

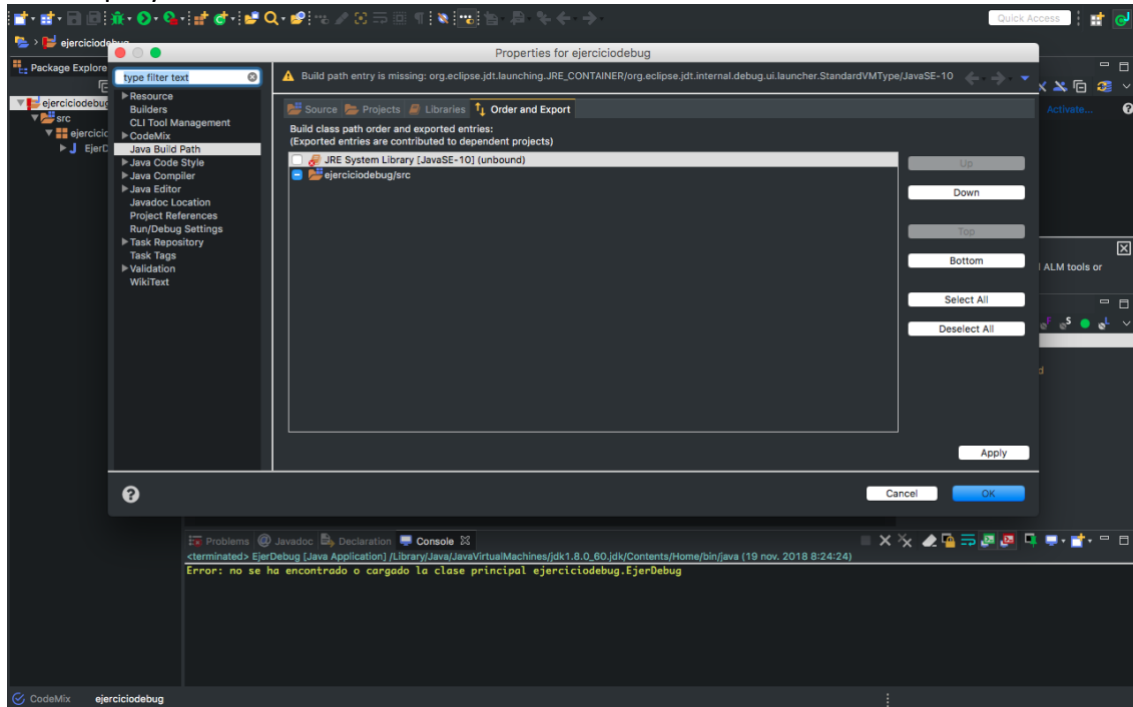
Nombre: Alexis Calderón

1DAW

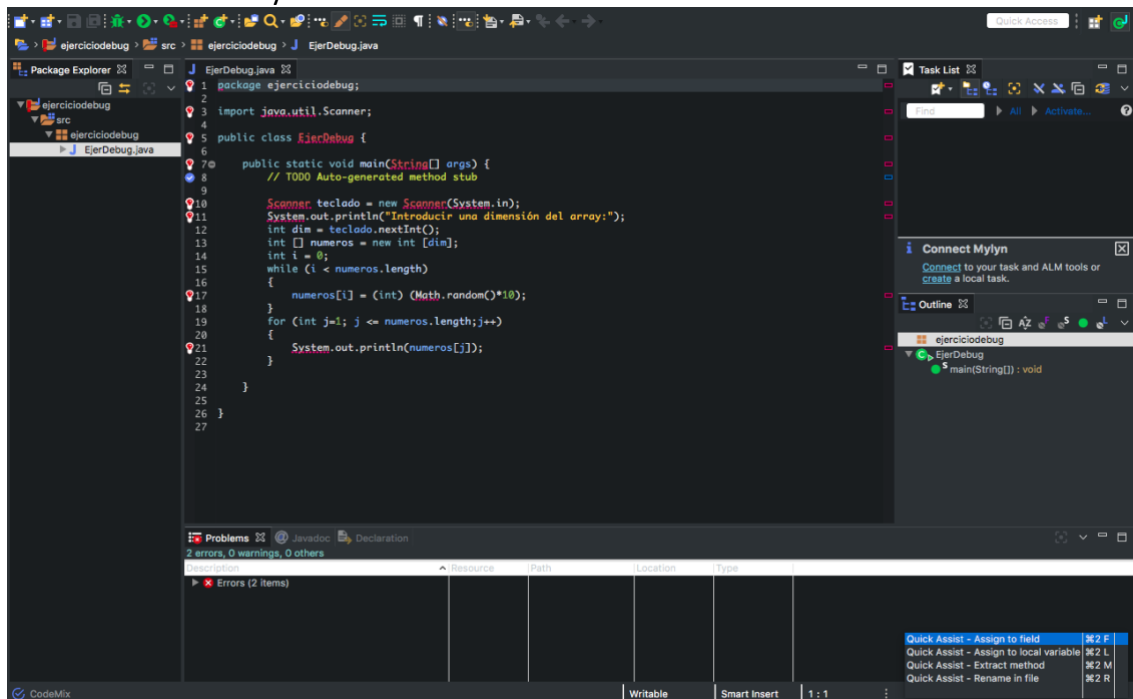
Ejercicio Debugger

Dado un proyecto en java usar el debugger para obtener el