ACH 2028
Qualidade de Software

Aula 08 - Teste de Software: Técnica Estrutural ou Caixa-Branca

Prof. Marcelo Medeiros Eler
marceloeler@usp.br

Escola de Artes, Ciências e Humanidades - Universidade de São Paulo

Objetivos

Apresentar a técnica de teste estrutural ou caixa-branca

Mostrar exemplos de casos de teste criados para satisfazer os principais critérios da técnica de teste estrutural

Propor exercícios para exercitar a técnica de teste estrutural

Como encontrar todas as falhas de um software?

Um conjunto de casos de teste com **qualidade ótima** é capaz de revelar todas as falhas de um software

Para encontrar todas as falhas e garantir que o software está livre de defeitos é necessário **testá-lo em todas as condições** e com **todos os dados de entrada possíveis**

Como encontrar todas as falhas de um software?

Entretanto, as condições, os cenários, os dados de entrada e suas combinações tendem a ser **infinitos ou muito grandes**

Portanto, criar um conjunto de casos de teste ótimo é geralmente **impossível** ou **impraticável**

Uma pergunta importante neste cenário

Como criar um conjunto de casos de teste que seja:

- Finito e factível?
- Capaz de revelar o maior número de falhas perceptíveis?
- Capaz de revelar as falhas mais críticas ou relevantes do software?

Esta pergunta esconde algumas outras:

- Como selecionar cenários/dados para criar bons casos de teste?
- Quantos testes são necessários para testar um software ou parte dele?
- Como saber se os testes criados são suficientes?

Técnicas e critérios de teste

Um bom caso de teste é aquele que tem uma alta probabilidade de revelar uma falha ainda não descoberta

Portanto, técnicas e critérios de teste foram propostos para orientar o testador na tarefa de derivar os casos de teste com base nos artefatos do projeto (especificação, código, modelos, etc):

- Teste Adhoc
- Teste Exploratório
- Teste Funcional ou Caixa-Preta
- Teste Estrutural ou Caixa-Branca
- Teste Baseado em Erros

Critérios de teste

Definem requisitos de teste que devem ser satisfeitos pelos casos de teste

Um critério é satisfeito somente quando todos os requisitos que ele define são satisfeitos

Ajudam a responder às questões:

- Como selecionar valores de entrada para criar bons casos de teste?
- Quantos casos de teste devem ser criados?
- Quando parar de testar?

Teste funcional ou caixa-preta (aula anterior)

Teste funcional ou caixa-preta (aula anterior)

Tem o objetivo de verificar se uma funcionalidade, operação, função, método, classe, programa está de acordo com os requisitos especificados

Os casos de teste são definidos com base na especificação do software (descrições, casos de uso, requisitos, diagramas, etc)

Exemplos de critérios desta técnica:

- Classes de equivalência
- Análise de valor-limite
- Tabela de decisão

Teste funcional ou caixa-preta (aula anterior)

No teste funcional, sabemos que o conjunto de casos de teste são suficientes (ou possuem qualidade adequada) quando eles:

- Exercitam todas as classes de equivalência conforme determina o critério
- Exercitam todos os valores-limite
- Exercitam todas as combinações expressas por meio de tabelas de decisão

Teste funcional ou caixa-preta (aula anterior)

Entretanto, os casos de testes criados com a técnica funcional, ainda que satisfaçam todos os critérios, podem não ser o suficientes.

Uma possível limitação desta técnica é a seguinte:

- Eu não sei se existem estruturas de código que não foram executadas pelos casos de teste, pois só consigo garantir que eu testei aquilo que está especificado
- Se alguma regra ou funcionalidade foi implementada e não está completamente especificada, pode haver condições que precisam ser testadas ainda

Teste estrutural ou teste caixa-branca

Também conhecido como teste caixa-branca

Tem o objetivo de verificar se os testes exercitam todas as estruturas de programação

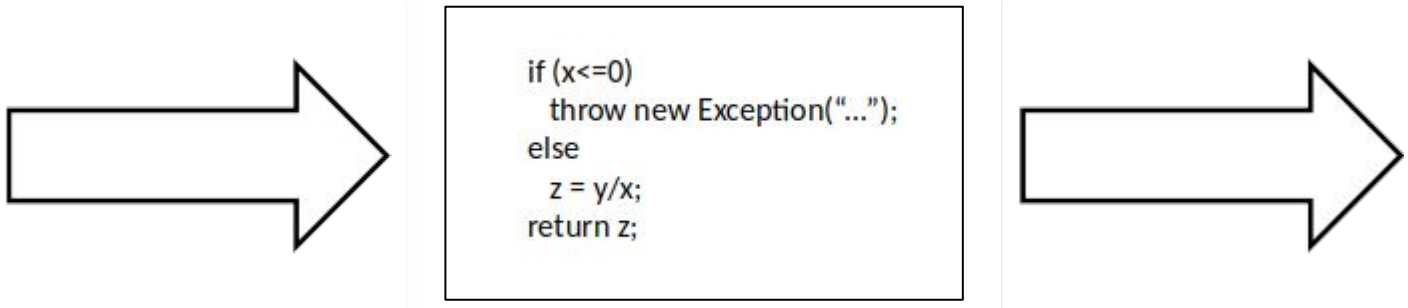
Seu principal objetivo é encontrar falhas no software por meio da análise de sua estrutura interna

Teste estrutural ou teste caixa-branca

Também conhecido como teste caixa-branca

Tem o objetivo de verificar se os testes exercitam todas as estruturas de programação

Seu principal objetivo é encontrar falhas no software por meio da análise de sua estrutura interna



Cobertura de código

Na prática, a maioria das pessoas verifica duas métricas para saber o quanto de seus códigos foram exercitados pelos casos de teste:

- Linhas de código executadas
- Branches (desvios de fluxo)

Quando analisadas em termos de porcentagem de linhas de código ou branches executadas, essas métricas são comumente chamadas de “cobertura de código”

O termo cobertura de código, portanto, refere-se à porcentagem do código que foi executada (de acordo com cada métrica)

Um pouco de teoria: critérios do teste estrutural

Os critérios da técnica de teste estrutural podem ser divididos em duas categorias:

- Critérios de fluxo de controle
- Critérios de fluxo de dados

O enfoque desta aula será critérios de fluxo de controle bem conhecidos:

- Todos os caminhos
- Todos os caminhos básicos
- Todos os nós
- Todas as arestas

Um pouco de teoria: estruturas de representação

É comum que no teste caixa-branca o elemento de programação sob teste seja representado por abstrações que facilitam entender sua estrutura e derivar requisitos de teste

Uma estratégia comum neste cenário é utilizar Grafos de Fluxo de Controle (GFC)

- Os nós do grafo representam blocos de instruções e as arestas representam os possíveis desvios de fluxo
- Blocos de instruções devem ser identificados de tal forma que não existam desvios de fluxo entre sua primeira e sua última instrução, desconsiderando situações de exceção.

Grafo de Fluxo de Controle

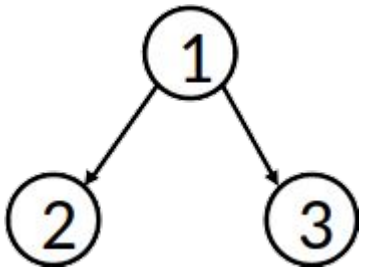
Exemplos de estruturas de programação e respectivo GFC

```
int p = Integer.valueOf(args[0]);
int q = Integer.valueOf(args[1]);
int n = p*q;
if (n>18)
   System.out.println("OK");
else
   System.out.println("ERRO");
```

Grafo de Fluxo de Controle

Exemplos de estruturas de programação e respectivo GFC

```
1  int p = Integer.valueOf(args[0]);
   int q = Integer.valueOf(args[1]);
   int n = p*q;
   if (n>18)
2     System.out.println("OK");
   else
3     System.out.println("ERRO");
```



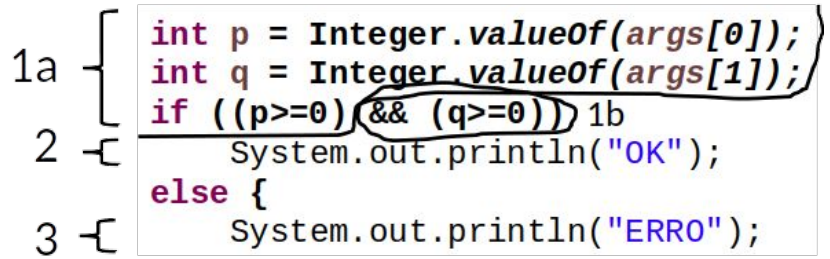
Grafo de Fluxo de Controle

Exemplos de estruturas de programação e respectivo GFC

```
int p = Integer.valueOf(args[0]);
int q = Integer.valueOf(args[1]);
if ((p>=0) && (q>=0))
    System.out.println("OK");
else {
    System.out.println("ERRO");
```

Grafo de Fluxo de Controle

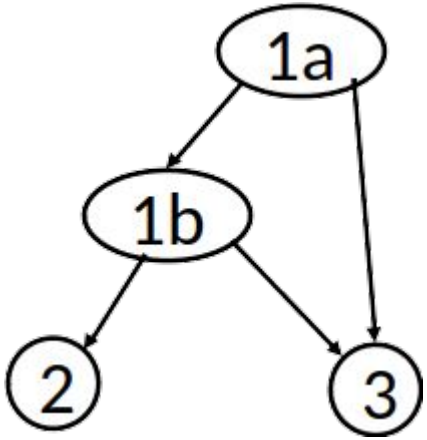
Exemplos de estruturas de programação e respectivo GFC



Grafo de Fluxo de Controle

Exemplos de estruturas de programação e respectivo GFC

```
1a  int p = Integer.valueOf(args[0]);
    int q = Integer.valueOf(args[1]);
    if ((p>=0) && (q>=0))                1b
2       System.out.println("OK");
    else {
3       System.out.println("ERRO");
```



Grafo de Fluxo de Controle

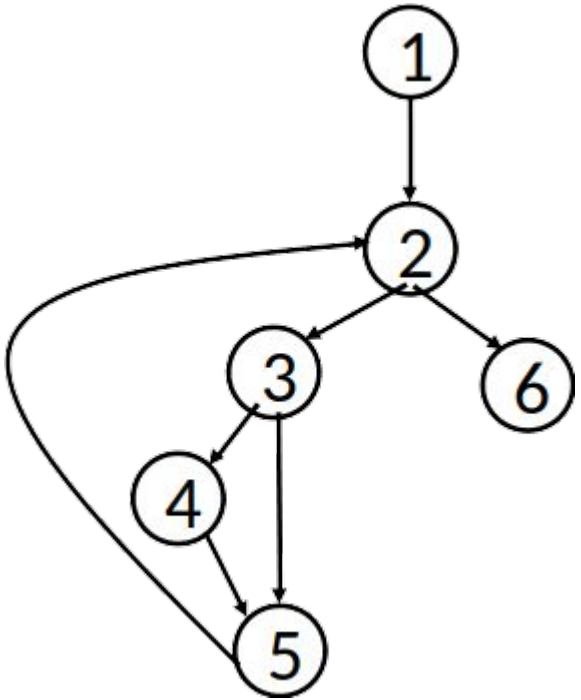
Exemplos de estruturas de programação e respectivo GFC

```
int cont=0;
int p = Integer.valueOf(args[0]);
int q = Integer.valueOf(args[1]);
int i = p;
while (i<=q){
  if (i % 2 == 0)
    cont++;
  i++;
}
System.out.println(cont + "pares");
```

Grafo de Fluxo de Controle

Exemplos de estruturas de programação e respectivo GFC

```
1  int cont=0;
   int p = Integer.valueOf(args[0]);
   int q = Integer.valueOf(args[1]);
   int i = p;
2  while (i<=q){
3    if (i % 2 == 0)
4      cont++;
5    i++;
   }
6  System.out.println(cont + "pares");
```



Exemplo 1 - Triângulo

Escreva casos de teste para testar um algoritmo que recebe como entrada três números inteiros positivos que representam os tamanhos dos três lados de um triângulo, e retorna o tipo de triângulo formado (Equilátero, isósceles ou escaleno).

