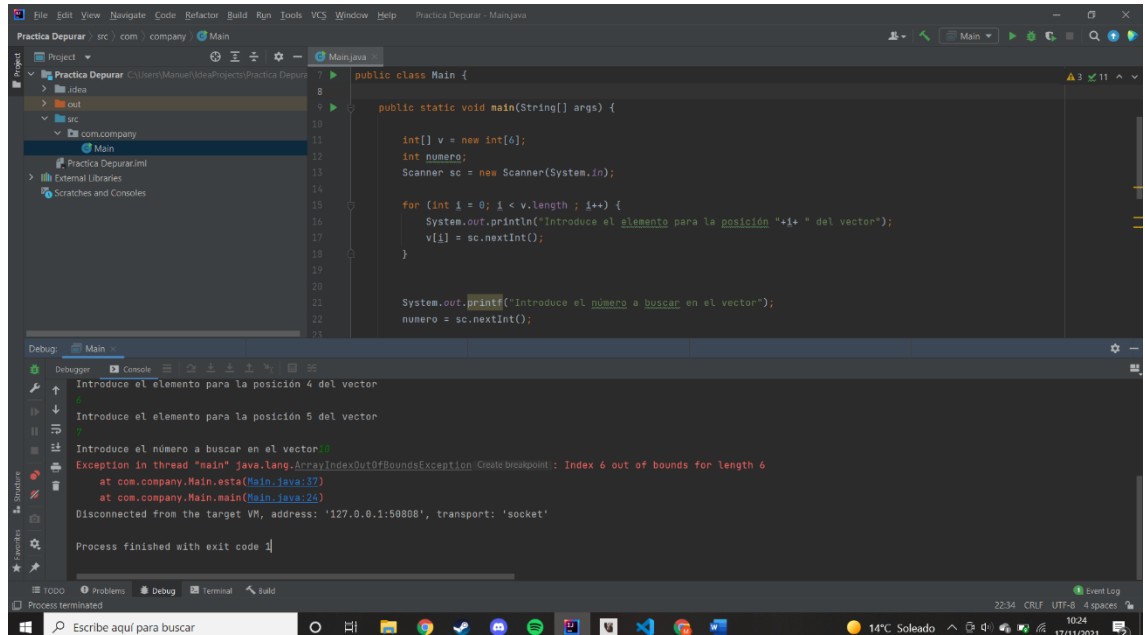


# Práctica Depurar

1. Ejecutar el programa , provocar el fallo del mismo y capturar un pantallazo del error que aparece. ¿Cuál es la hipótesis que según tu opinión ha provocado el fallo?



Lo que estaba provocando el fallo es que en el segundo bucle el vector para que al introducir un número que no este dentro del vector tenga salida (El elemento introducido no está en el vector), se debe poner en el bucle menor que el vector y no menor igual como está en el ejercicio.

2. Una vez corregido el error añade el código corregido al fichero y captura pantallazos de la consola donde se pueda ver que el programa funciona correctamente tanto para un caso en el que esté el elemento como para un caso en el que no esté el elemento.

```
3. package com.company;

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

public class Main {

    public static void main(String[] args) {

        int[] v = new int[6];
        int numero;
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < v.length ; i++) {
            System.out.println("Introduce el elemento para la
            posición "+i+ " del vector");
            v[i] = sc.nextInt();
        }
    }
}
```

```

        System.out.printf("Introduce el número a buscar en el
vector");
        numero = sc.nextInt();

        boolean resultado = esta(v,numero);

        if (resultado) {
            System.out.printf("El elemento "+numero+" está en el
vector "+ Arrays.toString(v));
        } else {
            System.out.printf("El elemento "+numero+" no está en
el vector"+ Arrays.toString(v));
        }
    }

    public static boolean esta(int[] vector, int elemento) {

        for (int i = 0; i < vector.length ; i++) {
            if (vector[i]==elemento) {
                return true;
            }
        }

        return false;
    }
}