posible fallo y solucionarlo.

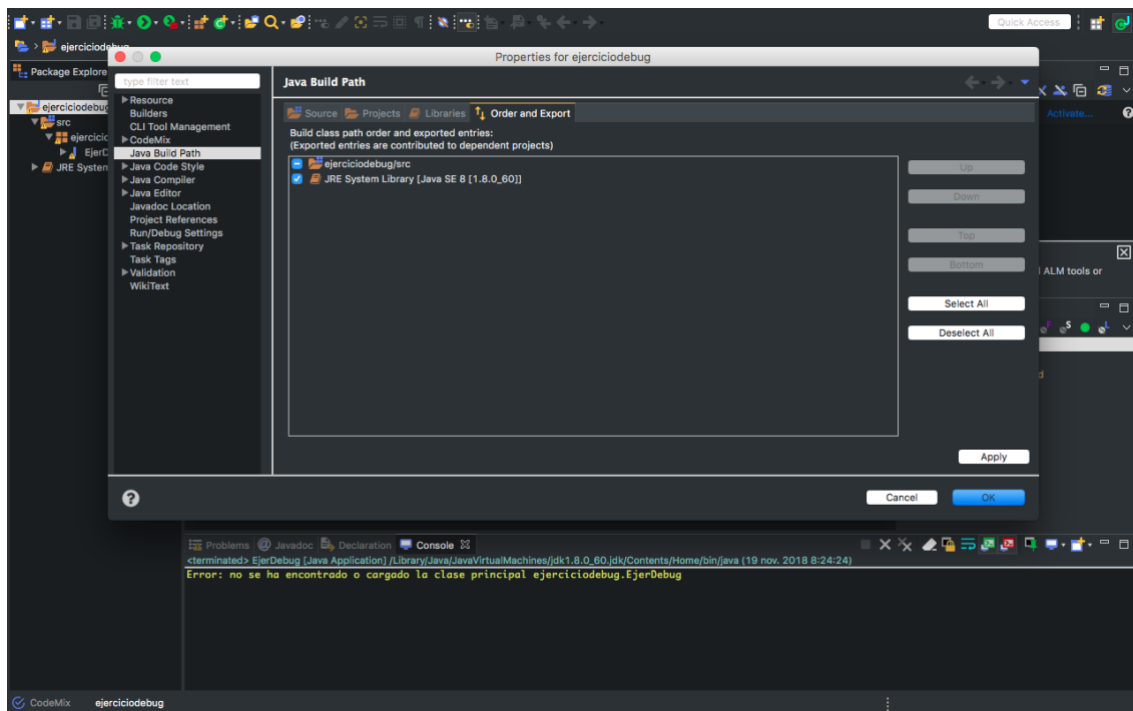
1. Cargamos el proyecto copiándolo a la carpeta del workspace y luego en cargar proyecto en la barra de herramientas.



