



# Executar Teste e Implantação de Aplicativos Computacionais

SENAC PE

08 de Outubro de 2024

# Teste Estrutural

- Contexto Geral



# Imagine o seguinte cenário

- Programa recebe uma idade, entre 1 e 100 e deve responder se o indivíduo é “maior” ou menor de idade.
- Baseado nessa especificação apenas, podemos elaborar alguns casos de teste:
- (1,menor), (17, menor), (18, maior), (19, maior), (100, maior)

# Imagine o seguinte cenário

```
def maioridade (idade):  
    if idade == 1:  
        print("Menor")  
    if idade == 2:  
        print("Menor")  
    .  
    .  
    .  
    if idade == 99:  
        print("Maior")  
    if idade == 100:  
        print("Maior")
```

# Definição

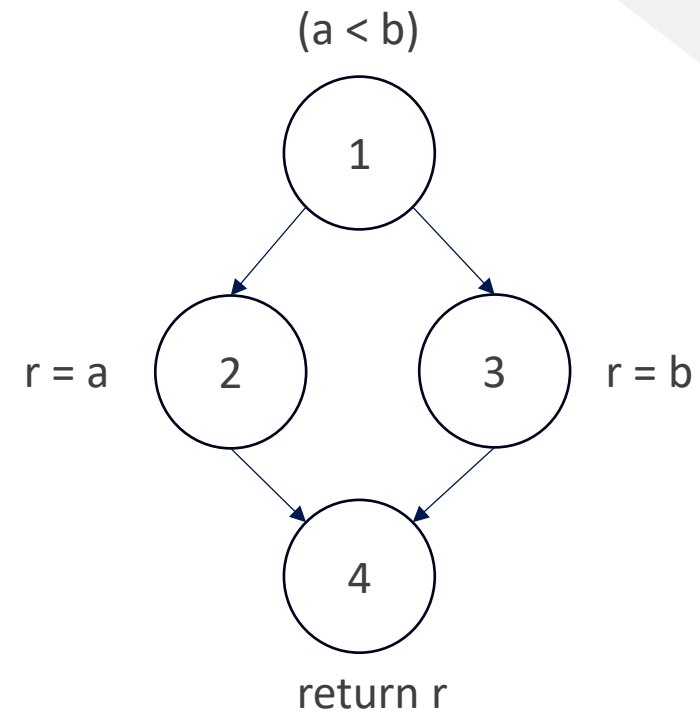
- O teste estrutural usa o código do programa para definir os requisitos de teste e deles derivar os casos de teste. O objetivo é observar as “estruturas” que compõem o programa e garantir que todas elas tenham sido executadas durante o teste.

# Abstração: Grafo de Fluxo de Controle

- Na verdade, para que a gente reconheça a estrutura do programa que vamos testar, vez de pensarmos nos seus comandos diretamente, é mais fácil fazermos uma abstração da estrutura do programa, e essa abstração é feita por meio do que chamamos de grafo de fluxo de controle, ou grafo de programa.
- Esse grafo é criado considerando os blocos de código que são sempre executados juntos como vértices do grafo e a transferência da execução de bloco a outro são as arestas do grafo.

