TESTE E QUALIDADE DE SOFTWARE

João Choma Neto

joao.choma@gmail.com

Unicesumar – Maringá – 2025/2

TESTE ESTRUTURAL

A CONTINUAÇÃO

CRITÉRIOS

- Cobertura de Linhas (Line Coverage)
- Cobertura de código
- OBJETIVO: garantir que cada linha de código seja executada pelo menos uma vez durante a execução dos casos de teste
- Isso ajuda a identificar partes do código que não foram testadas.

CRITÉRIOS

- Cobertura de Ramificações (Branch Coverage)
- OBJETIVO: garantir que todas as ramificações ou caminhos de decisão no código sejam exercidas
- Isso inclui a verificação de todas as instruções condicionais, como declarações "if" e "else".

CRITÉRIOS

- Cobertura de Condições (Condition Coverage)
- OBJETIVO: visa verificar cada condição dentro de instruções condicionais separadamente.
- Cada condição deve ser avaliada tanto como verdadeira quanto falsa.

CRITÉRIOS

- Cobertura de Caminhos (Path Coverage)
- OBJETIVO: Este é um critério mais abrangente que busca testar todos os caminhos possíveis através do código.
- Isso inclui todas as combinações de caminhos de decisão e loops.

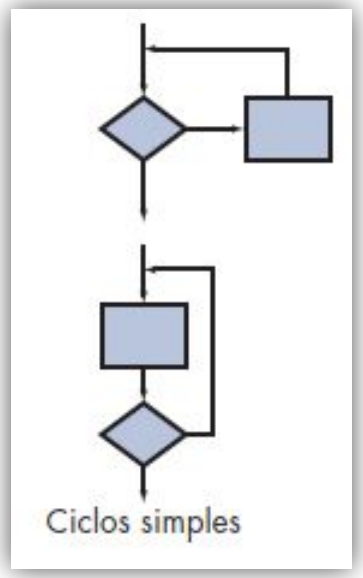
CRITÉRIOS

- Cobertura de ciclos (loop)
- OBJETIVO: Testar todos os ciclos.
- Podem ser definidas quatro diferentes classes de ciclos:

✔ Ciclos simples;
✔ Ciclos concatenados;
✔ Ciclos aninhados;
✔ Ciclos não estruturados.

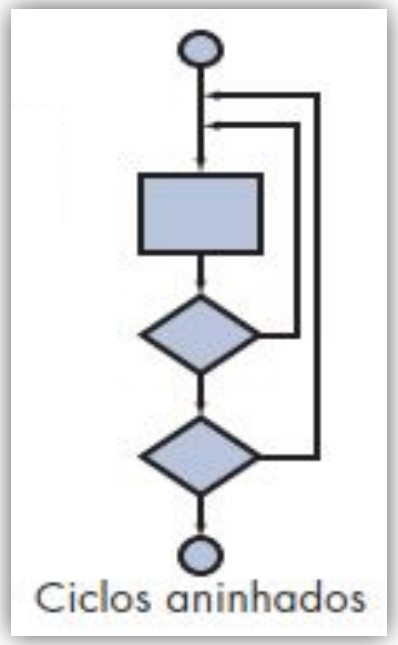
CICLO SIMPLES

- O seguinte conjunto de testes pode ser aplicado a onde ***n*** é o número máximo de passadas permitidas através do ciclo:
- 1. Pular o ciclo inteiramente.
- 2. Somente uma passagem pelo ciclo.
- 3. Duas passagens pelo ciclo.
- 4. m passagens através do ciclo onde m < n.
- 5. n – 1, n, n + 1 passagens através do ciclo.



CICLO ANINHADO

- Se fôssemos estender a abordagem de teste de ciclos simples para ciclos aninhados, o número de testes possíveis cresceria geometricamente à medida que o nível de aninhamento aumentasse.
- O resultado seria um número impossível de testes.
- Beizer (1990) sugere uma **abordagem** que ajudará a reduzir o número de testes.



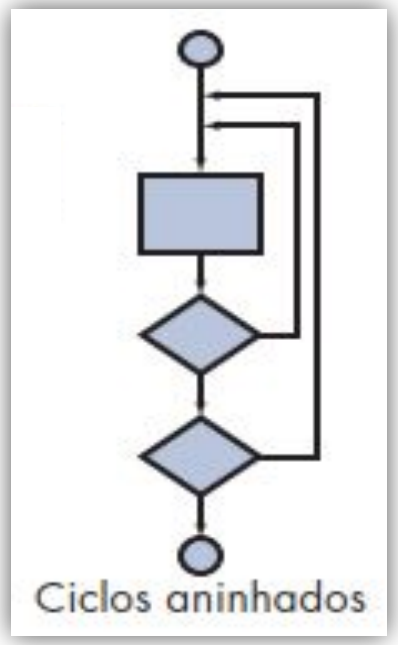
SOLUÇÃO

1. Comece pelo ciclo mais interno. Coloque todos os outros ciclos nos seus valores mínimos.

2. Faça os testes de ciclo simples para o ciclo mais interno mantendo os ciclos externos em seus parâmetros mínimos de iteração. Acrescente outros testes para valores fora do intervalo ou excluídos.

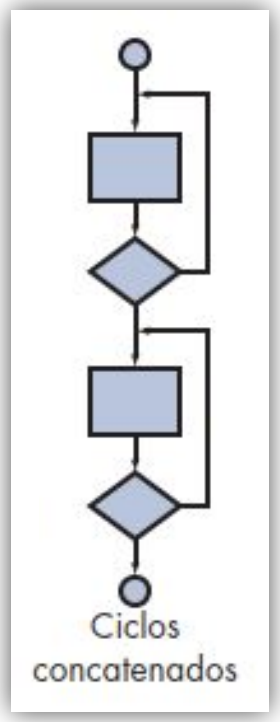
3. Trabalhe para fora, fazendo testes para o próximo ciclo, mas mantendo todos os outros ciclos externos nos seus valores mínimos.

4. Continue até que todos os ciclos tenham sido testados.



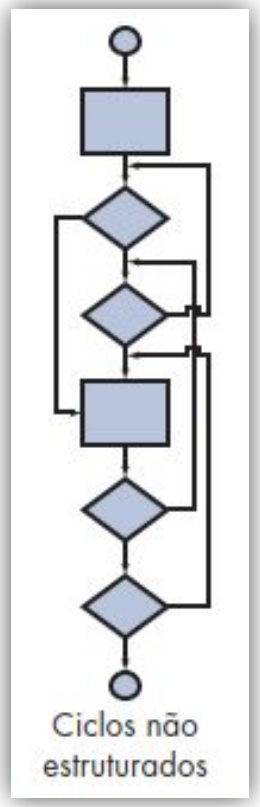
CICLO CONCATENADO

- Podem ser testados usando a abordagem definida para ciclos simples, se cada um for independente do outro.
- Se a contagem para o ciclo 1 for usada como valor individual para o ciclo 2, então os ciclos não são independentes.
- Nesse caso é recomendada a abordagem aplicada a ciclos aninhados.



CICLO NÃO ESTRUTURADO

- Sempre que possível, essa classe de ciclos deverá ser redesenhada para refletir o uso das construções de programação estruturada.


Ciclos não estruturados

TESTES EM OO

- Teste de unidade: métodos individualmente testados
- Teste de classe: testa a integração entre métodos de uma classe
- Teste de integração: testa a interação entre classes do sistema
- Teste de sistema: testa a funcionalidade do sistema como um todo