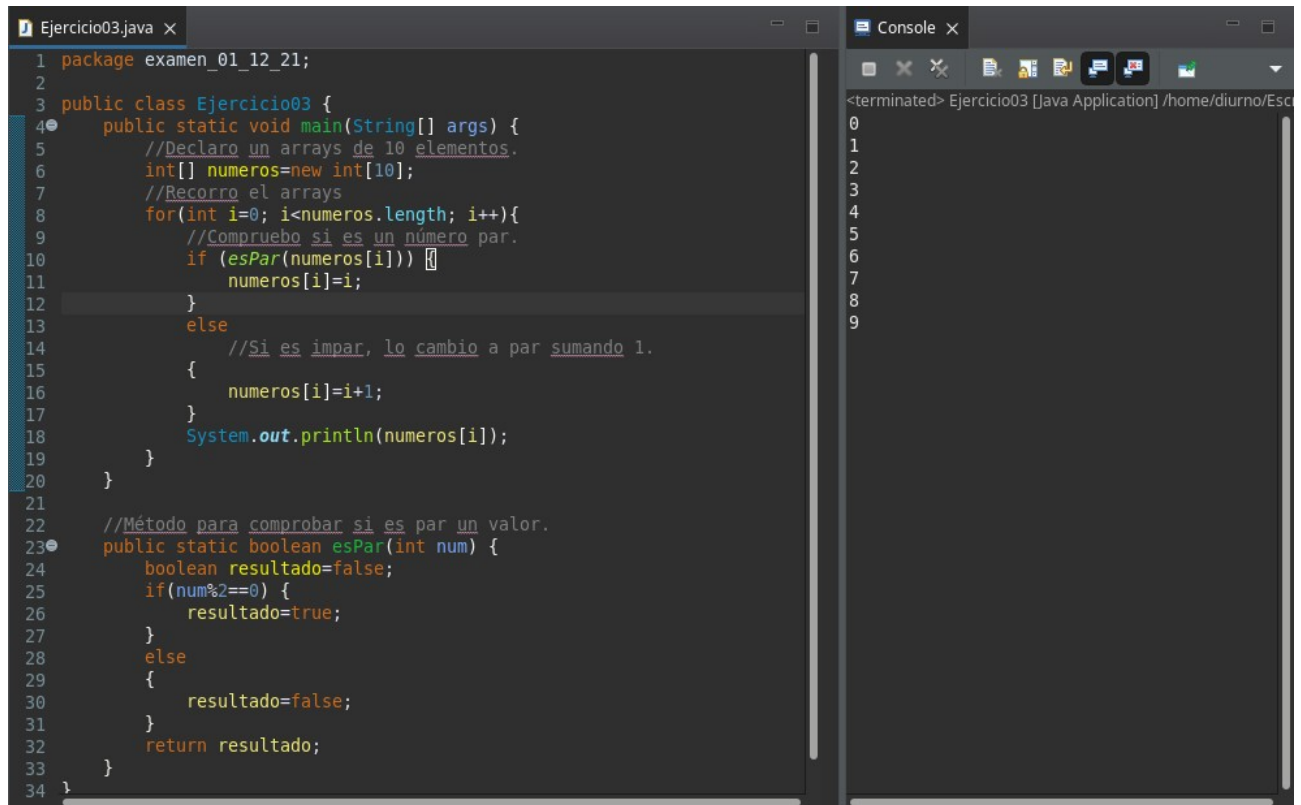


## Examen 01/12/2021 Ejercicio 03.-

En este ejercicio vamos a depurar el código intentando identificar cual es el problema, el primer paso será ejecutar el código para entender lo que está haciendo y, por lo tanto, mostrando en pantalla.