Se algum dos lados for negativo, o algoritmo deve informar que houve um erro porque um dos lados não tem valor válido

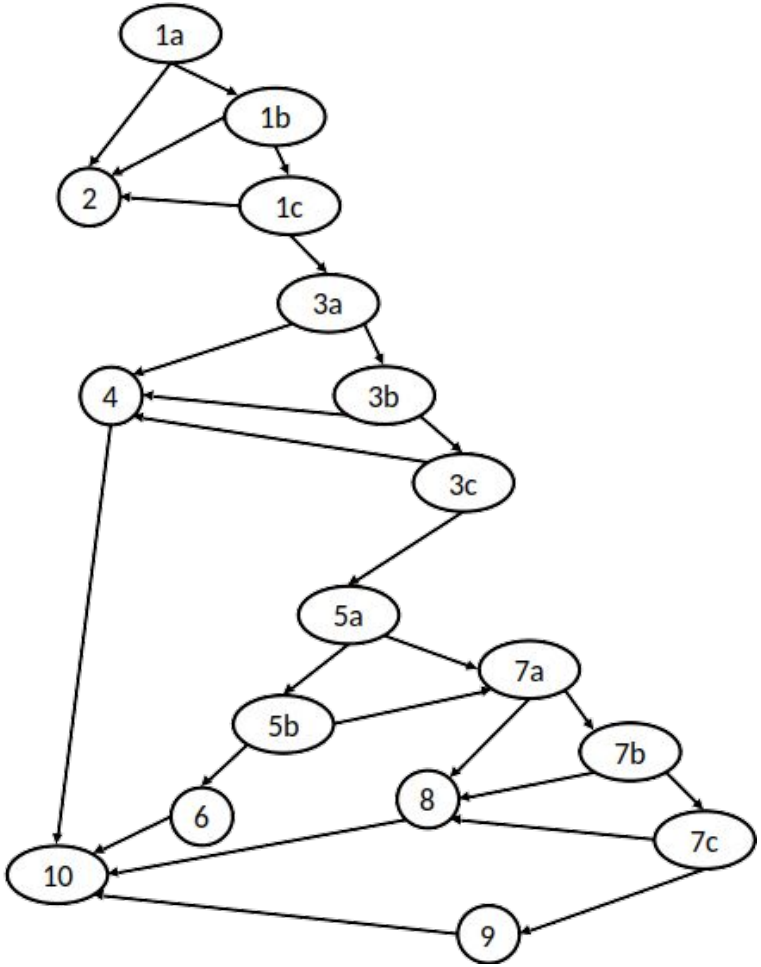
Se um dos lados tiver tamanho maior do que a soma dos outros três lados, então o algoritmo deve informar que esses valores não são suficientes para formar um triângulo

Exemplo 1 - Triângulo

```
public static String classificaTriangulo(int LA, int LB, int LC) throws LadoInvalidoException{
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("lado inválido");

  if ( (LA >= LB + LC) || (LB >= LA + LC) || (LC >= LA + LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```

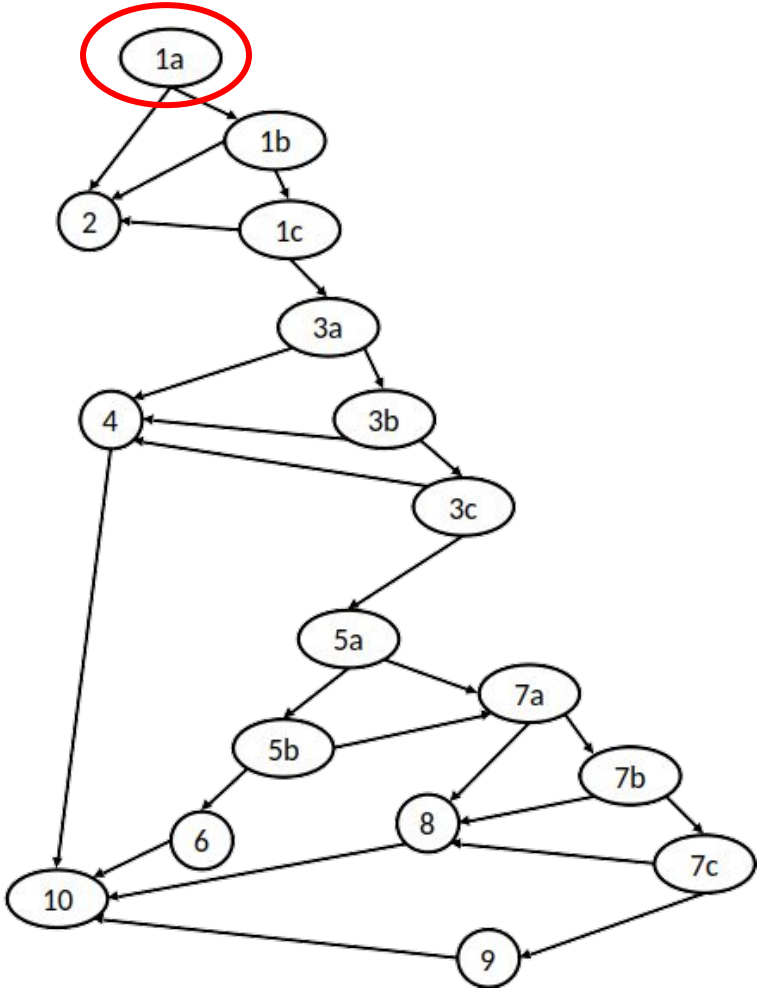
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



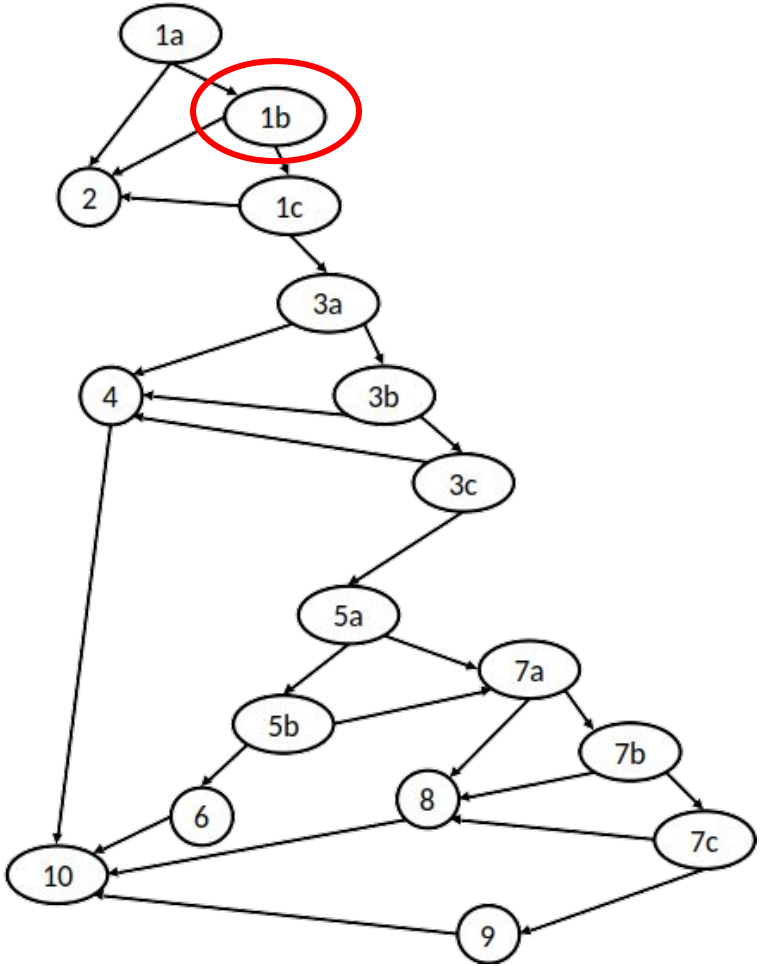
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if ( LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



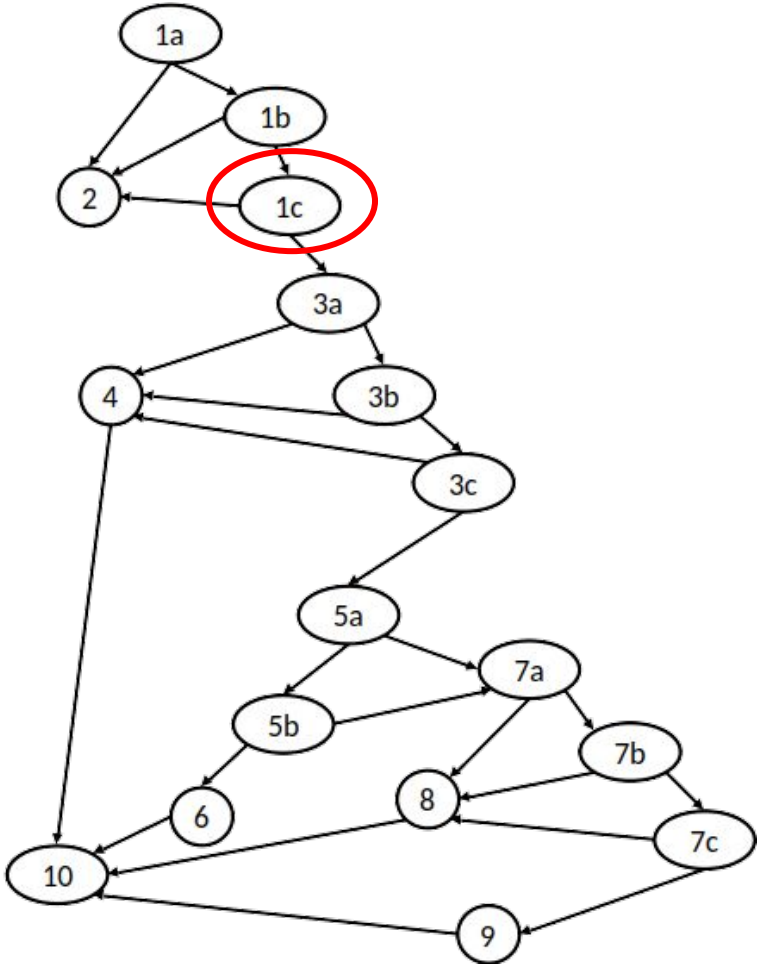
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



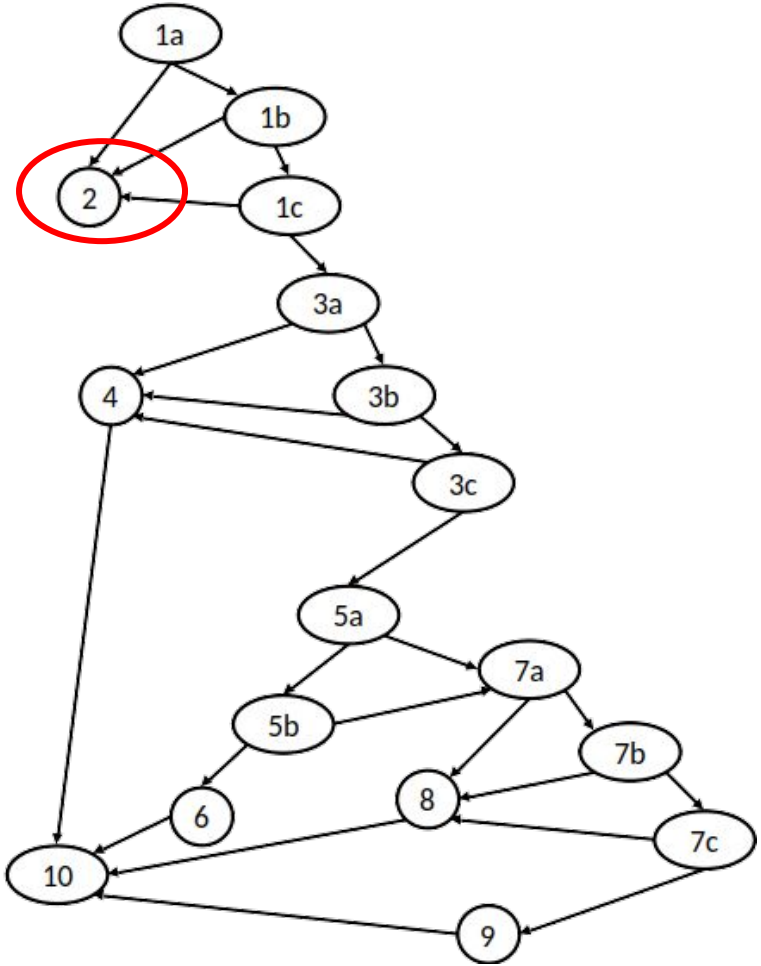
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



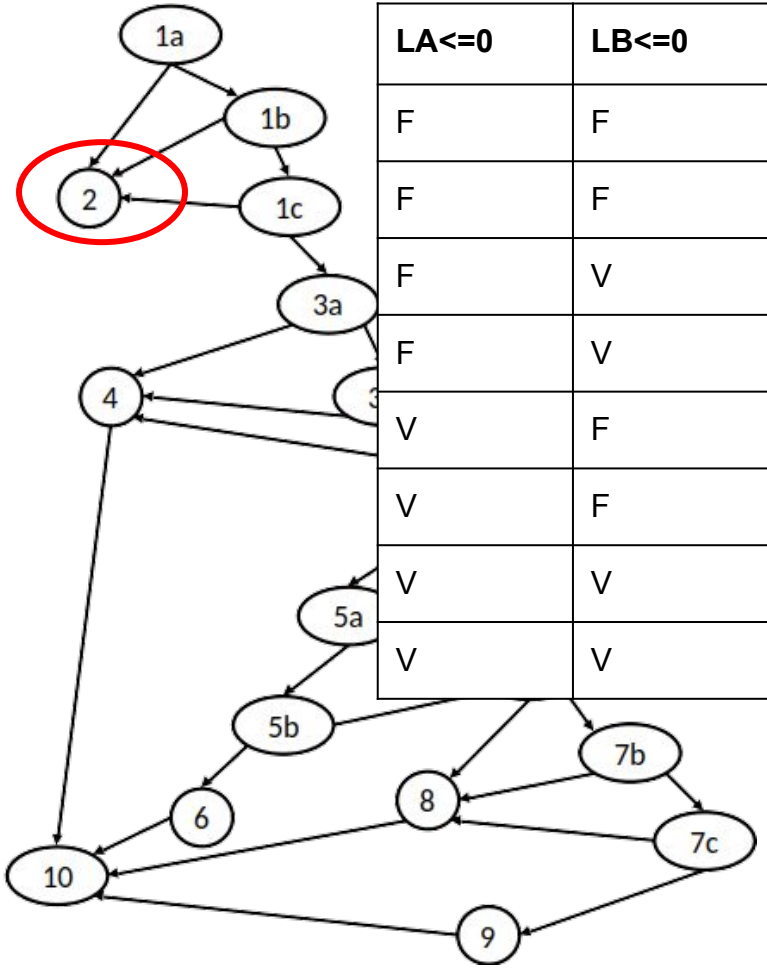
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



Exemplo 1 - Triângulo

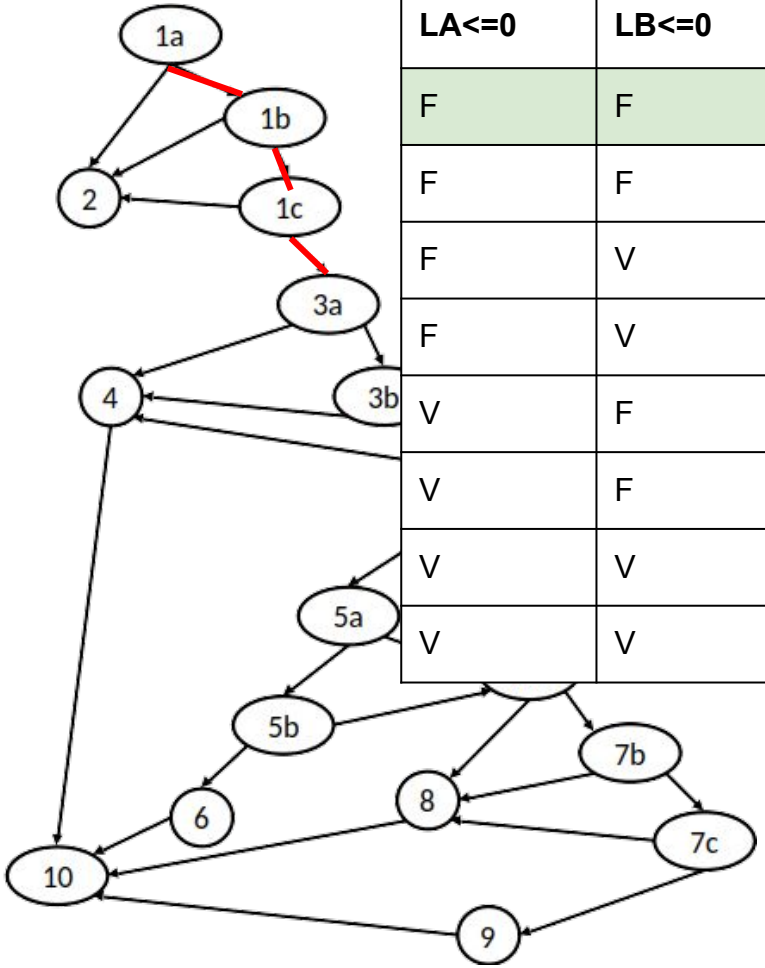
```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