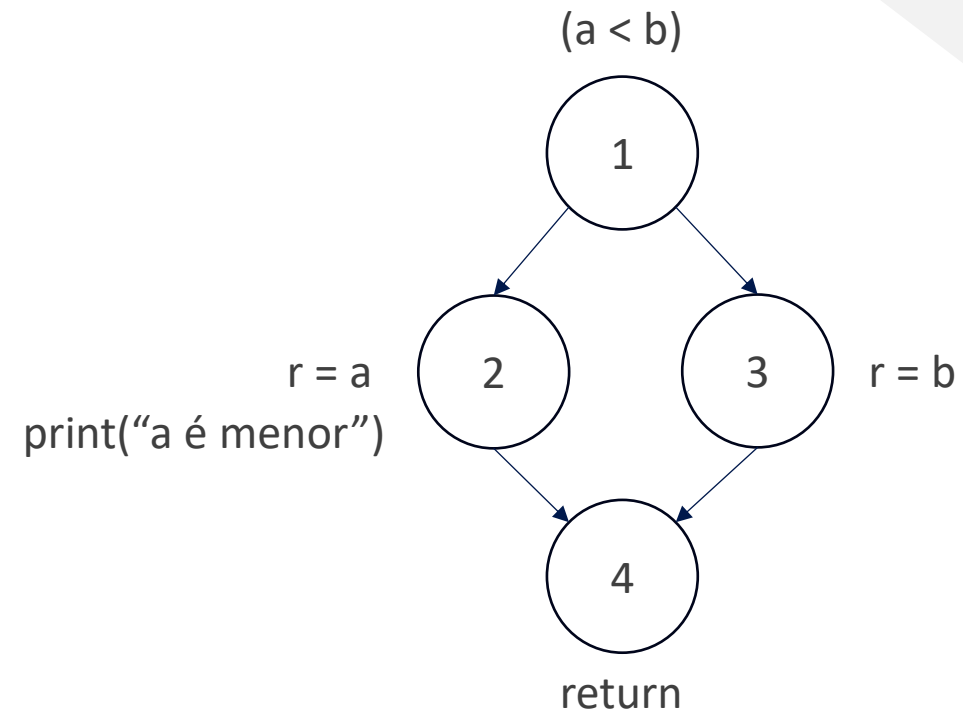
# GFC: Exemplo

```
def min(a, b):
    if a < b:
        return a
    else:
        return b
```



# GFC: Exemplo

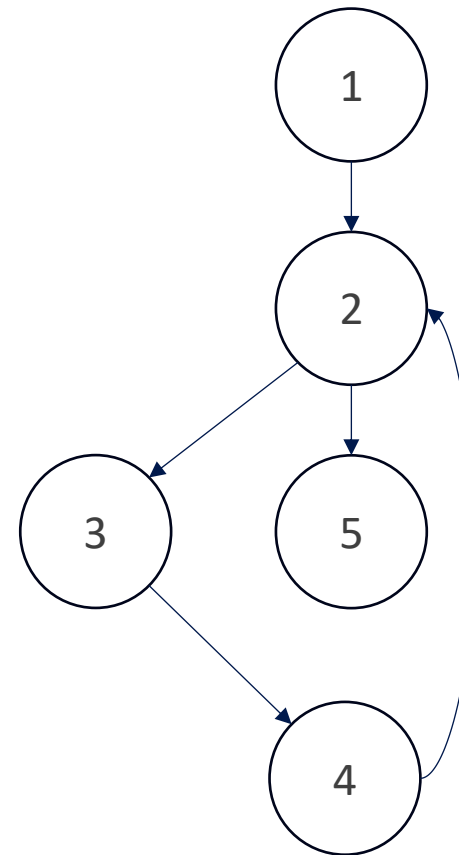
```
def min(a, b):
    r = 0;
    if a < b:
        print("a é menor")
        return a
    else:
        return b
```



