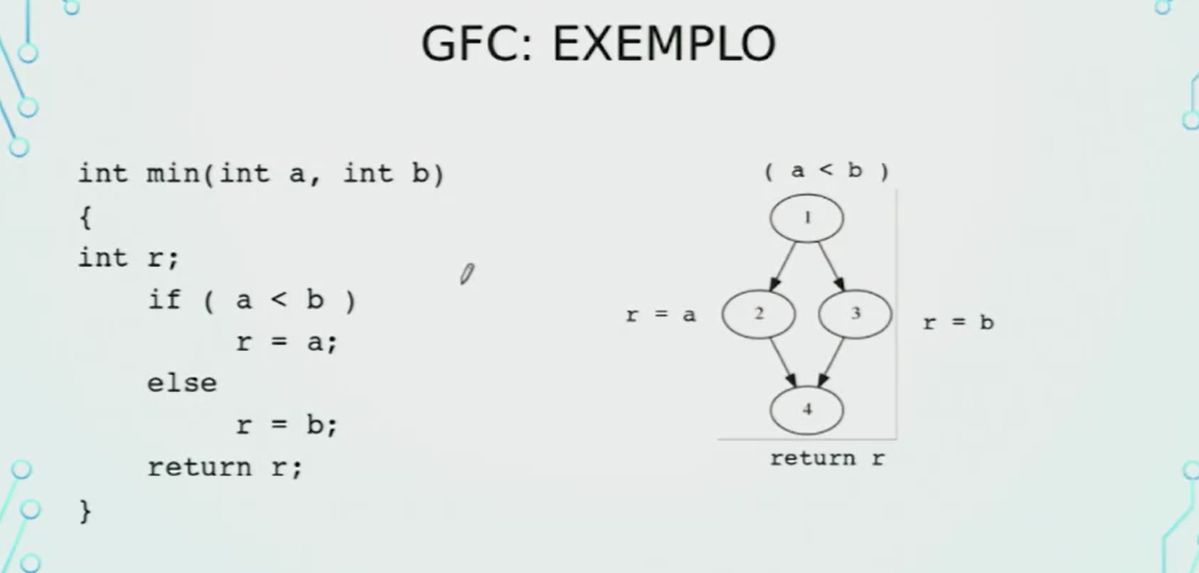
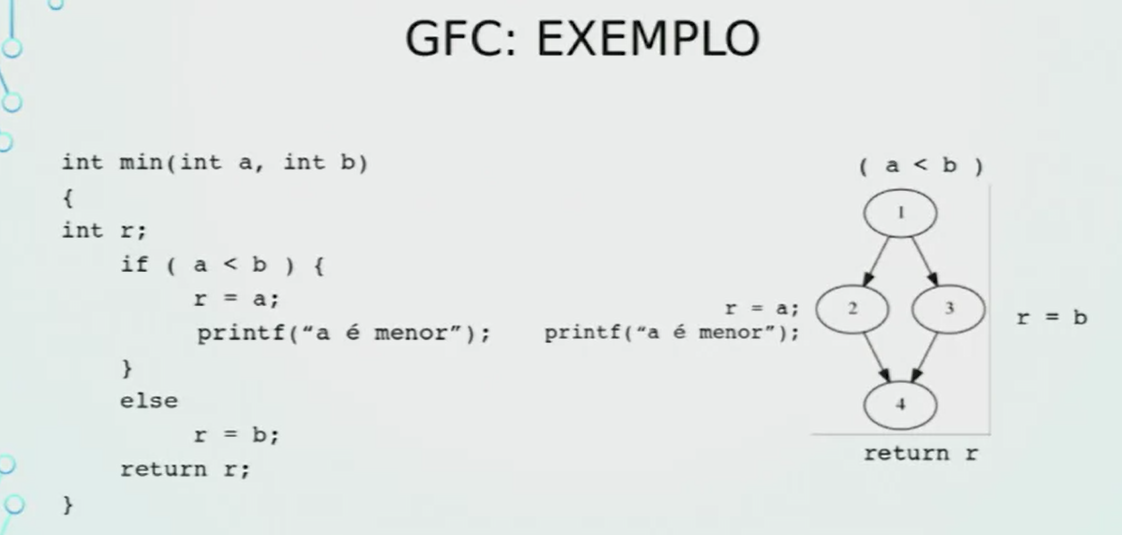
**QA – Teste Estrutural**

