Atividade de Projeto 8 - Casos de Teste Estrutural

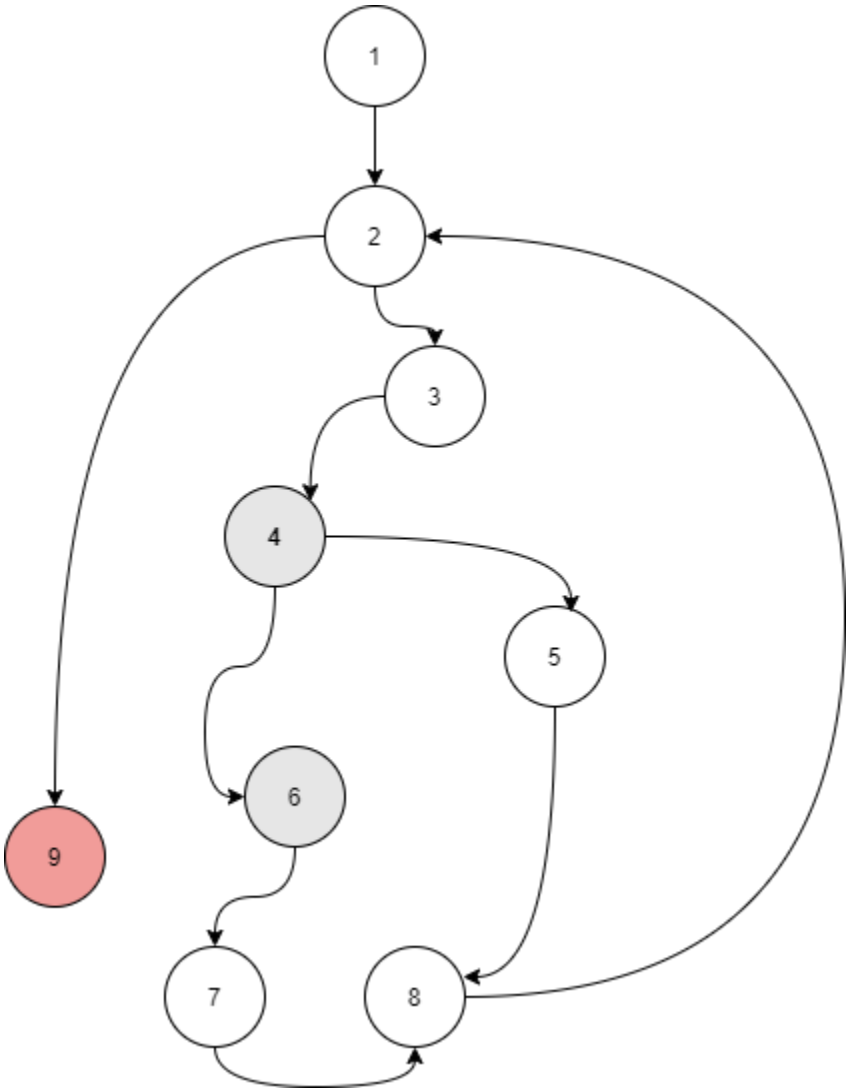
Grupo 9 - Integrantes
Felipe Veloso - 10349529
Rubens Galdino- 10786142
Marco Antonio - 10734175
João Victor Alves - 10734237
Victor Hugo de Lima Grecca - 10392185

Parte a) Elabore o grafo de fluxo para o programa. Numere o código de acordo com os nós do grafo.

```
/*1*/ int buscabinaria (int x, int n, int v[])
/*1*/ {
/*1*/ int esquerda, meio, direita;
/*1*/ esquerda = 0;
/*1*/ direita = n -1;

/*2*/ while (esquerda <= direita)
/*3*/ {
/*3*/     meio = (esquerda + direita) / 2;
/*4*/     if (x < v[meio])
/*5*/             direita = meio -1;
/*6*/     else if (x > v[meio])
/*7*/             esquerda = meio + 1;
/*7*/     else
/*8*/             return meio;
/*8*/ }

/*9*/ return -1;
/*9*/ }
```


1
2
3
4
5
6
7
8
9

Parte b) Elabore um conjunto de casos de teste para garantir que todos os nós foram exercitados. O conjunto deve ter como elementos tuplas compostas das entradas e saídas esperadas. Indique que nós foram cobertos por cada elemento do conjunto.

Nó #2
Fluxo Básico:
Checa se o valor de *esquerda* é menor ou igual ao valor *direita**.*** Caso seja, entra dentro do laço (while) e segue para o próximo nós.
Fluxo Alternativo:
Caso *direita* seja maior que *esquerda*, não entra no laço (while) e vai diretamente para a etapa *9,* encerrando a execução do programa.

Nó #4
Fluxo Básico:
Checa se X é menor que o valor obtido e inserido no vetor ***meio.*** Caso seja menor, entra dentro do laço e *direita* recebe *meio-1.*
Fluxo Alternativo:
Caso X seja maior que o valor obtido, segue para o próximo nó, #6.

Nó #6
Fluxo Básico:
Checa se X é maior que valor obtido e inserido no vetor ***meio.*** Caso seja maior, entra dentro do laço e *esquerda* recebe meio + 1.
Fluxo Alternativo:
Caso X seja menor que o valor obtido em meio, segue para o passo **#7**, retornando o valor inicial de *meio*.

Parte c) Comente: o que acharam dos casos de uso elaborados? Será que são mais efetivos para encontrar erros do que os casos de teste feitos pela técnica funcional, com critérios de classes de equivalência e análise do valor limite?

Os casos de uso elaborados demonstraram de maneira relativamente eficiente o poderio para realizar testagem no código. Entretanto, como ressaltado pela docente na vídeo aula gravada e disponibilizada anteriormente, pode ser um tanto quanto enfadonho realizar este processo de testes.
Por fim, concluímos que, ao utilizar-se dos casos de uso elaborados, podemos desencadear uma testagem mais clara, eficiente e objetiva.

Os critérios de teste em geral possuem uma abordagem sistemática e teoricamente fundamentada para conduzir uma atividade de teste. Aumentam a garantia de que os casos de testes irão revelar defeitos. Cada técnica depende da origem dos dados para criar os casos de testes. A técnica estrutural tem como base a implementação.

Entretanto, não existe um procedimento de teste de propósito geral para provar a correção de um programa. É indecidível se dois ou mais programas computam a mesma função; se dois caminhos de um mesmo programa, ou de programas diferentes, computam a mesma função; e se um dado caminho é executável e se existe um conjunto de dados de entrada que leve à execução desse caminho.