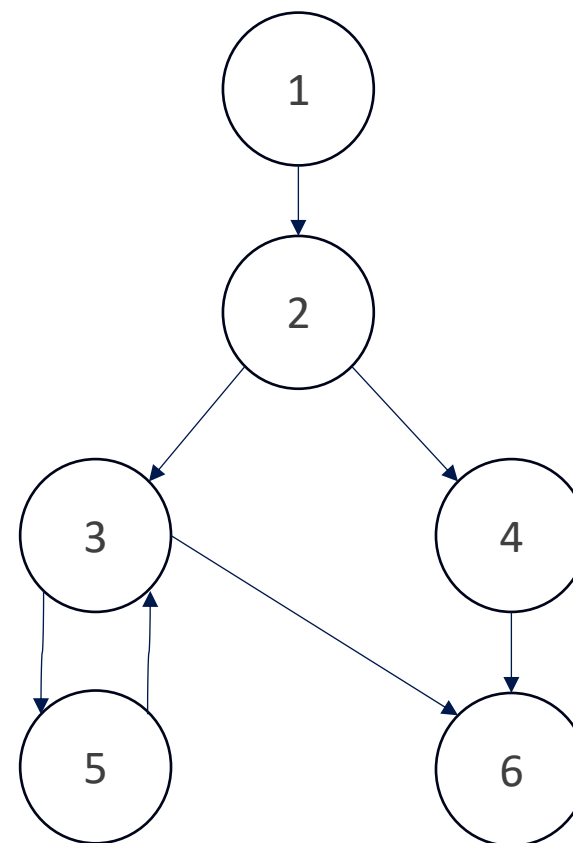
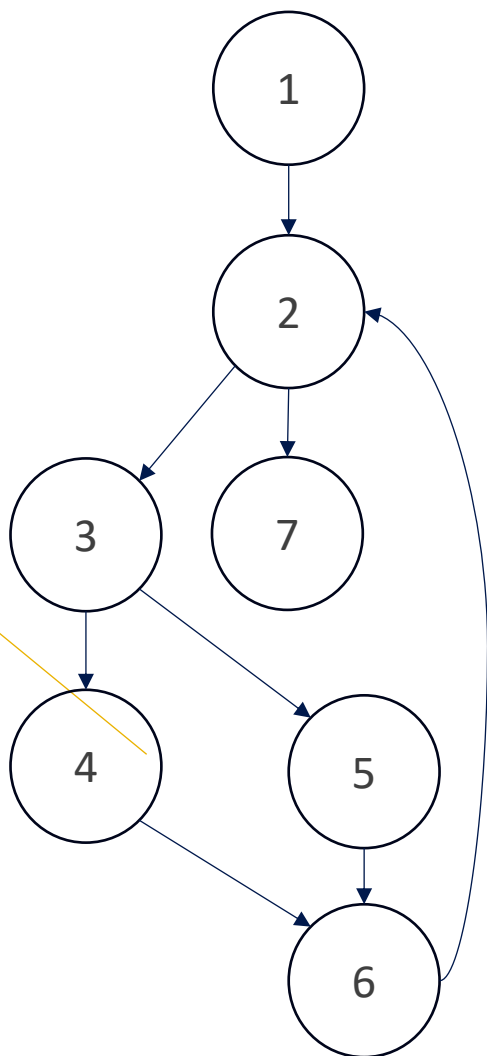


