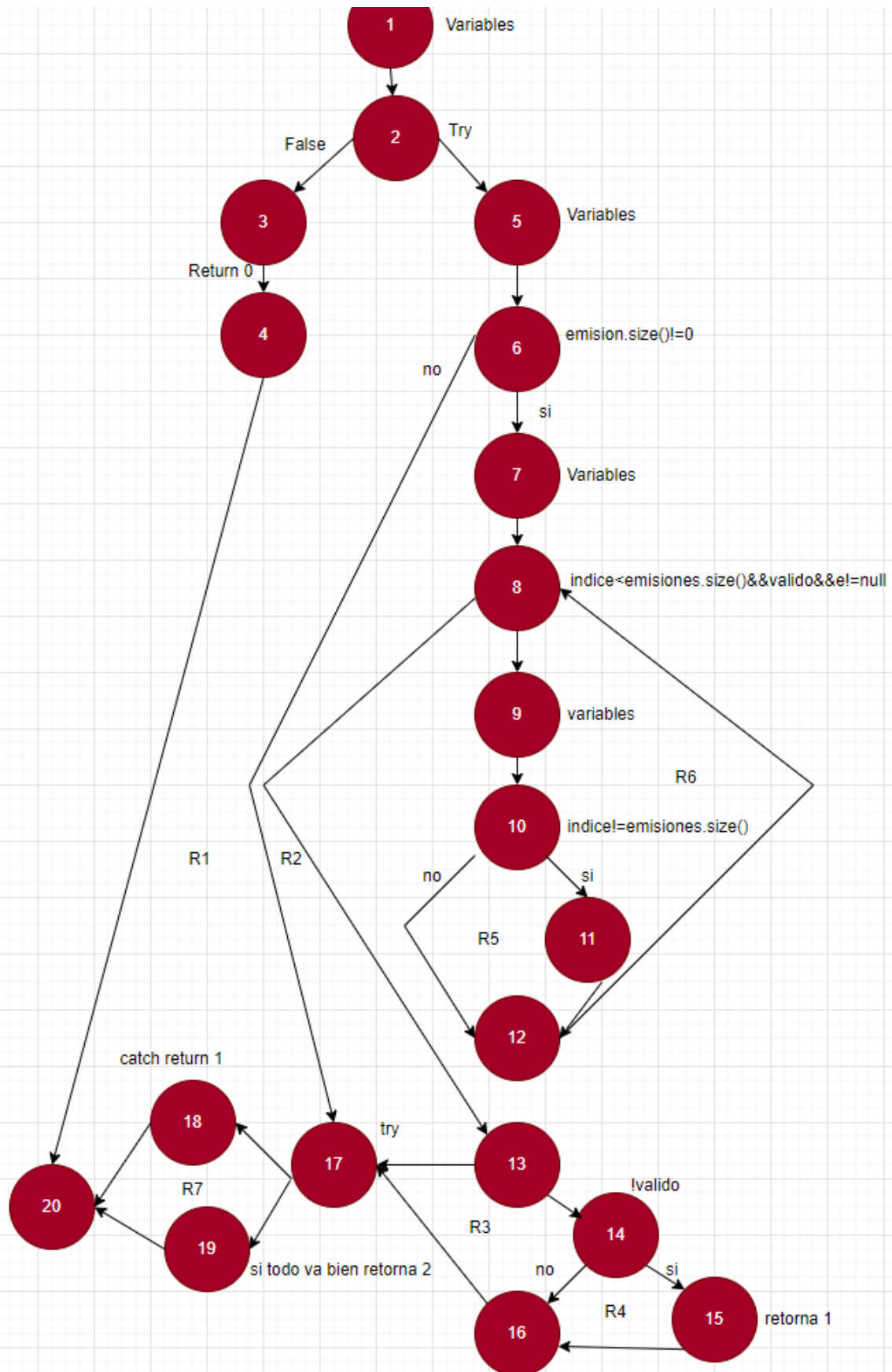


EJERCICIO DE CAMINO BÁSICO I

- Grafo de flujo.


```
1 String calificacionMoral = p.ObtenerClasificacionMoral();
2 int indice = 0;
3 int idcanal = can.obtenerID();
4 boolean valido = true;
5 Emision e;
6 Time horaE;
7 float duracionE;
8 Canal c;
9
10 // Comprobación de la calificación moral
11 try {
12     valido = comprobarCalificacionMoral(calificacionMoral, hora);
13 } catch (Exception a) {
14 }
15
16 // Si la calificación no es válida, se retorna 0
17 if (!valido)
18     return 0;
19
20 // Obtención de la duración del programa
21 float dur = p.obtenerDuracion();
22 ArrayList<Emision> emisiones;
23
24 // Obtención de las emisiones del canal en el día especificado
25 emisiones = AccesoBD.AccesoBDEmisiones.obtenerEmisiones(dia, can);
26
27 // Si hay emisiones en el canal en el día especificado, se verifica solapamiento
28 if (emisiones.size() != 0) {
29     indice = 0;
30     e = emisiones.get(indice);
31     valido = true;
32     while (indice < emisiones.size() && valido && e != null) {
33         horaE = e.ObtenerHora();
34         duracionE = e.ObtenerDuracion();
35         valido = comprobarSolapamiento(horaE, duracionE, hora, dur);
36         indice++;
37         if (indice != emisiones.size()) {
38             e = emisiones.get(indice);
39         }
40     }
41     // Si hay solapamiento, se retorna 1
42     if (!valido)
43         return 1;
44 }
45
46 // Creación y almacenamiento de la nueva emisión
47 int id = 0;
48 try {
49     Emision emi = new Emision(id, can, p, dur, hora, dia);
50     AccesoBD.AccesoBDEmisiones.incluir(emi);
51 } catch (Exception exception) {
52     // Si hay un error al incluir la emisión, se retorna 1
53     return 1;
54 }
55
56 // Si todo ha ido bien, se retorna 2
57 return 2;
```



