

Testes de Software

Professor: Aldo Moura

Aluna: Anna Ladewig

- Exercício de Fluxo de Controle e Teste de Cobertura –
- Exercício 07 -

```

public String apuracaoMedia() {
1 float media = 0;

2 if (this.percFrequencia < 75) {
3     this.situacao = "Reprovado por falta";
4 } else {
5     if (this.nota1 != 0 && this.nota2 != 0) {
6         // O estudante passou por duas avaliações
7         media = (this.nota1 + this.nota2) / 2;
8     } else if (this.nota1 == 0) {
9         // Estudante fez 2ª chamada para substituir a 1ª nota
10        media = (this.nota2 + this.notaSegChamada) / 2;
11    } else {
12        // Estudante fez 2ª chamada para substituir a 1ª nota
13        media = (this.nota1 + this.notaSegChamada) / 2;
14    }

15    this.mediaSemestre = media;

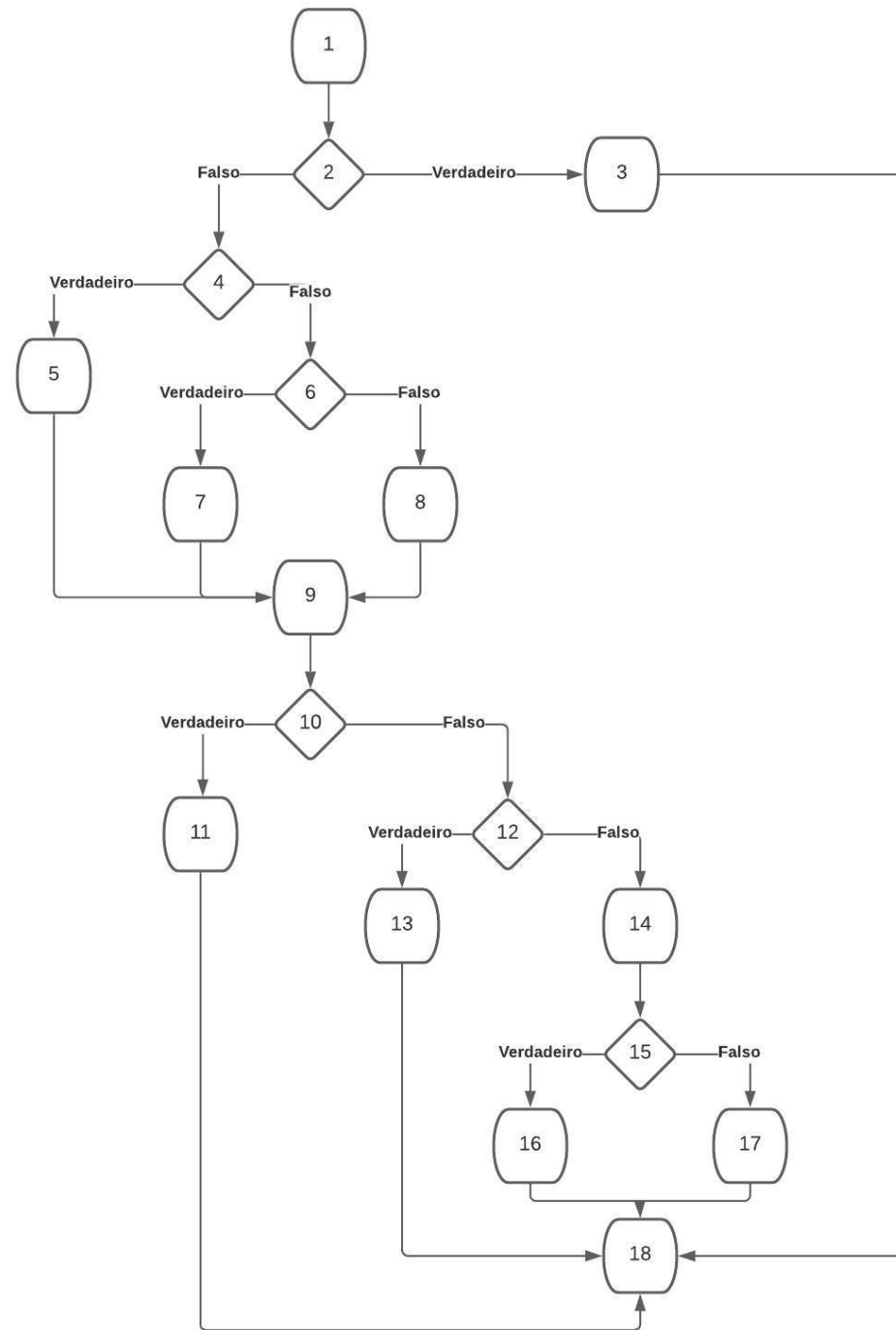
16    if (this.mediaSemestre >= 7) {
17        this.situacao = "Aprovado";
18    } else if (this.mediaSemestre < 3){
19        // Media menor que 3 não tem direito a final
20        this.situacao = "Reprovado";
21    } else {
22        this.mediaFinal = (this.mediaSemestre + this.notaFinal) / 2;
23        if (this.mediaFinal >= 5 ) {
24            // Na final a média passa a ser 5
25            this.situacao = "Aprovado na final";
26        } else {
27            this.situacao = "Reprovado na final";
28        }
29    }
30 }

31 return situacao;
32 }