# GFC: Laço

```
def mult(a, b):  
    r = 0  
    while a > 0:  
        r += b  
        a -= 1  
    return r
```

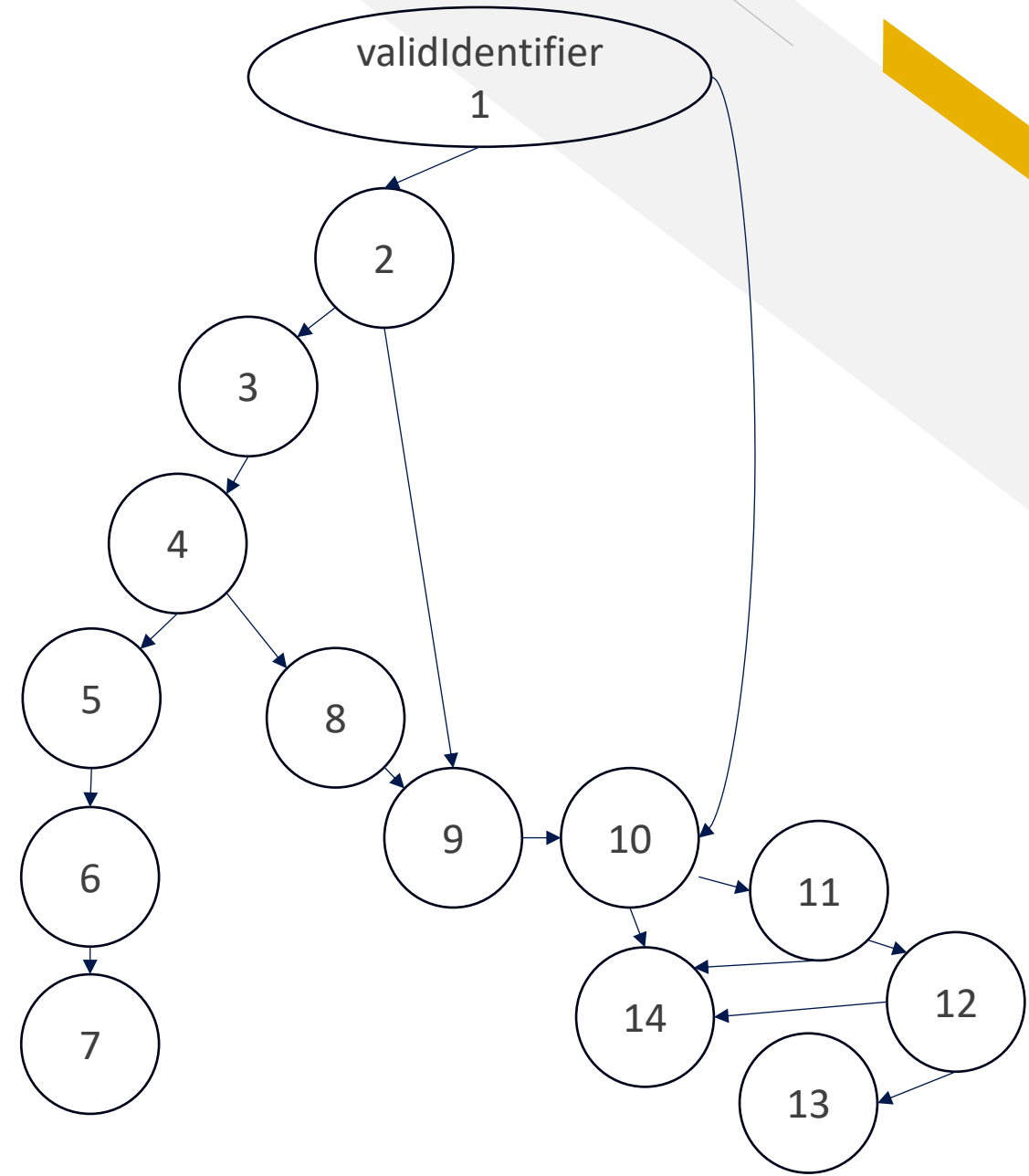


# GFC: Estruturas Aninhadas



# GFC: Identifier

```
int validateIdentifier (char* s) {
    char achar;
    int i, valid_id = FALSE;
    if (strlen(s) > 0) {
        achar = s[0];
        valid_id = valid_s(achar);
        if (strlen(s) > 1) {
            achar = s[1];
            i = 1;
            while (i < strlen(s)) {
                achar = s[i];
                if (!valid_f(achar))
                    valid_id = FALSE;
                i++;
            }
        }
    }
    if (valid_id && (strlen(s) >= 1) && (strlen(s) < 6))
        return TRUE;
    else
        return FALSE;
}
```



# Resumo

- O teste estrutural, ao contrário do funcional, tem como objetivo exercitar os comandos do nosso programa
- Vimos que para abstrair a estrutura dos comandos usamos o Grafo de Fluxo de Controle
- Como usar o GFC para derivar casos de teste para os programas?

