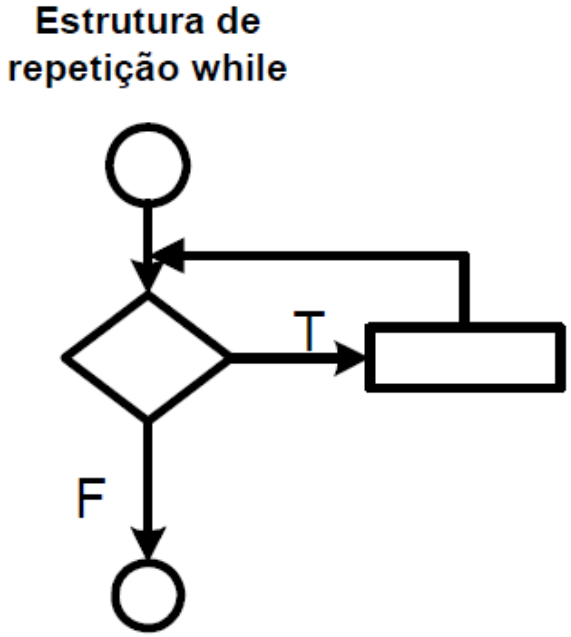
Estruturas de repetição

Quando utilizamos as estruturas de repetição?

Utilizamos as estruturas de repetição quando desejamos que um determinado conjunto de instruções ou comandos sejam executados um número definido ou indefinido de vezes, ou enquanto um determinado estado de coisas prevalecer ou até que seja alcançado.

Estrutura de repetição While

O bloco de instruções será executado enquanto a condição x for verdadeira. O teste da condição será sempre realizado antes de qualquer operação. Enquanto a condição for verdadeira o processo se repete.



Pseudocódigo *enquanto ... Faça -> While*

Sintaxe:

enquanto *<expressão-lógica>* faca
    <seqüência-de-comandos>
fimenquanto

Exemplo:

```
algoritmo "Números de 1 a 10 (com enquanto...faca)"
var j: inteiro
inicio
    j <- 1
    enquanto j <= 10 faca
        escreva (j:3)
        j <- j + 1
    fimenquanto
fimalgoritmo
```

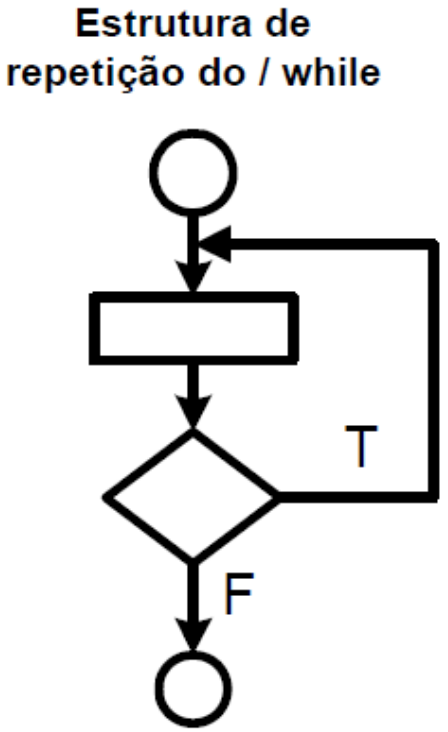
Estrutura de repetição While (sintaxe)

Sintaxe do comando "while" (enquanto) utilizada na linguagem C:

```
while (<condição>)
{

        Comando 1;

        Comando 2;

        Comando n;

}
```

Estrutura de repetição do ..... while

Neste caso primeiro são executados os comandos, e somente depois é realizado o teste da condição. Se a condição for verdadeira, os comandos são executados novamente, caso seja falso é encerrado o comando DO.



