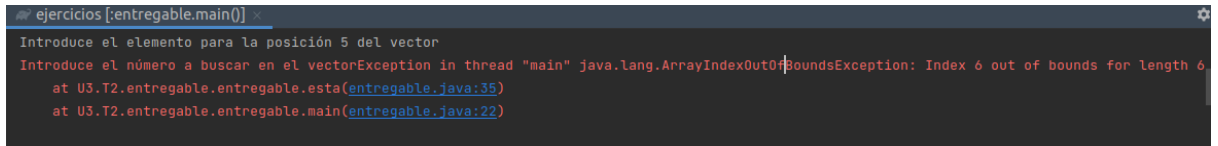


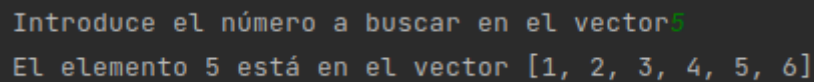
1. Ejecutar el programa , provocar el fallo del mismo y capturar un pantallazo del error que aparece. ¿Cuál es la hipótesis que según tu opinión ha provocado el fallo?



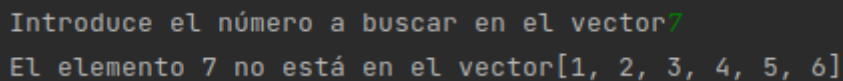
```
ejercicios [:entregable.main()] x
Introduce el elemento para la posición 5 del vector
Introduce el número a buscar en el vectorException in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 6 out of bounds for length 6
    at U3.T2.entregable.entregable.esta(entregable.java:35)
    at U3.T2.entregable.entregable.main(entregable.java:22)
```

El fallo se provoca porque el bucle intenta llegar hasta un número mayor que la cantidad de números que contiene.

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.



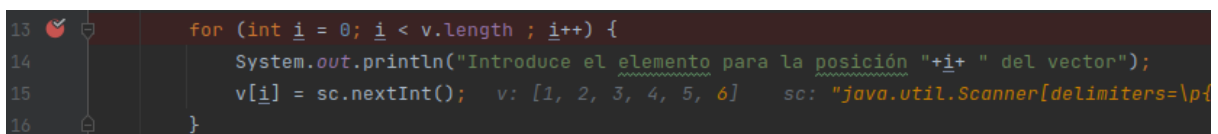
```
Introduce el número a buscar en el vector5
El elemento 5 está en el vector [1, 2, 3, 4, 5, 6]
```



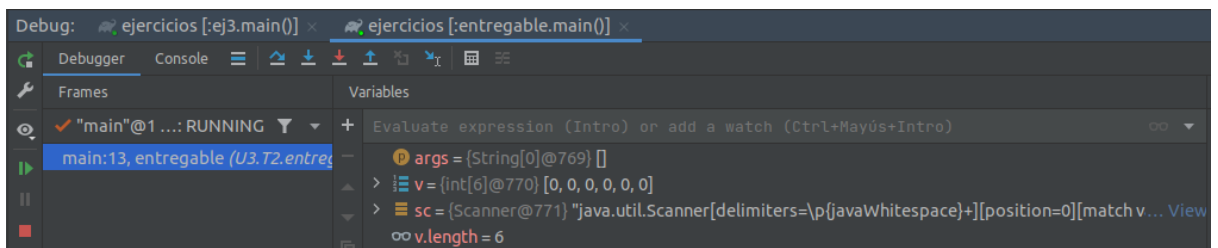
```
Introduce el número a buscar en el vector7
El elemento 7 no está en el vector[1, 2, 3, 4, 5, 6]
```

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.

Establezco el breakpoint en la línea 13, donde empieza el bucle para rellenar el vector.



```
13 for (int i = 0; i < v.length ; i++) {
14     System.out.println("Introduce el elemento para la posición "+i+ " del vector");
15     v[i] = sc.nextInt(); v: [1, 2, 3, 4, 5, 6] sc: "java.util.Scanner[delimiters=\p+
16 }
```



Debug: ejercicios [:ej3.main()] x ejercicios [:entregable.main()] x

Debugger Console

Frames

Variables

Evaluate expression (Intro) or add a watch (Ctrl+Mayús+Intro)

main:13, entregable (U3.T2.entregable)

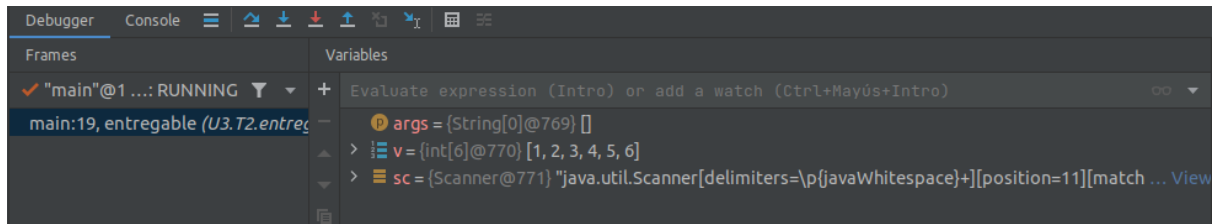
args = {String[0]@769} []

v = {int[6]@770} [0, 0, 0, 0, 0, 0]

sc = {Scanner@771} "java.util.Scanner[delimiters=\p+(javaWhitespace)+][position=0][match v... View

