




# DEBUGGING

ENTORNOS DE DESARROLLO

Mijael Tamayo Onofre

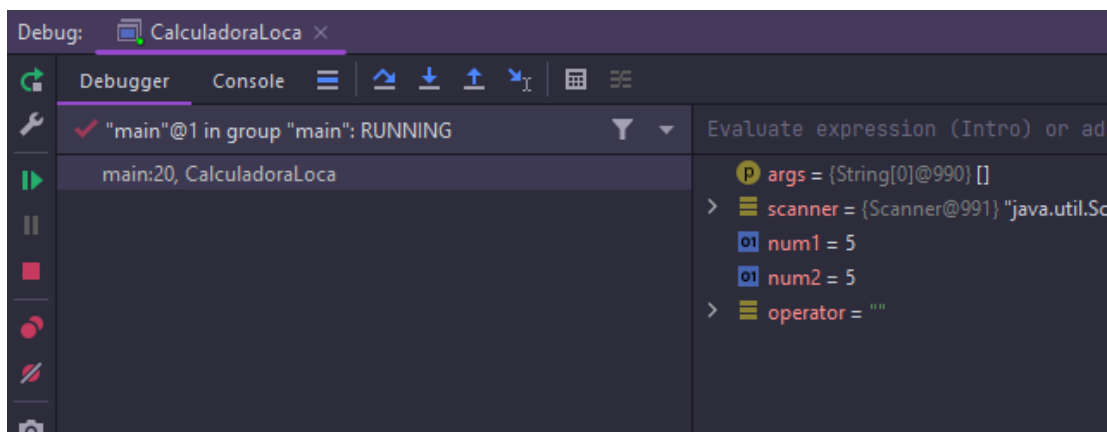


Al ejecutar la calculadora, solo nos lee los dos primeros números que introducimos. Podemos observar que no nos deja introducir el operador.

```
Introduce el primer número: 5
Introduce el segundo número: 5
Introduce el operador (+, -, *, /): Operador no válido!
Resultado: 0.0
El resultado es igual o menor de 10.
El resultado es 0.

Process finished with exit code 0
```

Que no nos lee el operador, nos sale “Operador no válido” sin preguntarnos que tipo de operador queremos. Lo que tenemos que hacer es cambiar el tipo de scanner y colocarle `scanner.next()`



Cambiamos el scanner de operator.

```
System.out.print("Introduce el primer número: ");
num1 = scanner.nextInt();

System.out.print("Introduce el segundo número: ");
num2 = scanner.nextInt();

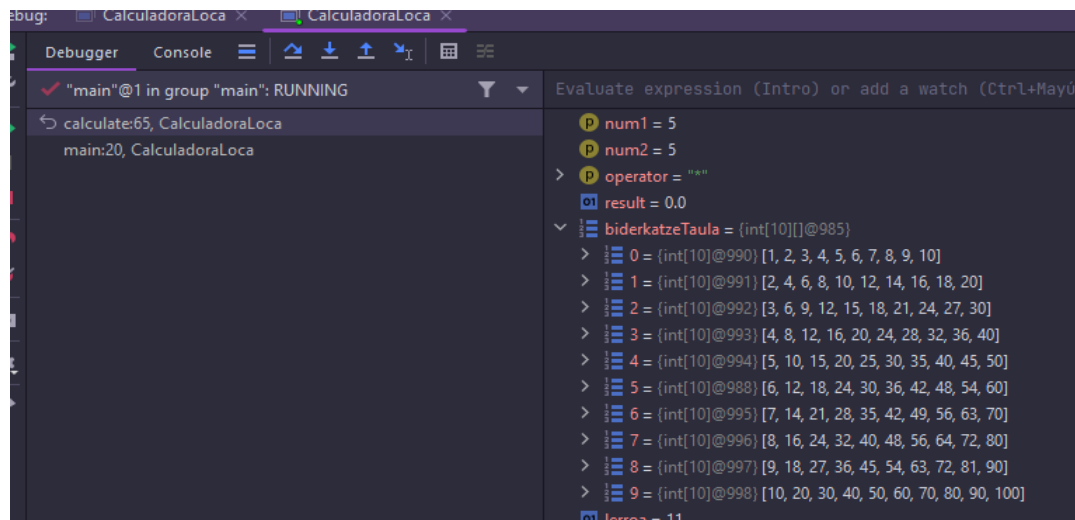
System.out.print("Introduce el operador (+, -, *, /): ");
operator = scanner.next();
```

