



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

ESTRUTURA COM DECISÃO COMPOSTA

Prof. Dr. Daniel Caetano

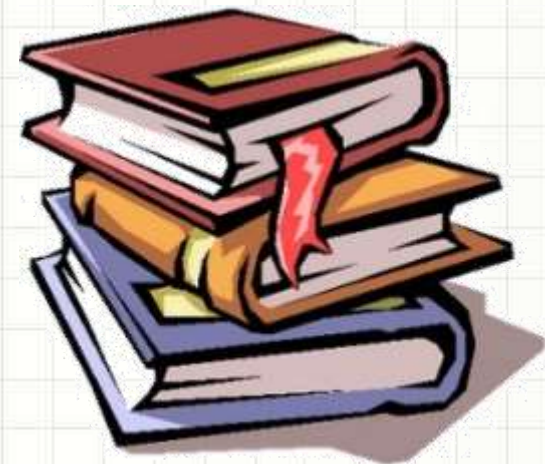
2018 - 1

Objetivos

- Entender o que são decisões compostas
- Compreender como implementar decisões compostas
- Capacitar para a criação de menus
- **Atividades Aula 9 – SAVA!**



Material de Estudo



Material

Acesso ao Material

Notas de Aula e
Apresentação

<http://www.caetano.eng.br/>
(Lógica de Programação para Eng. – Aula 9)

Material Didático

Lógica de Programação, págs 89 a 103.

Aula Online

Aula 6

Biblioteca Virtual

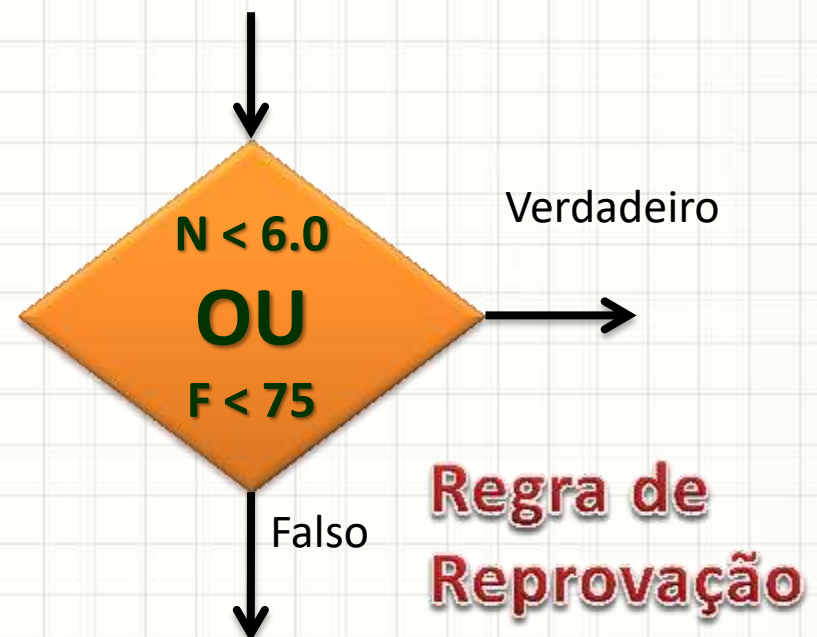
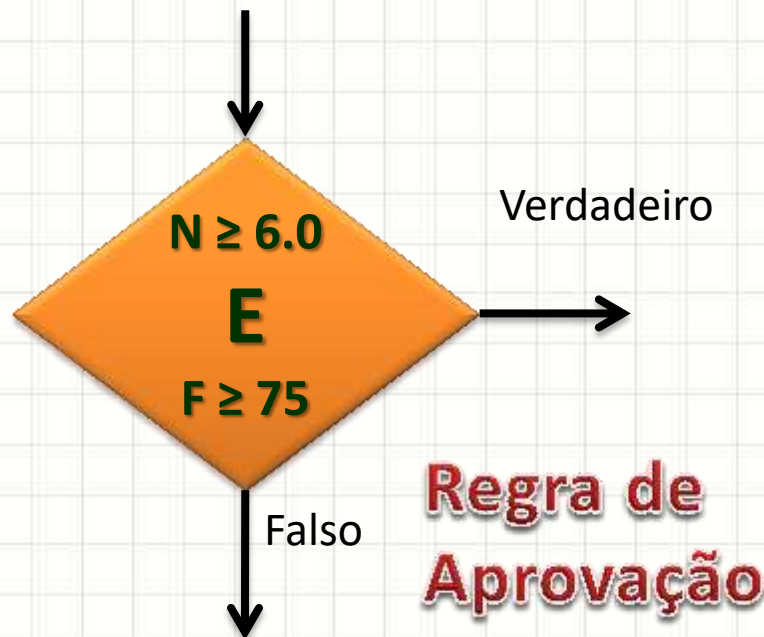
“Lógica de Programação – Fundamentos da
Programação de Computadores”, págs 50 a 92.



MOTIVAÇÃO

O que são decisões compostas?