```

The screenshot shows an IDE window titled 'Practica Depurar - Main.java'. The code editor displays the following Java code:

```

1 package con.company;
2
3
4 import java.util.Arrays;
5 import java.util.Scanner;
6
7 public class Main {
8
9     public static void main(String[] args) {
10
11         int[] v = new int[6];
12         int numero;
13         Scanner sc = new Scanner(System.in);
14
15         for (int i = 0; i < v.length ; i++) {
16             System.out.println("Introduce el elemento para la posición "+i+ " del vector");
17             v[i] = sc.nextInt();
18         }
19
20         System.out.printf("Introduce el número a buscar en el
vector");
21         numero = sc.nextInt();
22
23         boolean resultado = esta(v,numero);
24
25         if (resultado) {
26             System.out.printf("El elemento "+numero+" está en el
vector "+ Arrays.toString(v));
27         } else {
28             System.out.printf("El elemento "+numero+" no está en
el vector"+ Arrays.toString(v));
29         }
30     }
31
32     public static boolean esta(int[] vector, int elemento) {
33
34         for (int i = 0; i < vector.length ; i++) {
35             if (vector[i]==elemento) {
36                 return true;
37             }
38         }
39
40         return false;
41     }
42 }

```

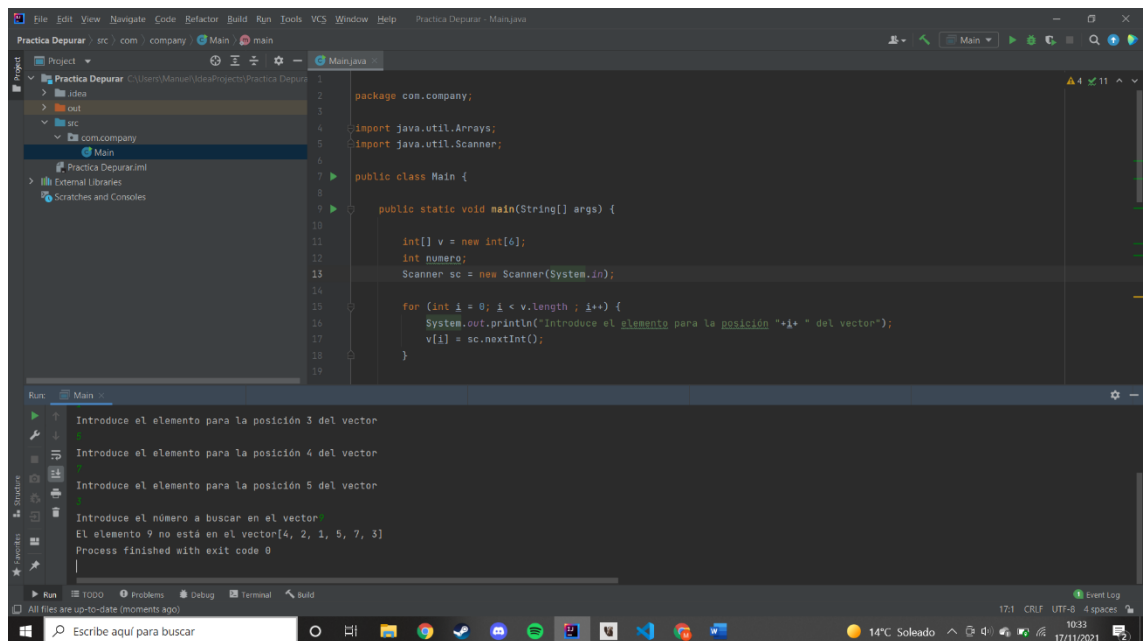
The Run console at the bottom shows the following output:

```

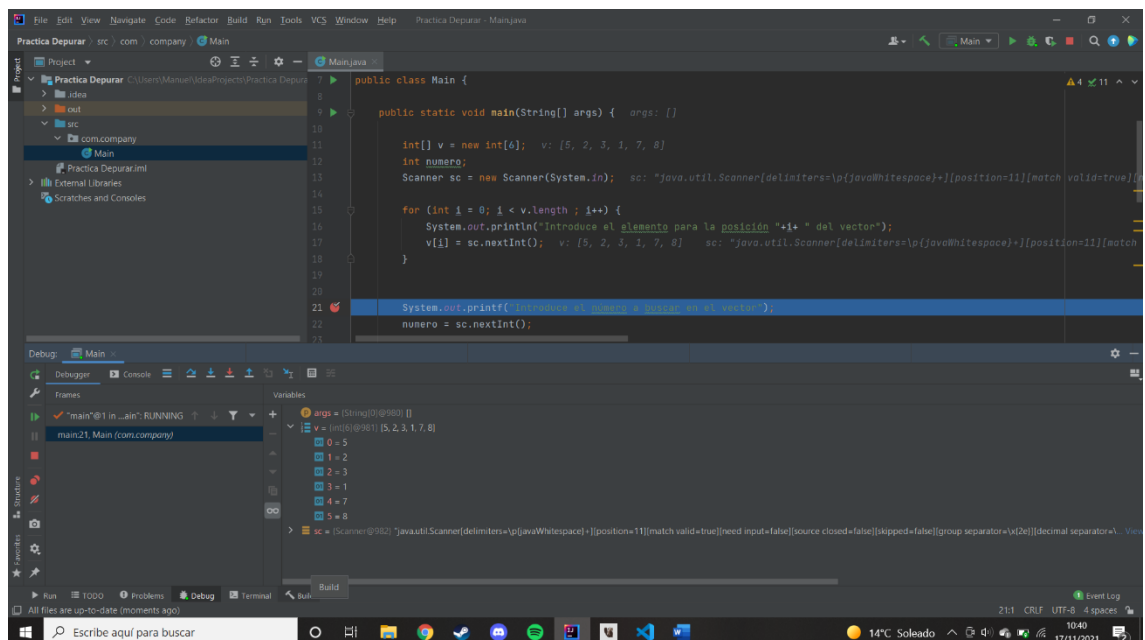
Introduce el elemento para la posición 3 del vector
3
Introduce el elemento para la posición 4 del vector
7
Introduce el elemento para la posición 5 del vector
2
Introduce el número a buscar en el vector:
2
El elemento 2 está en el vector [3, 6, 4, 1, 7, 2]
Process finished with exit code 0