Al observar la imagen, nos damos cuenta que el resultado dado, no es el correcto.

```
C:\Users\ik012982i9\.jdk\corretto-17.0.
Introduce el primer número: 5
Introduce el segundo número: 5
Introduce el operador (+, -, *, /): *
Resultado: 36.0
El resultado es mayor de 10.
El resultado es positivo.

Process finished with exit code 0
```

Utilizando el Debugger podemos ver de cómo está compuesto el for y que es lo que almacena cada posición.



El error aquí es que los for empezaban en 0, deberíamos de iniciarlos en 1 para que puedan funcionar, porque si lo dejamos así no va, ya que en cada posición hay un número y como empieza en 0 el for, no ser

```
case "*":
    int[][] biderkatzeTaula = new int[10][10];
    int lerroa = 1, zutabea = 1;

    for(int a = 1; a < biderkatzeTaula.length; a++)
    {
        for(int b = 1; b < biderkatzeTaula[a].length; b++)
        {
            biderkatzeTaula[a][b] = lerroa * zutabea;
            zutabea = zutabea + 1;
        }
        lerroa = lerroa + 1;
        zutabea = 1;
    }

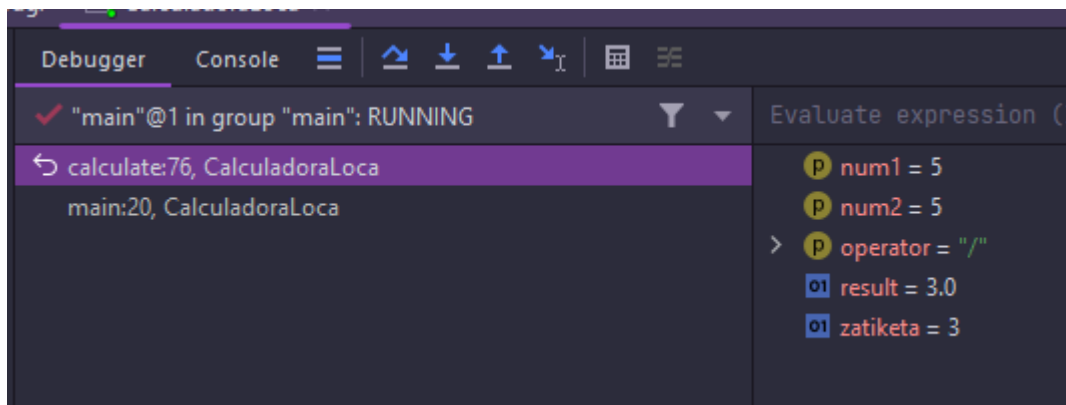
    result = biderkatzeTaula[num1][num2];
    break;
```

Al escoger el operador de división, nos salta un error en el resultado

```
Introduce el primer número: 20
Introduce el segundo número: 5
Introduce el operador (+, -, *, /): /
Resultado: 3.0
El resultado es igual o menor de 10.
El resultado es positivo.

Process finished with exit code 0
```

Al observar el código vemos que la variable `zaticeta` empieza en 0, debemos de inicializarla en 1, porque lo que hace esta variable es contar cuantas veces resta `num2` de `num1` y así sucesivamente hasta que `num1` sea menor que `num2`. Este `while` para cuando ambos sean iguales



```
case "/":
    int zaticeta=1;

    while(num1>num2){
        num1-=num2;
        ++zaticeta;
    }

    result = zaticeta;
    break;
```