LA<=0	LB<=0	LC<=0
F	F	F
F	F	V
F	V	F
F	V	V
V	F	F
V	F	V
V	V	F
V	V	V

Exemplo 1 - Triângulo

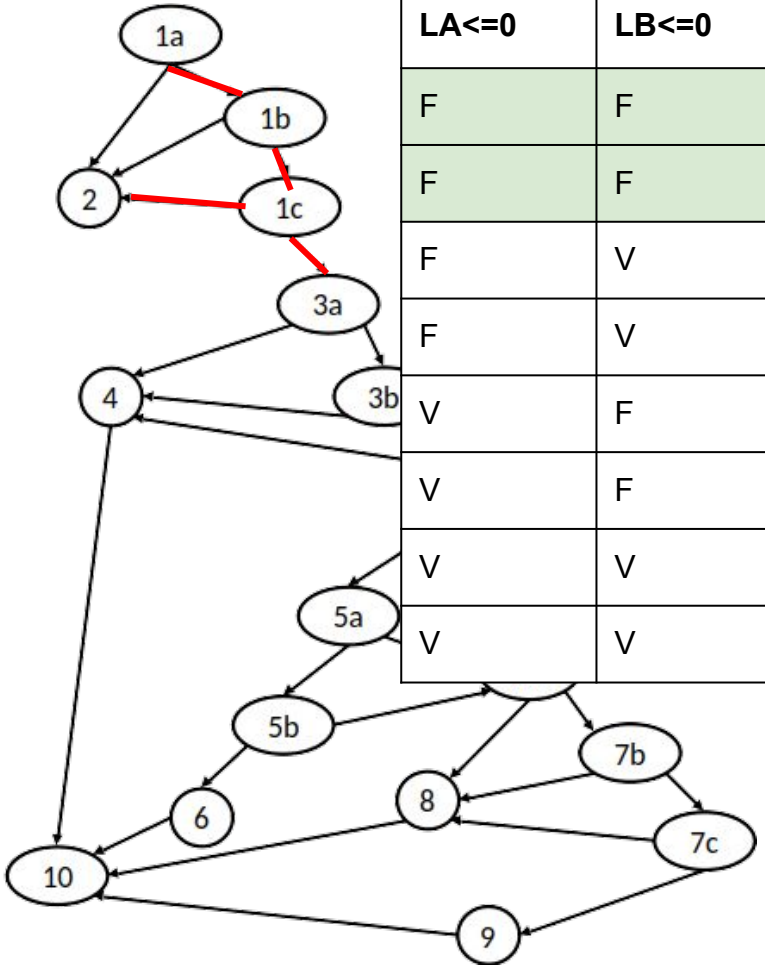
```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



LA<=0	LB<=0	LC<=0
F	F	F
F	F	V
F	V	F
F	V	V
V	F	F
V	F	V
V	V	F
V	V	V

Exemplo 1 - Triângulo

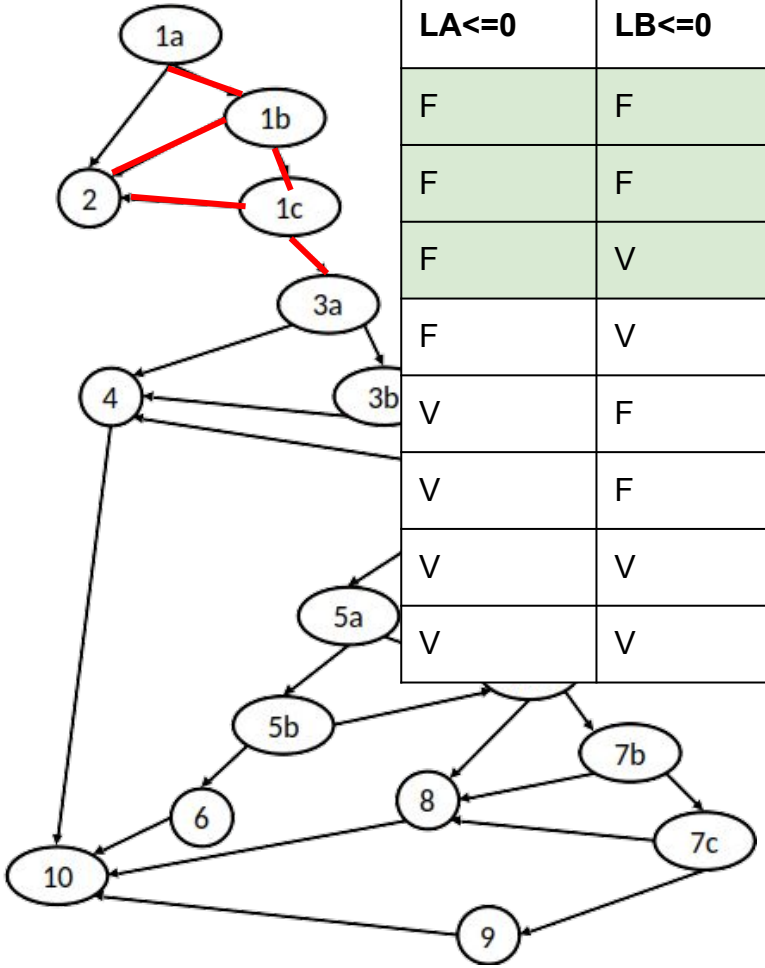
```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



LA<=0	LB<=0	LC<=0
F	F	F
F	F	V
F	V	F
F	V	V
V	F	F
V	F	V
V	V	F
V	V	V

Exemplo 1 - Triângulo

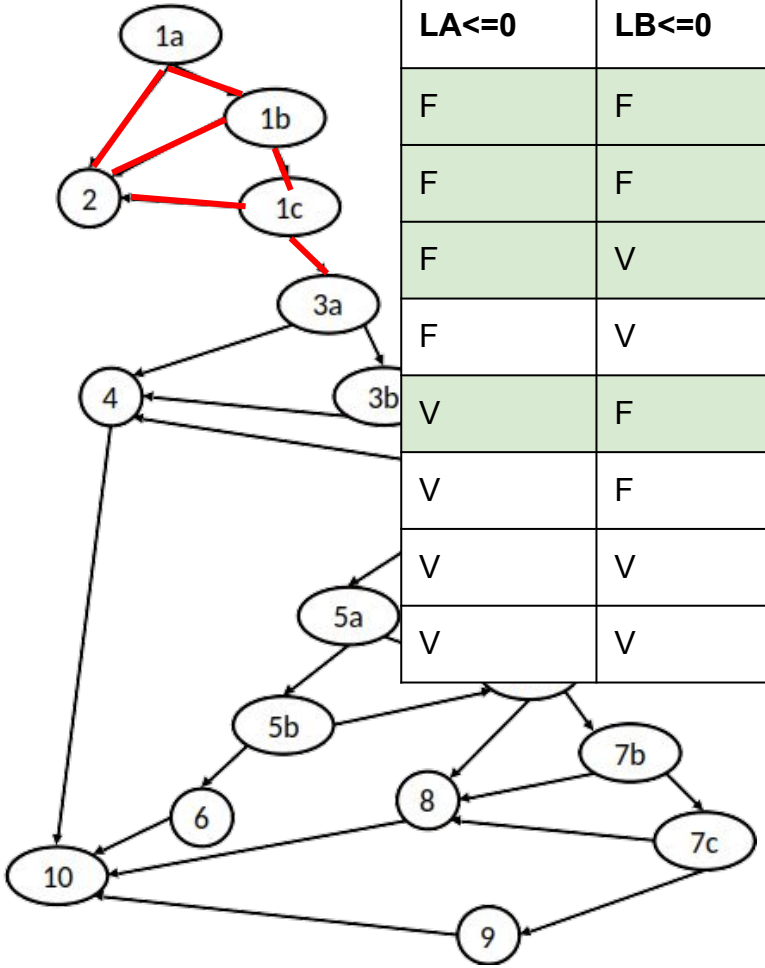
```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



LA<=0	LB<=0	LC<=0
F	F	F
F	F	V
F	V	F
F	V	V
V	F	F
V	F	V
V	V	F
V	V	V

Exemplo 1 - Triângulo

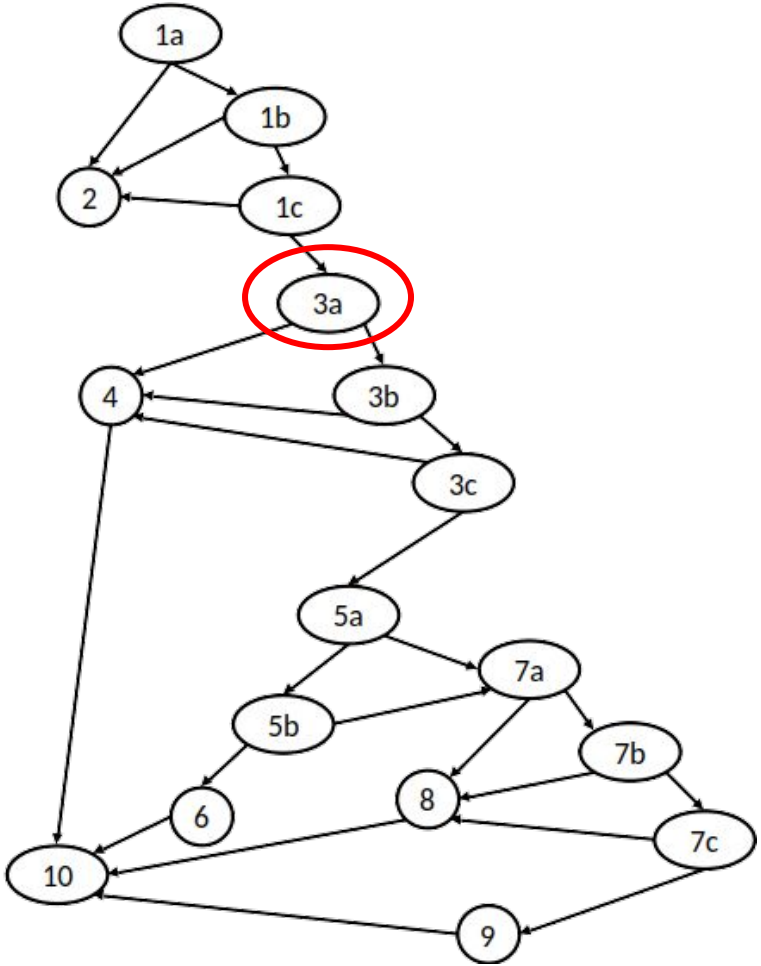
```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