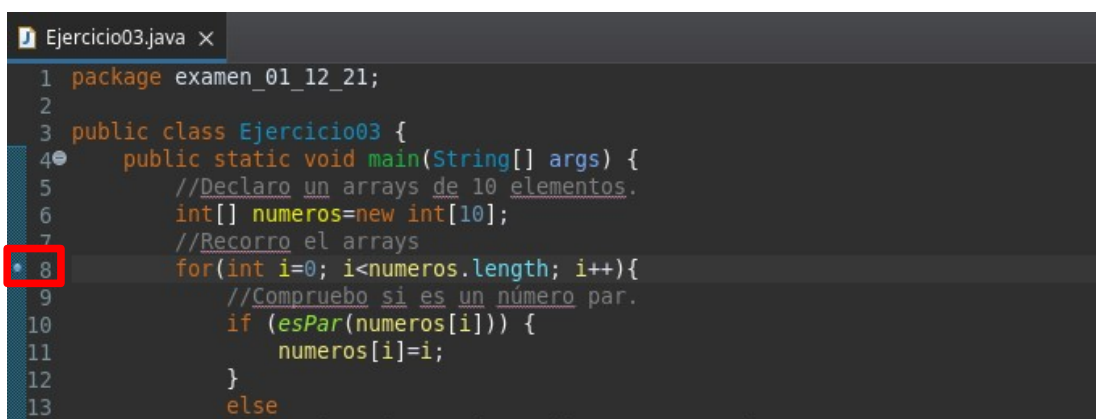


```
1 package examen_01_12_21;
2
3 public class Ejercicio03 {
4     public static void main(String[] args) {
5         //Declaro un arrays de 10 elementos.
6         int[] numeros=new int[10];
7         //Recorro el arrays
8         for(int i=0; i<numeros.length; i++){
9             //Compruebo si es un número par.
10            if (esPar(numeros[i])) {
11                numeros[i]=i;
12            }
13            else
14                //Si es impar, lo cambio a par sumando 1.
15                {
16                    numeros[i]=i+1;
17                }
18            System.out.println(numeros[i]);
19        }
20    }
21
22    //Método para comprobar si es par un valor.
23    public static boolean esPar(int num) {
24        boolean resultado=false;
25        if(num%2==0) {
26            resultado=true;
27        }
28        else
29        {
30            resultado=false;
31        }
32        return resultado;
33    }
34 }
```

Console

```
<terminated> Ejercicio03 [Java Application] /home/diurno/Esc
0
1
2
3
4
5
6
7
8
9
```

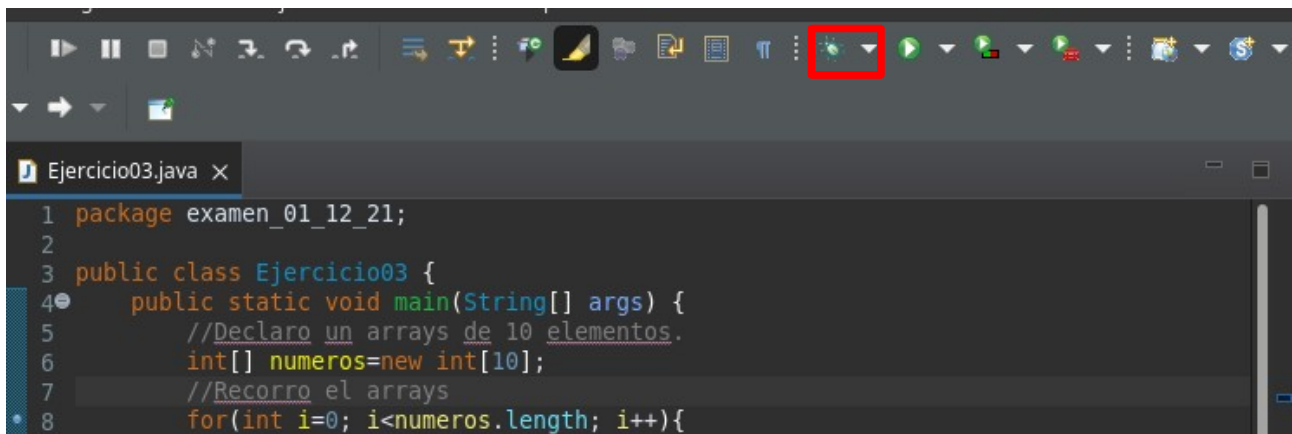
Como vemos, el problema es que los números impares no los suprime, vamos a proseguir depurando el código para identificar el problema. Para ello vamos a introducir un punto de interrupción, simplemente haciendo click derecho en la parte izquierda de la clase, encima el número de fila donde queremos posicionarlo, seleccionamos la opción “Toggle BreakPoint”.



