
Entornos de Desarrollo

PRUEBAS DE CAJA BLANCA::EJEMPLO FIBONACCI

ENUNCIADO

Dado el programa en Java que realiza el cálculo de la **serie de Fibonacci**, mostrando el resultado por pantalla, se pretende diseñar los casos de prueba de caja blanca.

La cantidad de números en la serie se introducirá por teclado y se comprobará que no sea un número mayor de 3.

El programa se cancelará cuando el usuario introduzca por teclado una 'S o 's'.

NOTA: La serie de Fibonacci empieza en cero, sigue por el uno y los números sucesivos se calculan como la suma de los dos anteriores.

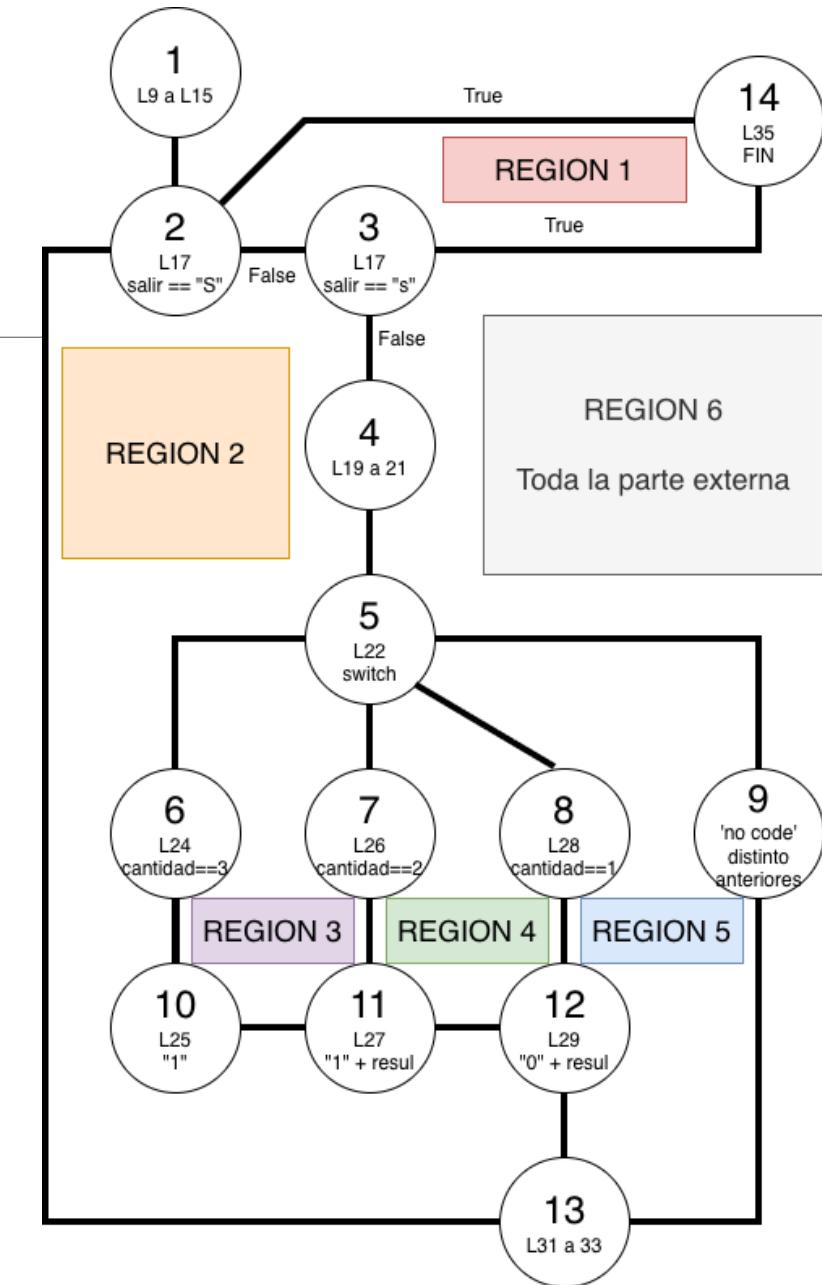
CÓDIGO JAVA

```
1 public class AppFibonacci {
2
3     public static void main(String [] args) {
4         AppFibonacci appFibonacci = new AppFibonacci();
5         appFibonacci.Fibonacci();
6     }
7
8     public void Fibonacci() {
9         Scanner scanner = new Scanner(System.in);
10        String salir = scanner.nextLine();
11
12        int cantidad = 0;
13        String cantidadAux;
14
15        String resultado;
16
17        while(!(salir.equals("S") || salir.equals("s")))
18        {
19            System.out.println("¿Cuantos numeros de la serie deseas mostrar?");
20            cantidadAux = scanner.nextLine();
21            cantidad = Integer.parseInt(cantidadAux);
22            switch(cantidad)
23            {
24                case 3:
25                    resultado = " 1";
26                case 2:
27                    resultado = " 1" + resultado;
28                case 1:
29                    resultado = " 0" + resultado;
30            }
31            System.out.println("Los " + cantidad + " numeros son: " + resultado);
32            System.out.println("Si deseas salir, pulsa: S o s");
33            salir = scanner.nextLine();
34        }
35    }
36 }
```