```

The IDE interface includes a Project Explorer on the left showing the project structure, a Run console at the bottom, and a status bar at the very bottom indicating the build status and system information.



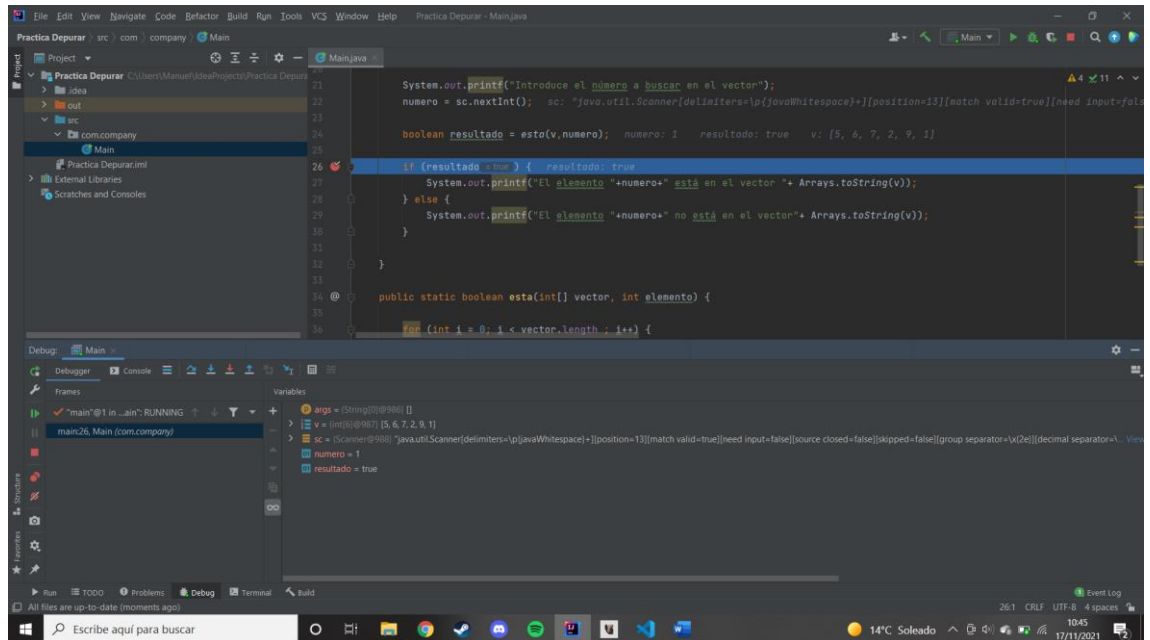
3. Establecer un breakpoint para comprobar que `v` se rellena correctamente. Justifica por qué has elegido establecer el breakpoint en esa línea de código e incluye una captura de pantalla donde se pueda comprobar que se ha parado la ejecución y que muestre en el área de depuración "Variables" todos los elementos del vector.



He establecido el breakpoint en la línea 21 porque es cuando el bucle acaba, y donde ya puedo comprobar si se ha rellenado el vector.