**Teste Estrutural** – Utilizado para **projetar testes baseando-se na estrutura interna (código) e na implementação de um programa. Tem por objetivo observar as estruturas que compõem o programa e garantir que todas essas estruturas sejam executadas, ao menos uma vez, durante os testes.**

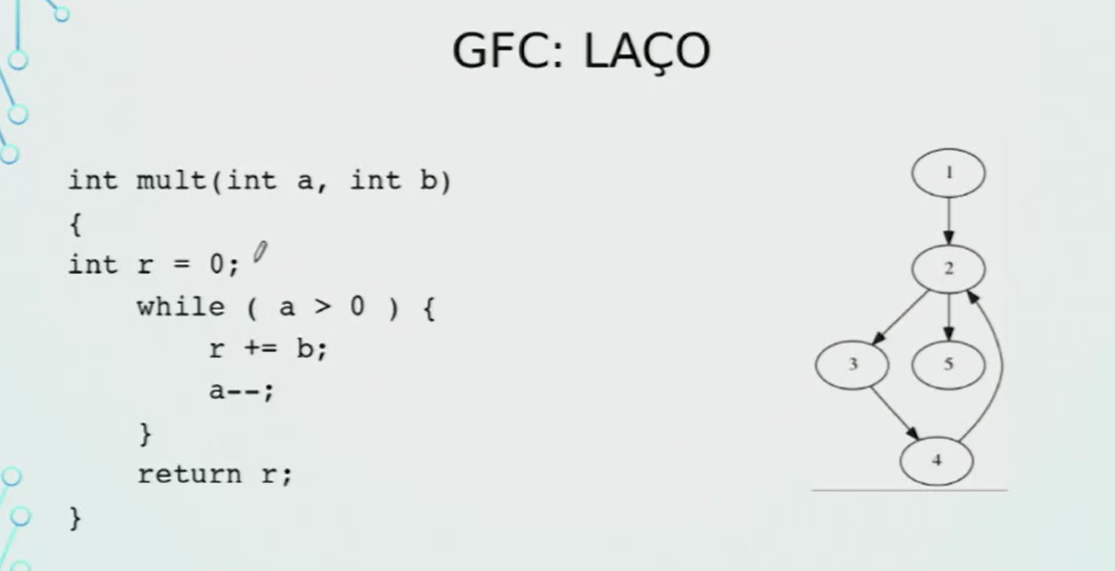
**Grafo de Fluxo de Controle (ou Grafo de Programa) – GFC**