- Complejidad ciclomática

1. Aristas = 26 Nodos = 20 (26-20+2=8)






2. Nodos predicados = 7 (7+1=8)






3. Regiones =7 (7+1=8)

- Caminos

1 Camino: 1-2-3-4-20 **2** Camino: 1-2-5-6-17-18-20 **3** Camino: 1-2-5-6-17-19-20 **4** Camino: 1-2-5-6-7-8-9-10-11-12-8-13-17-18-20 **5** Camino: 1-2-5-6-7-8-9-10-11-12-8-13-17-19-20 **6** Camino:1-2-5-6-7-8-9-10-11-12-8-13-14-15-16-17-18-20 **7** Camino: 1-2-5-6-7-8-9-10-11-12-8-13-14-15-16-17-19-20 **8** Camini: 1-2-5-6-7-8-9-10-11-12-8-13-14-16-17-18-20 **9** Camino: 1-2-5-6-7-8-9-10-11-12-8-13-14-16-17-19-20 **10** Camino: 1-2-5-6-7-8-17-18-20

- Tabla donde se indique: Camino, caso de prueba y Resultado esperado.

Caminos	Casos de prueba	Resultado esperado
1	Try devuelve la variable valido. Si valido no es igual a true, entra y devuelve 0.	 Devuelve el valor que tiene return =0
2	Si try devuelve true, procederemos a verificar la condicion <code>emision.size()!=0</code> . Si esto no se cumple, pasamos al siguiente try. Si hay errores al incluir emision, esto retornará 1.	 Devuelve el valor que tiene return =1
3	Si try devuelve true, entonces procederemos a verificar la condicion <code>emision.size()!=0</code> . Si esta condición no se cumple, pasaremos al siguiente try. Si todo ha ido bien, devolveremos 2.	 Devuelve el valor que tiene return =2
4	Si <code>if(!valido)</code> no es válido, pasa al siguiente <code>if(emisiones.size()!=0)</code> si la lista de emisiones no está vacía.Comprueba la condicion del bucle <code>indice<emision.size()</code> && <code>valido&&e!=null</code> mientras el índice sea menor que el tamaño de la lista de emisiones, y si aún es válido y la emisión en esa posición no es nula. Realiza las tareas correspondientes dentro de este bucle. Al terminar el bucle con el incremento, sale y pasa al try. Si hay algún error durante la ejecución del código dentro del try, se captura en el catch correspondiente y termina la ejecución.	 Devuelve el return 1
5	Si <code>if(!valido)</code> no se cumple, pasa al siguiente if donde se verifica si <code>if(emisiones.size()!=0)</code> . Si esta condición se cumple, ingresa en el bucle que se ejecuta mientras <code>indice < emision.size()</code> y <code>valido</code> y <code>e != null</code> . Cumple esta condición, se procede a comprobar <code>if(indice != emision.size())</code> , si cumple, realiza las tareas correspondientes. se pasa a un try. Si todo el código dentro del bloque try se ejecuta sin errores, se retorna el valor 2.	 Devuelve el return 2

Caminos	Casos de prueba	Resultado esperado
6	Si lvalido no se cumple, se pasa al siguiente if donde se verifica si indice != emision.size(). Si esta condición se cumple, se ingresa a un bucle while. Dentro de este bucle, se verifica si indice != emisiones.size(), y si es así, se llevan a cabo las tareas correspondientes. Después de salir del bucle, se comprueba la condición if (lvalido) que se encuentra al terminar del bucle. Si esta condición se cumple, se retorna 1.Después de esta verificación, se pasa al try. Si hay errores durante la ejecución del código dentro del try", se ingresa al catch	 Devuelve un 1
7	Si lvalido no se cumple, se pasa al siguiente if donde se verifica si indice != emision.size(). Si esta condición se cumple, se ingresa a un bucle while. Dentro de este bucle, se verifica si indice != emisiones.size(), y si es así, se llevan a cabo las tareas correspondientes. Después de salir del bucle, se comprueba la condición if (lvalido) que se encuentra al terminar del bucle. Si esta condición se cumple, se retorna 1.Después de esta verificación, se pasa al try. Si todo va bien NO entra al catch	 Devuelve un 2
8	Si lvalido no se cumple, se pasa al siguiente if donde se verifica si indice != emision.size(). Si esta condición se cumple, se ingresa a un bucle while. Dentro de este bucle,se comprueba la condición if (lvalido) que se encuentra al terminar del bucle. Si esta condición NO cumple, pasa al try. Si hay errores entra al catch	 Devuelve un 1
9	Si lvalido no se cumple, se pasa al siguiente if donde se verifica si indice != emision.size(). Si esta condición se cumple, se ingresa a un bucle while. Dentro de este bucle,se comprueba la condición if (lvalido) que se encuentra al terminar del bucle. Si esta condición NO cumple, pasa al try. Si NO entra al catch	 Devuelve un 2
10	Si lvalido no se cumple, se pasa al siguiente if donde se verifica si indice != emision.size(). Si esta condición se cumple, Combprueba el bucle como esto no se cumple, pasa al try , verifica el error y termina el programa	 Devuelve un 1