LA<=0	LB<=0	LC<=0
F	F	F
F	F	V
F	V	F
F	V	V
V	F	F
V	F	V
V	V	F
V	V	V

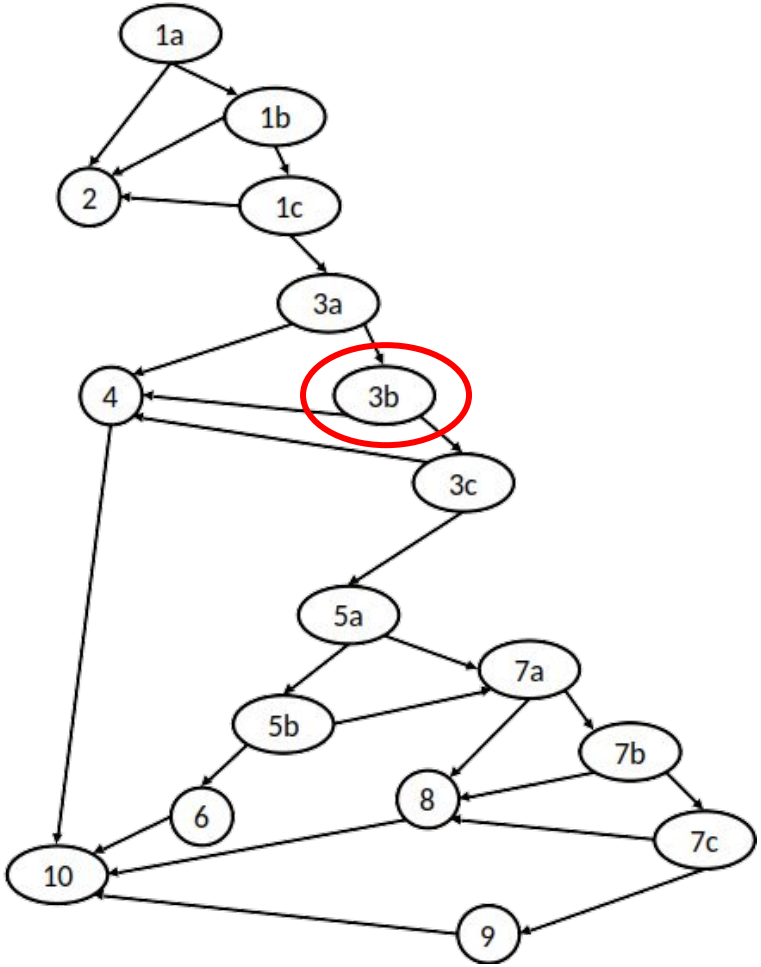
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



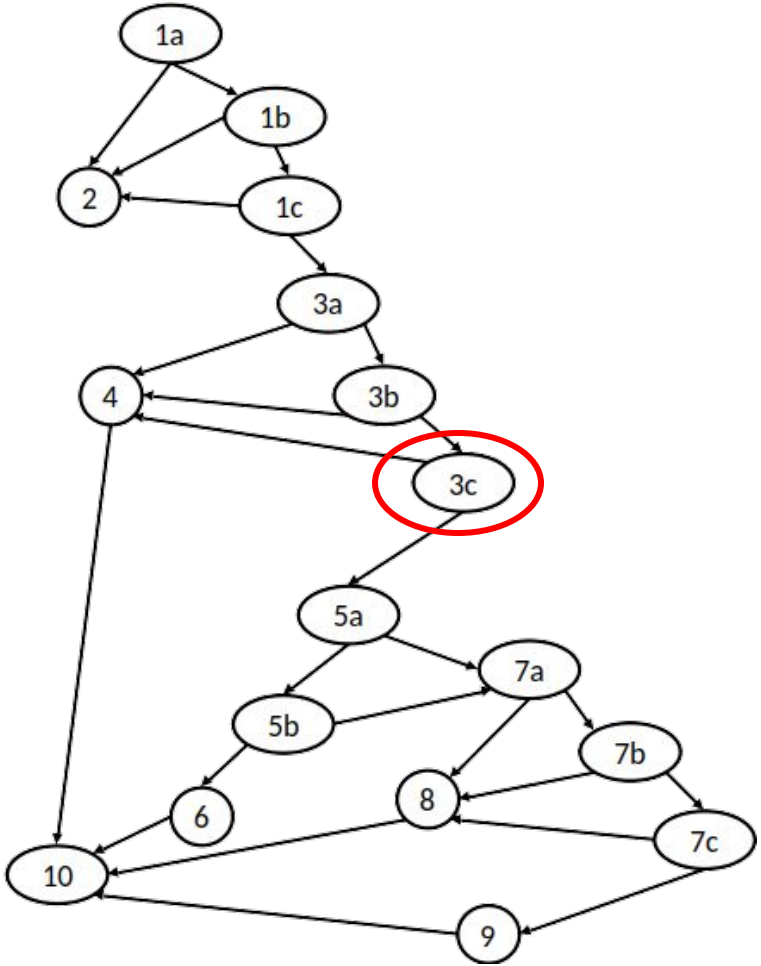
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



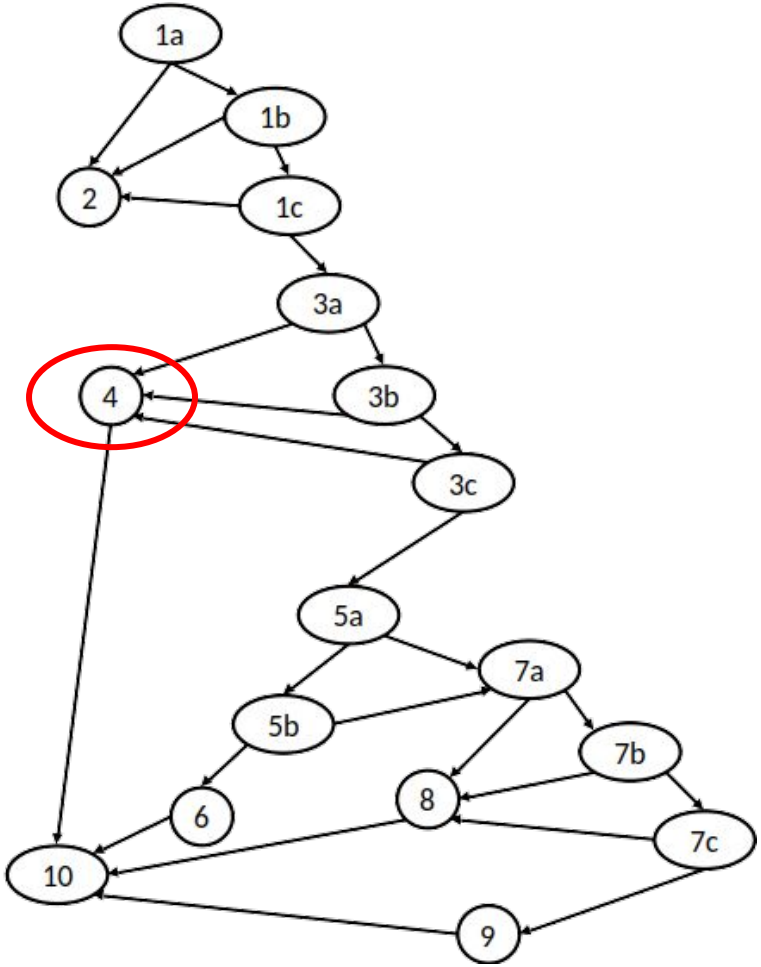
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



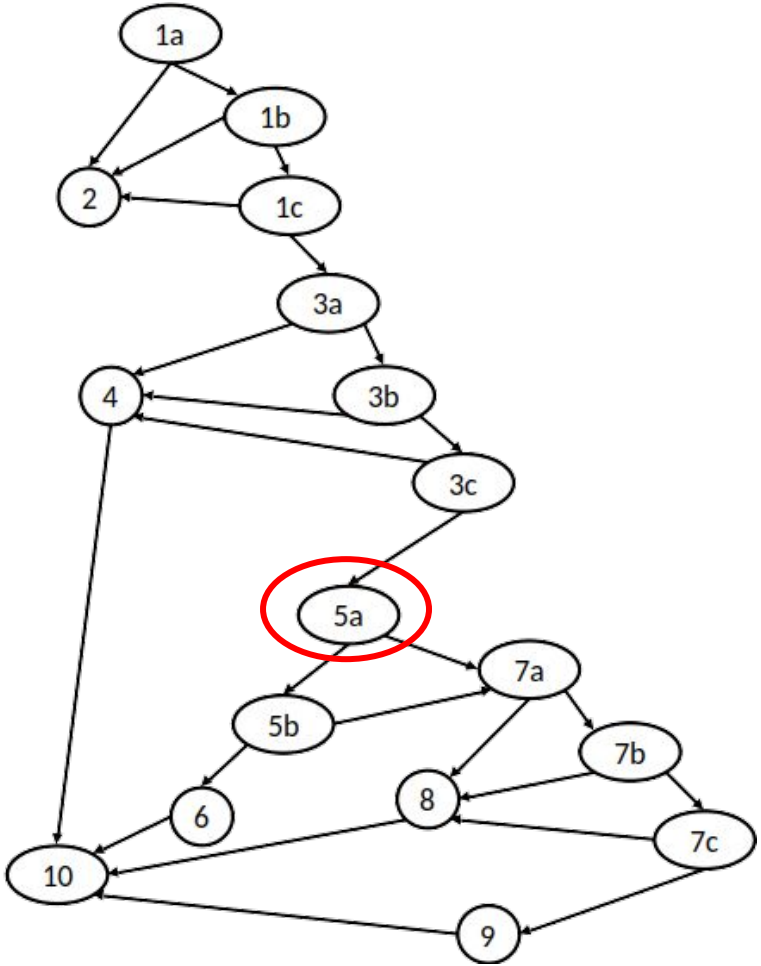
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



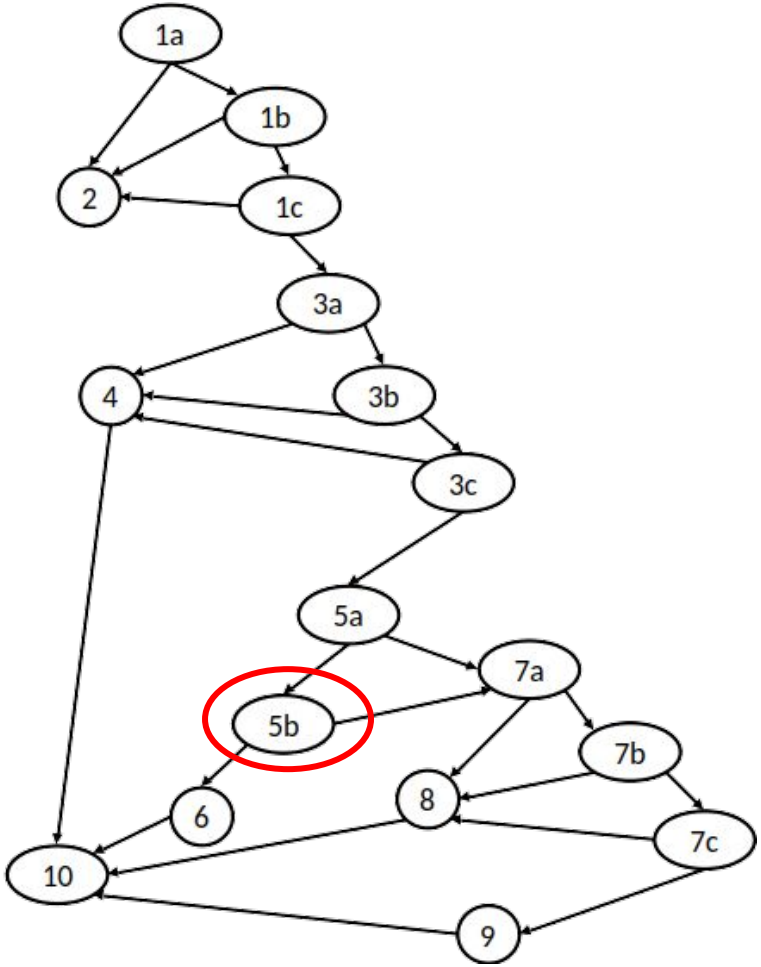
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
         resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