# Critérios

- Fluxo de Controle e de Dados

# Fluxo de Controle

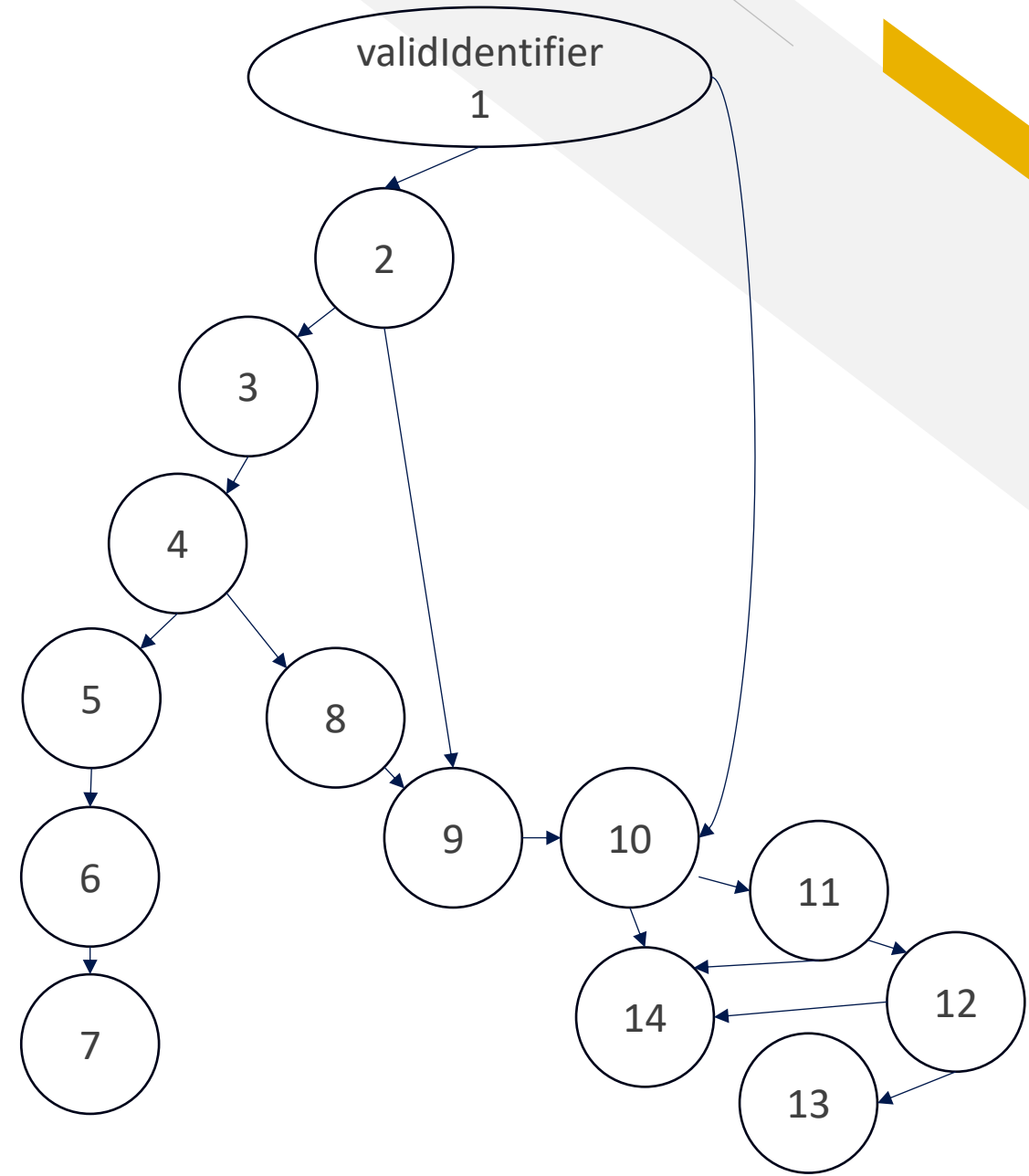
- Fluxo de controle significa que teremos que achar casos de teste que exercitem os elementos do GFC
- Todos-nós
- Todos-arcos

# Todos NÓS

- Todos-nós significa que devemos achar casos de teste que façam com que todos-nós do grafo do fluxo de controle sejam executados pelo menos uma vez. Quando conseguirmos isso, temos conjunto de teste adequado.

