```java
public RunExperimentBerlin(String directory) throws Exception {
1       if(directory == null) 1
2           directory = "dataNodes"; 2

3       SAXBuilder sb = new SAXBuilder();
4       File config = new File("..//configureExecution.xml");
5       Document d = sb.build(config);
6       Element root = d.getRootElement();                              3

7       Iterator<Element> it = root.getChildren().iterator();
8       while(it.hasNext()) {                           4
9           Element e = it.next();
10          String dns = e.getValue();                                  5
11          Connection conn = new Connection(dns);
12          System.out.println("Creating ssh connection to "+dns);
13          conn.connect();
14          File pem = new File("explore.pem");//new File("matching.pem");
15          boolean isAuthenticated = conn.authenticateWithPublicKey("ubuntu", pem, null);
16          if (isAuthenticated == false) 6
17              throw new IOException("Authentication failed."); 7

18          SCPClient scp = conn.createSCPClient();
19          SAXBuilder sb2 = new SAXBuilder();
20          File config2 = new File("..//copyFiles.xml");
21          Document d2 = sb2.build(config2);                           8
22          Element files = d2.getRootElement();
23          Iterator<Element> it2 = files.getChildren().iterator();

24          while(it2.hasNext()) {              9
25              Element ele = it2.next();
26              String name = ele.getAttributeValue("name");
27              String source = ele.getAttributeValue("local")+name;
28              String realName = source;
29              System.out.println("Source: "+realName);
30              String target = ele.getAttributeValue("aws");          10
31              System.out.println("Target: "+target);
32              scp.put(realName, target);
33              System.out.println("Source: " +directory+ "/"+dns);
34              conn.close();
35          }                                   11( fim do 9)
36      }                                   12( fim do 4)
37  }
```
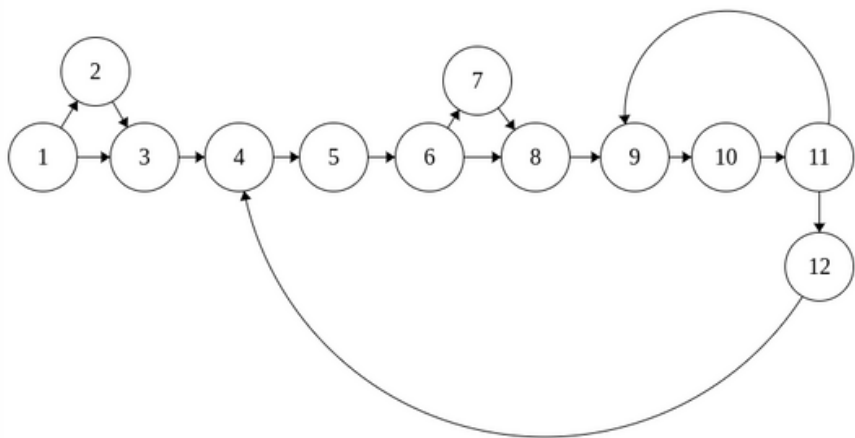
Caminhos independetes:
Total = 5;
(1): 1-3-4-5-6-8-9-10-11-12 (direto)
(2): 1-2-3-4-5-6-8-9-10-11-12 (primerio if)
(3): 1-3-4-5-6-8-7-9-10-11-12 (segundo if)
(4): 1-3-4-5-6-8-9-10-11-9-[..] (primeiro while)
(5): 1-3-4-5-6-8-9-10-11-12-4-[..] (segundo while)