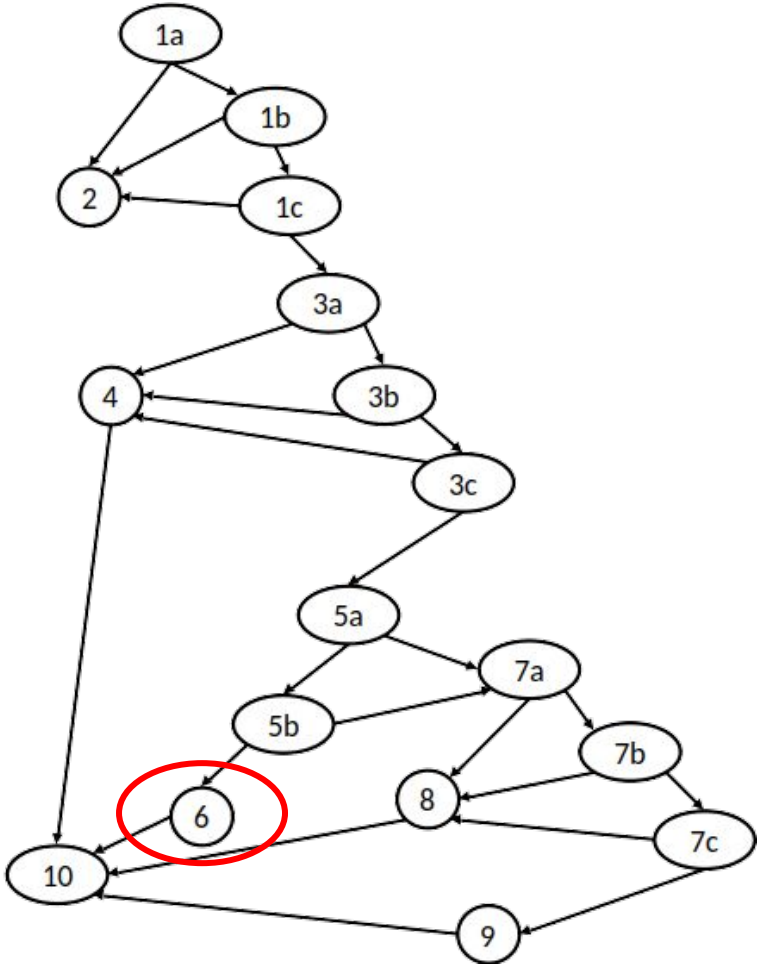
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



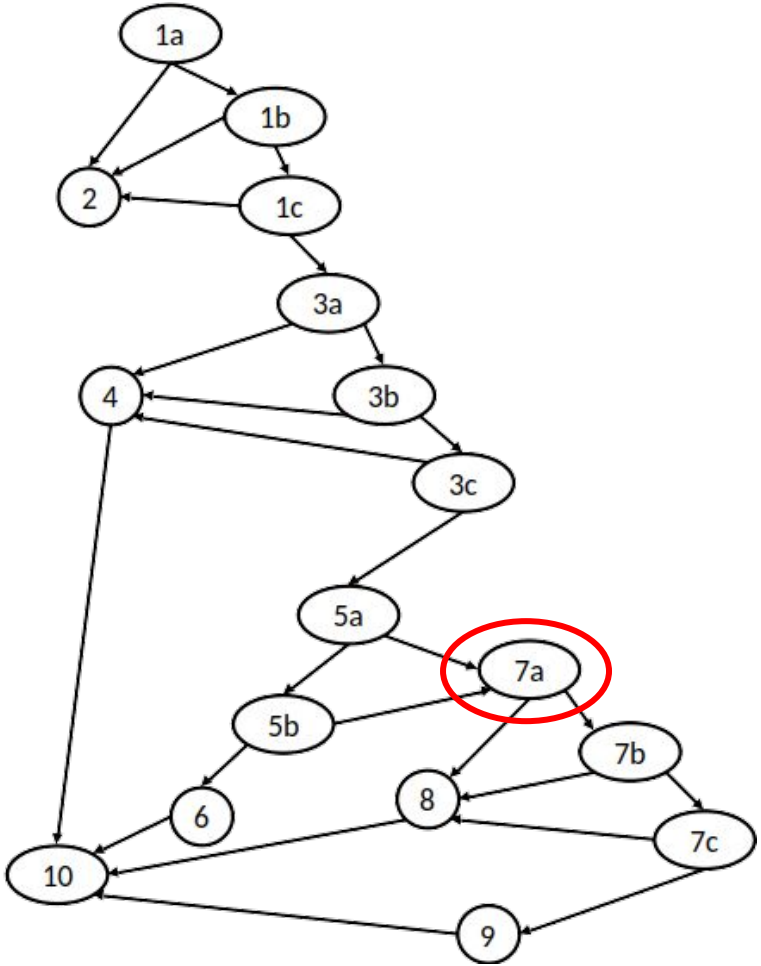
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
         resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



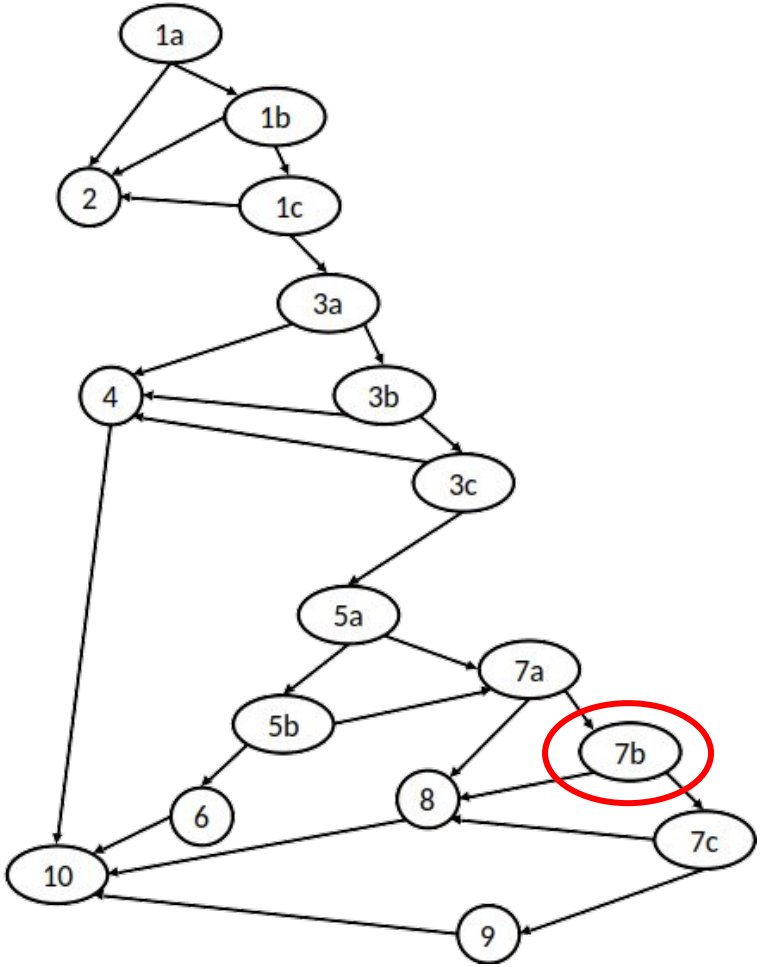
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



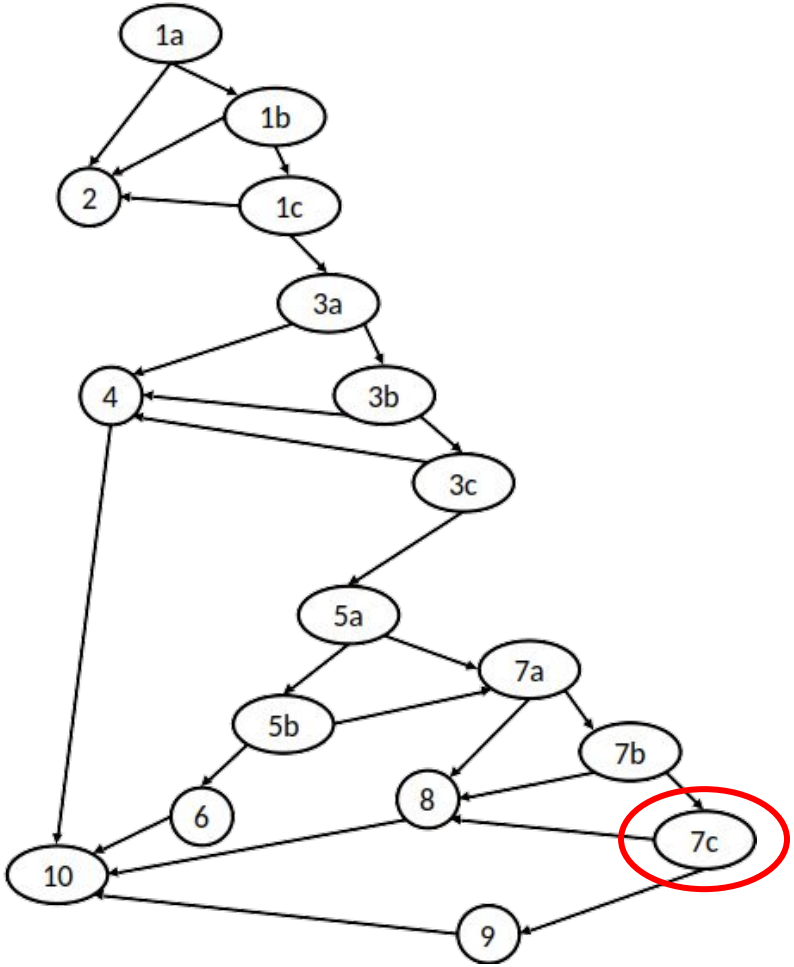
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



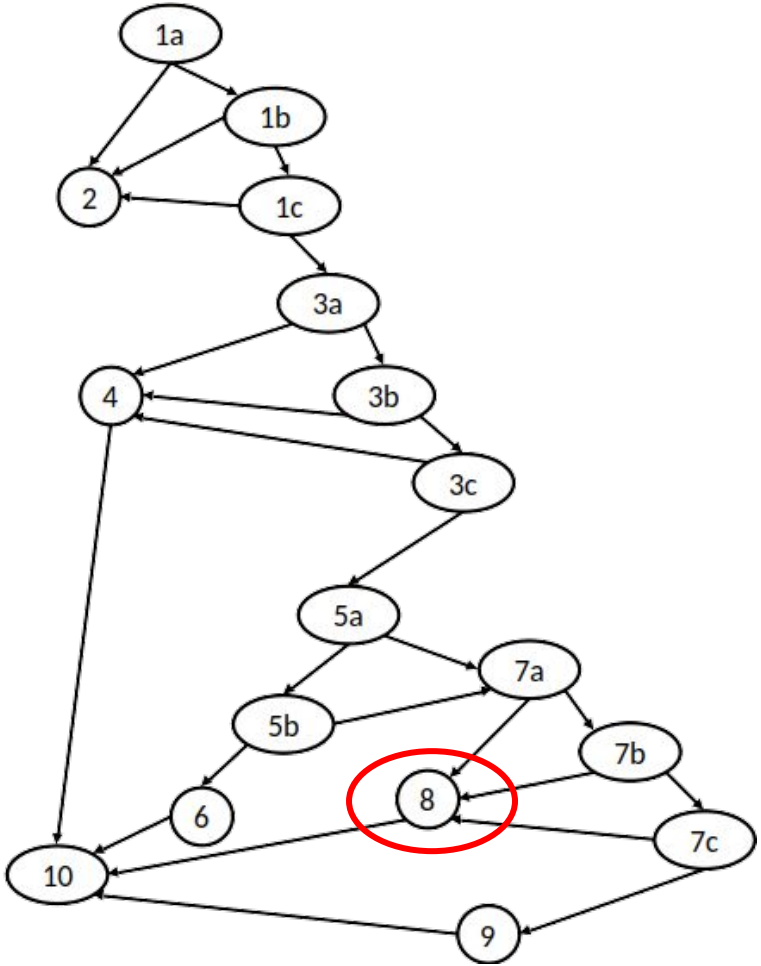
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



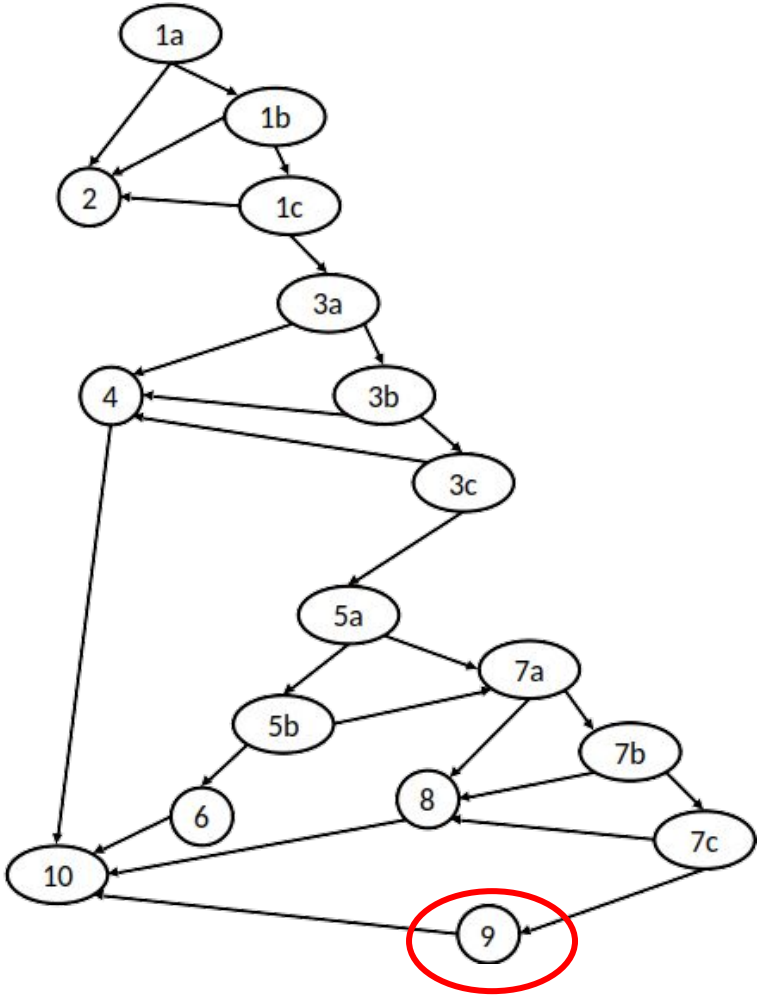
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