Pseudocódigo *repita ... até -> do ... while*

Sintaxe:

repita

    <seqüência-de-comandos>

ate *<expressão-lógica>*

Exemplo:

```
algoritmo "Números de 1 a 10 (com repita)"
var j: inteiro
inicio
    j <- 1
    repita
        escreva (j:3)
        j <- j + 1
    ate j > 10
fimalgoritmo
```

Estrutura de repetição do...while (sintaxe)

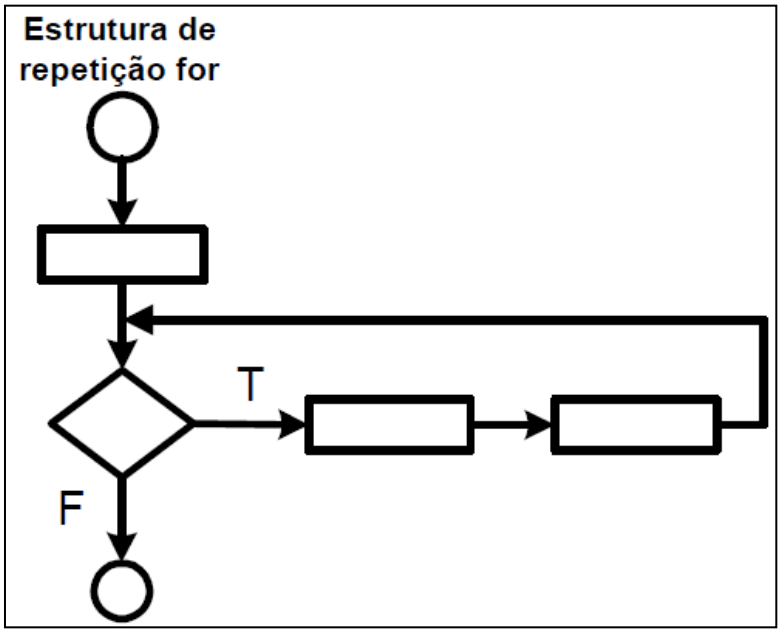
Sintaxe em "C" para realização da repetição com teste no final:

```
do
{

        comandos;

} while (condição);
```

Estrutura de repetição For

O FOR é utilizado quando sabemos exatamente a quantidade de vezes que determinado comando ou bloco de comandos serão executados.
A "expressão1" é executada incondicionalmente uma única vez no início da estrutura de repetição, na sequência será executado o teste da condição, e irá executar o comando ou bloco de comando se a condição for "verdadeira", e finalmente a "expressão2" será executada com um comando de incremento/decremento.

FOR (expressão1 ; condição ; expressão2)



Pseudocódigo *para ... faça -> for*

Sintaxe:

para *<variável>* de *<valor-inicial>* ate *<valor-limite>* [passo *<incremento>*] faca
    <seqüência-de-comandos>
fimpara

Exemplo:

```
algoritmo "Números de 1 a 10"
var j: inteiro
inicio
    para j de 1 ate 10 faca
        escreva (j:3)
    fimpara
fimalgoritmo
```

Estrutura de repetição For (sintaxe)

Sintaxe do comando "for" (para) utilizada na linguagem C:

```
for(inicialização; condição final; incremento)
{
comandos;
}
```

Na aplicação do comando "for", **há três expressões separadas por ponto e vírgula: inicialização**, **condição final** e **incremento**.

Estrutura de repetição For (sintaxe)

```
1.  #include <stdio.h>
2.  int main(void)
3.  {
4.    int contador;
5.    printf("\nDigite um numero para contagem regressiva\n\n");
6.    scanf("%d", &contador);
7.      for (contador; contador >= 1; contador--)
8.    {
9.        printf("%d ", contador);
10.   }
11.   getch();
12.   return(0);
13. }
```

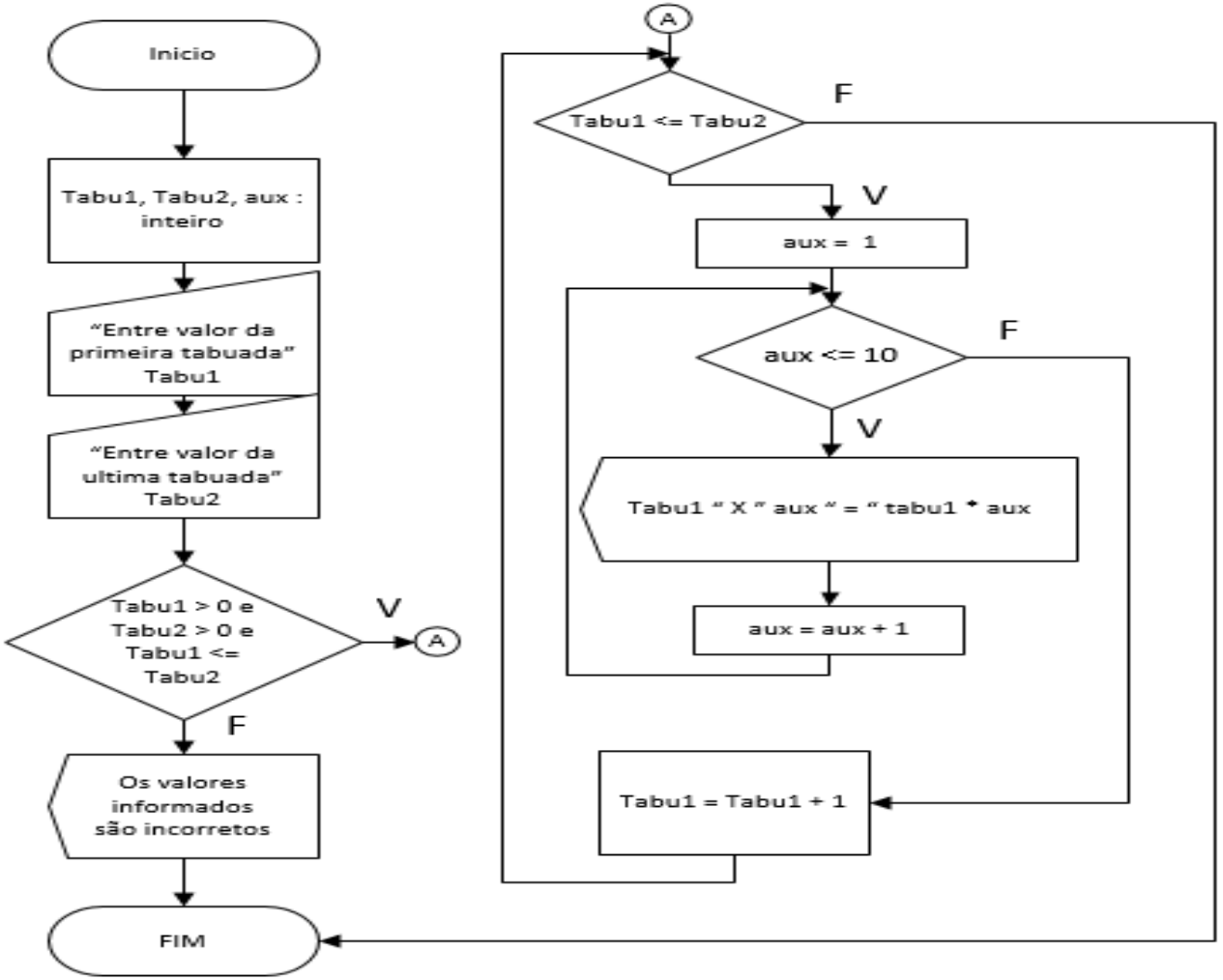
Fonte: autor

Exercício 1 - Tabuada

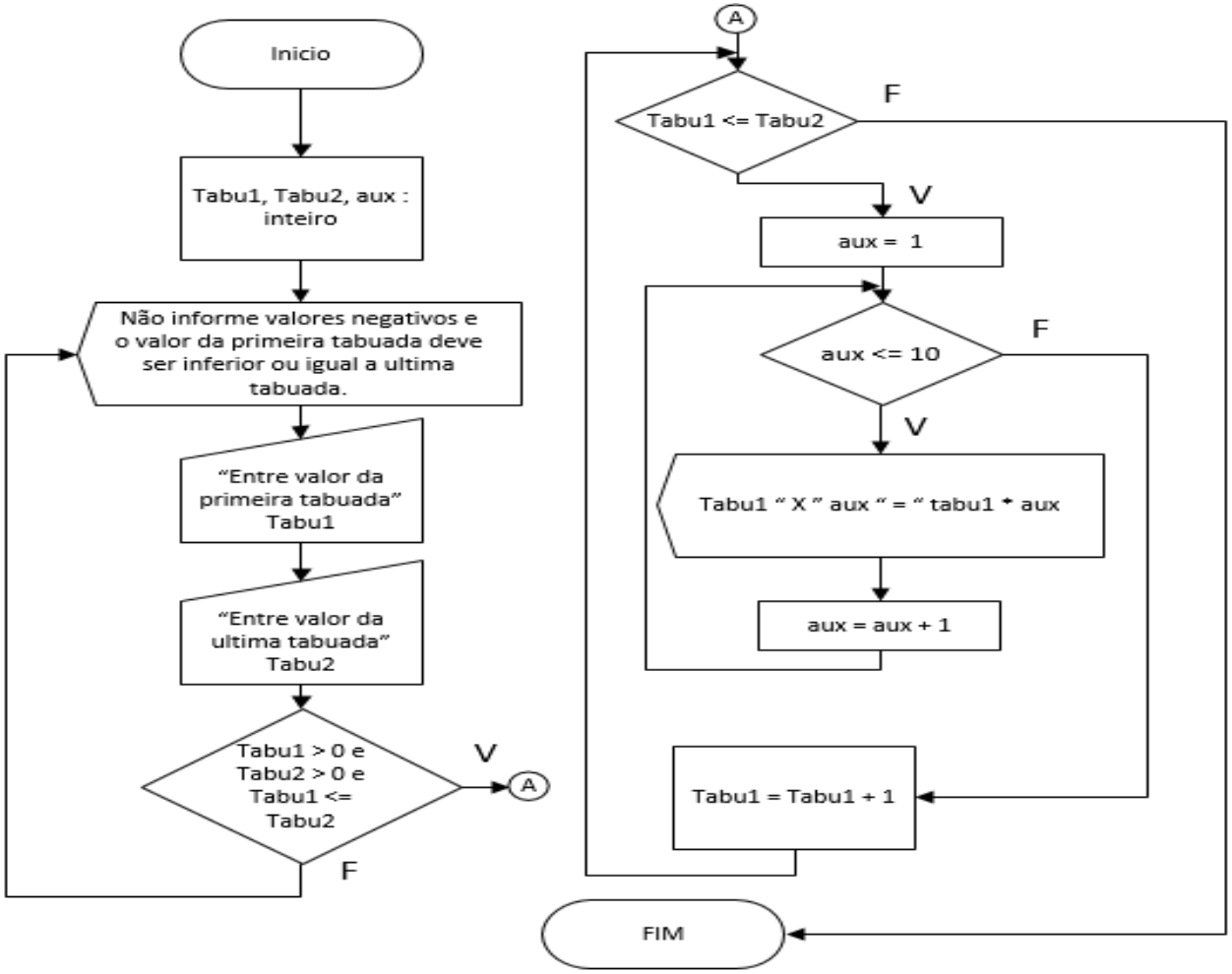
Crie o seguinte programa

Desenvolva um algoritmo onde o usuário deverá entrar com o valor da tabuada inicial e o valor da tabuada final. Disponibilizar na tela todas as tabuadas solicitadas. Utilize as estruturas de repetição que achar conveniente.

Exercício 1 – Tabuada resolvido Versão 1



Exercício 1 – Tabuada resolvido Versão 2



Exercício 2 – Número par

Desenvolva o seguinte programa em C

Desenvolva um algoritmo onde o usuário deverá entrar com dois valores numéricos que correspondem a um intervalo de números qualquer, sendo que, após esta entrada de dados o processamento deverá apresentar para o usuário todos os números pares deste intervalo.

Exercício 3 – Maior Número

Desenvolva o seguinte programa em C

Faça um algoritmo que determine o maior entre **N** números. A condição de parada é a entrada de um valor 0, ou seja, o algoritmo deve ficar calculando o maior até que a entrada seja igual a 0 (ZERO).

Exercício 4 – Números múltiplos

Desenvolva o seguinte programa em C

Faça um algoritmo que conte de 1 a 100 e a cada múltiplo de 10 emita uma mensagem: "X é múltiplo de 10"