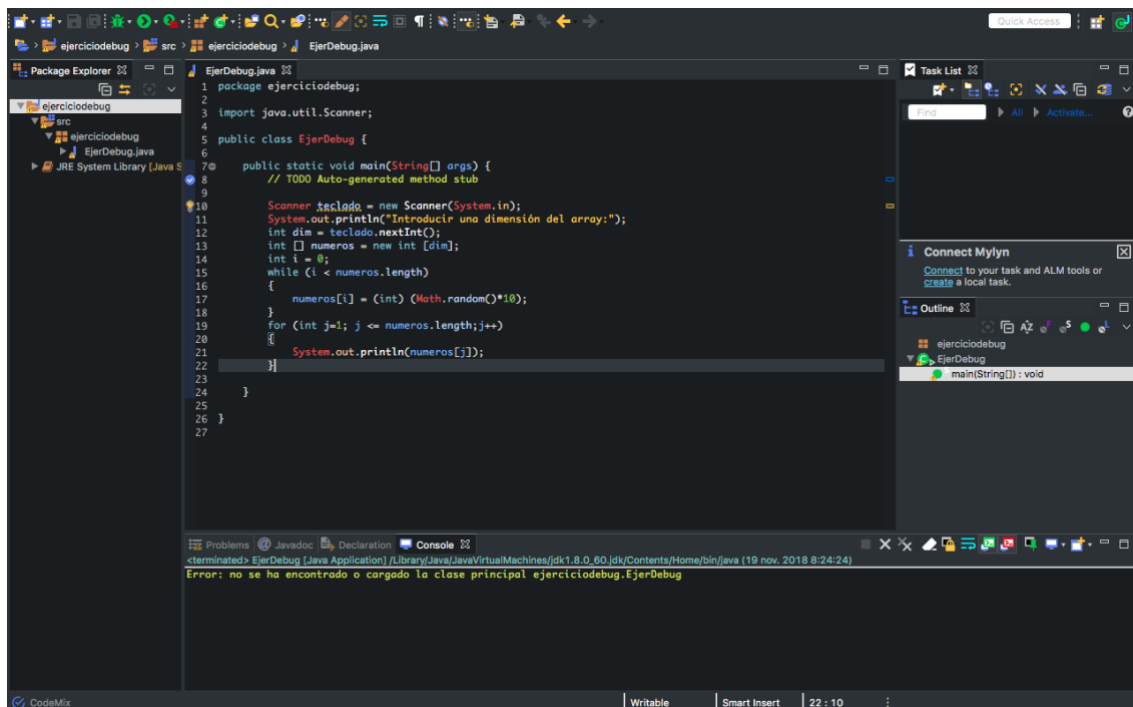
2. Observamos que nos aparece un símbolo de exclamación en el nombre del proyecto, procedemos a cambiar la versión del JRE ya que este esta en su versión 1.10 y la nuestra es 1.8.