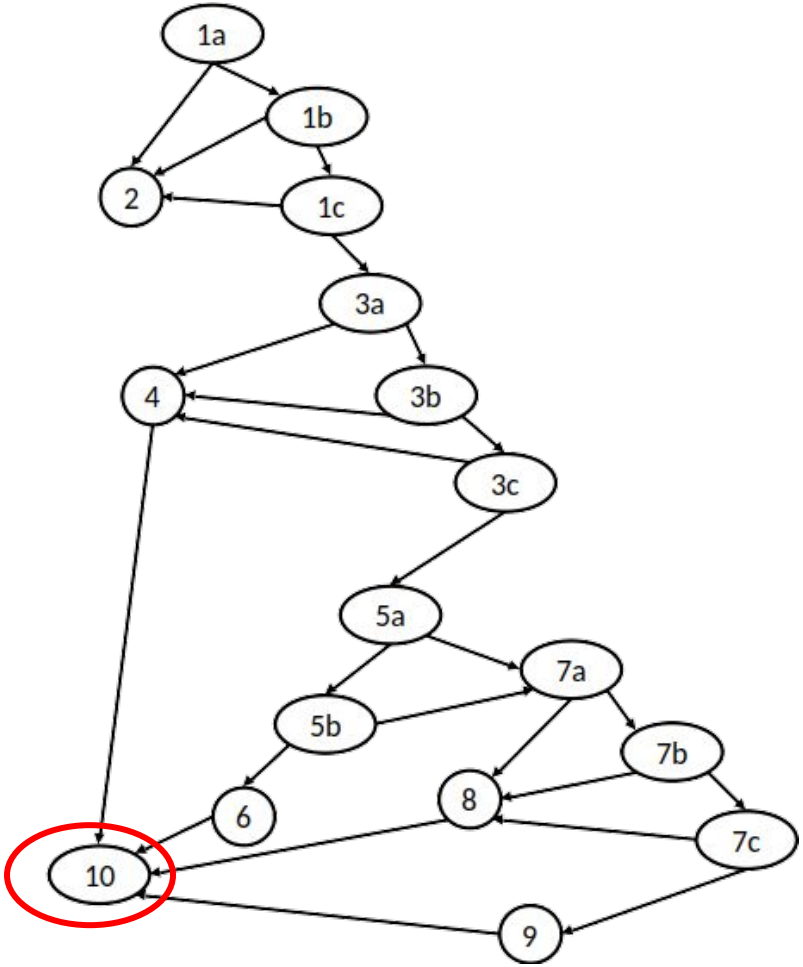
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



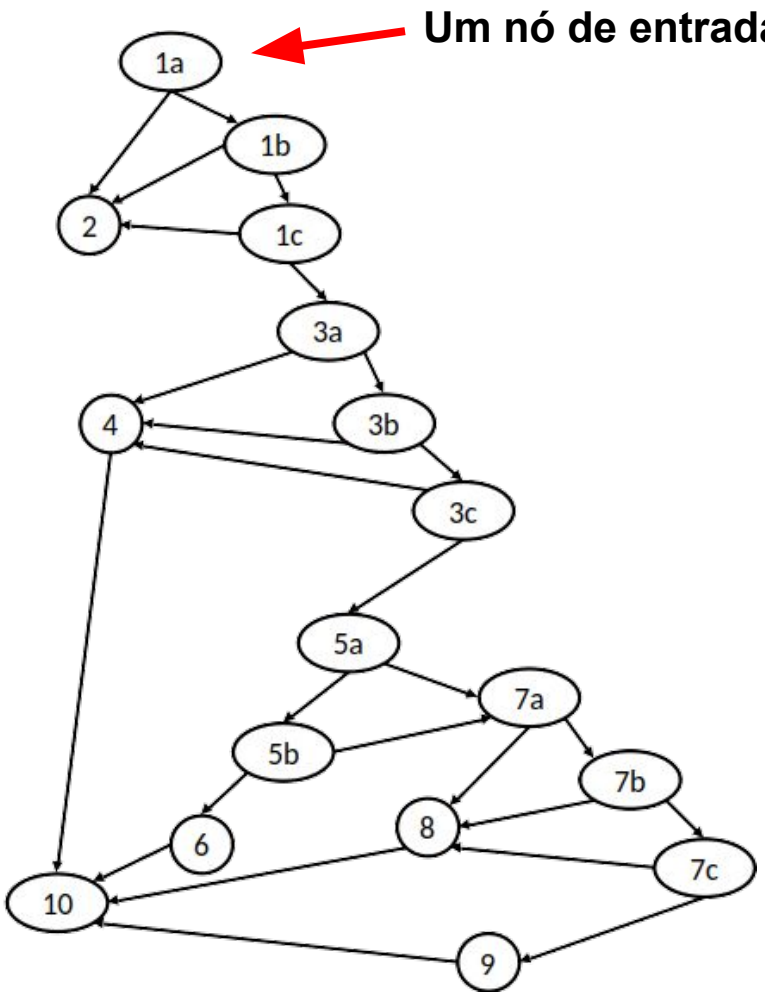
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
     }
  return resposta;
}
```



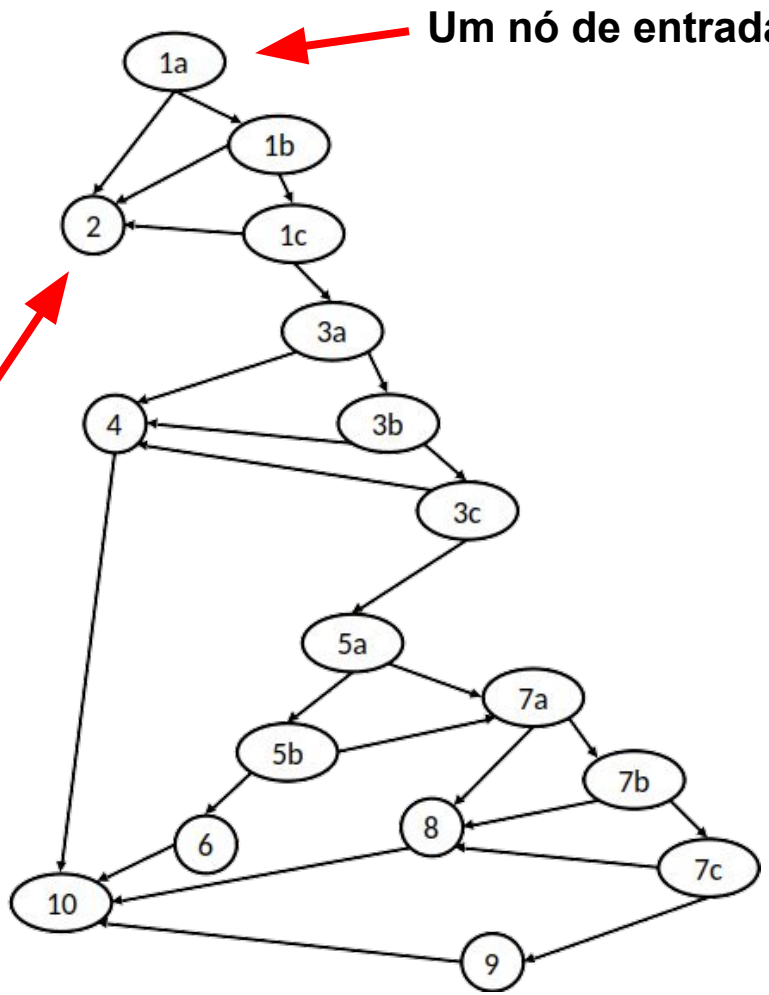
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



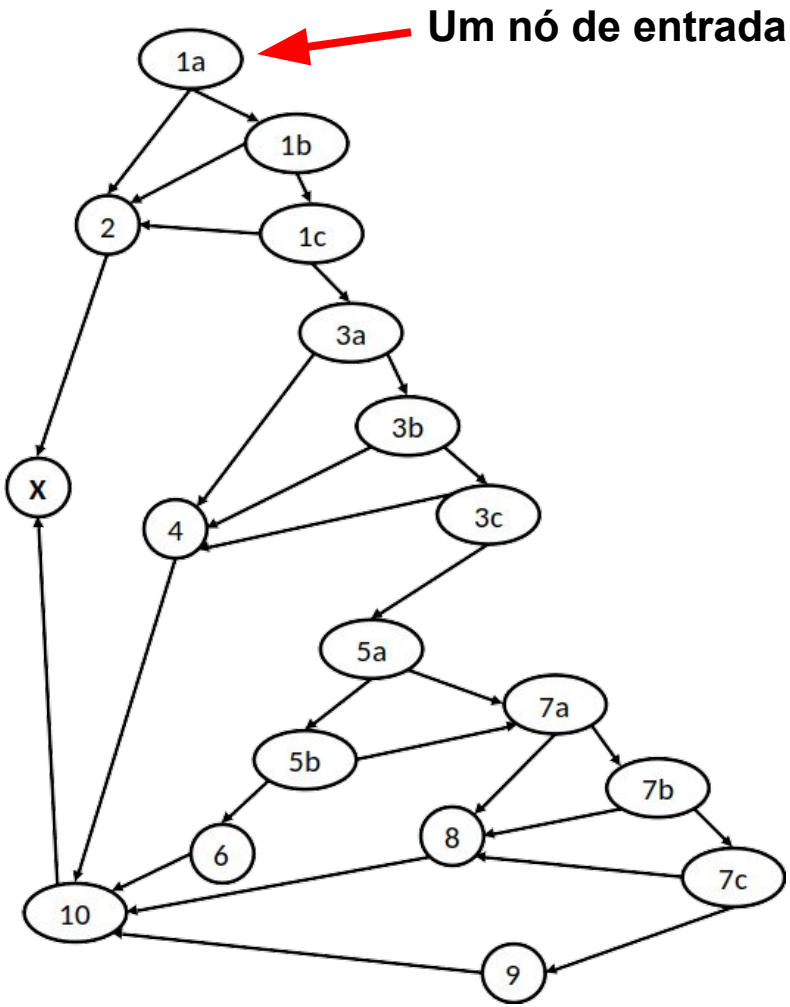
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



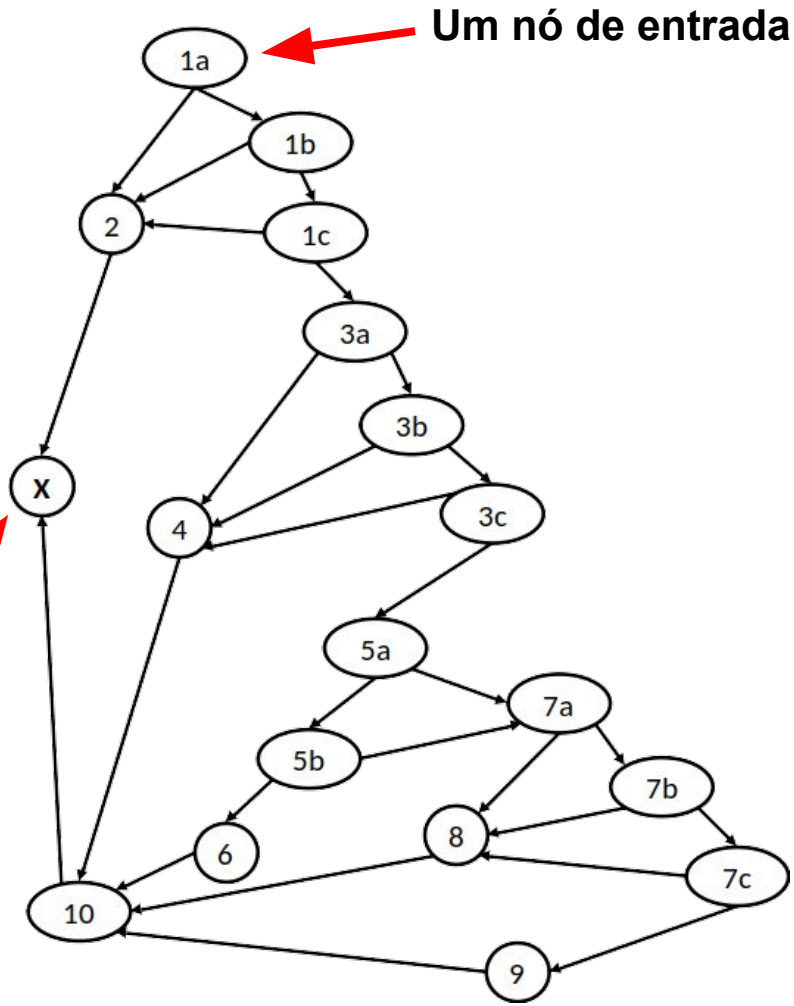
Exemplo 1 - Triângulo

