

FGA-0238 - Testes de Software

Turma	01	Semestre	2024.1
--------------	----	-----------------	--------

Equipe	NeverMind
---------------	-----------

Nome	Matrícula
Elias Faria de Oliveira	221007706
João Eduardo Pereira Rabelo	180053299
Yan Lucas Souza Guimarães	222006220
Daniel dos Santos Barros de Sousa	211030980

Método:

```
9 public boolean isTeamEligible(List<Employee> employees) {
10     boolean isEligible = false;
11     int managerCount = 0;
12     int seniorCount = 0;
13     int juniorCount = 0;
14     if (employees == null || employees.isEmpty()) {
15         return false;
16     }
17     for (Employee employee : employees) {
18         if (employee.getRole().equals("Manager")) {
19             managerCount++;
20         } else if (employee.getRole().equals("Senior")) {
21             seniorCount++;
22         } else if (employee.getRole().equals("Junior")) {
23             juniorCount++;
24         }
25     }
26     if ((managerCount >= 1 && seniorCount >= 2 && juniorCount >= 3) || (managerCount >= 2)) {
27         isEligible = true;
28     }
29     return isEligible;
30 }
```

Tabela 01: Decisões e Condições

ID	Decisão (Linha)	Condição	Situação para Verdadeiro	Situação para Falso
01	14	<code>employees == null</code>	Não existir a lista <code>employees</code>	Existir a lista <code>employees</code>
02	14	<code>employees.isEmpty()</code>	A lista <code>employees</code> está vazia.	A lista <code>employees</code> ter no mínimo um funcionário dentro dela.
03	18	<code>employee.getRole().equals("Manager")</code>	Na lista de <code>employees</code> , ter um funcionário cujo cargo seja "Manager".	Na lista de <code>employees</code> , não existir um funcionário cujo cargo seja "Manager".
04	20	<code>employee.getRole().equals("Senior")</code>	Na lista de <code>employees</code> , ter um funcionário cujo cargo seja "Senior".	Na lista de <code>employees</code> , não existir um funcionário cujo cargo seja "Senior".
05	22	<code>employee.getRole().equals("Junior")</code>	Na lista de <code>employees</code> , ter um funcionário cujo cargo seja "Junior".	Na lista de <code>employees</code> , não existir um funcionário cujo cargo seja "Junior".
06	26	<code>managerCount >= 1</code>	A quantidade de Manager for maior ou igual a 1	A quantidade de Manager for 0
07	26	<code>seniorCount >= 2</code>	A quantidade de Senior for maior ou igual a 2	A quantidade de Senior for menor ou igual a 1
08	26	<code>juniorCount >= 3</code>	A quantidade de Junior for maior ou igual a 3	A quantidade de Junior for menor ou igual a 2
09	26	<code>managerCount >= 2</code>	A quantidade de Manager for maior ou igual a 2	A quantidade de Manager for menor ou igual a 1

Fonte: Grupo NeverMind

Tabela 02: Tabela Verdade (MC/DC) - Linha 14

Teste	employees == null	employees.isEmpty()	Resultado
01	Verdadeiro	Falso	Verdadeiro
02	Falso	Verdadeiro	Verdadeiro
03	Falso	Falso	Falso

Fonte: Grupo NeverMind

- Pares de Independência
 - Para employees == null -> 1 e 3
 - Para employees.isEmpty() -> 2 e 3

Tabela 03: Casos de Teste - Linha 14

Casos de Teste	Entrada	Saída Esperada
1	employees == null && employees != Empty()	Retorna "False" imediatamente.
2	employees != null && employees == Empty()	Retorna "False" imediatamente.
3	employees != null && employees != Empty()	Da continuidade na função e retorna se o time é elegível.

Fonte: Grupo NeverMind

Tabela 04: Tabela Verdade - Linha 26

Teste	managerCount >= 1	seniorCount >= 2	juniorCount >= 3	managerCount >= 2	Resultado
00	Falso	Falso	Falso	Falso	Falso
01	Falso	Falso	Falso	Verdadeiro	Verdadeiro
02	Falso	Falso	Verdadeiro	Falso	Falso
03	Falso	Falso	Verdadeiro	Verdadeiro	Verdadeiro
04	Falso	Verdadeiro	Falso	Falso	Falso
05	Falso	Verdadeiro	Falso	Verdadeiro	Verdadeiro
06	Falso	Verdadeiro	Verdadeiro	Falso	Falso
07	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro
08	Verdadeiro	Falso	Falso	Falso	Falso
09	Verdadeiro	Falso	Falso	Verdadeiro	Verdadeiro
10	Verdadeiro	Falso	Verdadeiro	Falso	Falso
11	Verdadeiro	Falso	Verdadeiro	Verdadeiro	Verdadeiro
12	Verdadeiro	Verdadeiro	Falso	Falso	Falso
13	Verdadeiro	Verdadeiro	Falso	Verdadeiro	Verdadeiro
14	Verdadeiro	Verdadeiro	Verdadeiro	Falso	Verdadeiro
15	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro

Fonte: Grupo NeverMind

Tabela 05: Tabela Verdade (MC/DC) - Linha 26

Teste	managerCount >= 1	seniorCount >= 2	juniorCount >= 3	managerCount >= 2	Resultado
14	verdade	verdade	verdade	Falso	Verdadeiro
06	Falso	Verdadeiro	Verdadeiro	Falso	Falso

10	Verdadeiro	Falso	Verdadeiro	Falso	Falso
12	Verdadeiro	Verdadeiro	Falso	Falso	Falso
07	Falso	Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro

Fonte: Grupo NeverMind

- Pares de Independência
 - Para `managerCount >= 1` -> 14 e 06
 - Para `seniorCount >= 2` -> 14 e 10
 - Para `juniorCount >= 3` -> 14 e 12
 - Para `managerCount >= 2` -> 6 e 7

Tabela 06: Casos de Teste - Linha 26

Casos de Teste	Entrada	Saída Esperada
1	<code>managerCount >= 1 && seniorCount >= 2 && juniorCount >= 3 && managerCount < 2</code>	O time, com estes parâmetros, é elegível, retornando “verdadeiro” ao final da função.
2	<code>managerCount == 0 && seniorCount >= 2 && juniorCount >= 3 && managerCount < 2</code>	O time, com estes parâmetros, não é elegível, retornando “falso” ao final da função.
3	<code>managerCount >= 1 && seniorCount < 2 && juniorCount >= 3 && managerCount < 2</code>	O time, com estes parâmetros, não é elegível, retornando “falso” ao final da função.
4	<code>managerCount >= 1 && seniorCount >= 2 && juniorCount < 3 && managerCount < 2</code>	O time, com estes parâmetros, não é elegível, retornando “falso” ao final da função.
5	<code>managerCount == 0 && seniorCount >= 2 && juniorCount >= 3 && managerCount >= 2</code>	O time, com estes parâmetros, é elegível, retornando “verdadeiro” ao final da função.

Fonte: Grupo NeverMind