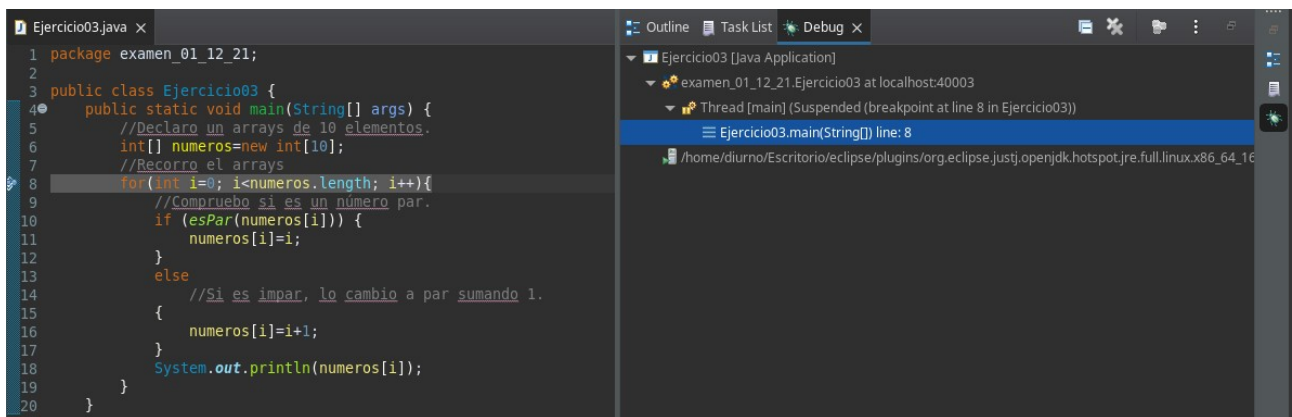
```
1 package examen_01_12_21;
2
3 public class Ejercicio03 {
4     public static void main(String[] args) {
5         //Declaro un arrays de 10 elementos.
6         int[] numeros=new int[10];
7         //Recorro el arrays
8         for(int i=0; i<numeros.length; i++){
9             //Compruebo si es un número par.
10            if (esPar(numeros[i])) {
11                numeros[i]=i;
12            }
13            else
```

Empezaremos por la línea 8, ya que es lo primero que está en el código además de la declaración del array.

En las opciones de arriba vamos a seleccionar la señalada, ya que es la herramienta de debug.

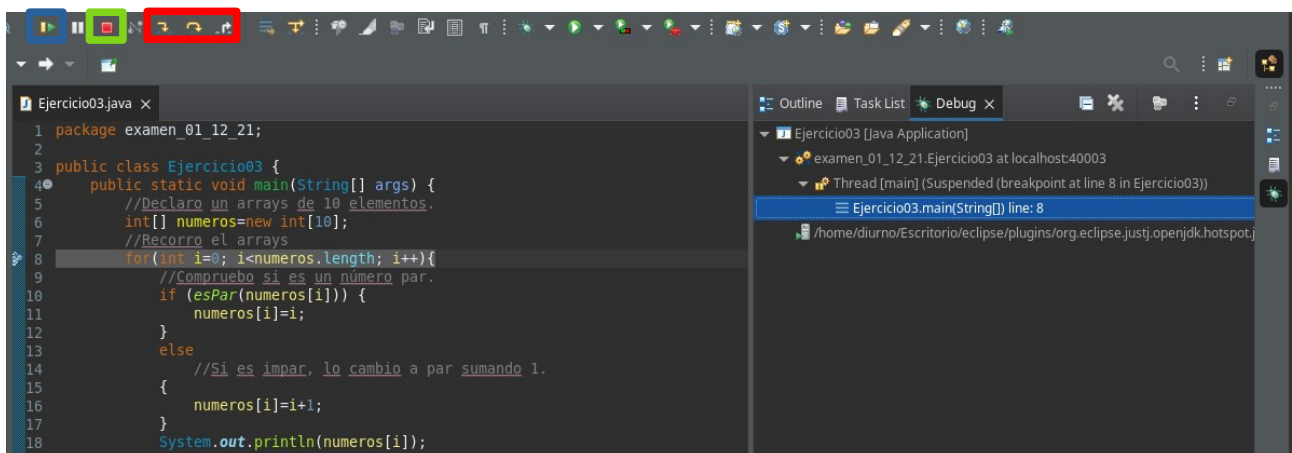


Esta es la interfaz de la herramienta de debug, que indica la información necesaria.