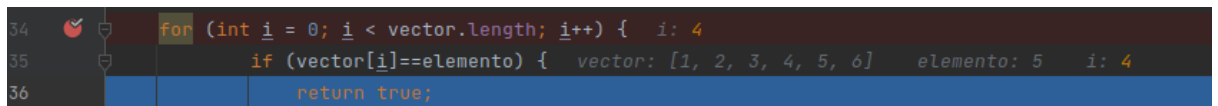
v.length = 6

Vemos como el programa se para correctamente antes de pedirnos que introduzcamos los datos en el array.

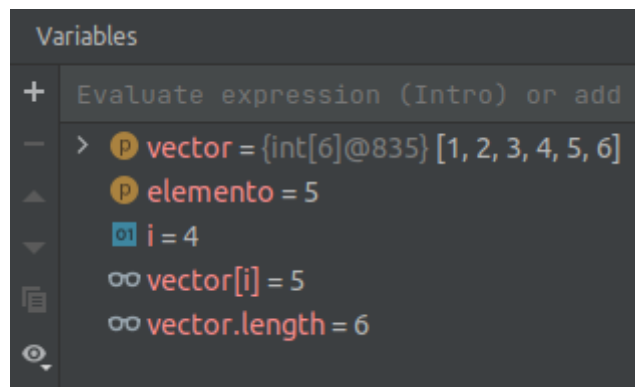


En esta imagen, vemos como quedaría el vector, cuando ya hemos rellenado los datos, uno a uno presionando F8 para avanzar de paso.

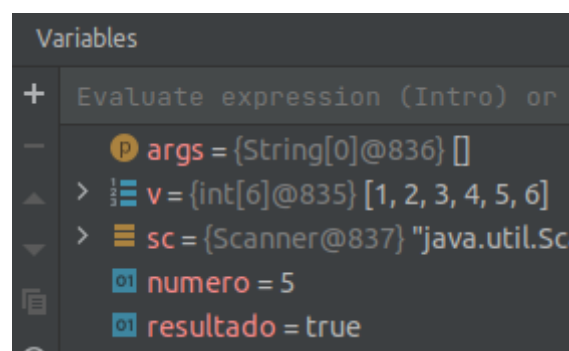
4. Establecer un breakpoint en el programa principal que nos permita comprobar que la función está devolviendo un valor true (deberás introducir los valores necesarios para así sea). Justificar por qué has elegido establecer el breakpoint en esa línea de código. Incluye un pantallazo donde se pueda comprobar que efectivamente se ha devuelto el resultado esperado (true).



Elijo el breakpoint en la línea 34, ya que es donde comienza el bucle para comprobar si el número se encuentra dentro del vector, y en ese caso devolver 'true'.



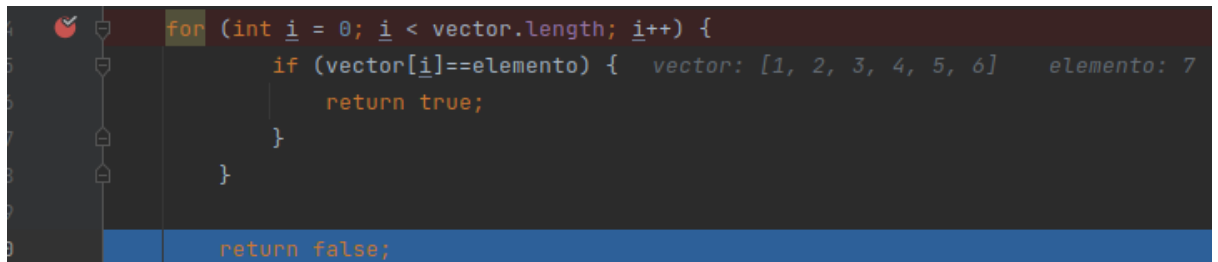
Vemos como 'elemento' vale 5 y 'vector[i]', también vale 5 por lo que nos devuelve 'true'.



Aquí comprobamos como el resultado del ejercicio es 'true'.