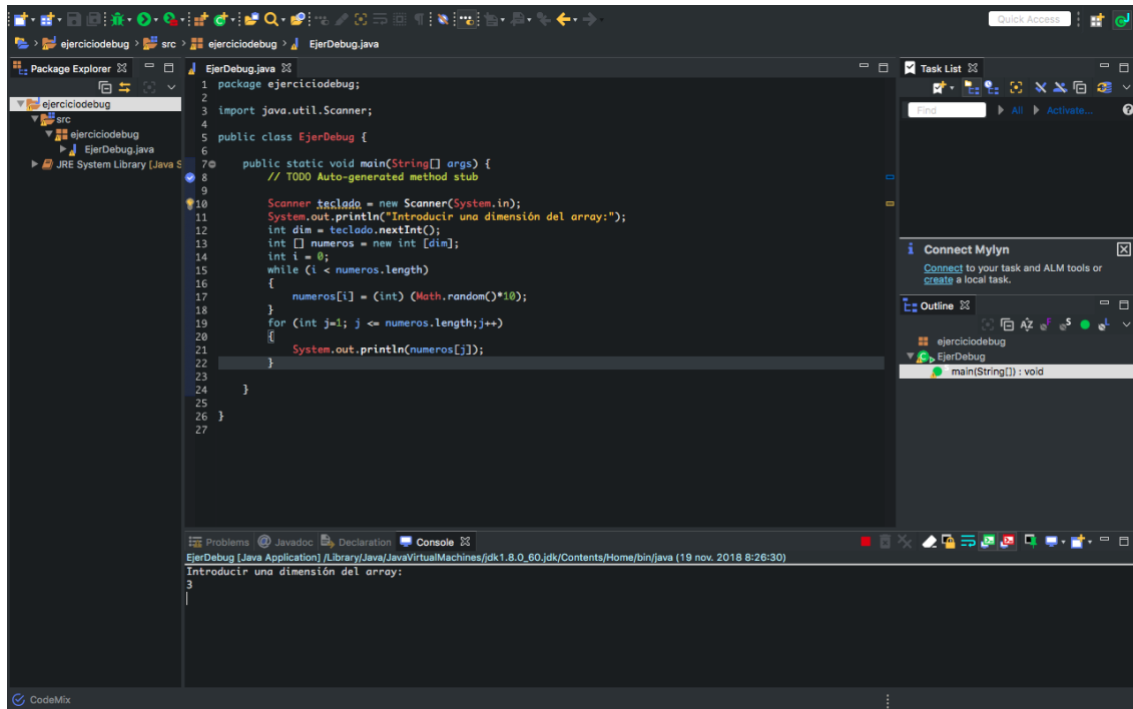
3. Cambiamos la versión a la que es compatible con nuestro Java.



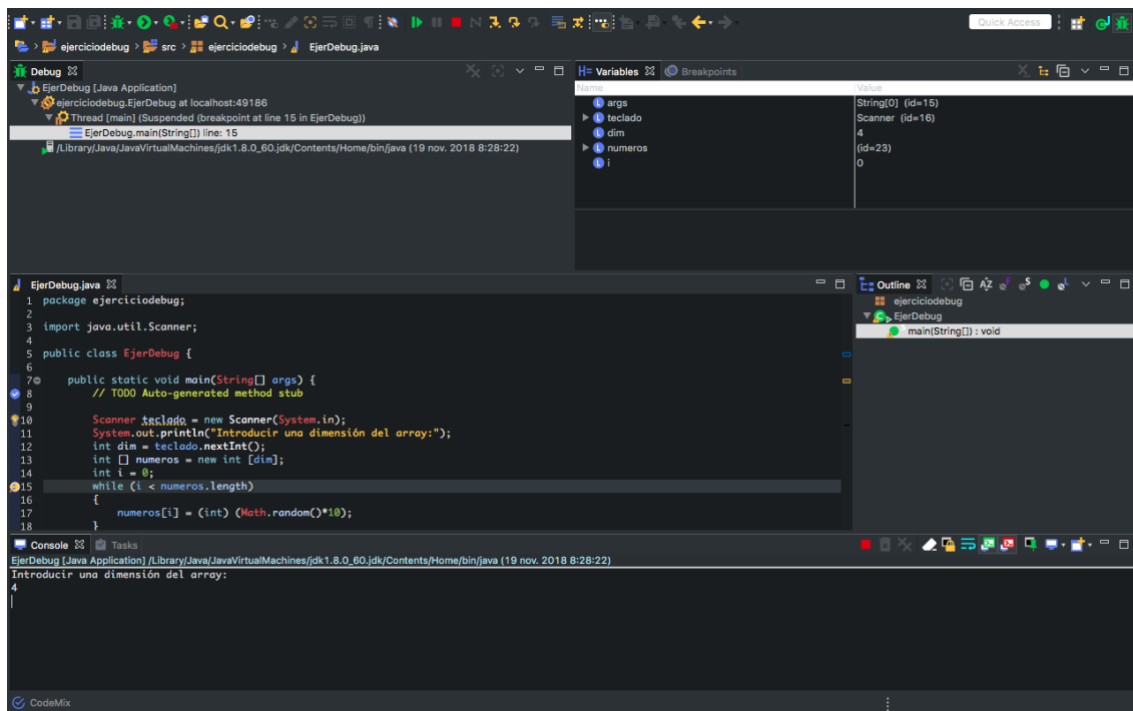
4. Aceptamos y vemos que ahora no nos muestra la exclamación en pantalla y tampoco ningún mensaje por lo que procedemos a ejecutarlo.