```

Grafo de Fluxo de Controle

Método:
apuracaoMedia()



Cálculo da Complexidade Ciclomática

$$C = A(\text{num de arestas}) - N(\text{num de nós}) + 2$$

$$C = 23 - 18 + 2$$

$$C = 7$$

Logo, existem 7 caminhos a serem percorridos.

Porém, professor, eu achei um resultado diferente quando fui analisar os caminhos...

Mapeamento dos Caminhos

- 1 – 2 – 3 – 18
- 1 – 2 – 4 – 5 – 9 – 10 – 11 – 18
- 1 – 2 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 18
- 1 – 2 – 4 – 6 – 8 – 9 – 10 – 14 – 15 – 16 – 18
- 1 – 2 – 4 – 6 – 8 – 9 – 10 – 14 – 15 – 17 – 18

Total: 05

Caminhos necessários para ter a cobertura de 100%.

Valores:

- 1 – 2 – 3 – 18
Valores: (1, 10f, 10f, 0f, 0f, 74)
- 1 – 2 – 4 – 5 – 9 – 10 – 11 – 18
Valores: (1, 9f, 10f, 0f, 0f, 75)
- 1 – 2 – 4 – 6 – 7 – 9 – 10 – 12 – 13 – 18
Valores: (1, 0f, 2f, 3f, 0f, 75)
- 1 – 2 – 4 – 6 – 8 – 9 – 10 – 14 – 15 – 16 – 18
Valores: (1, 6f, 0f, 6f, 8f, 75)
- 1 – 2 – 4 – 6 – 8 – 9 – 10 – 14 – 15 – 17 – 18
Valores: (1, 5f, 0f, 5f, 4f, 75)

Finished after 0,042 seconds

Runs: 1/1 Errors: 0 Failures: 0

testeApuracaoNotasAlunos.ApuracaoTest [Runner: JUnit 4] (0,004 s)

Failure Trace

```
1 package testeApuracaoNotasAlunos;
2
3 import static org.junit.Assert.*;
4
5
6
7 public class ApuracaoTest {
8
9     @Test
10    public void testApuracaoMedia() {
11        Apuracao apuracao1 = new Apuracao(1, 10f, 10f, 0f, 0f, 74);
12        assertEquals("Reprovado por falta", apuracao1.apuracaoMedia());
13
14        Apuracao apuracao2 = new Apuracao(1, 9f, 10f, 0f, 0f, 75);
15        assertEquals("Aprovado", apuracao2.apuracaoMedia());
16
17        Apuracao apuracao3 = new Apuracao(1, 0f, 2f, 3f, 0f, 75);
18        assertEquals("Reprovado", apuracao3.apuracaoMedia());
19
20        Apuracao apuracao4 = new Apuracao(1, 6f, 0f, 6f, 8f, 75);
21        assertEquals("Aprovado na final", apuracao4.apuracaoMedia());
22
23        Apuracao apuracao5 = new Apuracao(1, 5f, 0f, 5f, 4f, 75);
24        assertEquals("Reprovado na final", apuracao5.apuracaoMedia());
25    }
26
27 }
28
```

Coverage x

ApuracaoTest (12/11/2020 18:17:11)

Element	Coverage	ed Instructions	ed Instructions	tal Instructions
▸ exercicio07	100,0 %	198	0	198