

# BLU3101 – Introdução à Informática para Automação

Prof. Mauri Ferrandin  
Prof. Alex Roschildt Pinto

# Programação de Computadores

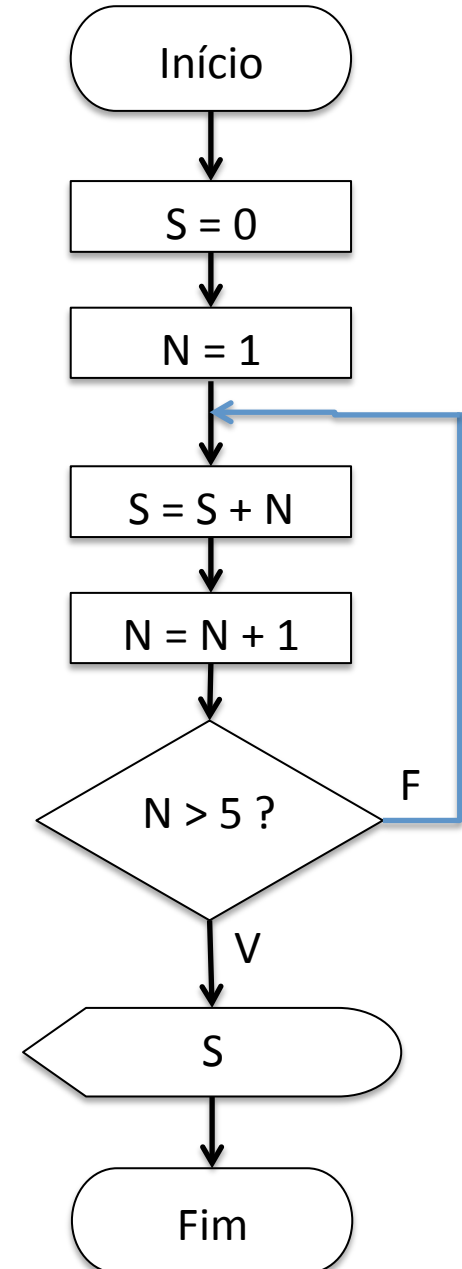
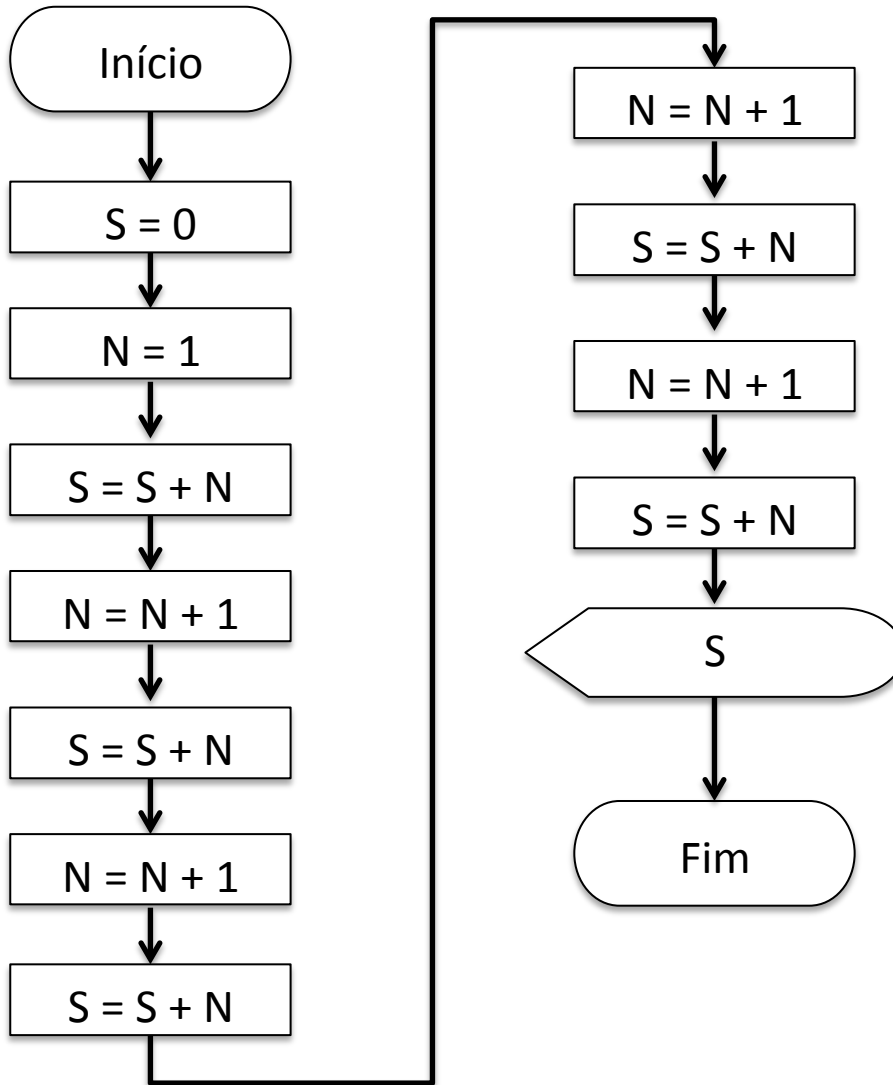
- Linguagem C
  - Iteração/repetição/laços
    - while
    - do while
    - for
  - Interrompendo blocos/loops
    - break;