```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



Exemplo 1 - Triângulo

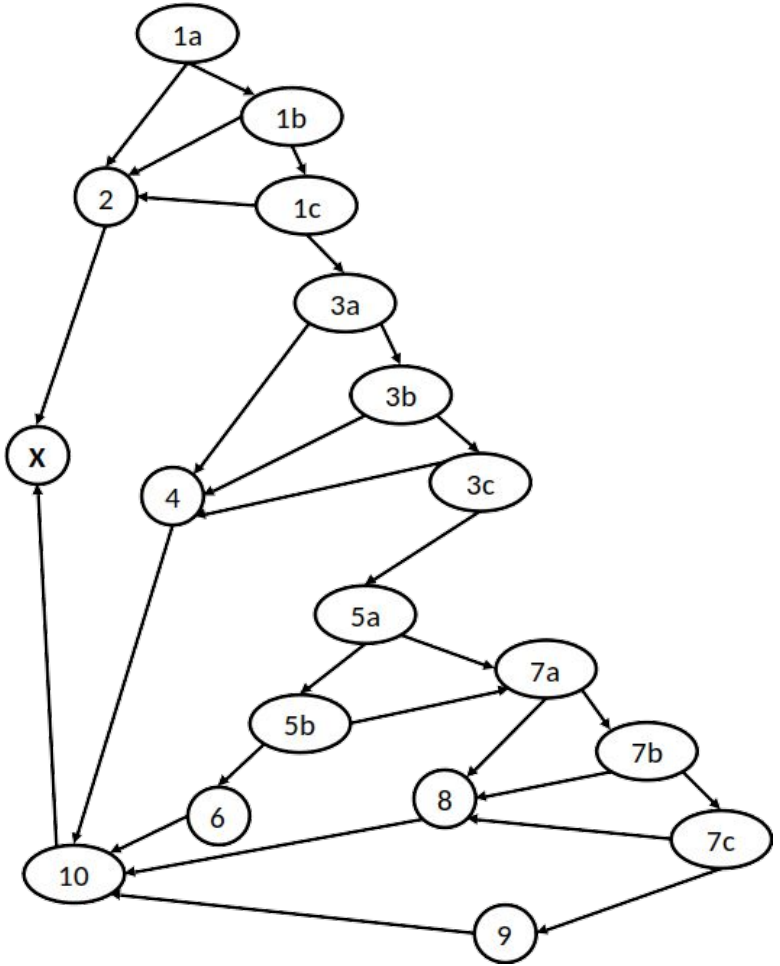
```
classificaTriangulo(int LA, int LB, int LC)
  String resposta="";
  if (LA<=0 || LB <=0 || LC <=0)
    throw new LadoInvalidoException("inválido");

  if ( (LA>=LB+LC) || (LB>=LA+LC) || (LC>=LA+LB))
    resposta = "NAO FORMA TRIANGULO";
  else {
     if (LA==LB && LB==LC)
        resposta = "EQUILATERO";
     else {
          if (LA==LB || LB==LC || LA==LC)
              resposta = "ISOSCELES";
          else
              resposta = "ESCALENO";
          }
      }
  return resposta;
}
```



Exemplo 1 - Triângulo

Critério todos os nós:

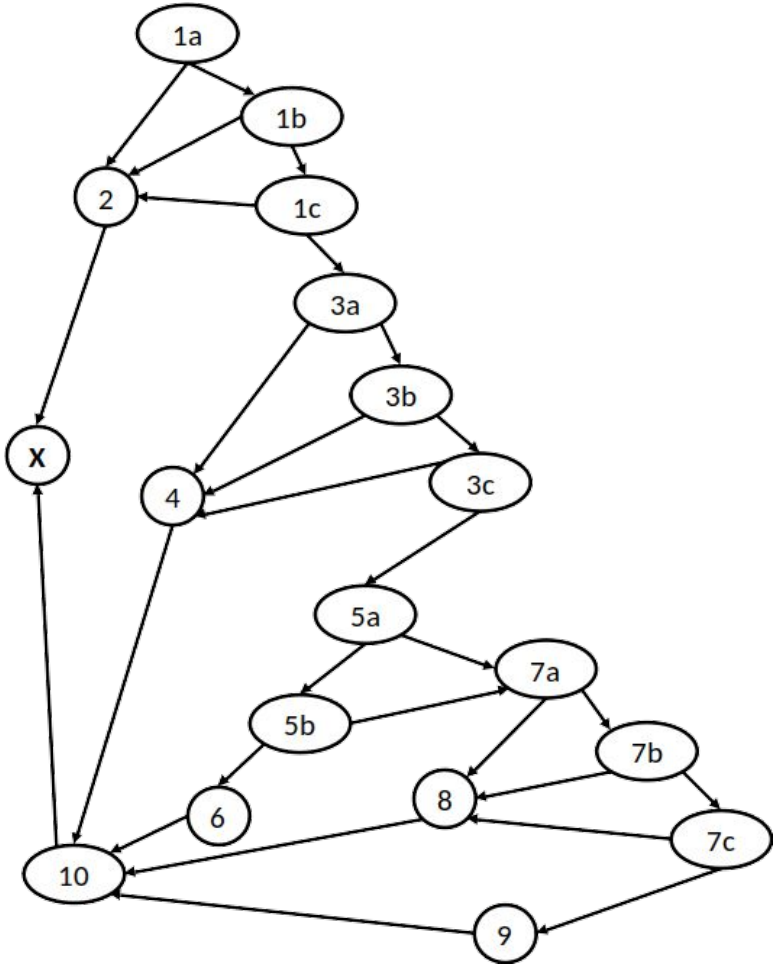
- 1a, 1b,1c,2,
- 3a,3b,3c,4,
- 5a,5b,6,
- 7a,7b,7c,
- 8,9,10,
- X



Exemplo 1 - Triângulo

Critério todas as arestas:

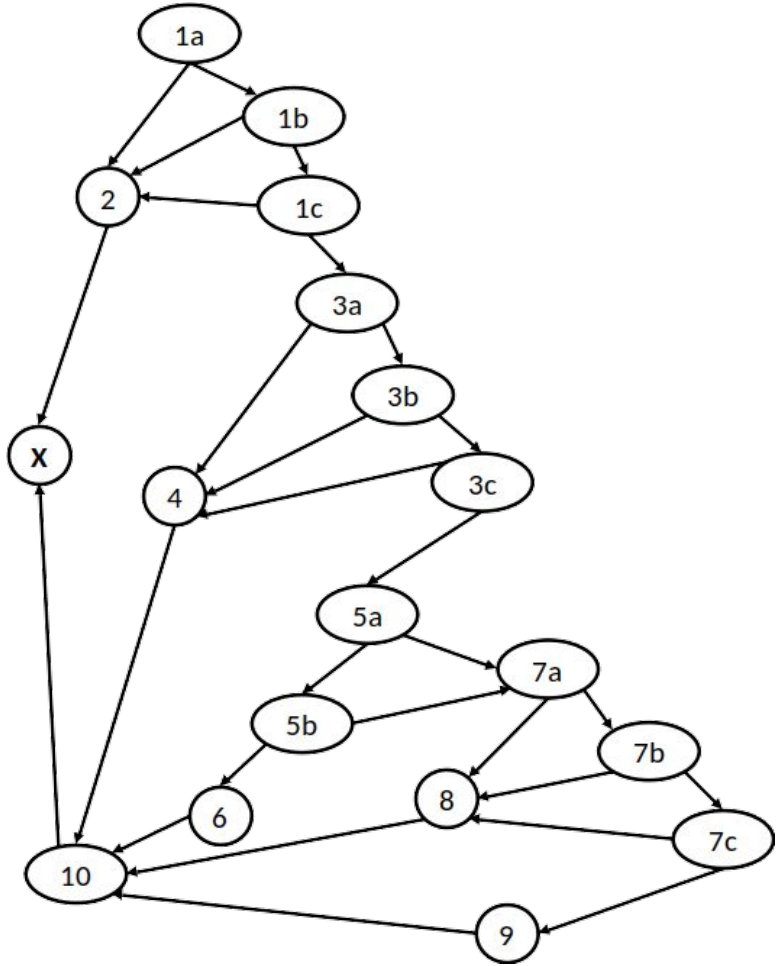
- (1a,1b), (1a,2), (1b,2),(1b,1c),
- (1c,2),(2,X),(1c,3a), (3a,3b),
- (3a,4),(3b,4),(3b,3c),(3c,4),
- (3c,5a),(4,10), (5a,7a),(5a,5b),
- (5b,7a),(5b,6),(6,10), (7a,7b),
- (7a,8),(7b,8),(7b,7c),(7c,8),
- (7c,9), (8,10), (9,10), (10,X)



Exemplo 1 - Triângulo

Critério todos os caminhos básicos

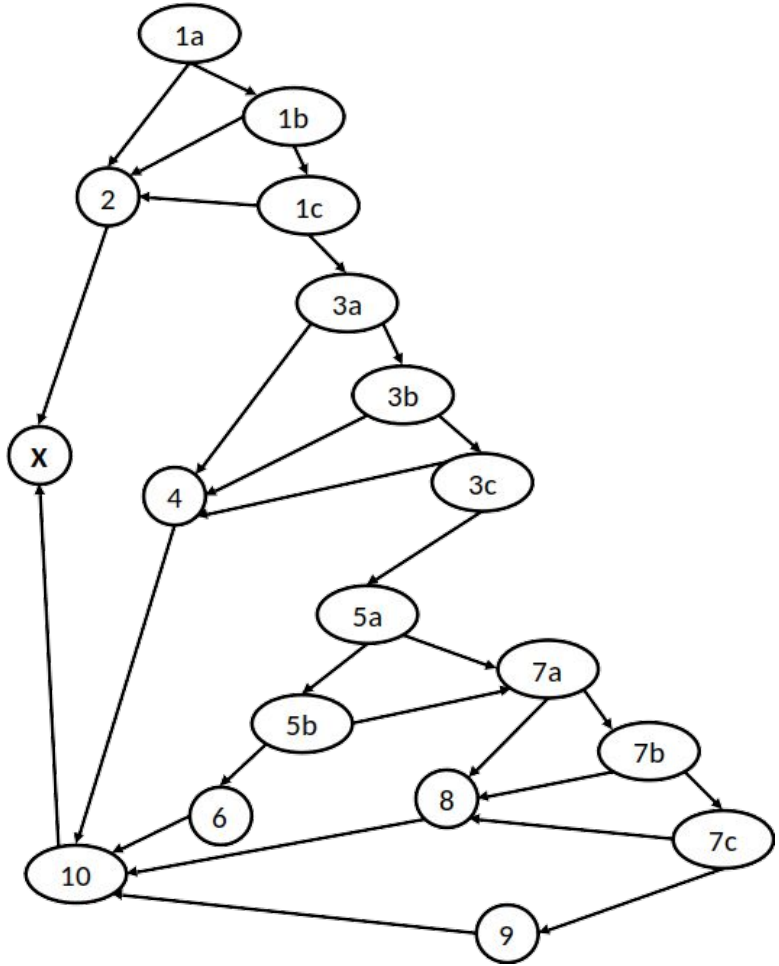
- Calcular complexidade ciclomática
- Complexidade ciclomática = A – N + 2
- Complexidade ciclomática = P + 1



Exemplo 1 - Triângulo

Critério todos os caminhos básicos

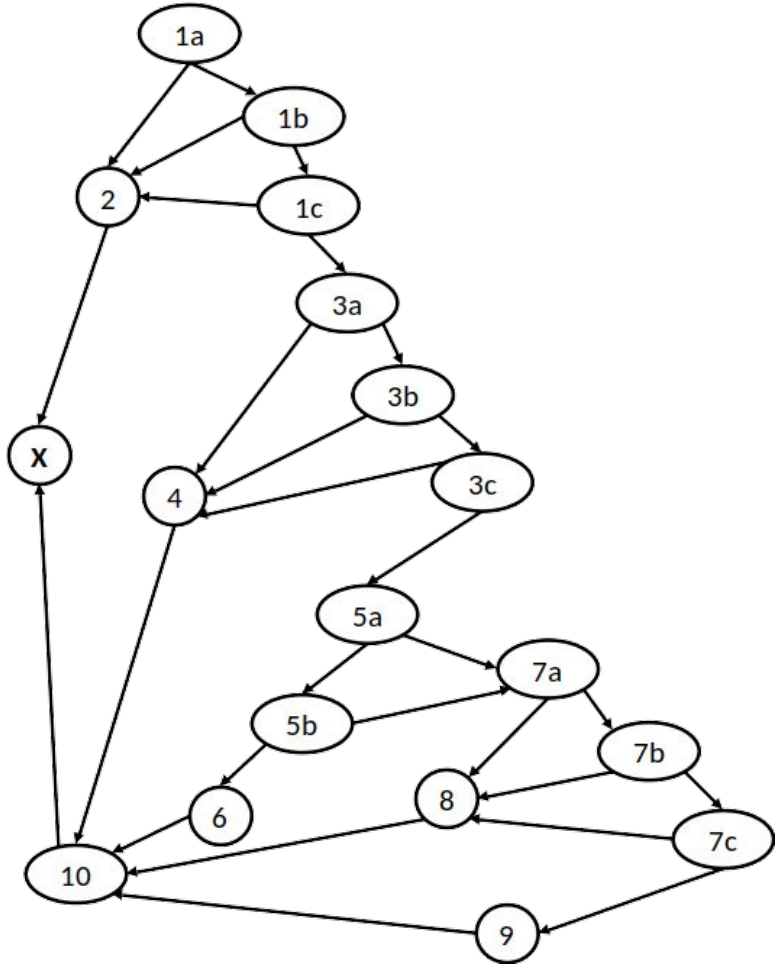
- Calcular complexidade ciclomática
- Complexidade ciclomática = A – N + 2
  - Arestas = 28
  - Nós = 18
  - CC = 28 - 18 + 2 => 12
- Complexidade ciclomática = P + 1
  - Predicados = 11
  - CC = 11 + 1