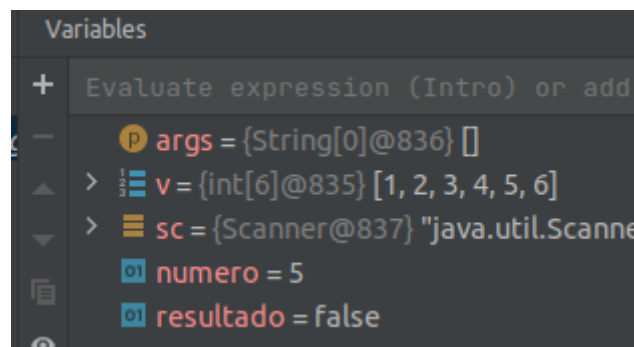
5. Modificar el valor de de resultado obtenido en el apartado anterior a false desde la zona de Variables. Describir el proceso realizado y capturar un pantallazo donde se pueda comprobar que el mensaje ya no es el adecuado "No está" y donde se pueda ver en la zona de variables que efectivamente se ha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

En la zona de Variables, pulsamos en la que queremos modificar, y apretamos botón derecho del ratón y le damos a 'Set value' o solo presionamos F2.



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

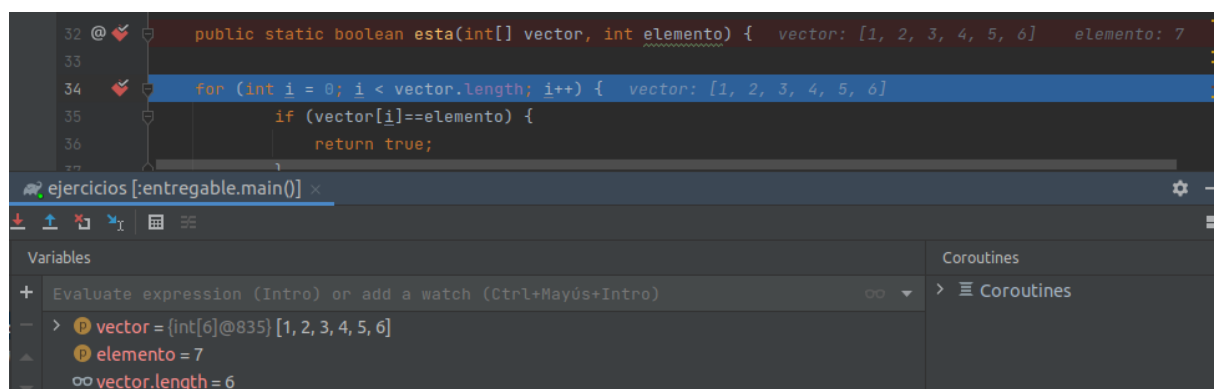
Hemos cambiado el número de '5' a '7'.



Variables	
+	Evaluate expression (Intro) or add
-	args = {String[0]@836} []
>	v = {int[6]@835} [1, 2, 3, 4, 5, 6]
>	sc = {Scanner@837} "java.util.Scanned"
01	numero = 5
01	resultado = false

Aquí vemos como el resultado de la función es 'false'.

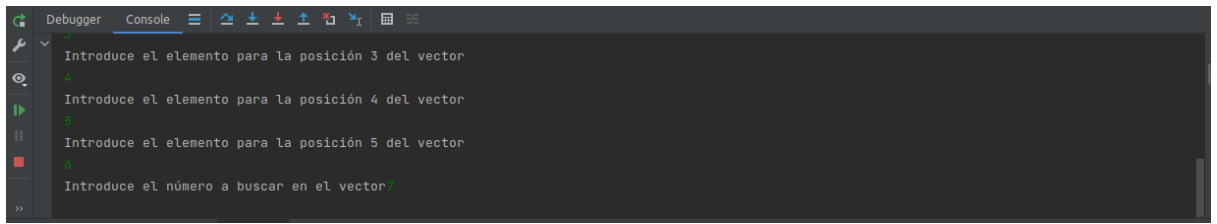
6. Establecer un breakpoint dentro de la función y realizar un pantallazo donde se pueda comprobar que se ha parado la ejecución y el contenido del vector recibido.



```
public static boolean esta(int[] vector, int elemento) {  
    for (int i = 0; i < vector.length; i++) {  
        if (vector[i] == elemento) {  
            return true;  
        }  
    }  
}
```

Variables		Coroutines
+	Evaluate expression (Intro) or add a watch (Ctrl+Mayús+Intro)	> Coroutines
-	vector = {int[6]@835} [1, 2, 3, 4, 5, 6]	
	elemento = 7	
	vector.length = 6	

Pongo el breakpoint en la línea 34, donde ya ha recibido el contenido del vector, y vemos como se ha parado la ejecución.



7. 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.

En el vector, pulsamos botón derecho, 'Inspect' y en la ventana emergente, cambiamos cualquier parámetro, y le damos el valor del elemento, para que en el resultado fuera 'true', en lugar de 'false' que es lo que tendría que haber devuelto anteriormente, dado que el valor '7' no estaba entre los valores del vector.