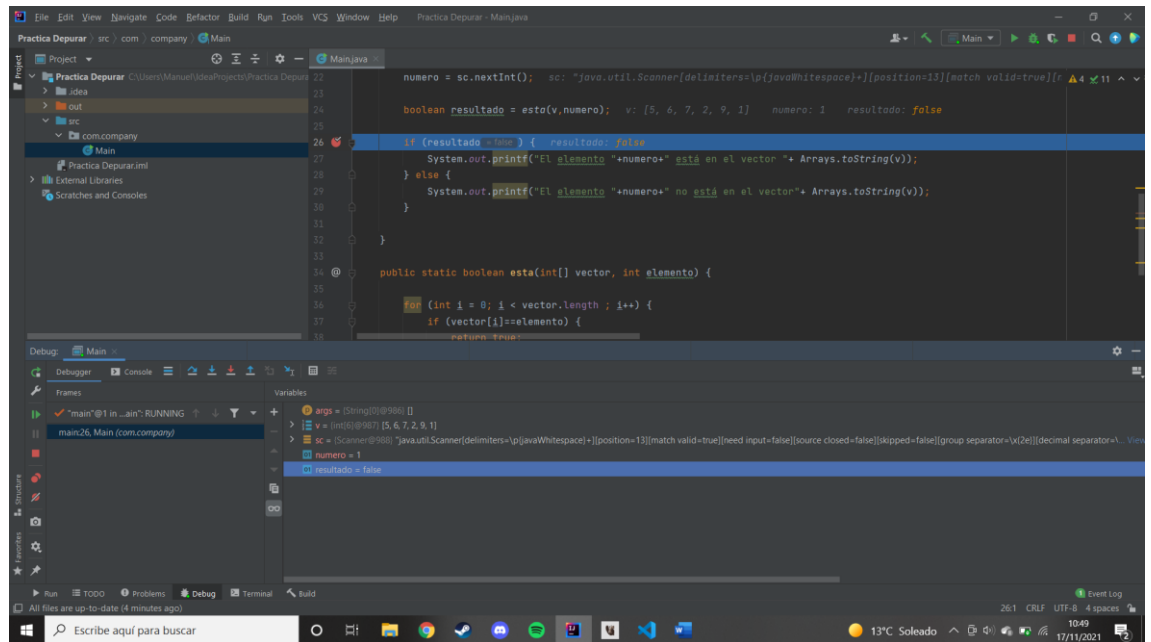
4. Establecer un breakpoint en el programa principal que nos permita comprobar que la función está devolviendo un valor true (deberás

introducir los valores necesarios para así sea). Justificar por qué has elegido establecer el breakpoint en esa línea de código. Incluye un pantallazo donde se pueda comprobar que se efectivamente se ha devuelto el resultado esperado (true).



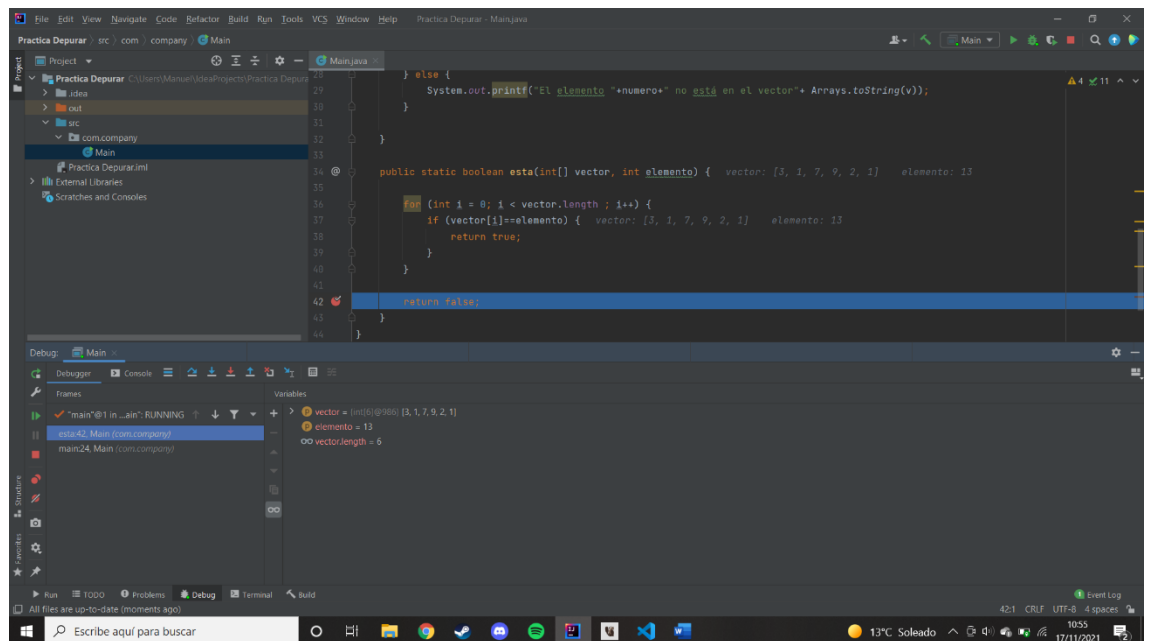
Establezco el breakpoint en la línea 26 porque es donde se visualiza el resultado de la condición del boolean si es true o false.