Como vemos a la derecha nos indica información varia de la herramienta además de donde está ahora mismo detenido el programa, como vemos en lo señalado en azul, en la línea 8.

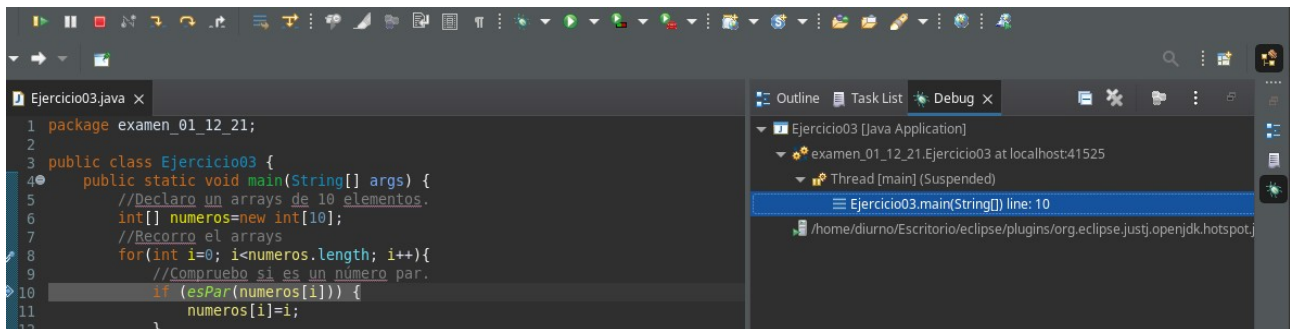
Lo que necesitamos ahora es saltar a las siguientes líneas para ir poco a poco analizando línea a línea para saber cual es el problema. Para ellos vemos tres botones que hacen lo siguiente.



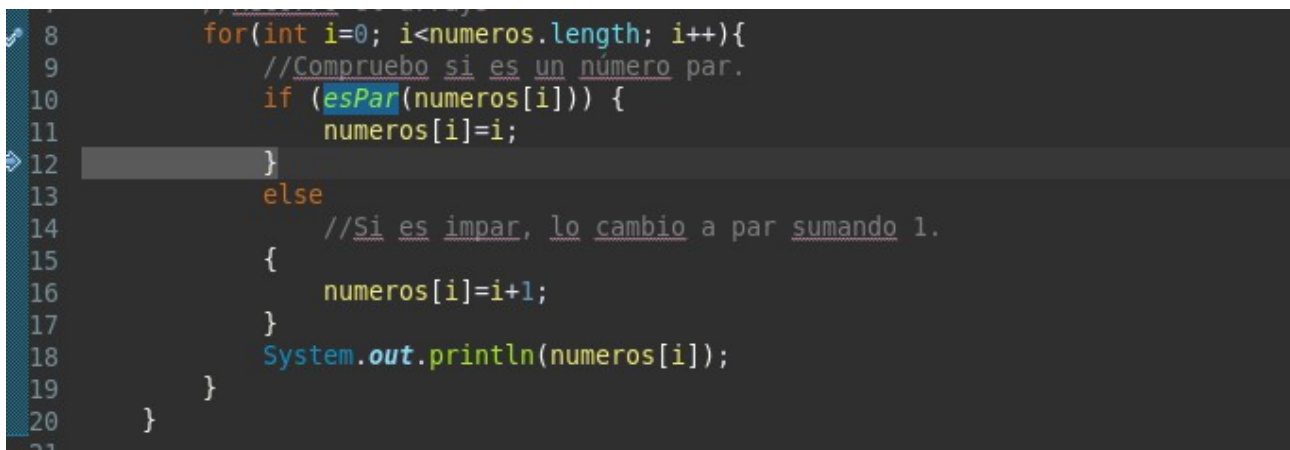
Nosotros nos vamos a enfocar en el segundo “Step Over” que su funcionamiento es saltar de línea a línea en nuestro código a partir de punto de interrupción indicado.

Si queremos saber si el programa está en ejecución es tan simple como mirar el botón rojo cuadrado (señalado en verde), utilizado para parar la depuración, al lado vemos el botón “Resume” (señalado en azul), este sirve para pausar o reanudar la depuración.