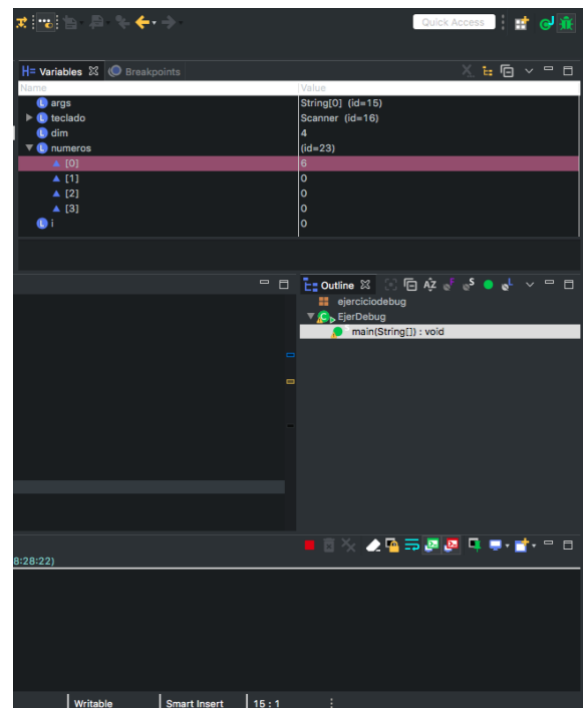
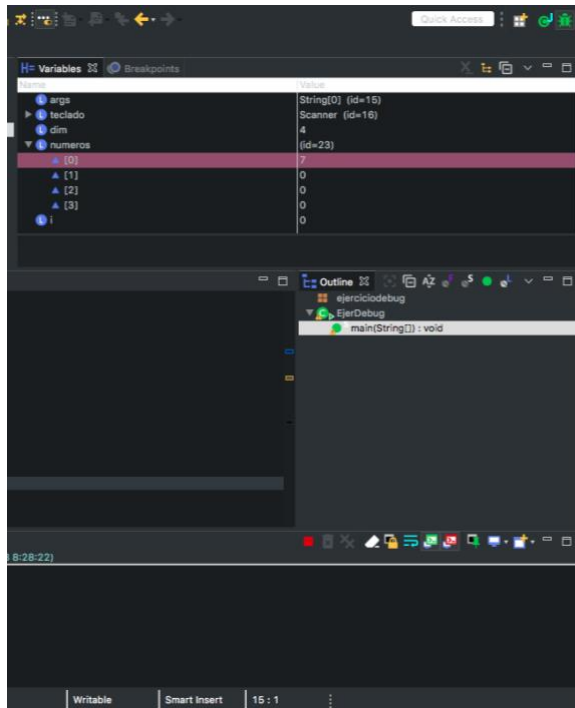
5. Al ejecutarlo observamos que después de pedir el número de tamaño del array no realiza nada mas (ya que el código nos dice que mostrara un array de números aleatorios)