GRAFO

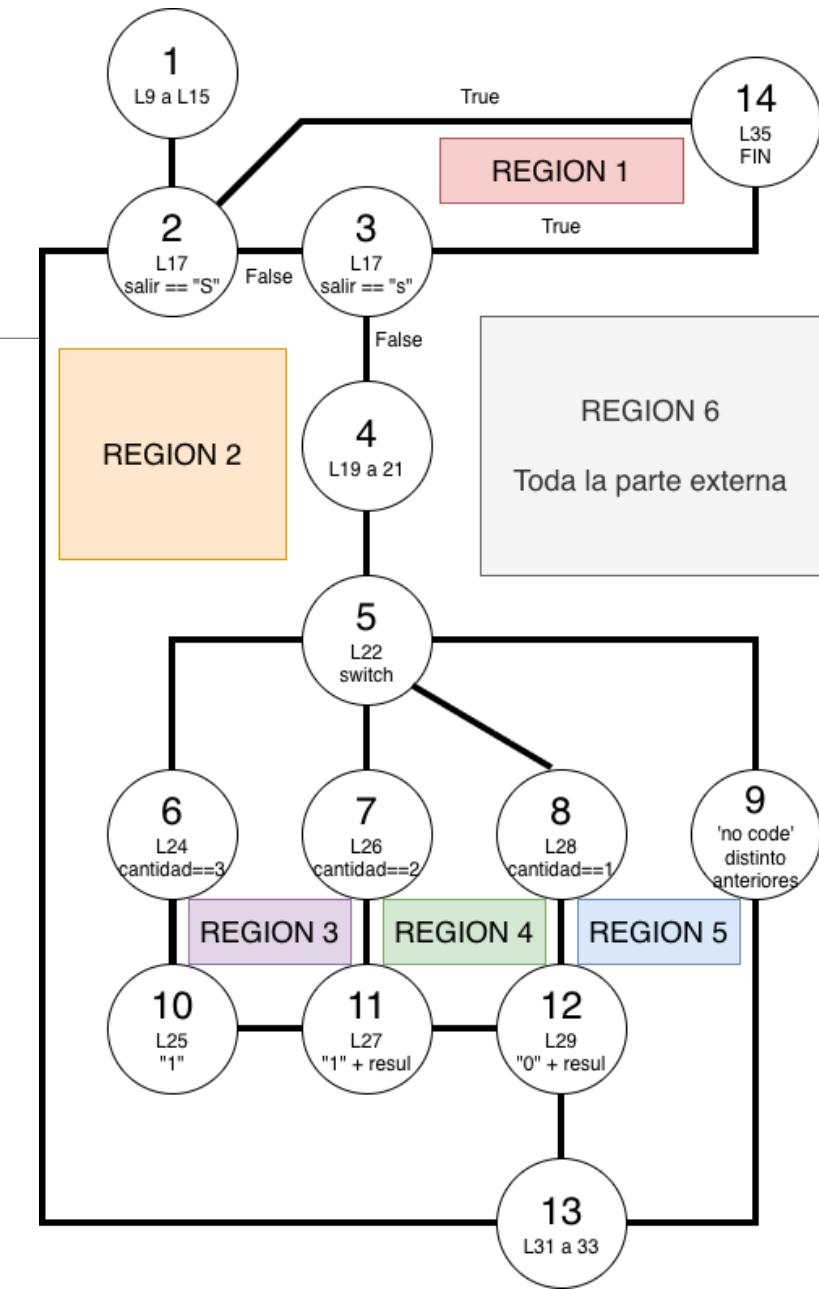
Los nodos 4 y 5 podrían agruparse en uno, pues no añaden complejidad ciclomática.

Lo mismo sucede con los nodos 6 y 10, 7 y 11, 8 y 12.



```

1  public class AppFibonacci {
2
3      public static void main(String [] args) {
4          AppFibonacci appFibonacci = new AppFibonacci();
5          appFibonacci.Fibonacci();
6      }
7
8      public void Fibonacci() {
9          Scanner scanner = new Scanner(System.in);
10         String salir = scanner.nextLine();
11
12         int cantidad = 0;
13         String cantidadAux;
14
15         String resultado;
16
17         while(!(salir.equals("S") || salir.equals("s")))
18     {
19             System.out.println("¿Cuantos numeros de la serie deseas mostrar?");
20             cantidadAux = scanner.nextLine();
21             cantidad = Integer.parseInt(cantidadAux);
22             switch(cantidad)
23     {
24                 case 3:
25                     resultado = " 1";
26                 case 2:
27                     resultado = " 1" + resultado;
28                 case 1:
29                     resultado = " 0" + resultado;
30             }
31             System.out.println("Los " + cantidad + " numeros son: " + resultado);
32             System.out.println("Si deseas salir, pulsa: S o s");
33             salir = scanner.nextLine();
34     }
35 }
36 }
```



COMPLEJIDAD CICLOMÁTICA

Método de cálculo	Complejidad	Comentarios
N. ^o regiones	6	Contar la región exterior
N. ^o aristas – n. ^o nodos + 2	$18 - 14 + 2 = 6$	No contar las que no añaden complejidad
N. ^o de condiciones + 1	$5 + 1 = 6$	Nodos 2, 3, 6, 7 y 8

CAMINOS DE PRUEBA

- Camino 1: 1-2-14
- Camino 2: 1-2-3-14
- Camino 3: 1-2-3-4-5-9-13-14
- Camino 4: 1-2-3-4-5-8-12-13-14
- Camino 5: 1-2-3-4-5-7-11-12-13-14
- Camino 6: 1-2-3-4-5-6-10-11-12-13-14

CASOS DE PRUEBA

Caso de prueba	Valor ‘salir’	Valor “cantidad”	Resultado esperado
Camino 1	“S”	No aplica	Fin del programa
Camino 2	“s”	No aplica	Fin del programa
Camino 3	Distinto de “S” y “s”	4	No muestra resultado
Camino 4	Distinto de “S” y “s”	1	Imprime en pantalla “0”
Camino 5	Distinto de “S” y “s”	2	Imprime en pantalla “0 1”
Camino 6	Distinto de “S” y “s”	3	Imprime en pantalla “0 1 1”