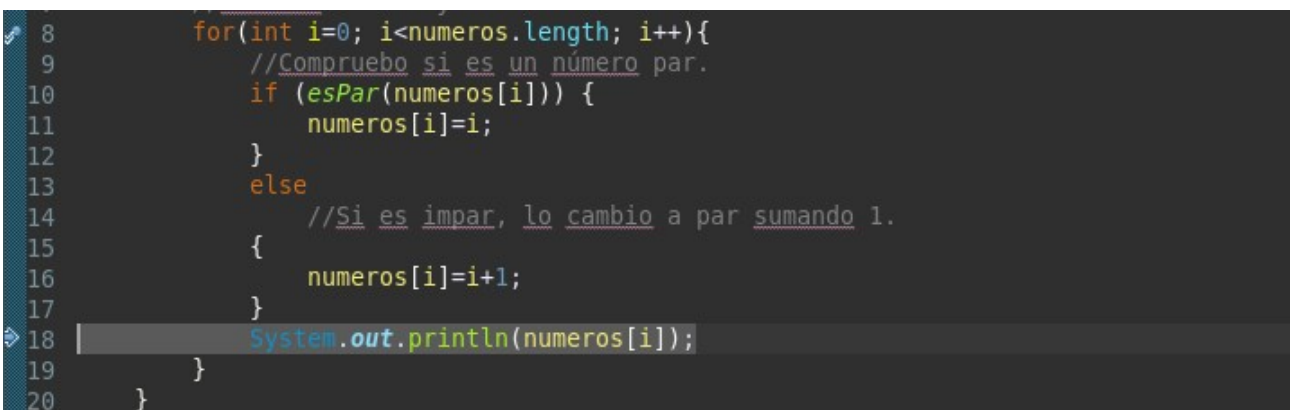
Continuamos con el “Step Over” o pulsando la tecla del teclado f6.



Como vemos hemos saltado a la siguiente línea, ahora lo que queda es comprobar que está mal. Al depurar un poco nos damos cuenta que nunca se mete en el else, siempre salta de esta línea:



a esta otra:



Vamos a introducir un nuevo punto de interrupción en la línea 23 para entrar al segundo método, así podremos comprobar que esta ocurriendo exactamente.

```
22 //Método para comprobar si es par un valor.  
23 public static boolean esPar(int num) {  
24     boolean resultado=false;  
25     if(num%2==0) {  
26         resultado=true;  
27     }  
28     else  
29     {  
30         resultado=false;  
31     }  
32     return resultado;  
33 }  
34 }  
35
```

Ahora vemos que cuando la depuración se detiene en el “if” del main la siguiente interrupción va al segundo método comprobando los valores y devolviendo un valor, cumpliendo su correcto funcionamiento, esto nos daría una pista de que lo que está fallando está en el main.

```
8 for(int i=0; i<numeros.length; i++){  
9 //Compruebo si es un número par.  
10 if (esPar(numeros[i])) {  
11     numeros[i]=i;  
12 }  
13 else  
14 //Si es impar, lo cambio a par suma  
15 {  
16     numeros[i]=i+1;  
17 }  
18 System.out.println(numeros[i]);  
19 }  
20 }  
21  
22 //Método para comprobar si es par un valor.  
23 public static boolean esPar(int num) {  
24     boolean resultado=false;  
25     if(num%2==0) {  
26         resultado=true;  
27     }  
28     else  
29     {  
30         resultado=false;  
31     }  
32     return resultado;  
33 }  
34 }  
35
```

```
8 for(int i=0; i<numeros.length; i++){  
9 //Compruebo si es un número par.  
10 if (esPar(numeros[i])) {  
11     numeros[i]=i;  
12 }  
13 else  
14 //Si es impar, lo cambio a par suma  
15 {  
16     numeros[i]=i+1;  
17 }  
18 System.out.println(numeros[i]);  
19 }  
20 }  
21  
22 //Método para comprobar si es par un valor.  
23 public static boolean esPar(int num) {  
24 boolean resultado=false;  
25 if(num%2==0) {  
26     resultado=true;  
27 }  
28 else  
29 {  
30     resultado=false;  
31 }  
32 return resultado;  
33 }  
34 }  
35
```

Vamos a centrarnos en el main, especialmente en el if, ya que si nunca entra en el else es porque probablemente el error se encuentre en este punto.