5. **Modificar el valor de resultado obtenido en el apartado anterior a false desde la zona de Variables. Describir el proceso realizado y capturar un pantallazo donde se pueda comprobar que el mensaje ya no es el adecuado "No está" y donde se pueda ver en la zona de variables que efectivamente se ha modificado el resultado de la función.**



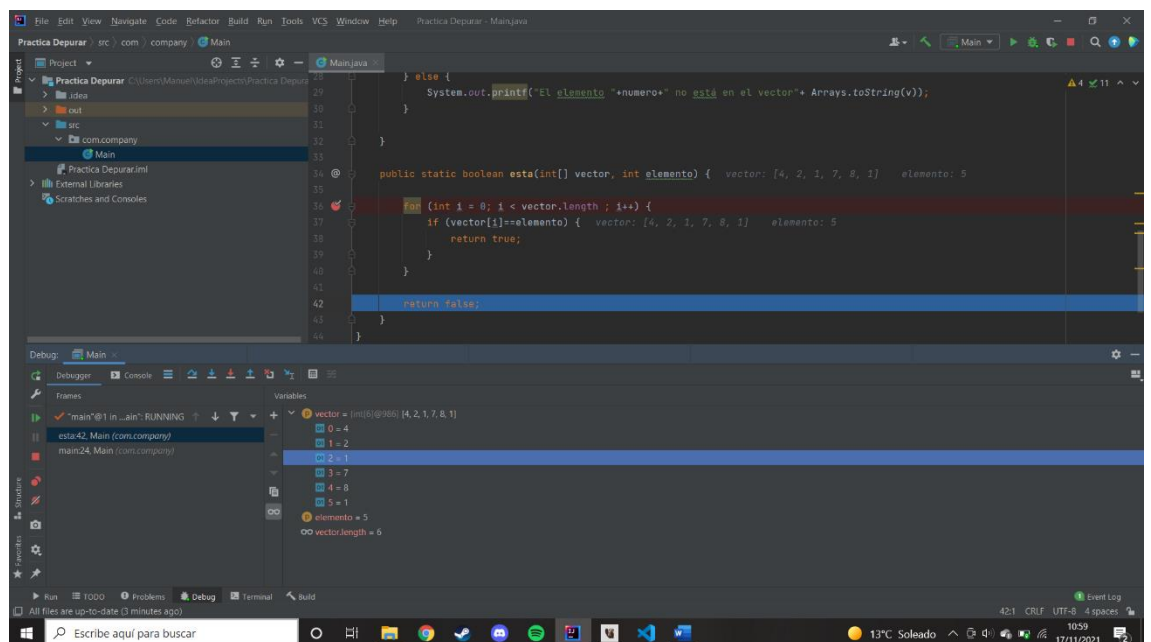
El proceso que he hecho es el siguiente:

1. Establecer el breakpoint en la condición booleana.
2. Situarse en el *Debugger*.
3. Hacer click derecho en el resultado del boolean y seleccionar *Set Value*.
4. Dentro en vez de true introduce false.
5. Por ultimo hacer un Step Over.
6. Establecer un breakpoint dentro de la función y realizar un pantallazo donde se pueda comprobar que se ha parado la ejecución y el contenido del vector recibido.



- Utilizando la zona de variables y una vez se ha parado en el breakpoint anterior modificar el contenido del vector para modificar el resultado previsto en la función. Si debía de ser true pasará a ser false y al revés. Incluid los pantallazos necesarios para justificar dicho cambio.

Antes del cambio:



Después del cambio:

