

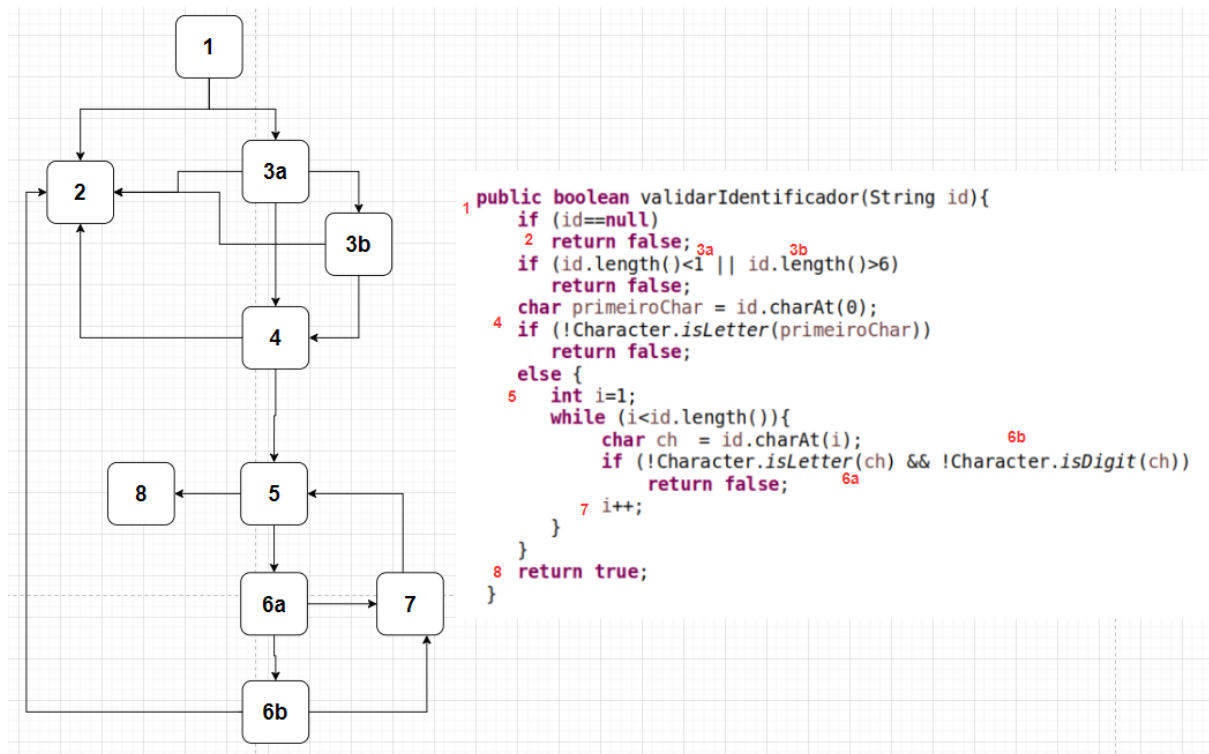
ESI - Prof. Marcelo Eler

Ilan Sousa - 11832920

Sungwon Yoon - 9822261

Questão 1

1 e 4.



2. CC = 8 + 1 = 9

3.

1-2

1-3a-2

1-3a-3b-2

1-3a-4-2

1-3a-3b-4-2

1-3a-4-5-6a-6b-2

1-3a-3b-4-5-6a-6b-2

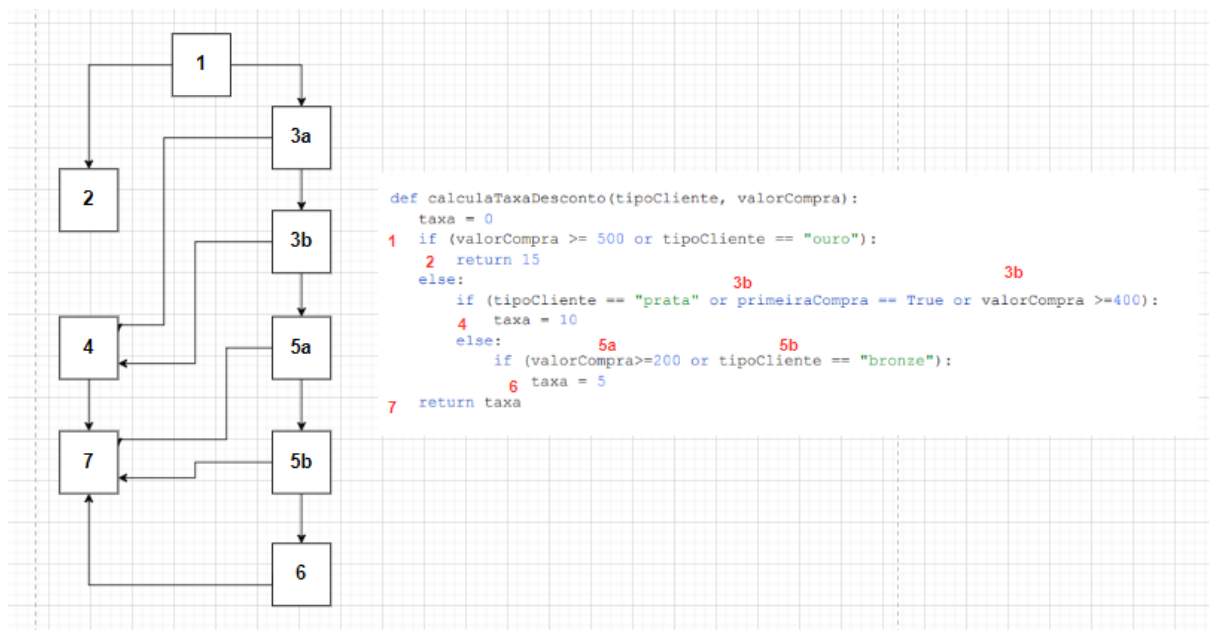
1-3a-4-5-6a-7-5-6a-6b-2

1-3a-3b-5-6a-7-5-6a-6b-2

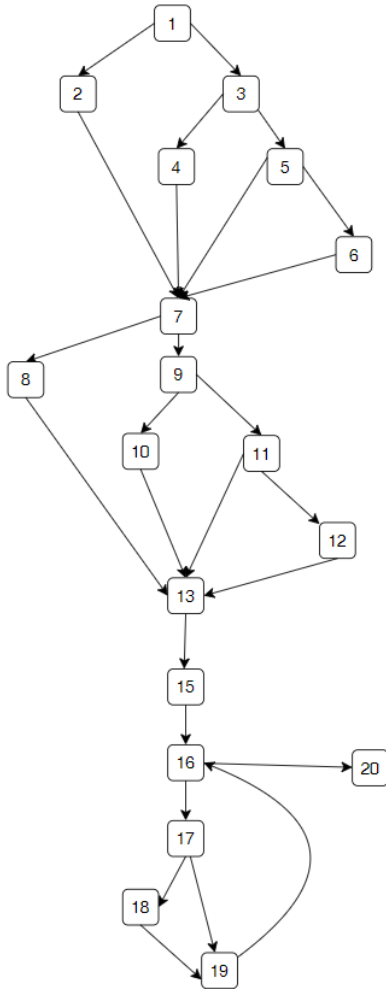
1-3a-4-5-6a-6b-7-5-6a-6b-2
 1-3a-3b-4-5-6a-6b-7-5-6a-6b-2
 1-3a-4-5-8
 1-3a-3b-4-5-8
 1-3a-4-5-6a-7-5-8
 1-3a-3b-4-5-6a-7-5-8
 1-3a-4-5-6a-6b-7-5-8
 1-3a-3b-4-5-6a-6b-7-5-8

5. Não existem novos casos de teste.

Questão 2



CC = 7+1 = 8



```

def calculaTaxaDesconto2(primeiraCompra, tipoCliente, valorCompra):
    taxas = []
    1 if (valorCompra>=500):
    2     taxas.append(15)
    else:
    3     if (valorCompra>=400):
    4         taxas.append(10)
    else:
    5         if (valorCompra>=200):
    6             taxas.append(5)
    7 if (tipoCliente=="ouro"):
    8     taxas.append(15)
    else:
    9     if (tipoCliente=="prata"):
    10         taxas.append(10)
    else:
    11         if (tipoCliente=="bronze"):
    12             taxas.append(5)
    13 if (primeiraCompra==True):
    14     taxas.append(10)

    maiorTaxa = 0
    indice = 0
    16 while (indice<len(taxas)):
    17     if (taxas[indice]>maiorTaxa):
    18         maiorTaxa = taxas[indice]
    19         indice = indice + 1
    20 return maiorTaxa
  
```

$$CC = 20 + 1 = 21$$

Obviamente, dada a quantidade de testes, o primeiro resultado é bem menos complexo. Mas para confirmarmos, fizemos o cálculo de complexidade ciclomática para atestarmos a complexidade superior da segunda implementação.