**Texto base**

**10**

**Estrutura de Repetição Encadeada**

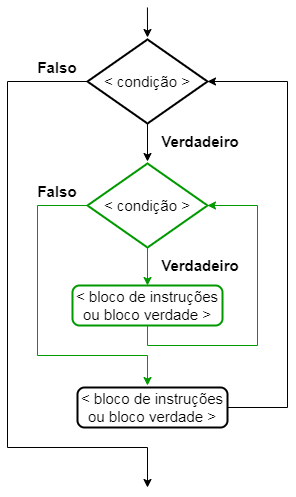
Gilberto Alves Pereira

***Resumo***

*Neste texto é discutido o conceito de estrutura de repetição encadeada ou aninhada. Nessa construção um laço é inserido dentro de outro laço. Forma-se assim uma repetição de uma repetição. Como exemplificação mostra-se uma implementação de um exemplo usando fluxogramas, um segundo exemplo usando fluxogramas juntamente com uma simulação. Ao final é mostrada a conversão do fluxograma para Python.*

# 1.1. Estruturas de Repetição Encadeada ou Aninhada

Na Estrutura de Repetição Encadeada um laço é inserido dentro de um outro laço. Observando a Figura 10.1 percebemos que existe um laço em verde completamente inserido em outro laço. O que ocorre neste caso é que para cada repetição do laço externo (para cada passo do laço externo) o laço interno (em verde) é executado por completo (todos os seus passos).

****

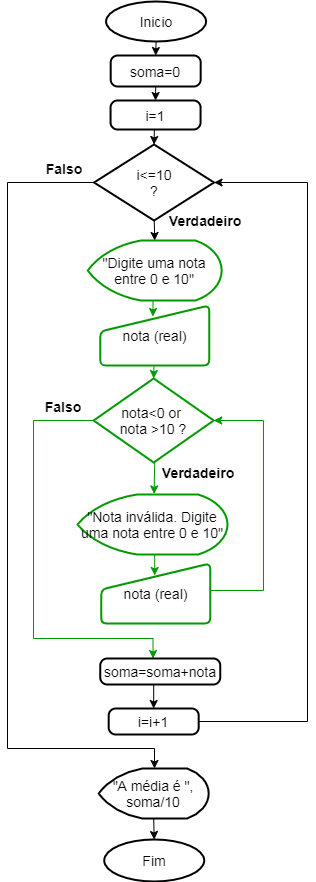
**Figura 10.1. Fluxograma de uma estrutura de repetição encadeada.**

**Fonte Autor.**

Veja que não fizemos nenhuma consideração em relação ao tipo de laço (definido ou indefinido). Essa construção pode ser usada para qualquer combinação desses tipos de laços.

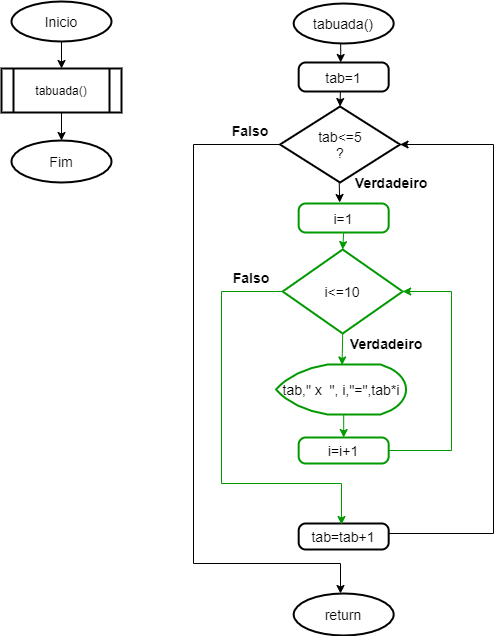
Vejamos o exemplo abaixo:

**Exemplo 1**: Desenhe um fluxograma que receba 10 notas de uma sala e exiba a média dessas notas. OBS.: A entrada da nota deve ser validada por um loop.

**Figura 10.2. Fluxograma do exemplo 1. Fonte Autor.**

Perceba que nesse exemplo temos o loop da validação em verde (loop indefinido), dentro de um loop definido (que executa 10 vezes).

**Exemplo 2**: Desenhe um fluxograma que exiba as tabuadas do 1 até 5.

****

**Figura 10.3. Fluxograma do exemplo 2 - Tabuada.**

**Fonte Autor.**

Criamos uma função chamada tabuada. Essa função tem o objetivo de exibir as tabuadas do 1 ao 5. Não utilizaremos parâmetros nessa função. Perceba que o loop em verde executa a exibição de uma única tabuada. Como queremos exibir várias tabuadas (do 1 ao 5) colocaremos esse loop dentro de outro loop executado 5 vezes, uma vez para a tabuada do 1, e até a tabuada do 5. Para cada execução do loop externo é exibida uma tabuada inteira (loop verde). O valor da variável **tab** define qual a tabuada será exibida. Foi criado também um fluxograma que chama a função tabuada. Veja que como a função tabuada não utiliza parâmetros e também não retorna nenhum valor (ele exibe na tela a tabuada) não foi criada nenhuma variável no programa. É apenas invocada a função **tabuada**.