# Todos NÓS

- "" (string vazio)
- a
- abc
- a\*c



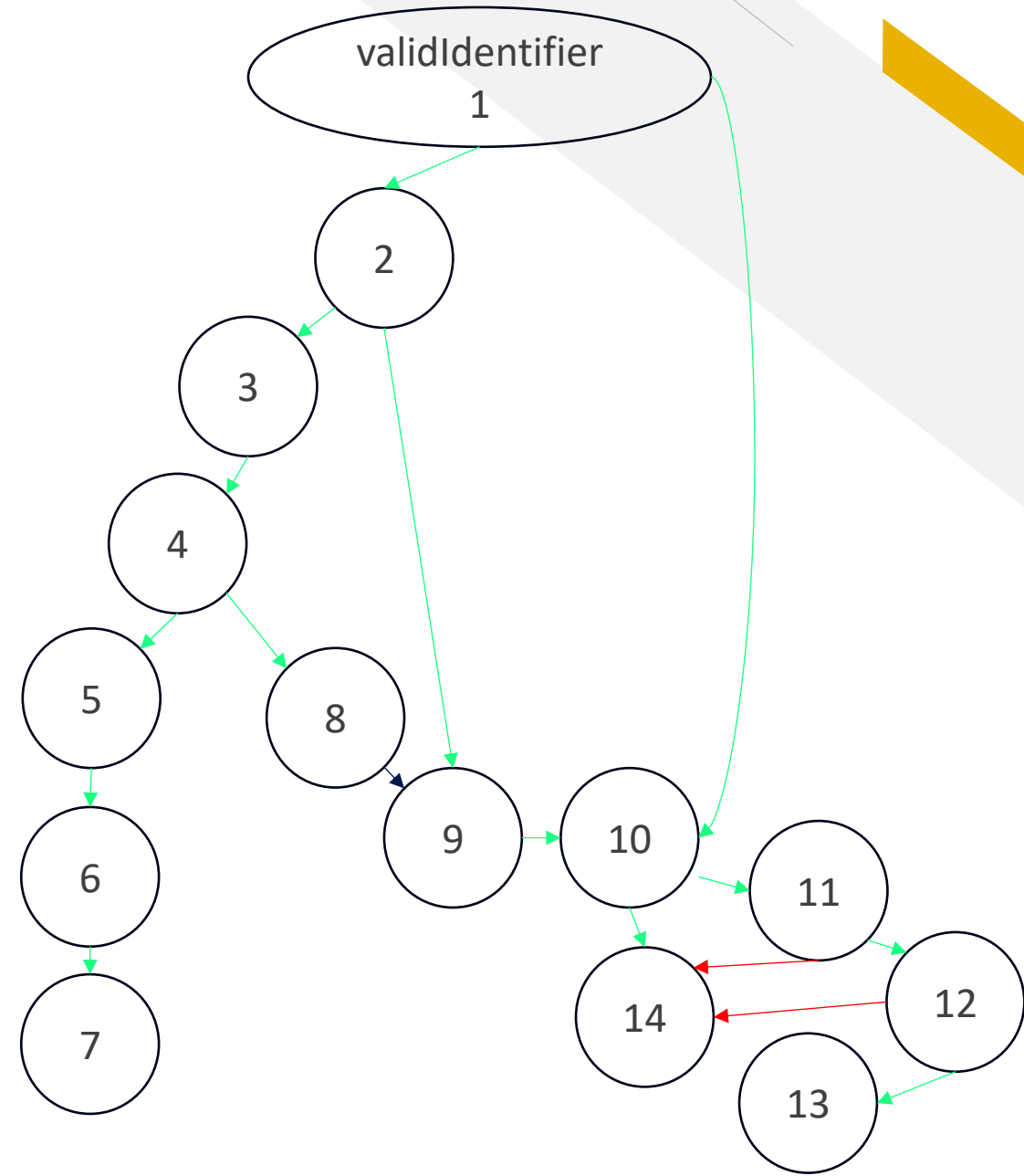


# Todos NÓS

- No critério Todos-arcos, estamos interessados em cobrir, executar, as arestas do grafo, ou seja, executar todos os desvios que representam o fluxo de execução do nosso programa.
- Por isso, também, chamamos esse tipo de teste de cobertura de ramos, ou cobertura de desvios.

# Todos NÓS

- "" (string vazio)
- a
- abc
- a\*c
- abcdefghi (mais de 6 caracteres)
  - nó 12 - 14
- nó 11 - 14 não é possível com os requisitos fornecidos
  - Requisito não executável (não há dado de teste que o faça ser executado)