Exemplo 1 - Triângulo

Critério todos os caminhos básicos

- 1a – 2 – X
- 1a – 1b – 2 - X
- 1a – 1b – 1c – 2 – X
- 1a – 1b – 1c – 3a – 4 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 4 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 3c – 4 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 3c – 5a – 5b – 6 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 3c – 5a – 7a – 8 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 3c – 5a – 7a – 7b - 8 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 3c – 5a – 5b – 7a – 8 – 10 – X
- 1a – 1b – 1c – 3a – 3b – 3c – 5a – 7a – 7b – 7c - 8 – 10 - X
- 1a – 1b – 1c – 3a – 3b – 3c – 5a – 7a – 7b – 7c - 9 – 10 - X



Derivando casos de teste

É preciso criar casos de teste para cobrir os requisitos de teste derivados de cada critério

Os casos de testes precisam passar por todos os nós

Os casos de testes precisam passar por todas as arestas

Os casos de testes precisam exercitar todos os caminhos básicos

Exemplo do Triângulo

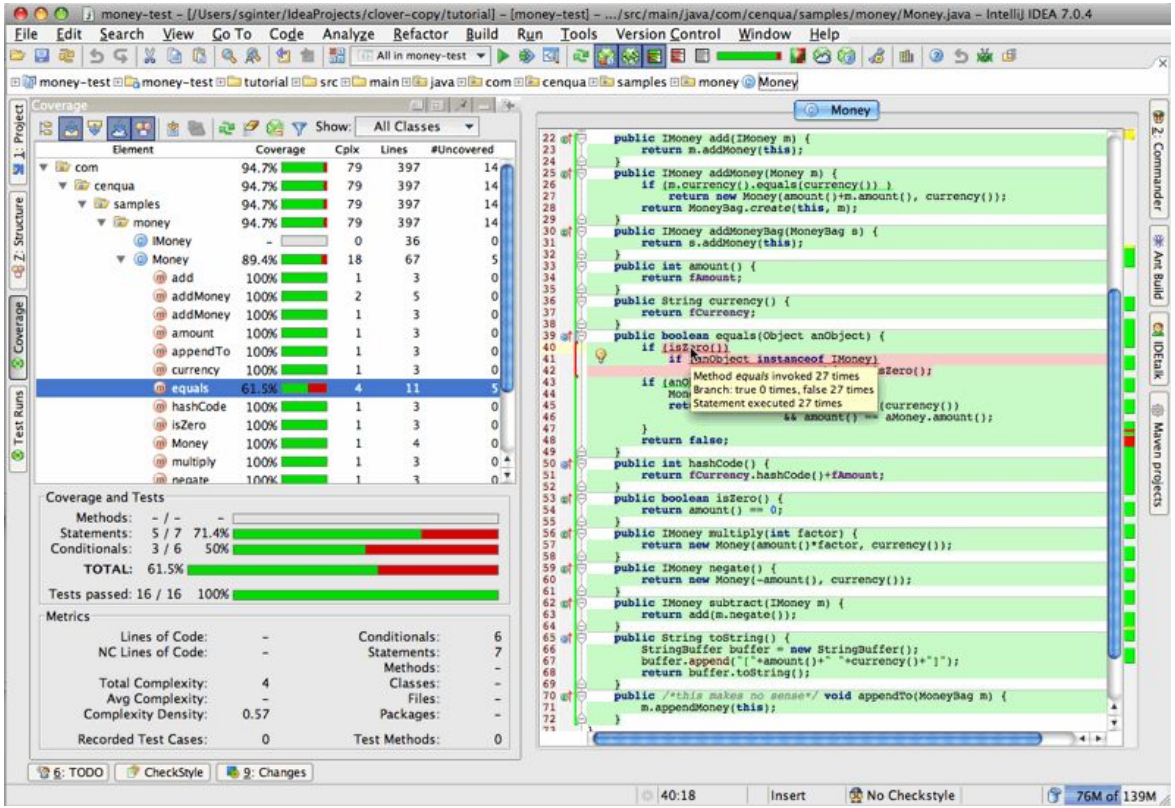
Ver arquivo:

- TesteEstrutural-ExemploTrianguloCobertura.pdf

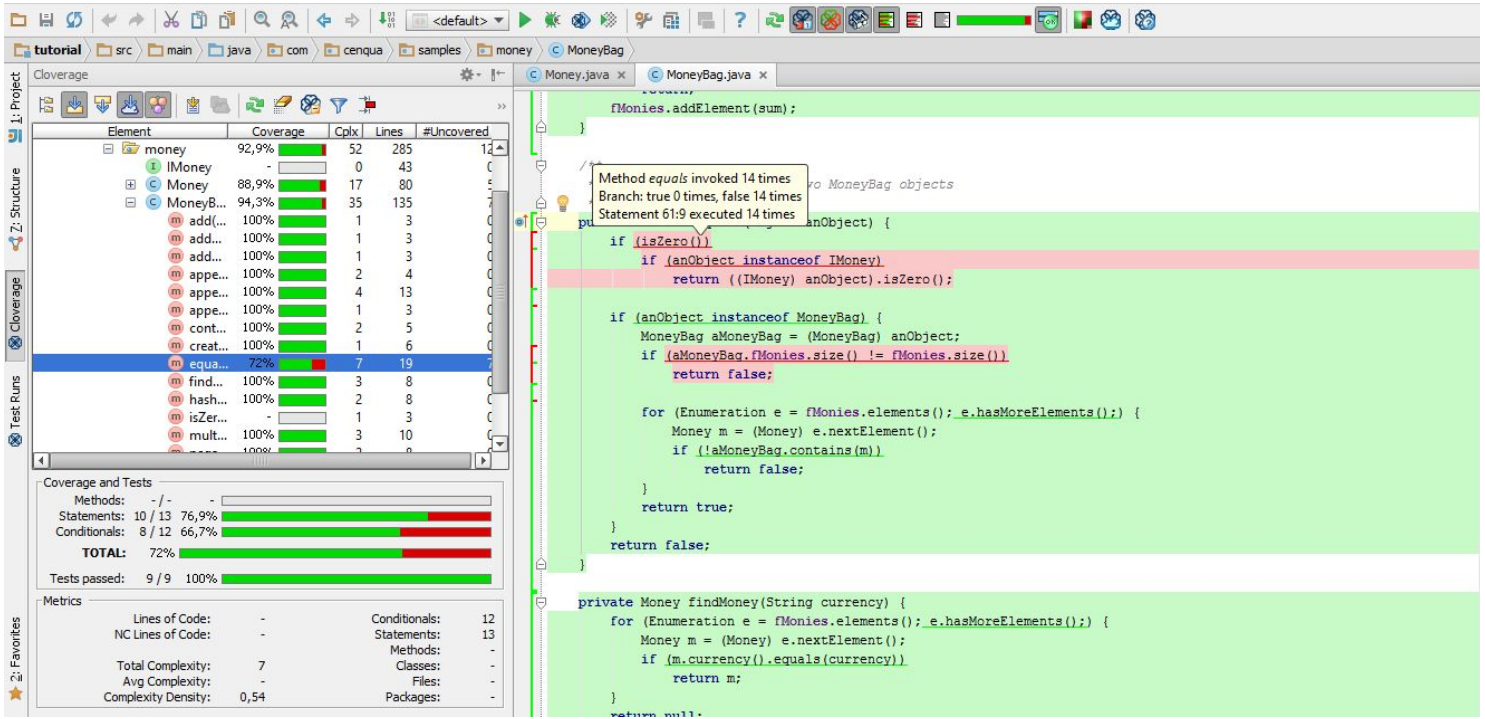
Link:

- https://drive.google.com/file/d/1IidjhH5RAgG-qwx2kL5ShGR3Q2wdMGMM/view?usp=sharing

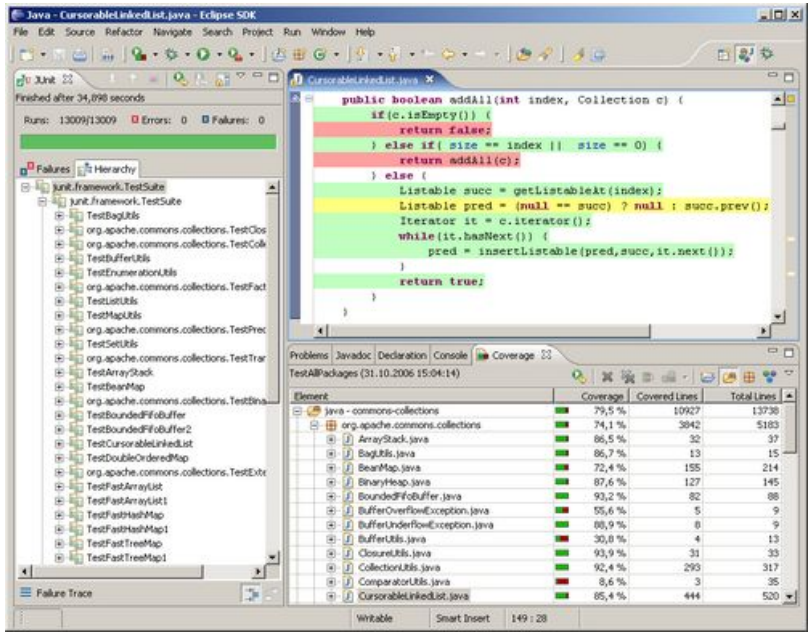
Ferramentas



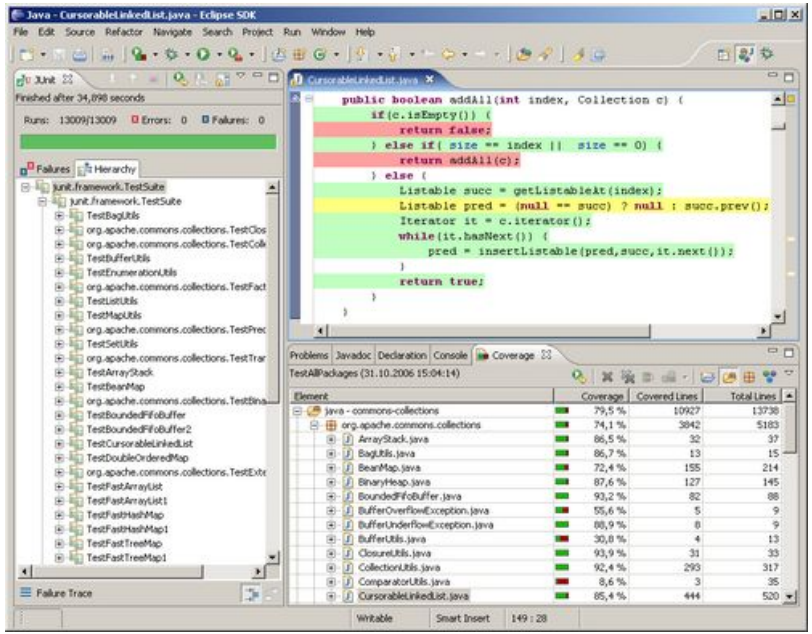
Ferramentas



Ferramentas



Ferramentas



Uma demonstração

Exemplo do triângulo

Vídeo: cobertura de código com IntelliJ + Gradle + Jacoco

```
public static TipoTriangulo classificaTriangulo (int A, int B, int C){
   if (temLadoInvalido(A,B,C))
       return TipoTriangulo.LADOINVALIDO;
   if (!formaTriangulo(A,B,C))
       return TipoTriangulo.NAOTRIANGULO;
   if (A==B && B==C)
       return TipoTriangulo.EQUILATERO;
   if (A==B || B==C || A==C)
       return TipoTriangulo.ISOSCELES;
   return TipoTriangulo.ESCALENO;
}
```

ACH 2028
Qualidade de Software

Aula 08 - Teste de Software: Técnica Estrutural ou Caixa-Branca

Prof. Marcelo Medeiros Eler
marceloeler@usp.br

Escola de Artes, Ciências e Humanidades - Universidade de São Paulo