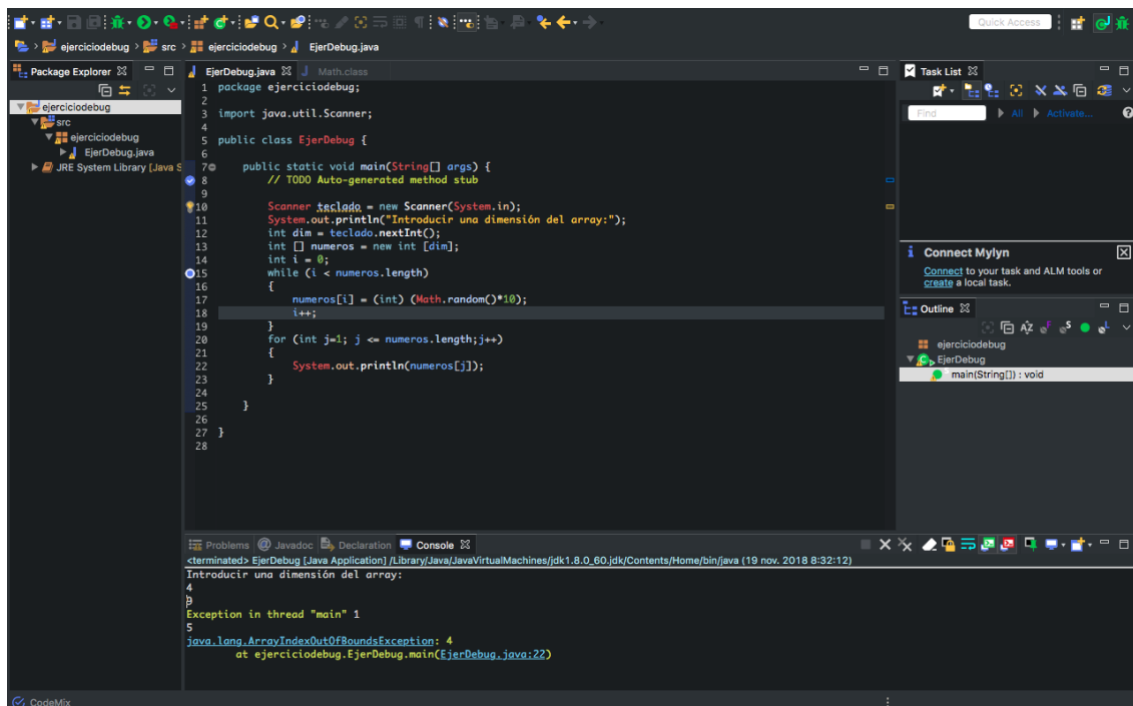
6. Para solucionarlo entramos al Debugger poniendo un breakpoint en donde creemos que puede ser el fallo en nuestro caso los bucles while y luego el for.



