Lógica de Programação

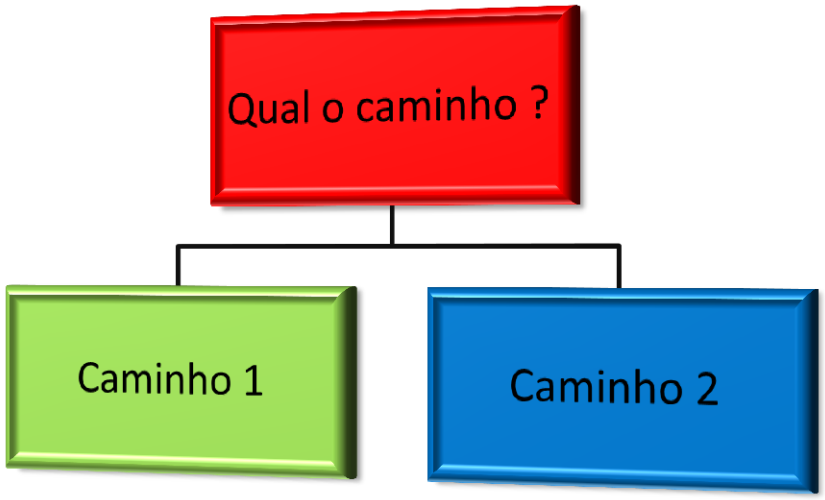
Fabrício Curvello Gomes

Rodrigo Dacome Lima

Programação Com Decisão

e

Operadores Lógicos



Programação Com Decisão

Decisões, Condições e Operadores Relacionais

Condição – É uma obrigação que se impõe e se aceita.

Decisão – É o ato ou efeito de decidir, de tomar uma decisão.

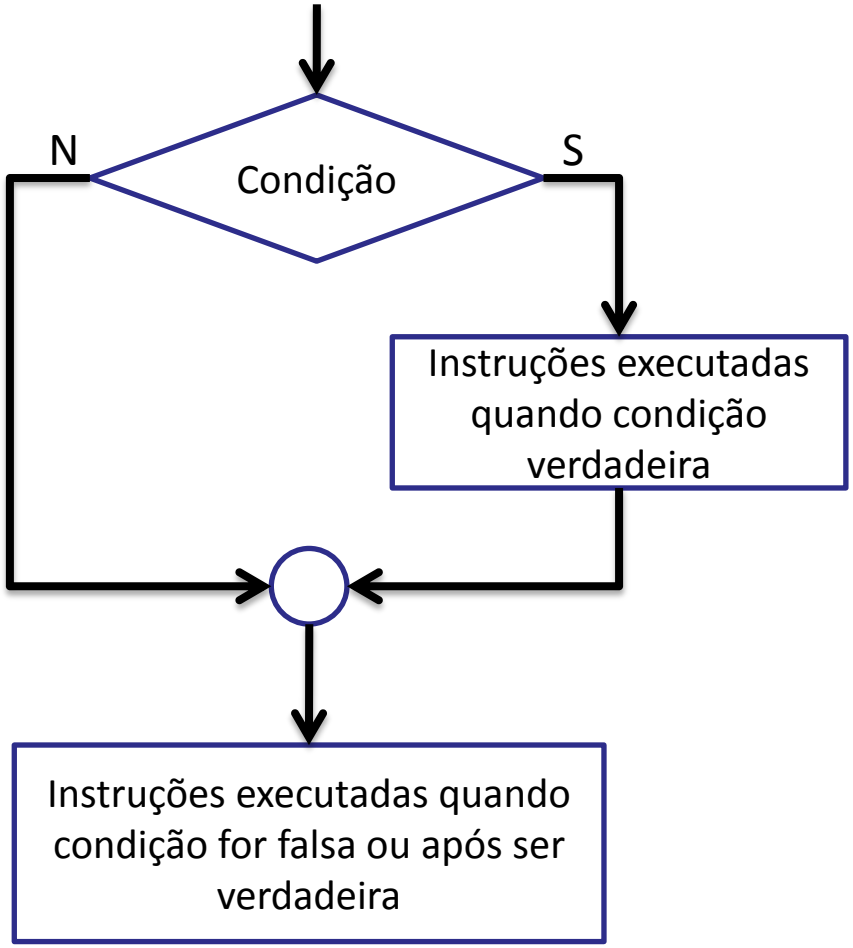
O ato de tomar uma decisão está calcado no fato de haver uma condição.

Do ponto de vista computacional, condição é uma expressão booleana cujo resultado é um valor lógico **falso** ou **verdadeiro**.

Para ter uma expressão booleana como condição, usa-se uma relação lógica entre dois elementos e um operador relacional.

Serão utilizados os operadores lógicos apresentados no capítulo 8.

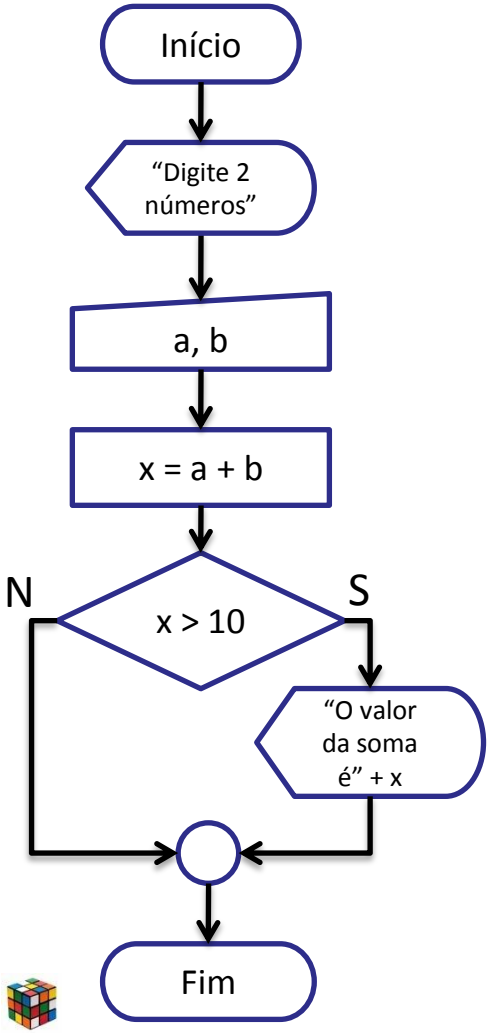
Desvio Condicional Simples




```
...
if (<condição>) {
  <instruções para
  condição verdadeira>
}
<instruções para condição
falsa ou após ser
verdadeira>
```

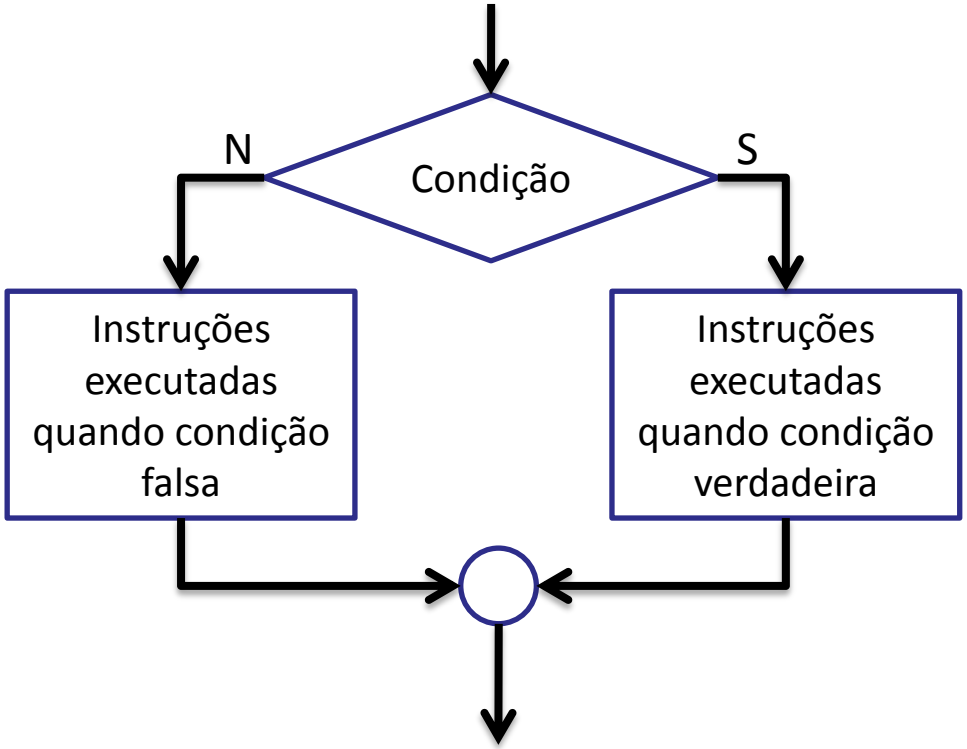
Desvio Condicional Simples (Cont.)

Exemplo: Algoritmo para ler dois valores numéricos, efetuar a adição e apresentar o resultado **se** o valor for maior que 10.




```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  int a, b, x;

  cout << "Primeiro número:";
  cin >> a;
  cout << "Segundo número:";
  cin >> b;

  x = a + b;

  if (x > 10){
    cout << "O resultado é " << x << endl;
  }

  return 0;
}
```

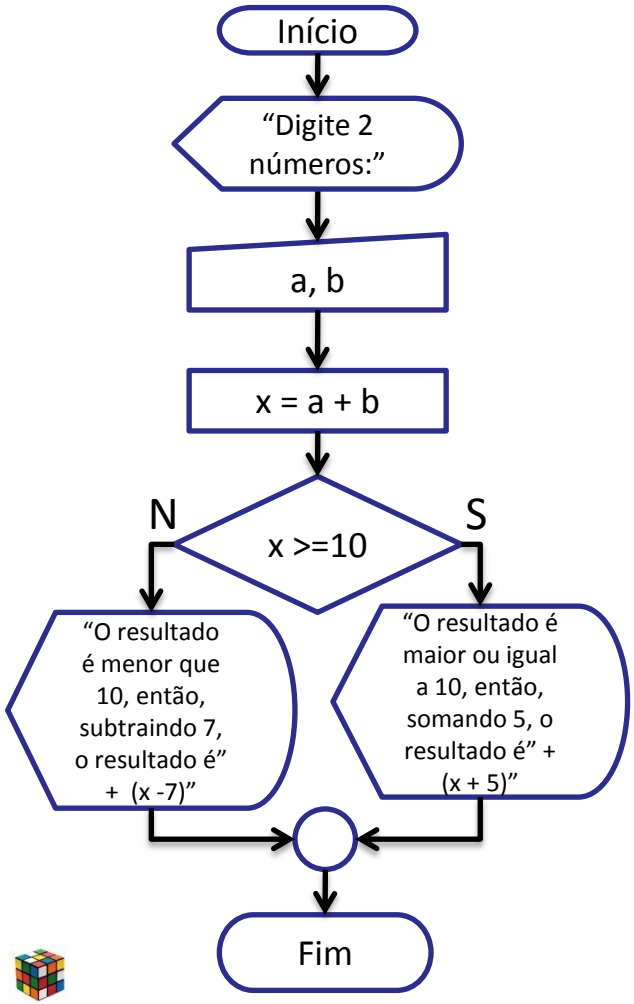
Desvio Condicional Composto




```
...
if (<condição>) {
   <instruções para
   condição verdadeira>
} else {
  <instruções para
   condição falsa>
}
```

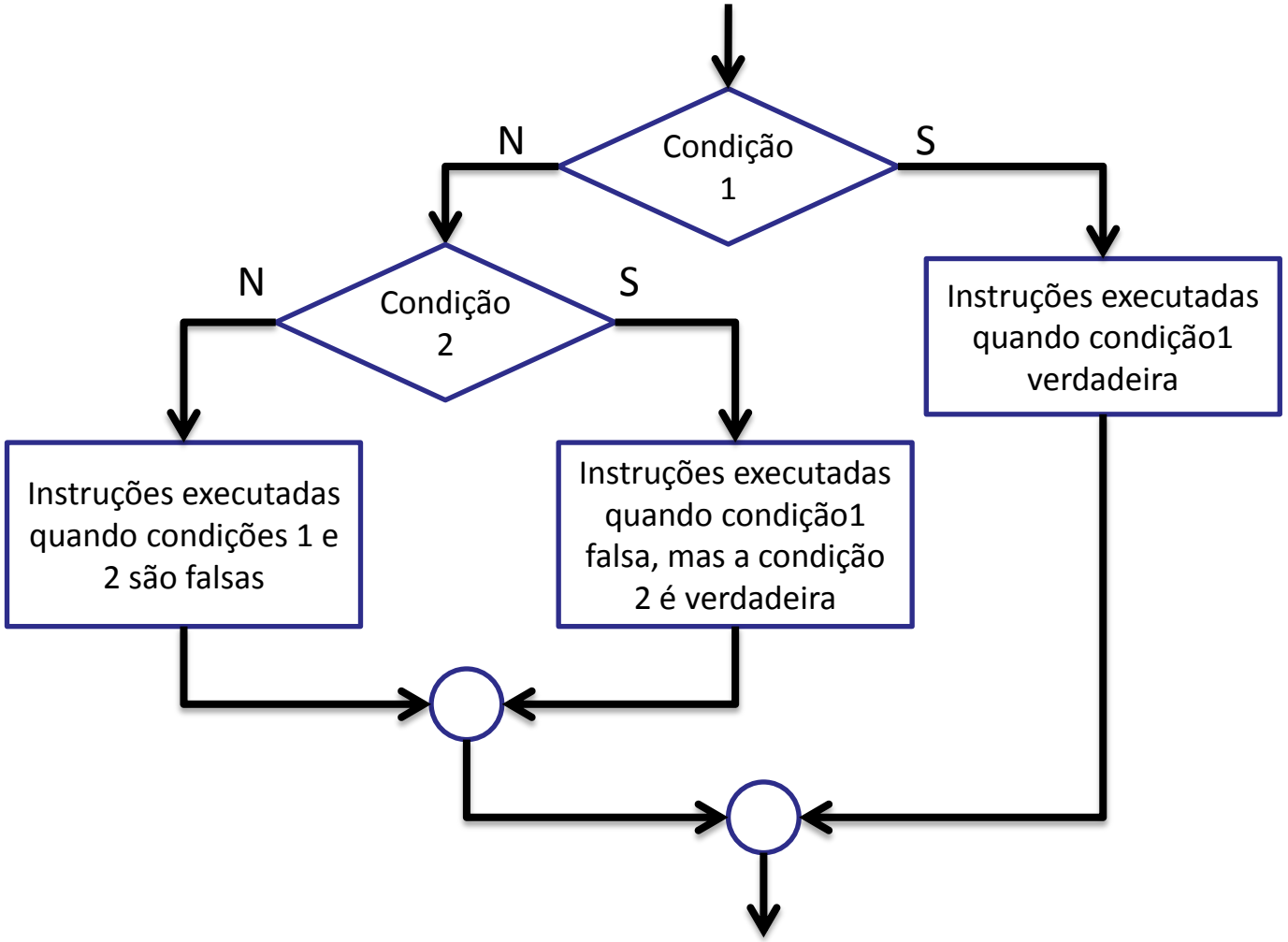
Desvio Condicional Composto (Cont.)

Exemplo: Algoritmo para ler dois valores numéricos, efetuar a adição. **Se** a soma for maior ou igual a 10, apresente o resultado somando 5, **senão**, apresente o resultado subtraindo 7.



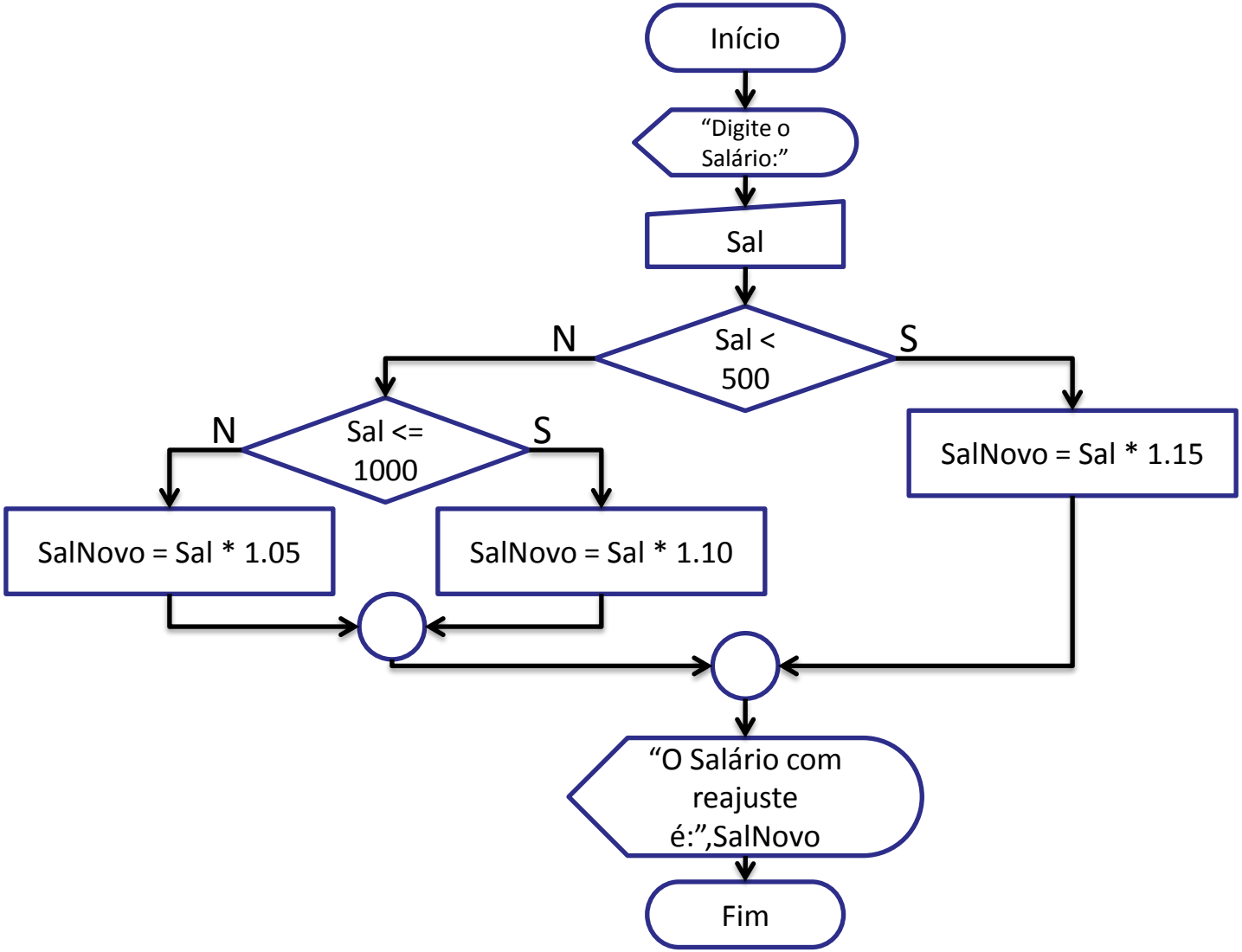

```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  int a, b, x;

  cout << "Primeiro número:";
  cin >> a;
  cout << "Segundo número:";
  cin >> b;

  x = a + b;

  if (x >= 10){
    cout << "O resultado + 5 é " << (x + 5) << endl;
  }else{
    cout << "O resultado -7 é " << (x - 7) << endl;
  }

  return 0;
}
```

Desvios Condicionais Encadeados



Desvios Condicionais Encadeados (Cont.)

Exemplo: Elaborar um programa que calcule o reajuste de salário de um funcionário, sob as seguintes condições: Reajuste de 15% **se** salário menor que R$ 500,00. Reajuste de 10% **se** salário entre R$ 500,00 e R$ 1000,00. Reajuste de 5% **se** salário acima de R$ 1000,00.



Desvios Condicionais Encadeados (Cont.)

Exemplo: Elaborar um programa que calcule o reajuste de salário de um funcionário, sob as seguintes condições: Reajuste de 15% **se** salário menor que R$ 500,00. Reajuste de 10% **se** salário entre R$ 500,00 e R$ 1000,00. Reajuste de 5% **se** salário acima de R$ 1000,00.

```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  double sal, salNovo;

  cout << "Digite o Salário: ";
  cin >> sal;

  if (sal < 500){
    salNovo = sal * 1.15;
  }else{
    if (sal <= 1000){
      salNovo = sal * 1.10;
    }else{
      salNovo = sal * 1.05;
    }
  }

  cout << "O Salário com reajuste é: " << salNovo << endl;
  return 0;
}
```

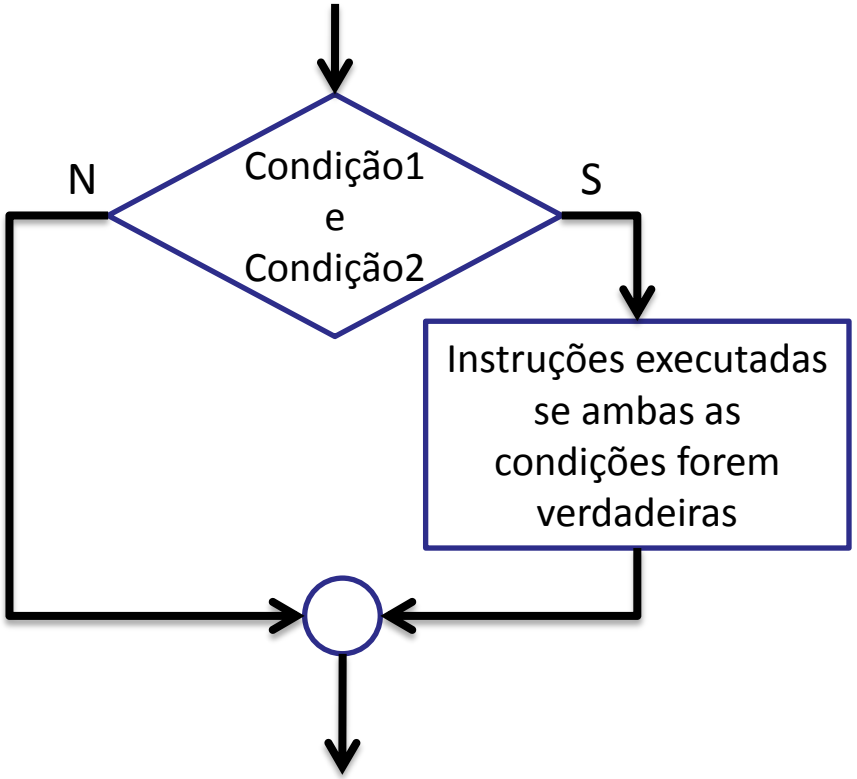
Operadores Lógicos

Operadores Lógicos

Conforme já foi visto no Capítulo 3, segue novamente a tabela verdade dos operadores lógicos:

A	B	NÃO A	NÃO B	A E B	A OU B	A XOU B
VERDADEIRO	**VERDADEIRO**	FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO
VERDADEIRO	**FALSO**	FALSO	VERDADEIRO	FALSO	VERDADEIRO	VERDADEIRO
FALSO	**VERDADEIRO**	VERDADEIRO	FALSO	FALSO	VERDADEIRO	VERDADEIRO
FALSO	**FALSO**	VERDADEIRO	VERDADEIRO	FALSO	FALSO	FALSO

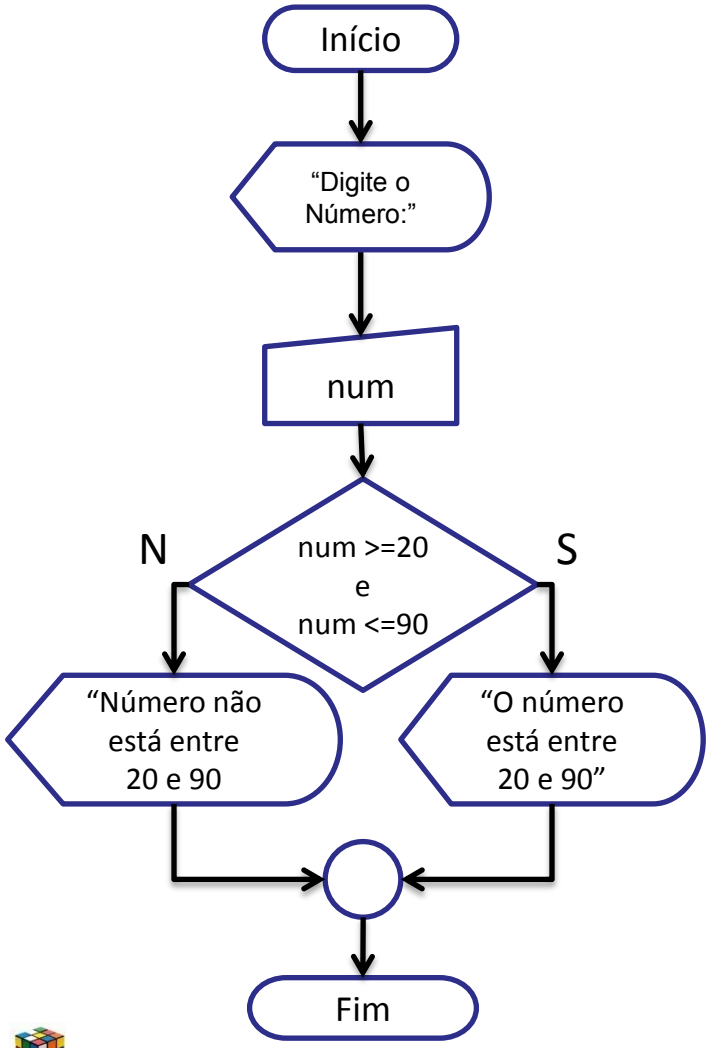
Operador Lógico E - &&




```
...
if ((<condição1>) &&
   (<condiçao2>)) {
   <instruções executadas se
   ambas forem verdadeiras>
}
```

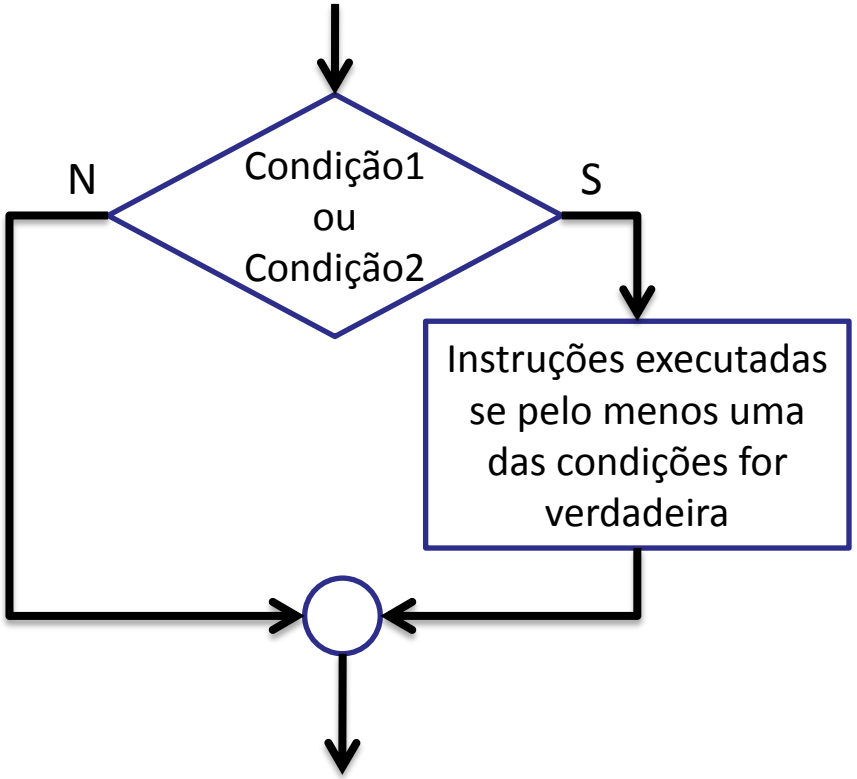
Operador Lógico E - && (Cont.)

Algoritmo que testa se um numero dado está na faixa de 20 a 90.




```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  int num;

  cout << "Digite um número:";
  cin >> num;

  if ((num >= 20) && (num <= 90)) {
    cout << "Número está entre 20 e 90" << endl;
  } else {
    cout << "Número não está entre 20 e 90" << endl;
  }

  return 0;
}
```

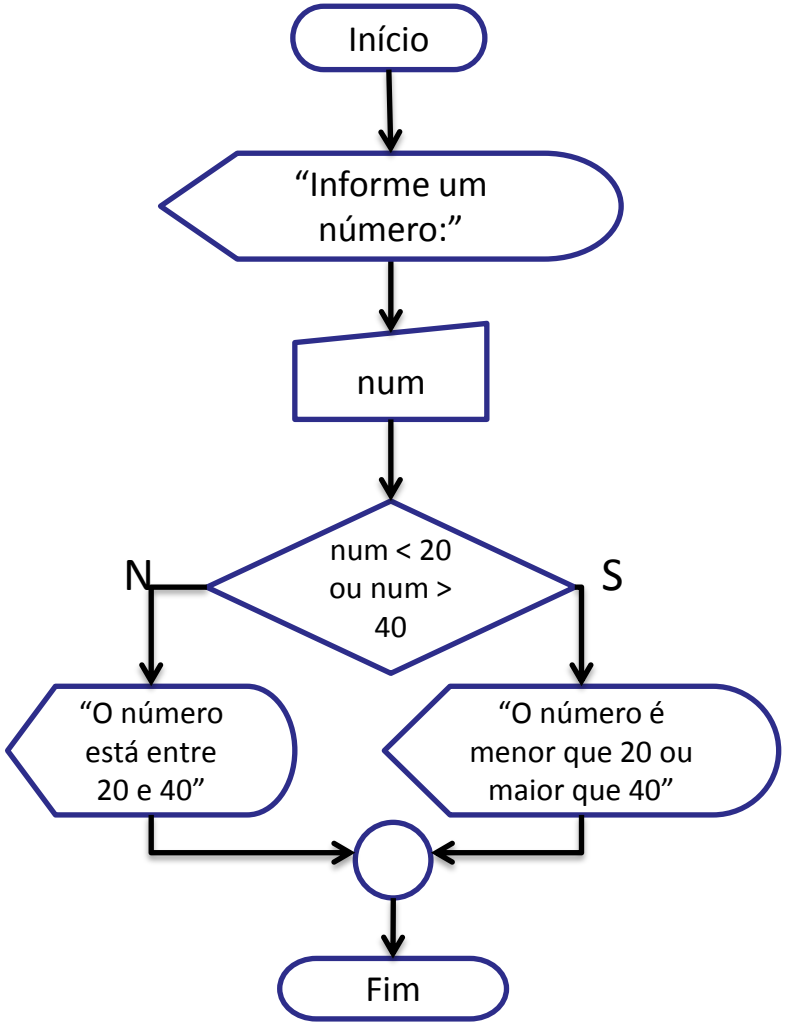
Operador Lógico OU - ||




```
...
if ((<condição1>) ||
   (<condiçao2>)) {
   <instruções executadas se
   pelo menos uma das
   condições for verdadeira>
}
```

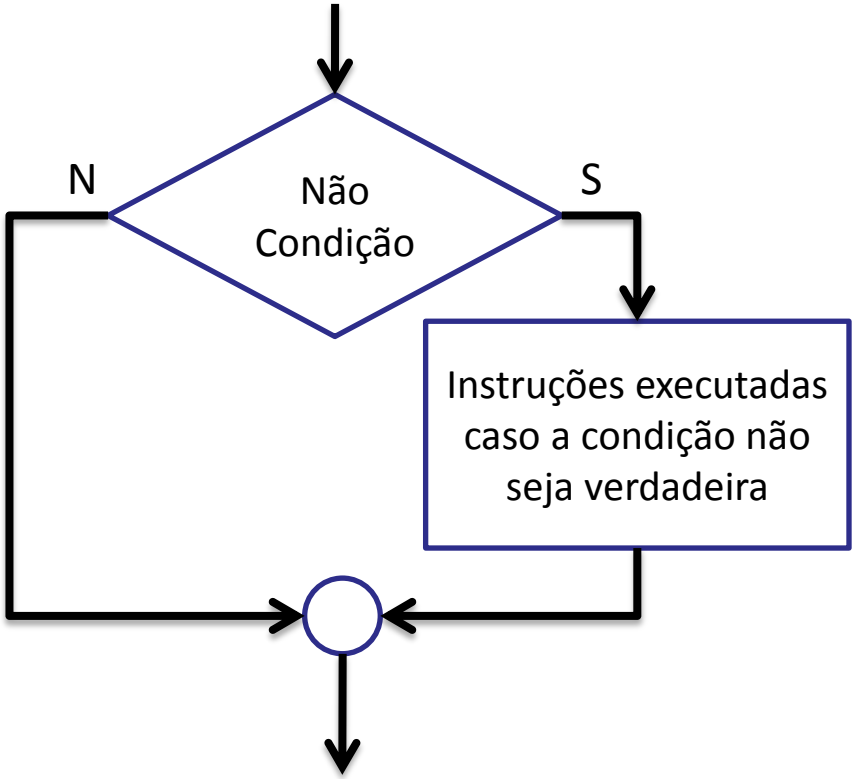
Operador Lógico OU - || (Cont.)

Algoritmo que testa se um número é menor que 20 ou maior que 40.




```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  int num;

  cout << "Digite um número:";
  cin >> num;

  if ((num < 20) || (num > 40)) {
    cout << "Menor que 20 ou maior que 40" << endl;
  } else {
    cout << "Está entre 20 e 40" << endl;
  }
  return 0;
}
```

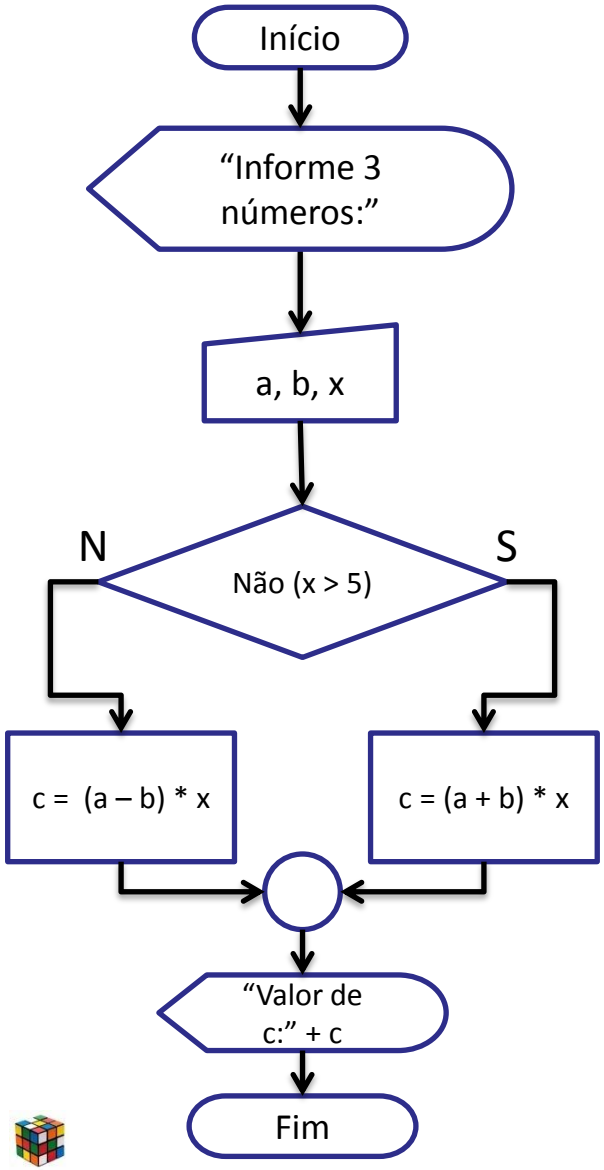
Operador Lógico NÃO - !




```
...
if (!(<condição>)) {
   <Instruções executadas
   caso a condição não seja
   verdadeira>
}
```

Operador Lógico NÃO - ! (Cont.)

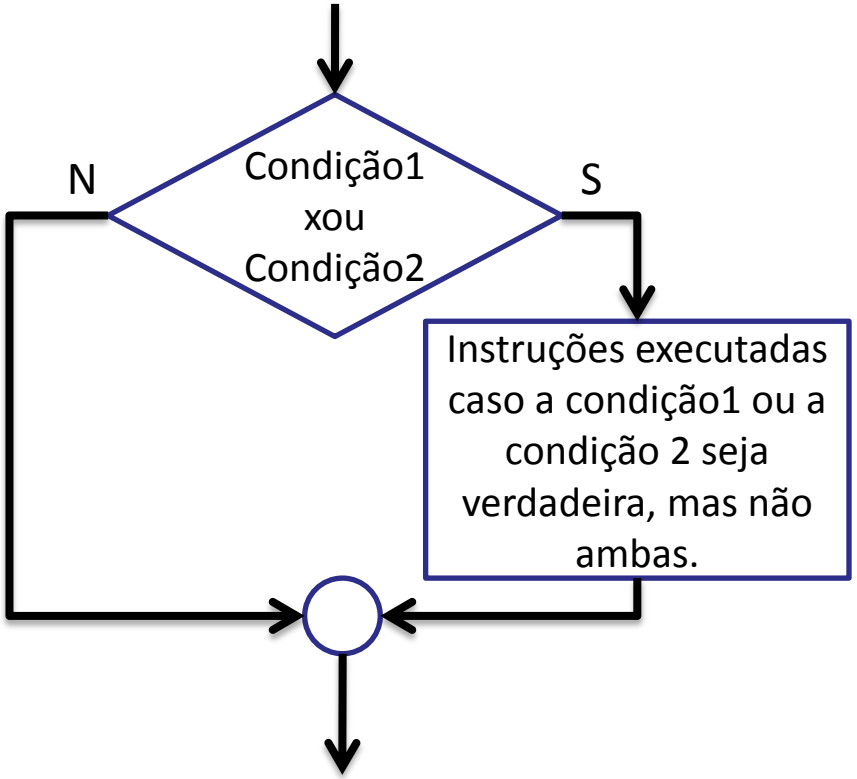
Observe o que este algoritmo faz:




```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  int a, b, c, x;

  cout << "Primeiro número:";
  cin >> a;
  cout << "Segundo número:";
  cin >> b;
  cout << "Terceiro número:";
  cin >> x;

  if (!(x > 5)){
    c = (a + b) * x;
  } else {
    c = (a - b) * x;
  }

  cout << "Valor de c:" << c << endl;
  return 0;
}
```

Na verdade, esta sentença equivale a:
if (x <= 5) {

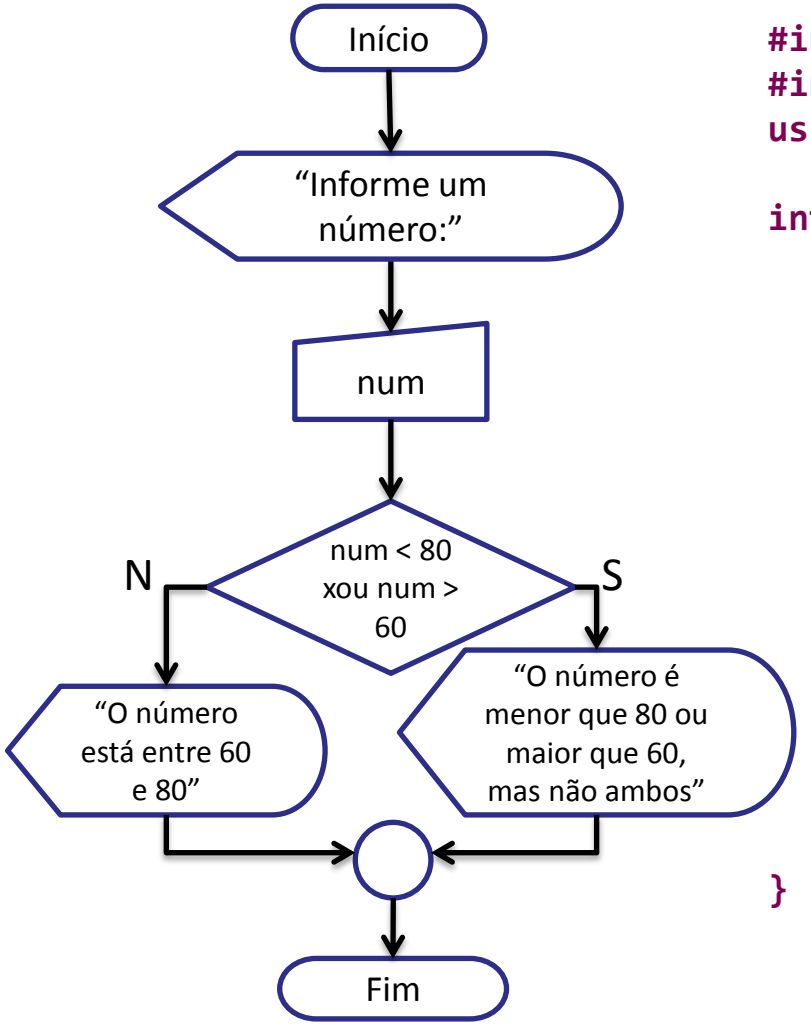
Operador Lógico XOU - ^




```
...
if ((<condição1>) ^
(<condição2>))
  <Instruções executadas
  caso a condição1 ou a
  condição2 seja verdadeira,
  mas não ambas>
}
```

Operador Lógico XOU - ^ (Cont.)

Algoritmo que testa se um número é menor que 80 ou maior que 60, mas não ambos, ou seja, números entre 60 e 80 serão descartados.




```
#include <iostream>
#include <locale>
using namespace std;

int main(){
  setlocale(LC_ALL,"Portuguese");
  int num;

  cout << "Informe um número:";
  cin >> num;

  if ((num < 80) ^ (num > 60)) {
    cout << "O número é menor que 80 ou maior que 60,"
    << " mas não ambos" << endl;
  } else {
    cout << "O número está entre 60 e 80" << endl;
  }

  return 0;
}
```

Dúvidas?

Bibliografia

Estudo Dirigido de Algoritmos
José Auguto N. G. Manzano e Jayr Figueiredo de Oliveira
Ed. Érica

Fundamentos de Computação e Orientação a Objetos Usando JAVA
Francisco A. C. Pinheiro
Ed. LTC