**Exemplos:**

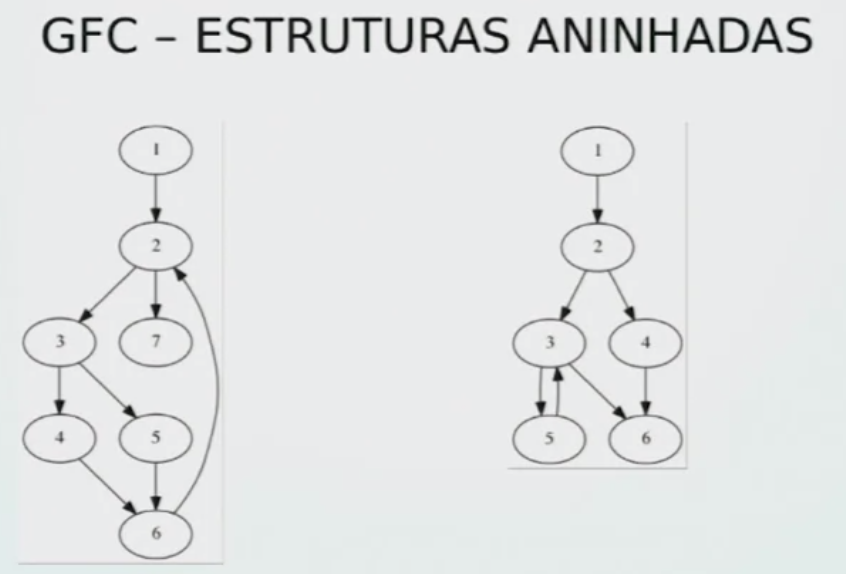
****

****

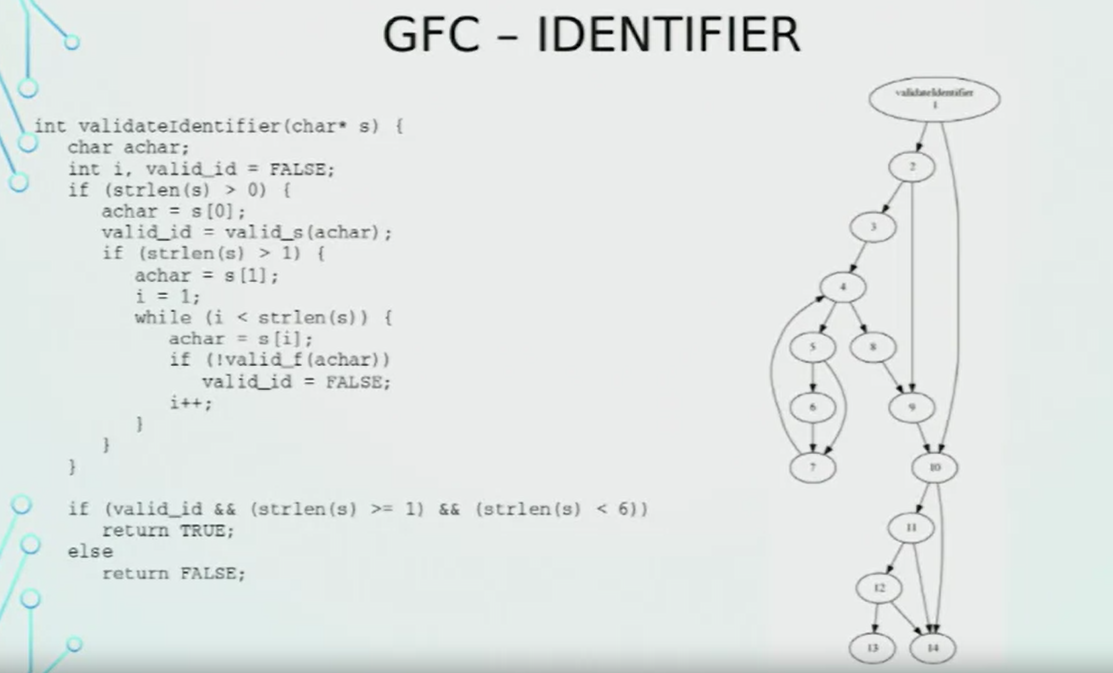
**GFC Laço:**

****

**GFC Estruturas Aninhadas:**

****

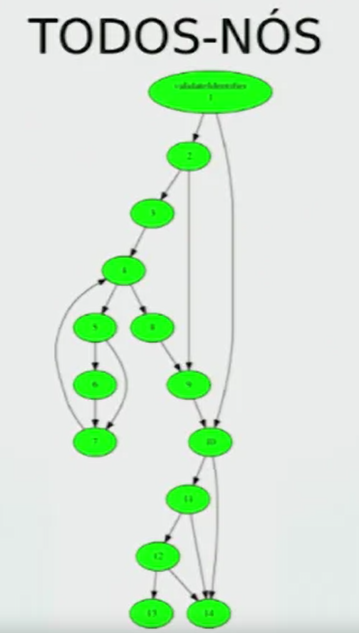
**GCF do Programa Identifier:**

****

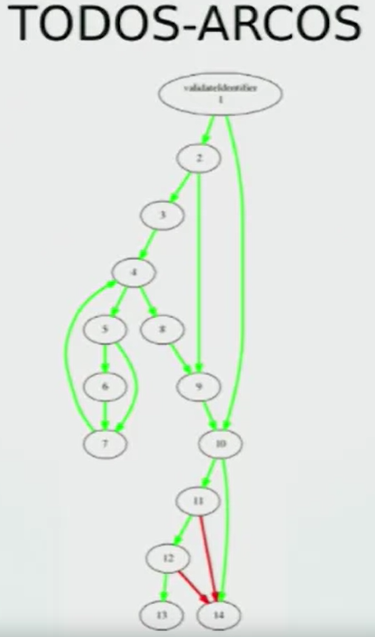
**Testes Baseados em Fluxo de Controles:** Encontrar casos de testes que exercitem os elementos do GFC.

**Critérios mais utilizados:**

**- Todos-Nós (ou Teste de Cobertura de Comandos)** – Casos de testes que executem todos os nós do fluxo de controle, pelo menos, uma vez cada:



**- Todos-Arcos (ou Teste de Cobertura de Ramos ou de Cobertura de Desvios)** – Executar todos os desvios que representam o fluxo de execução do programa:

****

**Código do identifier.c**

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;

}

**Testes Baseados em Fluxo de Dados:**