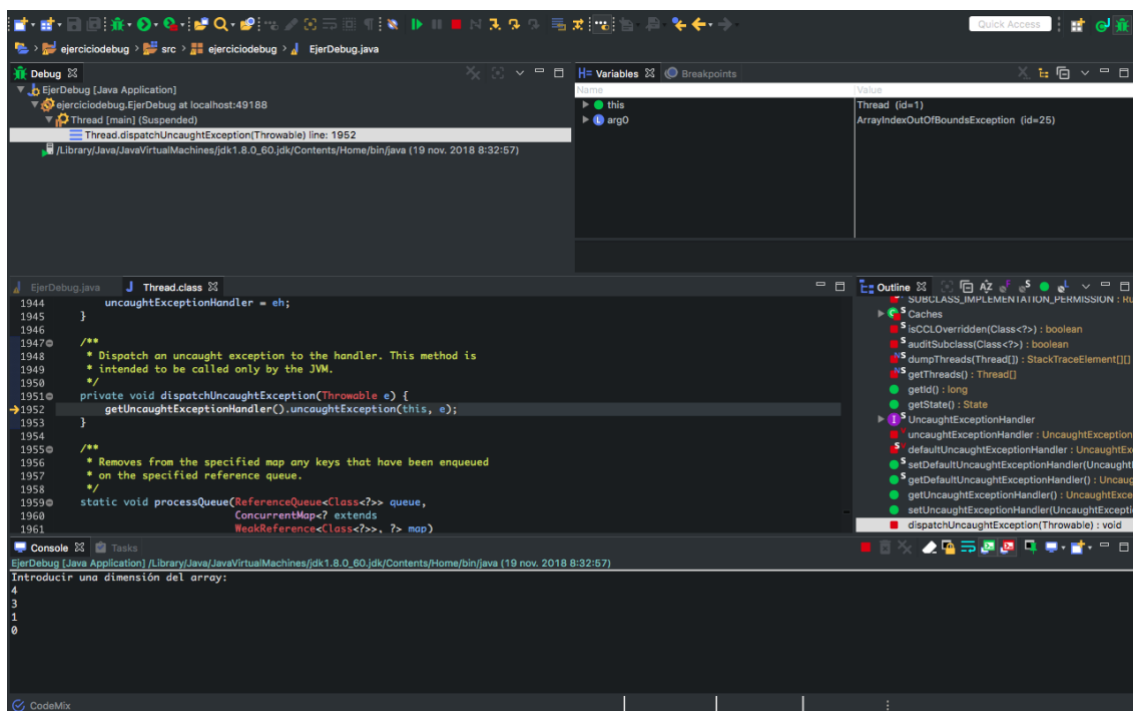
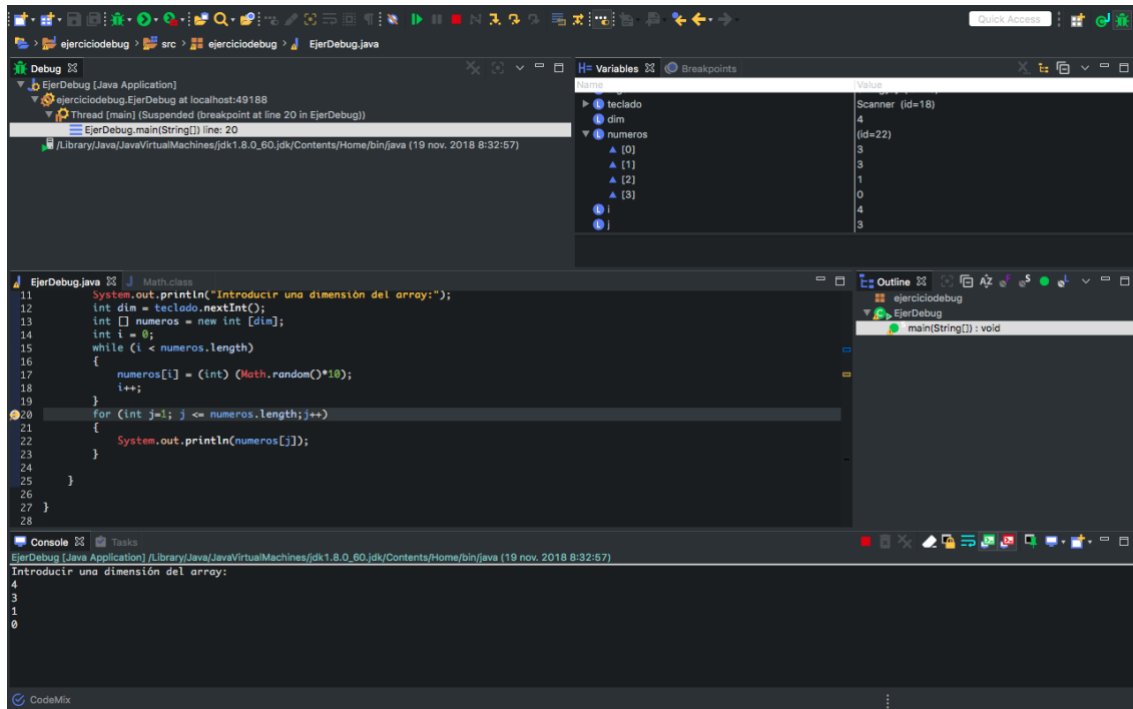
7. Podemos ver en la ventana de las variables que en cada bucle de while el único dato que se escribe es `numeros[0]` y este se sobrescribe cada vez en un bucle infinito ya que la condición es que `(i < numeros.length)` pero dentro del while, `i` no cambia de valor por lo que siempre estará cambiando `numeros[0]`.



8. Una vez arreglado eso volvemos a la ventana de edición de código y volvemos a ejecutar normalmente, pero vemos que una vez mas tenemos otro problema, en este caso una `Exception` al mostrar los números, ponemos un breakpoint en el `for` que muestra los resultados y volvemos a la ventana del debugger.