Segue uma simulação simplificada (Tabuadas do 4 ao 5 e ao invés de ir do 1 ao 10 para cada tabuada a simulação vai do 1 ao 3)

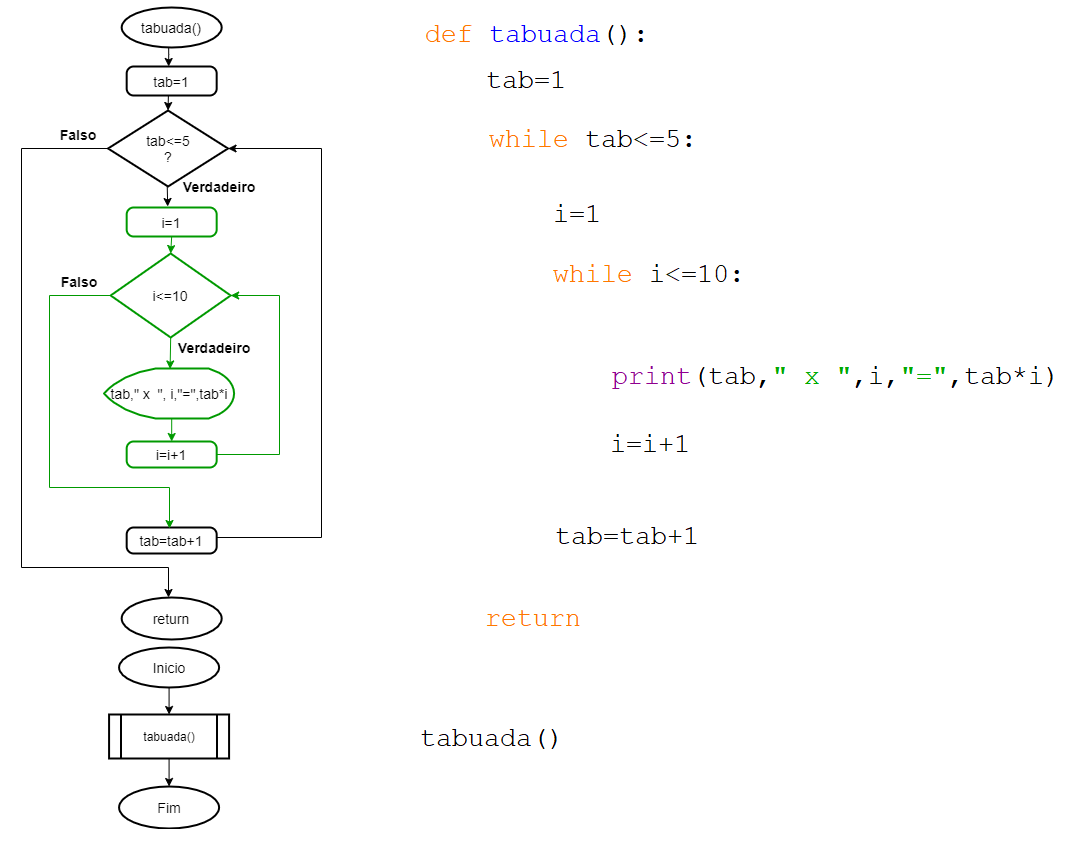
| **Tela** | **i** | **i<=3** | **tab** | **tab<=5** |
| --- | --- | --- | --- | --- |
| 4 x 1 = 4 | 1 | 1<=3 V | 4 | 4<=5 V |
| 4 x 2 = 8 | 2 | 2<=3 V | 5 | 5<=5 V |
| 4 x 3 = 12 | 3 | 3<=3 V | 6 | 6<=5 **F** |
| 5 x 1 = 5 | 4 | 4<=3 **F** |  |  |
| 5 x 2 = 10 | 1 | 1<=3 V |  |  |
| 5 x 3 = 15 | 2 | 2<=3 V |  |  |
|  | 3 | 3<=3 V |  |  |
|  | 4 | 4<=3 **F** |  |  |

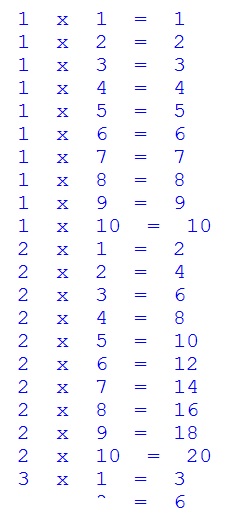
**Quadro 10.1. Simulação do Fluxograma da tabuada (simplificado).**

**Fonte Autor**

**1.2. Comparativo entre fluxograma e Python**

Abaixo podemos identificar um fluxograma do Exemplo 2 e sua conversão para Python. Na Figura 10.5 pode ser observado um trecho da saída do programa em Python.

**Figura 10.4. Esquema do fluxograma e seu respectivo código em Python referentes ao exemplo 2 (Tabuada). Fonte Autor.**

**Figura 10.5 Trecho da saída em Python. Fonte Autor.** 

**1.3. Referências**

DIERBACH, C. “Introduction to Computer Science Using Python: A Computational Problem-Solving Focus”1st Edition, New York: Wiley, 2012.