Al pensar un poco nos damos cuenta de que le estamos devolviendo al método un valor erróneo, ya que le estamos pasando el valor de la posición del array y le deberíamos de estar pasando la posición en sí. Así que si cambiamos esto:

```

7 //Recorro el arrays
8 for(int i=0; i<numeros.length; i++){
9     //Compruebo si es un número par.
10    if (esPar(numeros[i])) {
11        numeros[i]=i;
12    }
13    else
14        //Si es impar, lo cambio a par sumando 1.
15    {
16        numeros[i]=i+1;
17    }
18    System.out.println(numeros[i]);
19 }
20 }

```

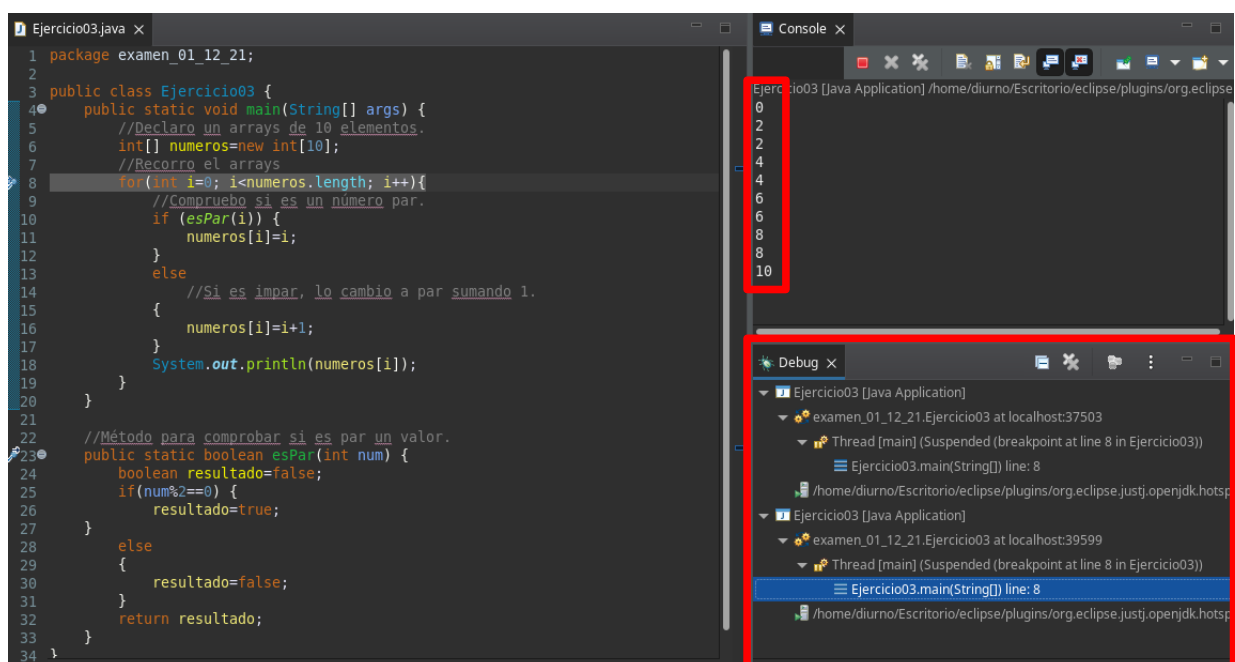
A esto:

```

7 //Recorro el arrays
8 for(int i=0; i<numeros.length; i++){
9     //Compruebo si es un número par.
10    if (esPar(i)) {
11        numeros[i]=i;
12    }
13    else
14        //Si es impar, lo cambio a par sumando 1.
15    {
16        numeros[i]=i+1;
17    }
18    System.out.println(numeros[i]);
19 }
20 }

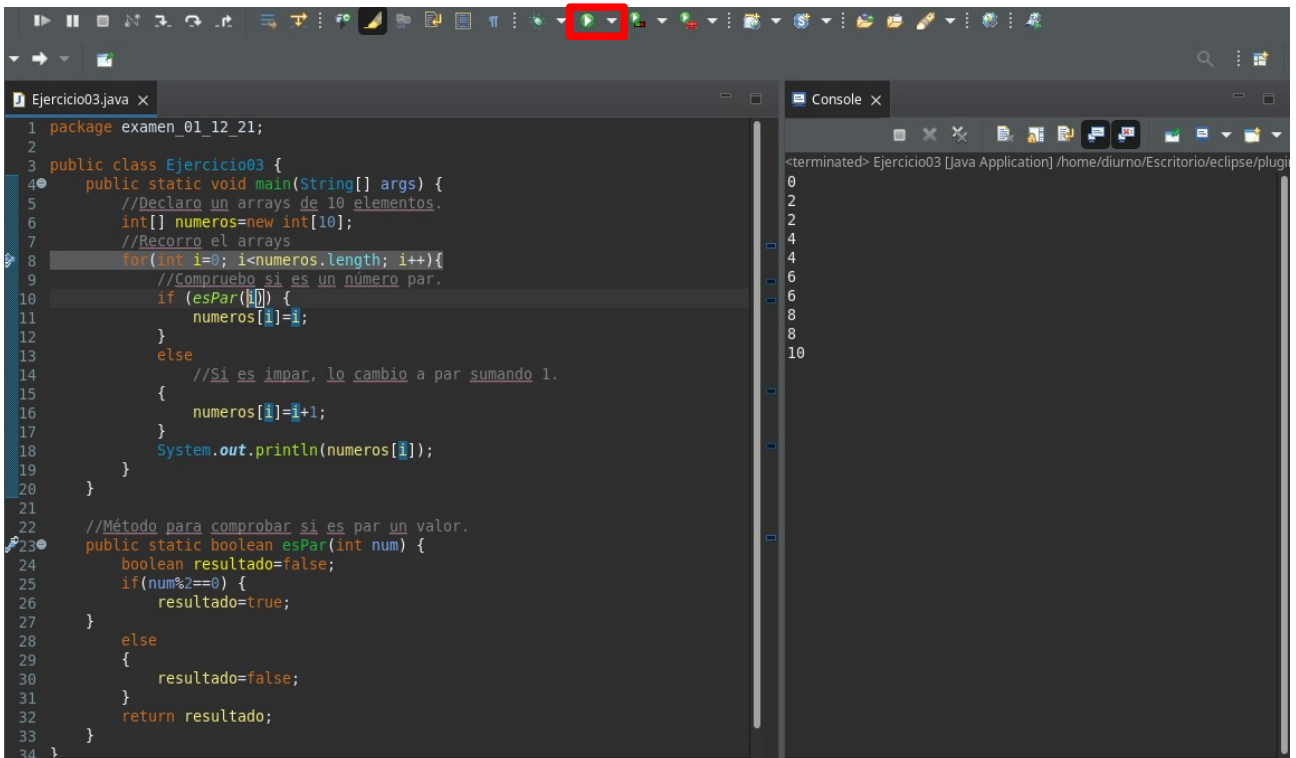
```

Y depuramos ahora si el código, vemos que funciona a la perfección, devolviendo en pantalla los valores indicados.





Una vez depurado y solucionado correctamente paramos la depuración y ejecutamos el programa con tranquilidad de que todo está correcto.



```
1 package examen_01_12_21;
2
3 public class Ejercicio03 {
4     public static void main(String[] args) {
5         //Declaro un array de 10 elementos.
6         int[] numeros=new int[10];
7         //Recorro el array
8         for(int i=0; i<numeros.length; i++){
9             //Compruebo si es un número par.
10            if (esPar(i)) {
11                numeros[i]=i;
12            }
13            else
14                //Si es impar, lo cambio a par sumando 1.
15                {
16                    numeros[i]=i+1;
17                }
18            System.out.println(numeros[i]);
19        }
20    }
21
22    //Método para comprobar si es par un valor.
23    public static boolean esPar(int num) {
24        boolean resultado=false;
25        if(num%2==0) {
26            resultado=true;
27        }
28        else
29        {
30            resultado=false;
31        }
32        return resultado;
33    }
34 }
```

<terminated> Ejercicio03 [Java Application] /home/diurno/Escritorio/eclipse/plugi  
0  
2  
2  
4  
4  
6  
6  
8  
8  
10