# Iteração ou repetição

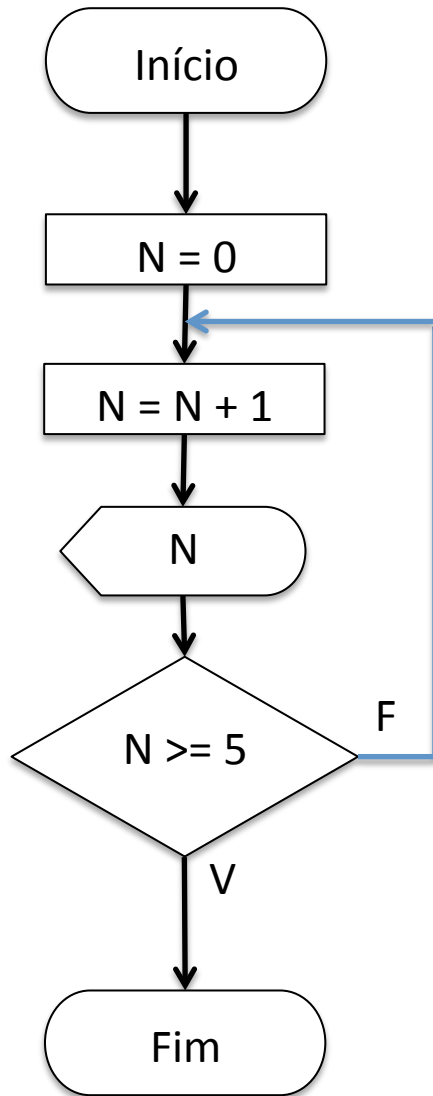
- Possibilita a repetição de um bloco de programa. O número de vezes em que o bloco será repetido é controlado através de uma expressão lógica.

# Fluxograma

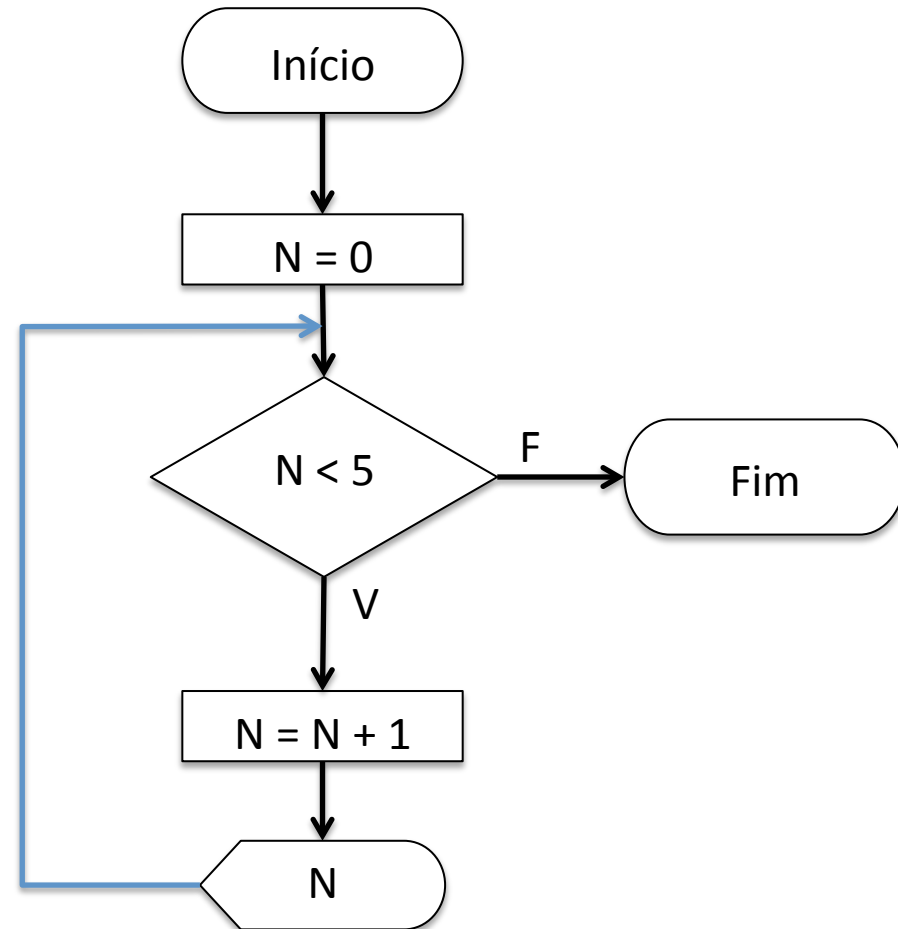
Algoritmos para somar  
os 5 primeiros números  
inteiros positivos



# Fluxograma



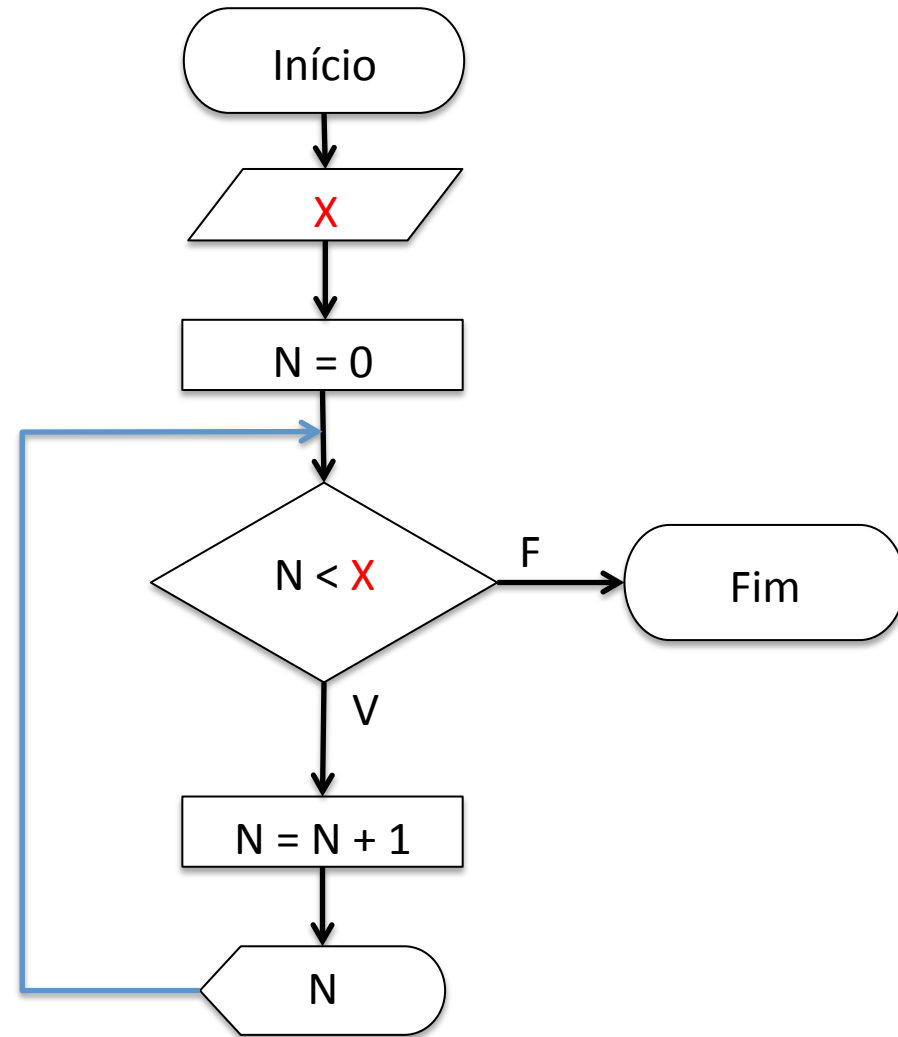
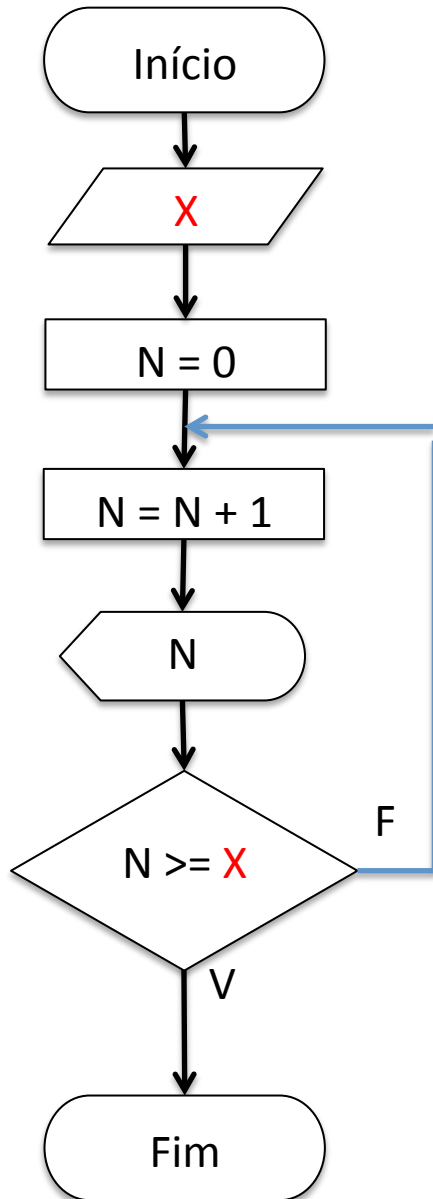
Algoritmos para mostrar os  
inteiros positivos de 1 a 5



Existem diferenças?

# Fluxograma

Algoritmos para mostrar os inteiros positivos de 1 a **X** (informado pelo usuário)



Existem Diferenças?

# Programação de Computadores

- Iteração/repetição/laços
  - while (enquanto faça)
    - Teste condicional no início do bloco a ser repetido
  - do while (faça enquanto)
    - Teste condicional no final do bloco a ser repetido
  - for (para)
    - Intervalo - número de repetições - conhecido (normalmente)
- Instrução break em laços;

# Iteração while em C

Possibilita repetir um bloco de comandos enquanto o resultado da expressão lógica for verdadeiro;

Sintaxe:

```
while (Expressão lógica) {  
    <sequência de comandos>  
}
```

```
/* mostrar os valores de 1 a 5 */  
int n = 0;  
while (n < 5) {  
    n = n + 1;  
    printf("%d\n",n);  
}
```



# Iteração do while em C

Possibilita repetir um bloco de comandos enquanto o resultado da expressão lógica for verdadeiro. O bloco sempre será executado pelo menos uma vez.

Sintaxe:

```
do {  
    <sequência de comandos>  
} while (Expressão lógica)
```

```
/* mostrar os valores de 1 a 5 */  
int n = 0;  
do {  
    n = n + 1;  
    printf("%d\n",n);  
} while (n < 5) ;
```

# Iteração for em C

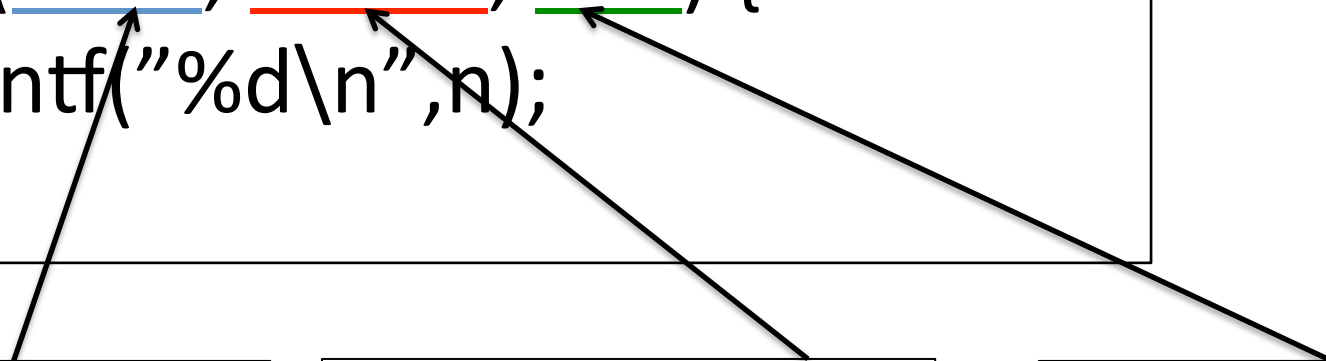
Possibilita repetir um bloco de comandos. O controle é feito através de variáveis definidas para o próprio laço. Normalmente empregado para situações em que se conhece o número de vezes que o bloco precisa ser executado.

Sintaxe:

```
for (Inicializações; Expressão lógica; incrementos) {  
    <sequência de comandos>  
}
```

# Iteração for em C

```
/* mostrar os valores de 1 a 5 */  
for (n = 1; n <= 5; n++) {  
    printf("%d\n",n);  
}
```



Inicialização  
das variáveis  
de controle  
da repetição

Expressão lógica

Incremento das  
variáveis

# Iteração for em C

```
/* estrutura complexa*/  
for (  
    a = 0, b = 0;  
    a * b < 1000 && a <= 30;  
    a = a + 1, b = b + 2  
) {  
    printf("a: %d b: %d\n",a,b);  
}
```

# break em laços de iteração

A instrução **break** possibilita a interrupção de um laço de repetição. Quando a execução encontra uma instrução break a repetição é interrompida e a execução passa para a próxima instrução após o bloco de repetição.

Sintaxe:

```
bloco de repetição{  
    break;  
}
```

# Exemplos break

```
/* n terá valor 15 */  
int n = 0;  
while (n < 100) {  
    if (n == 15) {  
        break;  
    }  
    n = n + 1;  
}  
printf("%d\n",n);
```

```
/* n terá valor 15 */  
int n;  
for (n = 0;n < 100, n++) {  
    if (n == 15) {  
        break;  
    }  
}  
printf("%d\n",n);
```

# Referências sobre C

- Programar em C (Wikibook)
  - [http://pt.wikibooks.org/wiki/Programar\\_em\\_C](http://pt.wikibooks.org/wiki/Programar_em_C)
- Livro aberto: Aprendendo a Programar: Programando na Linguagem C
  - <http://professor.ic.ufal.br/jaime/livros/Aprendendo%20a%20Programar%20Programando%20na%20Linguagem%20C.pdf>
- Mensagem de Mark Zuckerberg sobre repetição:
  - <https://www.youtube.com/watch?v=mgooqyWMTxk>