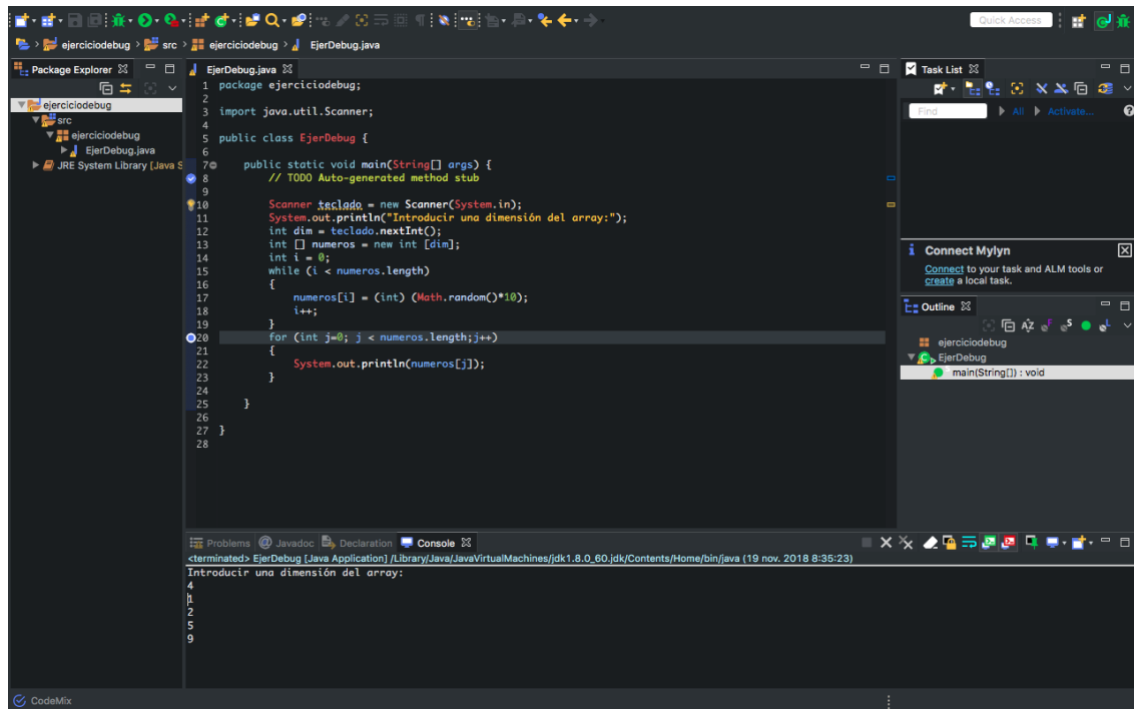


9. En las 2 próximas imágenes podemos ver como al tener $j=4$ e intentar mostrar el debugger sala a otro método o excepción diciendo que estamos saliéndonos de el array y mostrando otras cosas.



10. Una vez identificado el problema cambiamos la lógica del bucle for, diciéndole que empiece desde **j=0** y que continúe mientras **j<numeros.length** así mostrando todos los valores del array y también solo los contenidos en este sin salirnos de su espacio de memoria.

La imagen muestra la correcta ejecución del programa.



```
1 package ejerciciodebug;
2
3 import java.util.Scanner;
4
5 public class EjerDebug {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9
10        Scanner teclado = new Scanner(System.in);
11        System.out.println("Introducir una dimensión del array:");
12        int dim = teclado.nextInt();
13        int [] numeros = new int [dim];
14        int i = 0;
15        while (i < numeros.length)
16        {
17            numeros[i] = (int) (Math.random()*10);
18            i++;
19        }
20        for (int j=0; j < numeros.length;j++)
21        {
22            System.out.println(numeros[j]);
23        }
24    }
25
26
27 }
28
```

Console Output:

```
<terminated> EjerDebug [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home/bin/java (19 nov. 2018 8:35:23)
Introducir una dimensión del array:
4
1
2
5
9
```