FIAP GRADUAÇÃO

GOVERNANÇA E QUALIDADE DE SOFTWARE

DISCIPLINA: **PROJETO DE SISTEMAS APLICADO AS MELHORES PRÁTICAS EM QUALIDADE DE SOFTWARE E GOVERNANÇA DE TI**

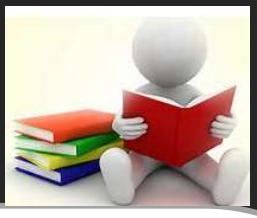
AULA **22 – TESTE UNITÁRIO DE SOFTWARE**
ESTUDO DE CASO

PROFESSOR:
RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR

Renato Parducci - YouTube

ESTUDO DE CASO SIMULADO

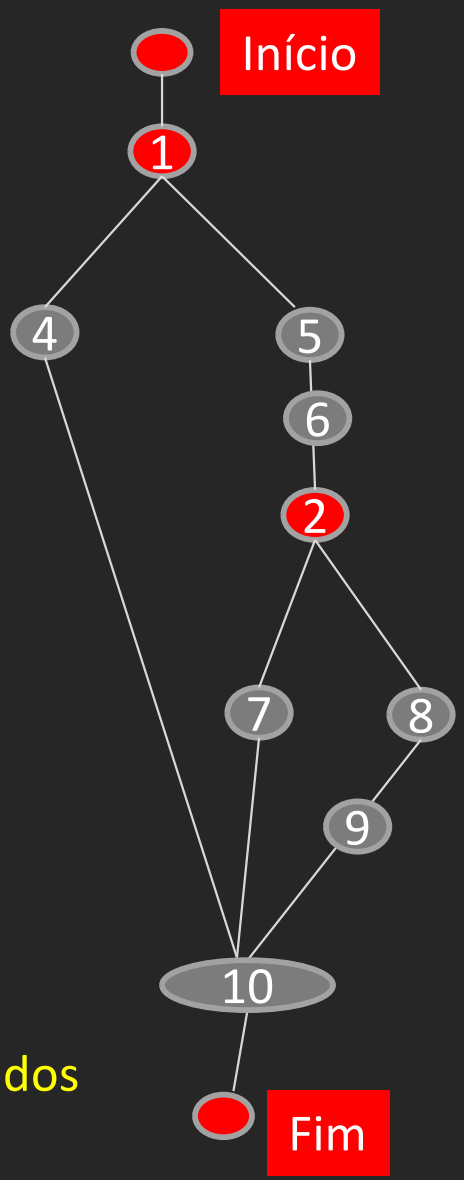


Desenhe o grafo de caminhos de teste e calcule a Complexidade Ciclomática do trecho de algoritmo que foi passado para você avaliar.
Calcule a Complexidade Ciclomática da transação.

```
SE x = False  ENTÃO
  y= y +1
SENÃO
  y=y -1
  SE w+z >= 72  ENTÃO
     y=y+2
  SENÃO
     y=y-2
   FIM-SE
FIM-SE
```

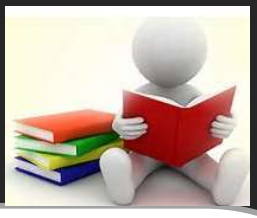
SOLUÇÃO

```
SE x = False  ENTÃO          1
  y= y +1                    4
SENÃO                        5
  y=y -1                     6
  SE w+z >= 72  ENTÃO        2
     y=y+2                   7
  SENÃO                      8
     y=y-2                   7
   FIM-SE                    9
FIM-SE                       10
```



Complexidade Ciclomática = 2 + 1 = 3 testes a serem produzidos

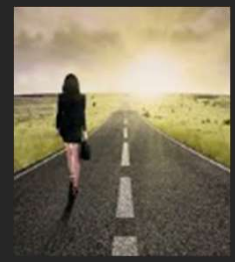
ESTUDO DE CASO SIMULADO



Crie agora, um conjunto de Casos de Testes com valores de Input e Output previsto para todas as variáveis envolvidas na transação, utilizando a regra de Caminhos Mínimos.

```
SE x = False  ENTÃO
  y= y +1
SENÃO
  y=y -1
  SE w+z >= 72  ENTÃO
     y=y+2
  SENÃO
     y=y-2
   FIM-SE
FIM-SE
```

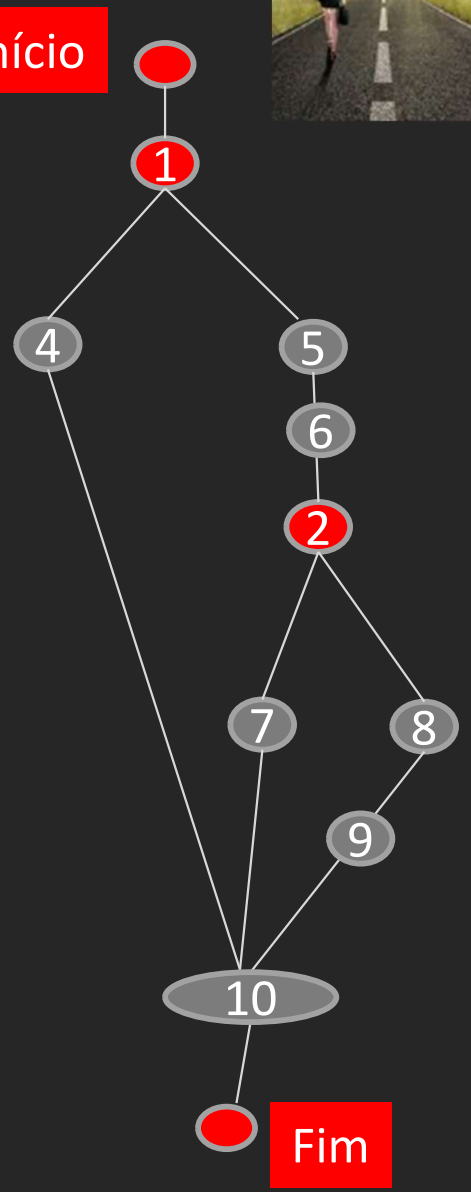
COMO PLANEJAR UM CASO DE TESTE



```
SE x = False  ENTÃO
   y= y +1
SENÃO
   y=y -1
   SE w+z >= 72  ENTÃO
      y=y+2
   SENÃO
      y=y-2
    FIM-SE
FIM-SE
```

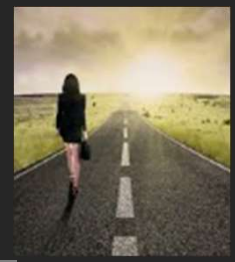
Nosso casos de teste mínimos para garantir que todos os trechos do código foram executados ao menos uma vez, poderiam ser:

1) ENTRADA:X = False, Y =1; SAÍDA: X=False, Y=2
2) ENTRADA:X = True, W=50, Z= 50, Y=1; SAÍDA: X=True, Y=2, W=50, Z= 50
3) ENTRADA: X = True, W=2, Z=1, Y=1; SAÍDA: X=True, W=2,Z=1; Y= -2



Complexidade Ciclomática = 2 + 1 = 3 testes a serem produzidos

COMO PLANEJAR UM CASO DE TESTE



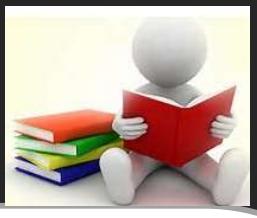
```
IF x = False THEN
   y= y +1
ELSE
   y=y -1
   IF w+z >= 72 THEN
       y=y+2
   ELSE
       y=y-2
    ENDIF
ENDIF
```

Escrevemos então as Fichas de Casos de Testes com as Entradas e Saídas previstas e aplicamos esses testes quando o código-fonte estiver pronto, para confirmarmos o funcionamento correto da aplicação de software:

1) ENTRADA:X = False, Y =1; SAÍDA: X=False, Y=2
2) ENTRADA:X = True, W=50, Z= 50, Y=1; SAÍDA: X=True, Y=2, W=50, Z= 50
3) ENTRADA: X = True, W=2, Z=1, Y=1; SAÍDA: X=True, W=2,Z=1; Y= -2

Complexidade Ciclomática = 2 + 1 = 3 testes a serem produzidos

ESTUDO DE CASO SIMULADO



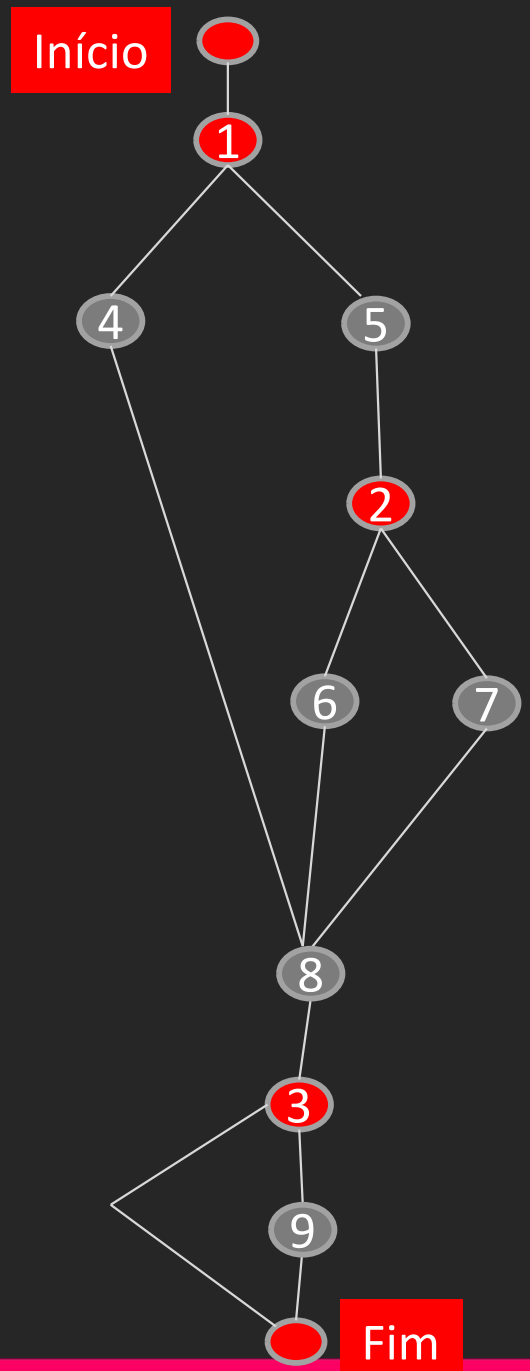
Desenhe o grafo de caminhos de teste e calcule a Complexidade Ciclomática do trecho de algoritmo que foi passado para você avaliar.
Calcule a Complexidade Ciclomática da transação.

```
IF x = False THEN
  y= y +1
ELSE
  y=y -1
  IF w+z >= 72 THEN
      y=y+2
  ELSE
      y=y-2
   ENDIF
ENDIF

r = r+y

IF r = w+z THEN
    r = 0
ENDIF
```

COMO PLANEJAR UM CASO DE TESTE

Estrutura de decisão	Estrutura não decisórias

```
IF x = False THEN                    (1)
   y= y +1                           (4)
ELSE
   y=y -1                            (5)
   IF w+z >= 72 THEN                 (2)
      y=y+2                          (6)
   ELSE
      y=y-2                          (7)
    ENDIF
ENDIF

r = r+y                              (8)

IF r = w+z THEN                      (3)
    r = 0                            (9)
ENDIF
```

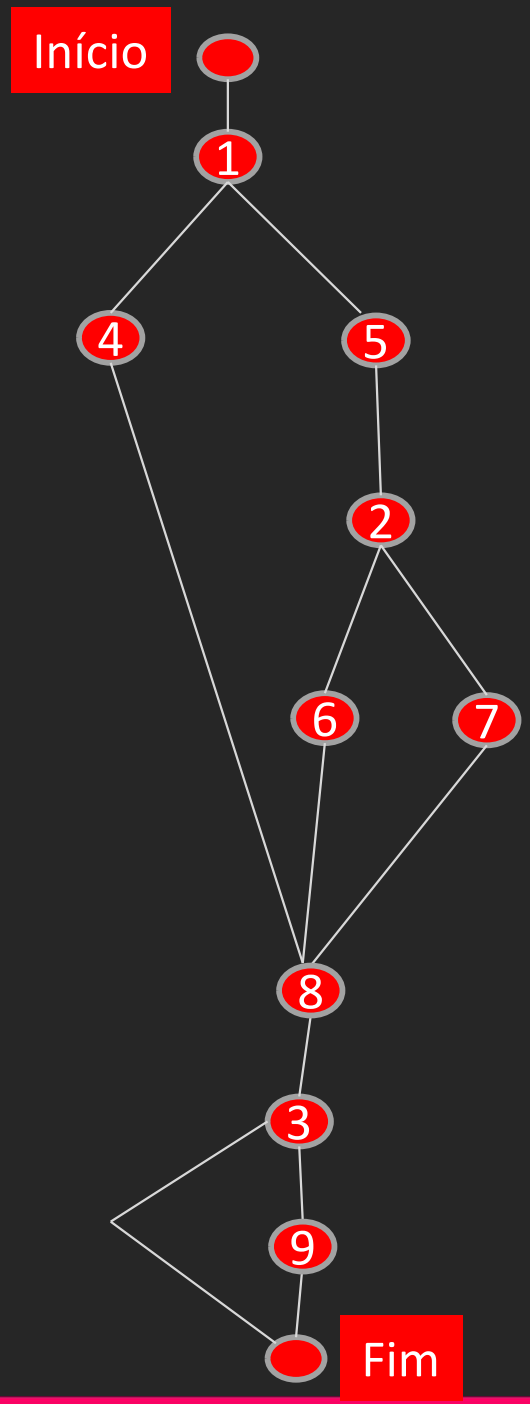


COMO PLANEJAR UM CASO DE TESTE

TOTAL DE 5 CASOS NECESSÁRIOS DE TESTES

SUBREDE PARALELA 1
2 PONTOS DE DECISÃO

SUBREDE PARALELA 2
1 PONTO DE DECISÃO



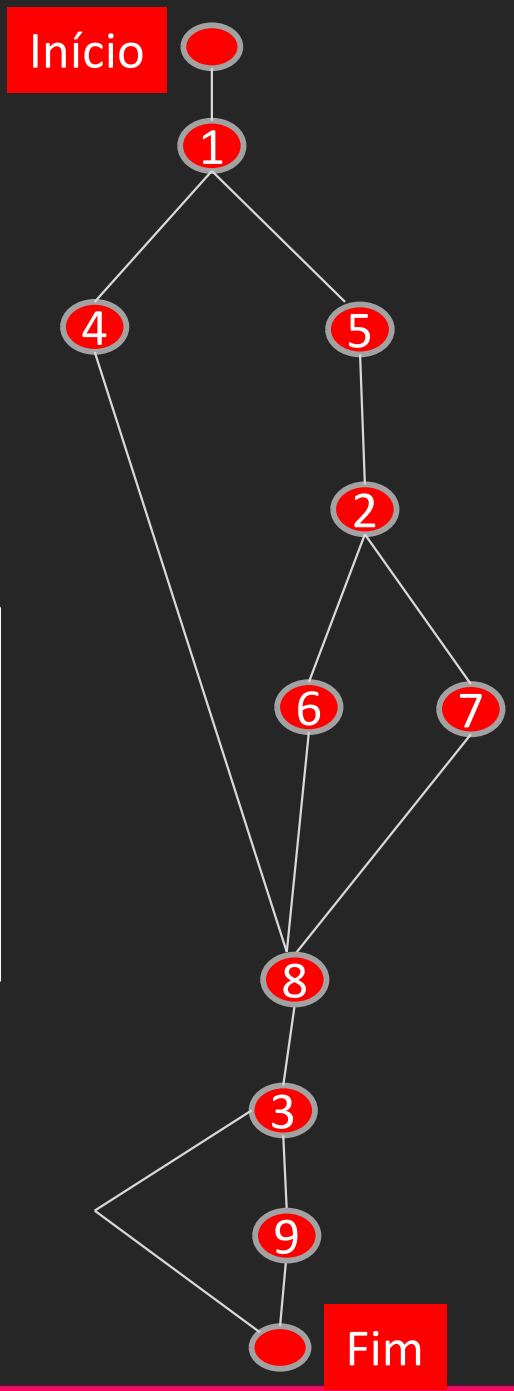
COMO PLANEJAR UM CASO DE TESTE

TOTAL DE 5 CASOS NECESSÁRIOS DE TESTES

SUBREDE PARALELA 1
2 PONTOS DE DECISÃO

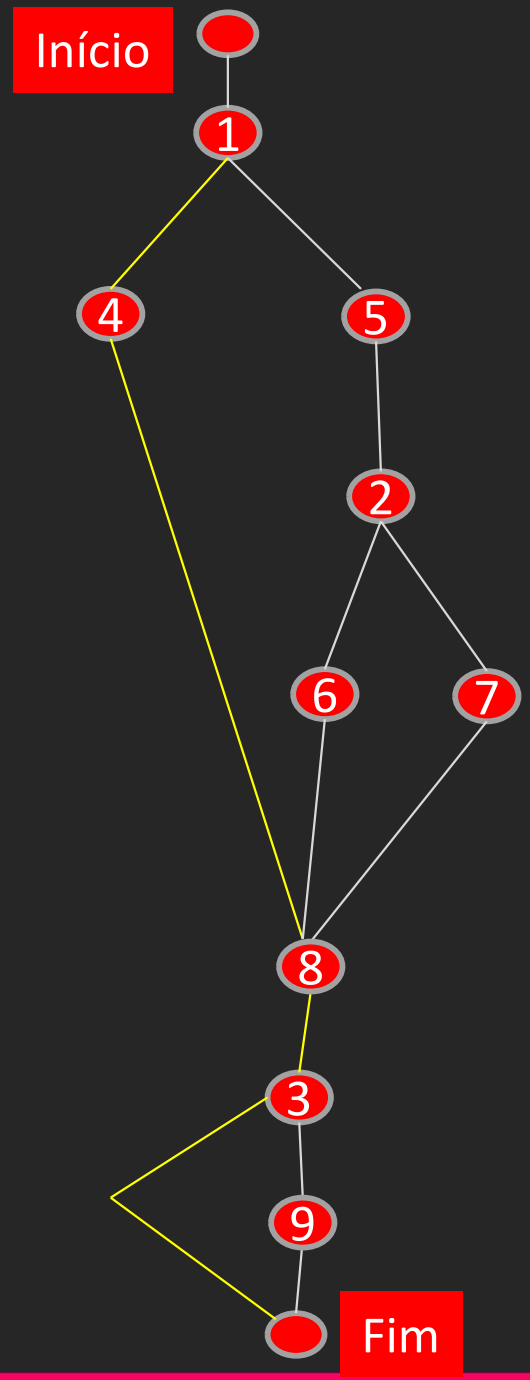
SUBREDE PARALELA 2
1 PONTO DE DECISÃO

TOTAL DE TESTES =
2+1 **+1 =**
4 CASOS



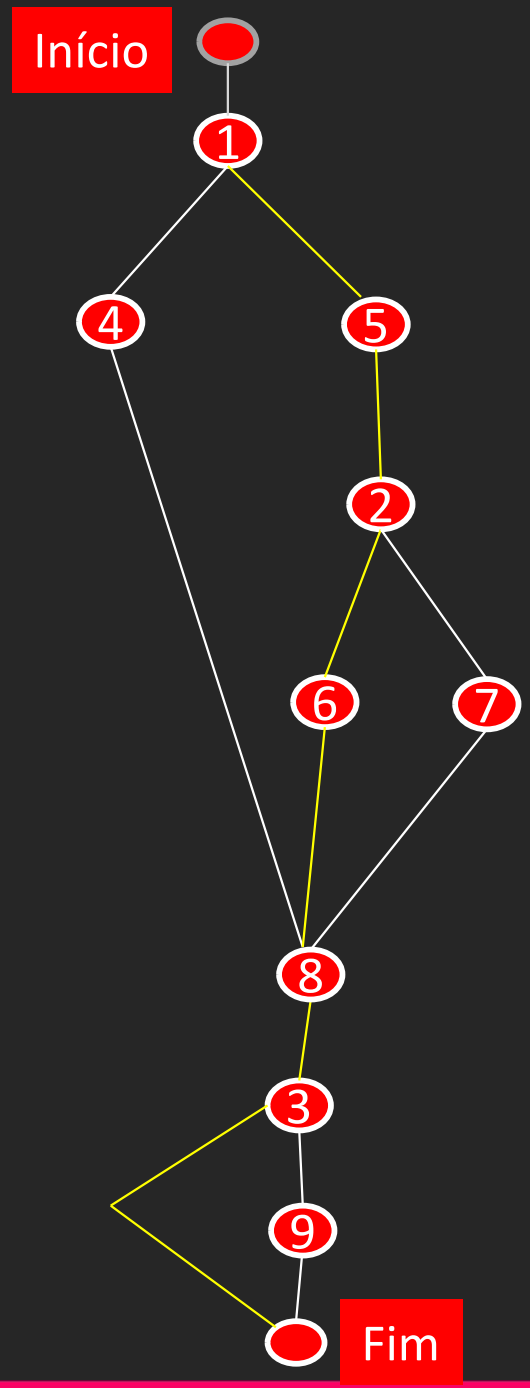
COMO PLANEJAR UM CASO DE TESTE

O caso 1 deve ter controle de entrada de dados que permita percorrer o caminho indicado ao lado



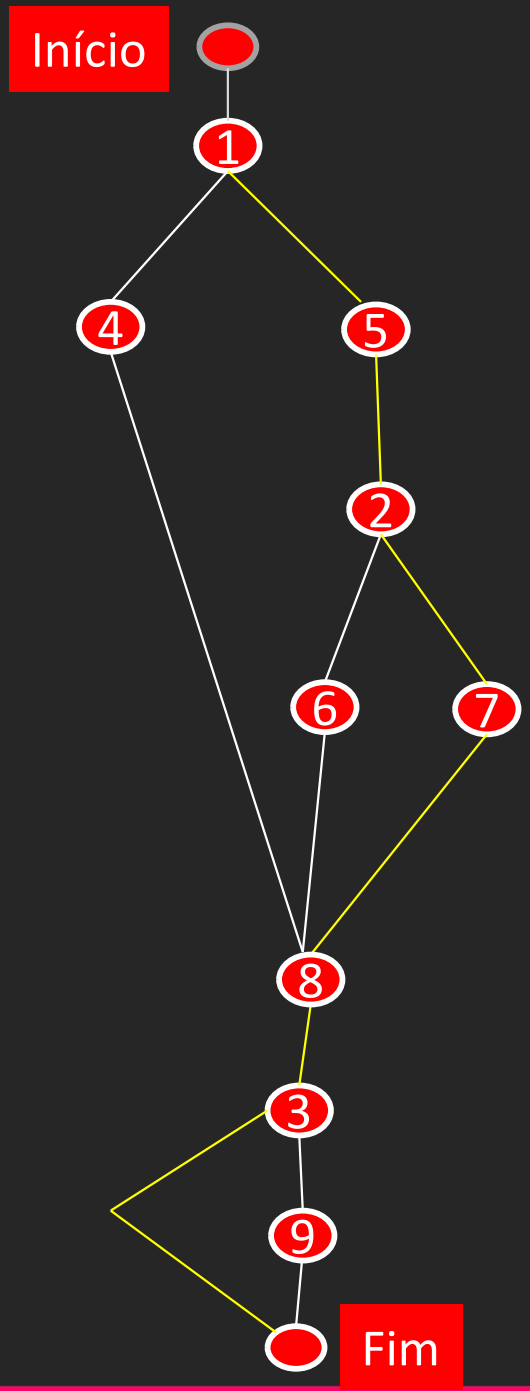
COMO PLANEJAR UM CASO DE TESTE

O caso 2 deve ter controle de entrada de dados que permita percorrer o caminho indicado ao lado



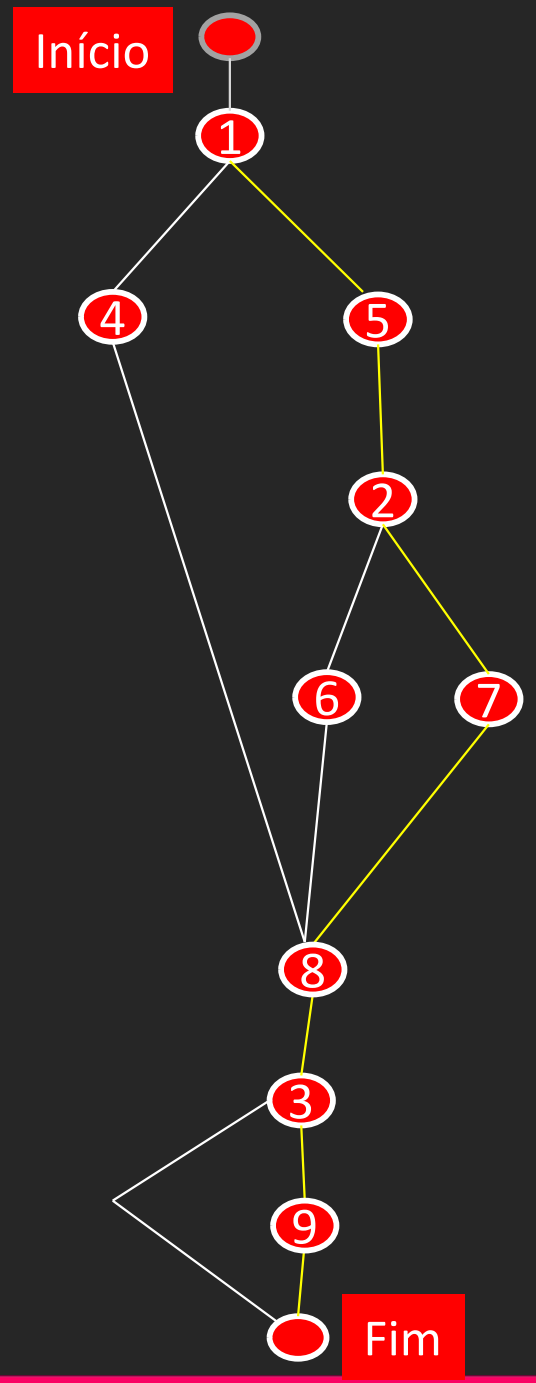
COMO PLANEJAR UM CASO DE TESTE

O caso 3 deve ter controle de entrada de dados que permita percorrer o caminho indicado ao lado

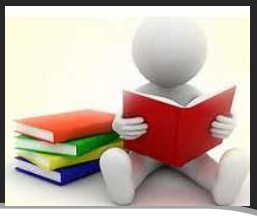


COMO PLANEJAR UM CASO DE TESTE

O caso 4 deve ter controle de entrada de dados que permita percorrer o caminho indicado ao lado



ESTUDO DE CASO SIMULADO



Crie os casos de teste unitários de caixa branca, com base na regra de caminhos mínimos.

```
IF x = False THEN
  y= y +1
ELSE
  y=y -1
  IF w+z >= 72 THEN
      y=y+2
  ELSE
      y=y-2
   ENDIF
ENDIF

r = r+y

IF r = w+z THEN
    r = 0
ENDIF
```

```
IF x = False THEN
   y= y +1
ELSE
   y=y -1
   IF w+z >= 72 THEN
        y=y+2
   ELSE
        y=y-2
    ENDIF
ENDIF

r = r+y

IF r = w+z THEN
     r = 0
ENDIF
```

ELABORE OS CASOS DE TESTES UNITÁRIOS DE CAIXA PRETA E FUNCIONAIS, BASEANDO-SE NA COMPLEXIDADE CICLOMÁTICA!

USE O TEMPLATE DE DEFINIÇÃO DE CASOS DE TESTES, PREPARANDO A MASSA DE DADOS DE ENTRADA E SAÍDA E OS PASSOS DO TESTE, IMAGINANDO QUE ESTE PROGRAMA É UM MÉTODO CALC_NUM DA CLASSE CLA_NUM

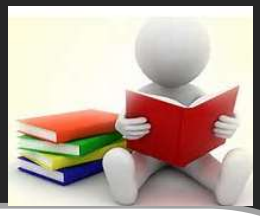
AS VARIÁVEIS SERÃO RECEBIDAS POR PARÂMETRO

```
IF x = False THEN
   y= y +1
ELSE
   y=y -1
   IF w+z >= 72 THEN
       y=y+2
   ELSE
       y=y-2
    ENDIF
ENDIF

r = r+y

IF r = w+z THEN
    r = 0
ENDIF
```

Nosso casos de teste mínimos para garantir que todos os trechos do código foram executados ao menos uma vez, poderiam ser:

1) ENTRADA:X = True, R=0; W=50, Z= 50, Y=1; SAÍDA: X=True, Y=2; R=2
2) ENTRADA: X = True, W=2, Z=1, Y=1; R=0; SAÍDA: X=True, W+Z=3; Y= -2; R=-2
3) ENTRADA: R = 1; W=1; Z=2,,X=False, Y=1; SAÍDA: R = 0; W=1; Z=2,,X=False, Y=2
4) ENTRADA: R = 3; W=1; Z=1,,X=True, Y=0; SAÍDA: R = 3; W=1; Z=1,,X=True, Y=1

ESTUDO DE CASO SIMULADO



Crie o grafo de caminhos e os casos de teste unitários de caixa branca, com base na regra de Mc Cabe, para o algoritmo de programa que lhe foi disponibilizado para teste.

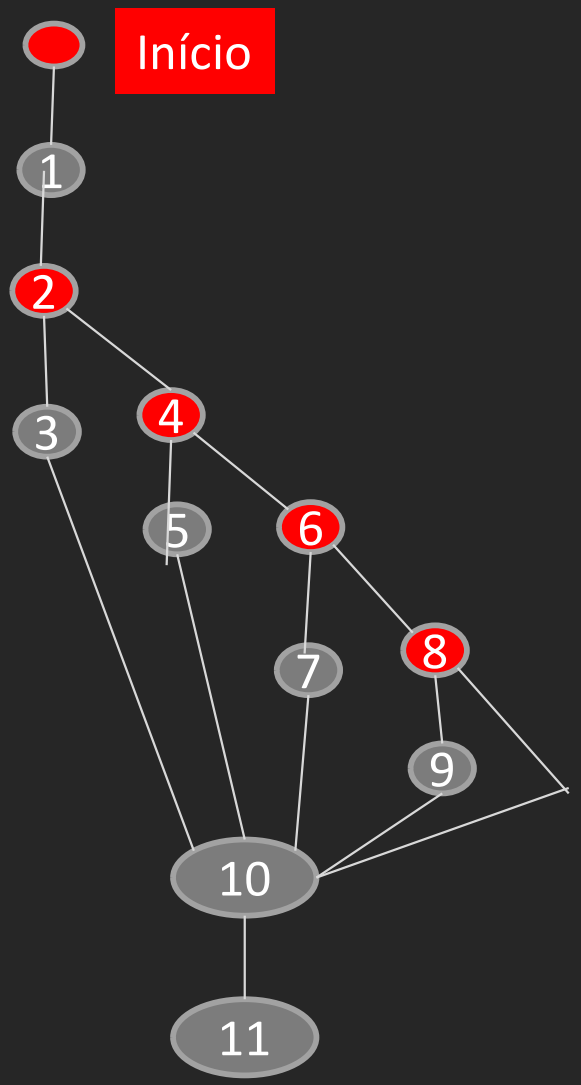
```
Caso X
=1 then X=X+1;
=2 then X=X+2;
=3 then X=X+3;
=4 then X=X+5;
Fim-Caso
X=X-1
```

COMO PLANEJAR UM CASO DE TESTE

COMPLEXIDADE CICLOMÁTICA – Aprendendo na prática

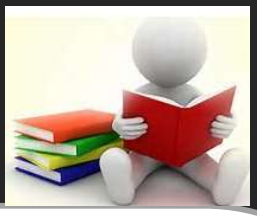
No caso de uma instrução CASE:

1. Caso X
2. =1
3. then X=X+1;
4. =2
5. then X=X+2;
6. =3
7. then X=X+3;
8. =4
9. then X=X+5;
10. Fim-Caso
11. X=X-1



5 testes

ESTUDO DE CASO SIMULADO



Defina os Casos de Teste com valores e entrada de dados e saída prevista para cada variável do programa que será desenvolvido com base no algoritmo a seguir.

```
REPETIR
   x= x+1

   REPETIR
      y=y+1
      z=z+100
   ATÉ  y > 20

      ENQUANTO  w < 50
         w=w+1
      FIM-ENQUANTO

ATÉ  x >= 100
```

ELABORE OS CASOS DE TESTES UNITÁRIOS DE CAIXA BRANCA E FUNCIONAIS, BASEANDO-SE NA AVALIAÇÃO DE ENLACES!

RELACIONE:
-Identificação/número do Caso de Teste;
-Descrição do objetivo
-Dados de entrada previstos
-Dados de saída previstos
-Preparação, se houver necessidade de manipular previamente dados de bancos de dados para que os testes funcionem

COMO PLANEJAR UM CASO DE TESTE

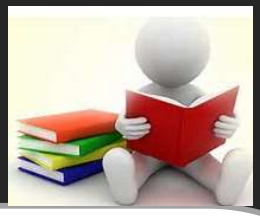
TESTE DE ENLACE – Aprendendo na prática

1º conjunto de testes a fazer: enlace mais interno (azul), sem fazer os loops mais externos.

```
LER (x, y, z, w);
REPETIR
    x= x+1

    REPETIR
        y=y+1
        z=z+100

        ENQUANTO  w < 50
            w=w+1
        FIM-ENQUANTO
    ATÉ  y > 20
ATÉ  x >= 100
```

TESTE1 – LOOP azul (não entra)	TESTE2– LOOP azul (entra e sai sem loop)	TESTE3– LOOP azul (entra e sai com loop)
.INPUT:	.INPUT:	.INPUT:
...X=100	...X=100	...X=100
...Y=20	...Y=20	...Y=20
...Z=0	...Z=0	...Z=0
...W=51	...W=49	...W=48
.OUTPUT:	.OUTPUT:	.OUTPUT:
...X=101	...X=101	...X=101
...Y=21	...Y=21	...Y=21
...Z=100	...Z=100	...Z=100
...W=51	...W=50	...W=50

ESTUDO DE CASO SIMULADO



Defina os Casos de Teste com valores e entrada de dados e saída prevista para cada variável do programa que será desenvolvido com base no algoritmo a seguir., considerando o método de avaliação de Limites

```
SE 12 <= a <= 19  ENTÃO
    a=a+1
SENÃO
   SE a<b ENTÃO
      a=a+b
   FIM-SE
FIM-SE
```

Compare o número de testes por esse critério em relação ao critério de complexidade ciclomátca

ELABORE OS CASOS DE TESTES UNITÁRIOS DE CAIXA BRANCA E FUNCIONAIS, BASEANDO-SE NA AVALIAÇÃO DE LIMITES!

RELACIONE:
-Identificação/número do Caso de Teste;
-Descrição do objetivo
-Dados de entrada previstos
-Dados de saída previstos
-Preparação, se houver necessidade de manipular previamente dados de bancos de dados para que os testes funcionem

COMO PLANEJAR UM CASO DE TESTE

TESTE DE LIMITES: Aprendendo na prática

```
SE 12 <= a <= 19  ENTÃO
    a=a+1
SENÃO
    SE a<b ENTÃO
        a=a+b
    FIM-SE
FIM-SE
```

Casos de testes para verificar se entra ou não no primeiro SE:

- **a = 11;**
- **a = 12**
- **a = 19**
- **a = 20**

Casos de testes para verificar se entra ou não no segundo SE:

- **a = 11 e b = 13**
- **a = 11 e b = 11**

Total de 6 casos de teste pela análise de limites

COMO PLANEJAR UM CASO DE TESTE

TESTE DE LIMITES: Aprendendo na prática

```
SE 12 <= a <= 19  ENTÃO
    a=a+1
SENÃO
    SE a<b ENTÃO
        a=a+b
    FIM-SE
FIM-SE
```

Casos de testes:

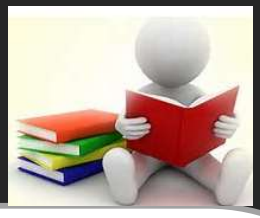
- **a = 11**
- **a = 12**
- **a = 19**
- **a = 20**

- **a = 11 e b = 13**
- **a = 11 e b = 11**

Total de 6 casos de teste pela análise de limites

Número de testes segundo a complexidade ciclomática = = número de pontos de decisão na subrede (aninhado) +1 =3

ESTUDO DE CASO SIMULADO



Defina os Casos de Teste com valores e entrada de dados e saída prevista para cada variável do programa que será desenvolvido com base no algoritmo a seguir., considerando o método de avaliação de Condição e Equivalência

```
SE Serie_Nota_Fiscal = “U” OU  “U1”
   ISS := “Isento”
SENÃO
  ISS := Tributado”
FIM-SE
```

Compare o número de testes por esse critério em relação ao critério de complexidade ciclomátca

ELABORE OS CASOS DE TESTES UNITÁRIOS DE CAIXA BRANCA E FUNCIONAIS, BASEANDO-SE NA AVALIAÇÃO DE LIMITES!

RELACIONE:
-Identificação/número do Caso de Teste;
-Descrição do objetivo
-Dados de entrada previstos
-Dados de saída previstos
-Preparação, se houver necessidade de manipular previamente dados de bancos de dados para que os testes funcionem

COMO PLANEJAR UM CASO DE TESTE

TESTE DE PARTIÇÃO/CONDIÇÃO DE EQUIVALÊNCIA – Aprendendo na prática

```
SE Serie_Nota_Fiscal = “U” OU  “U1”
    ISS := “Isento”
SENÃO
    ISS := Tributado”
FIM-SE
```

Testar todos valores do conjunto válido e apenas um do conjunto inválido:
Exemplo:

- Teste 1=> Serie_Nota_Fiscal = “U”
- Teste 2=> Serie_Nota_Fiscal = “W12”

Com isso percorreríamos todos os trechos da aplicação!