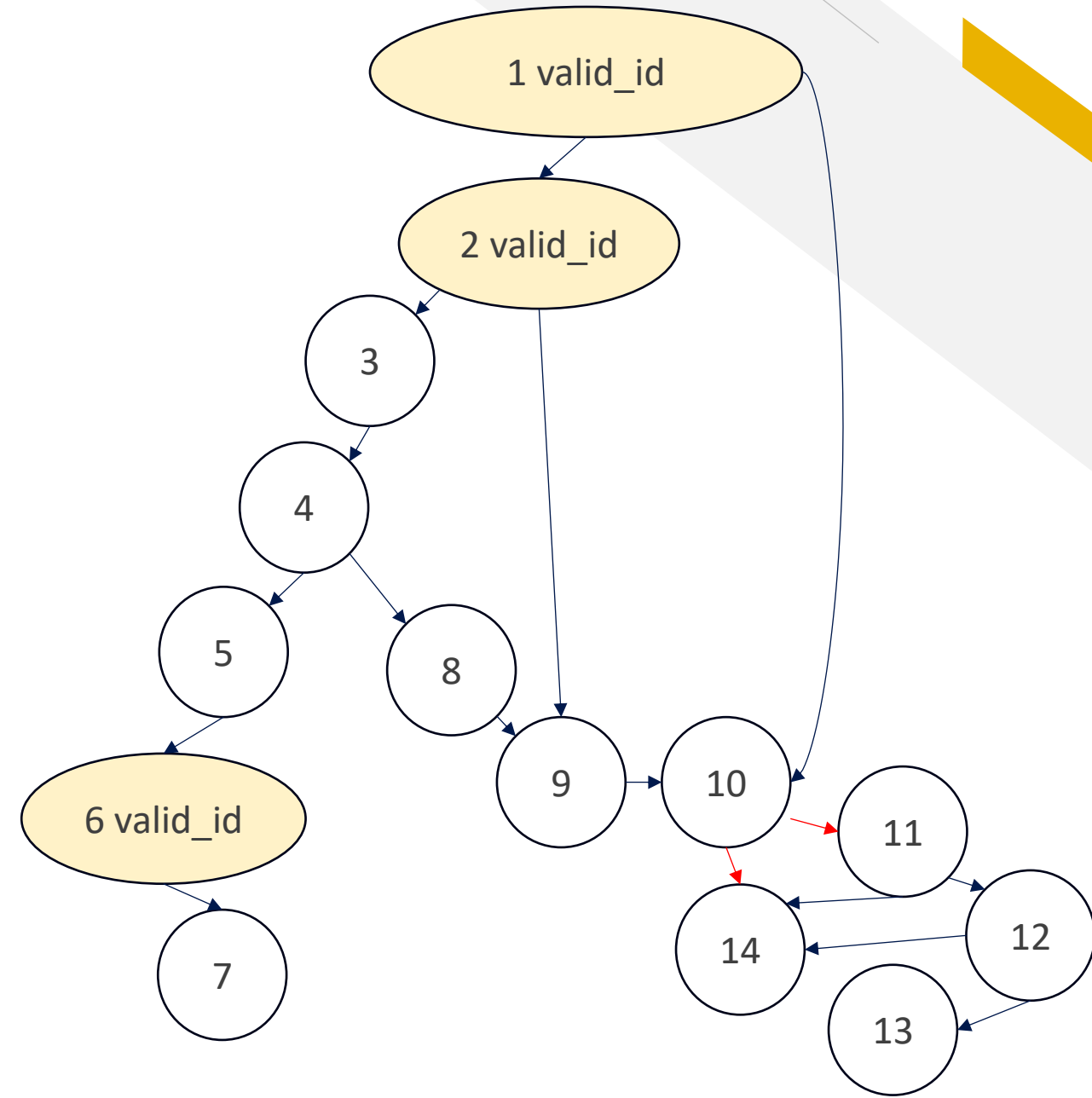


# Fluxo de Dados - Conceito

- Definição: Ponto em que uma variável recebe um valor
- Uso: Ponto em que um valor de variável é usado em um cálculo ou numa condição

# Fluxo de Dados - Conceito

<valid\_id,1, (10,11)>  
 <valid\_id,1, (10,14)>  
 <valid\_id,2, (10,11)>  
 <valid\_id,2, (10,14)>  
 <valid\_id,6, (10,11)>  
 <valid\_id,6, (10,14)>



# Fluxo de Dados - Comentários

- Casos de teste de melhor qualidade
- Conceitos mais complexos
- Custo mais alto
- Muitos requisitos não executáveis

# Exercício

- GFC

# Exercício – Considere o programa:

1. início
2. ler (N)
3. ler (M)
4. se  $N < M$
5. então
6.     se N for número par
7.         então  $NRO \leftarrow N+1$
8.         senão  $NRO \leftarrow N$
9.      $SOMA \leftarrow 0$
10.    enquanto ( $NRO \leq M$ )
11.      se  $NRO > 0$
12.         então  $SOMA \leftarrow SOMA + NRO$
13.          $NRO \leftarrow NRO + 2$
14.    fim-enquanto
15.    escrever (N, M, SOMA)
16.    senão
17.      escrever (INTERVALO INCORRETO)
18. fim-programa

# Exercício

- a) Elabore o GFC.
- b) Identifique todos os requisitos de teste considerando-se o Critério Todos-Nós.
- c) Projete o conjunto de casos de teste para os requisitos do item b.
- d) Identifique todos os requisitos de teste considerando-se o Critério Todos-Arcos.
- e) Projete o conjunto de casos de teste para os requisitos do item d.





# Senac Pernambuco Educação Profissional Recife

Thiago Dias Nogueira

Instrutor Técnico

(81) 9 9627-0419

[thiago.nogueira@pe.senac.br](mailto:thiago.nogueira@pe.senac.br)