- Decisão Simples: uma única regra
- Decisão Composta: múltiplas regras
- Como? Exemplo: aprovação/reprovação

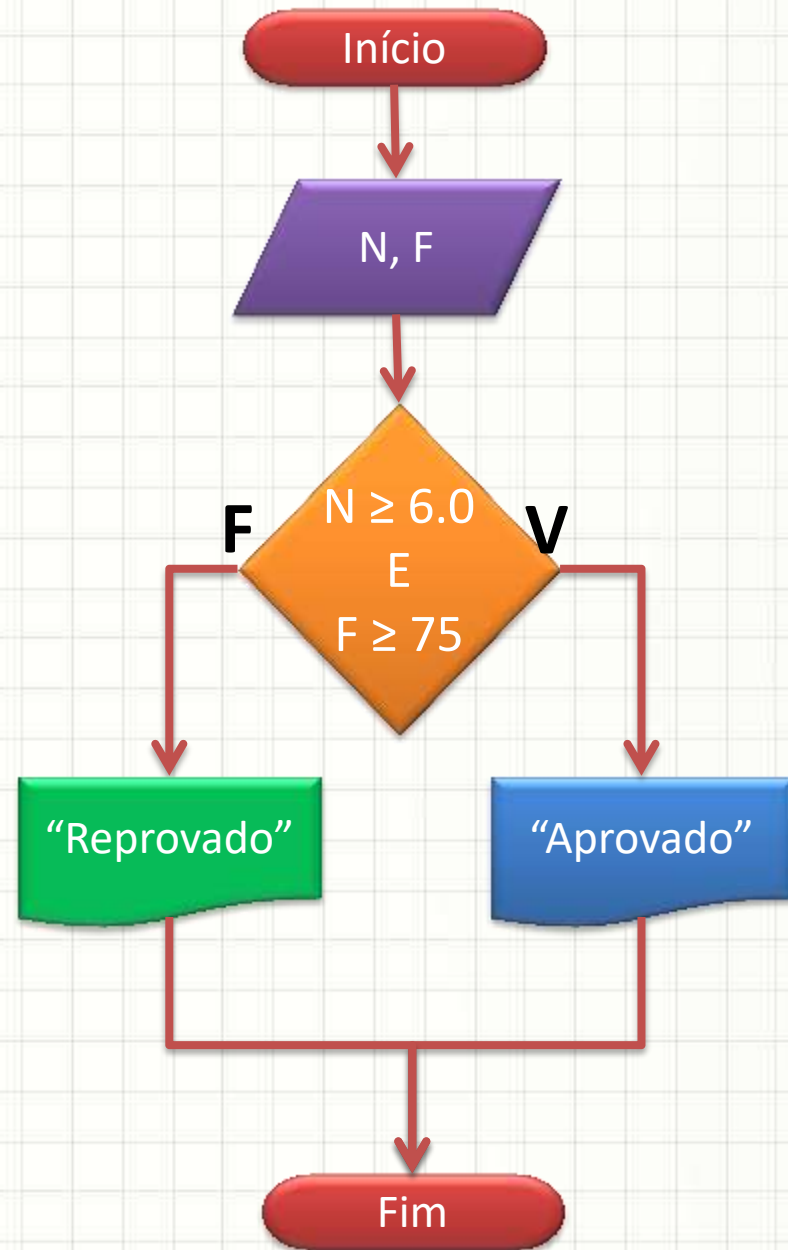




DECISÕES COMPOSTAS

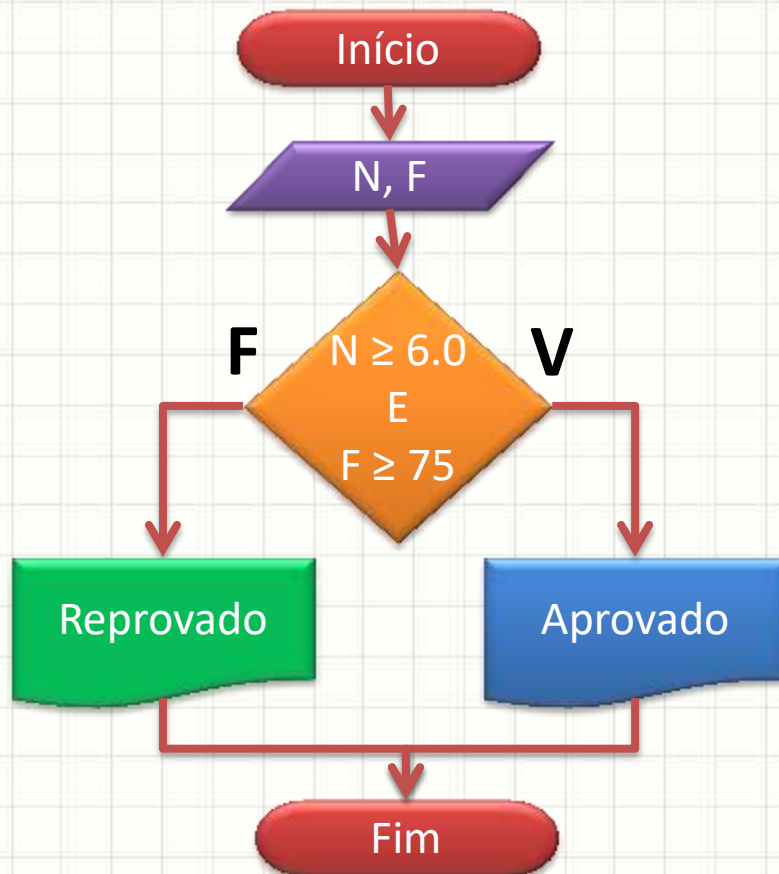
Decisão Composta

- Receba a **nota** de um aluno e a frequência dele e responda: **aprovado** caso a nota seja maior ou igual a 6,0 **E** a frequência seja maior ou igual a 75% e **reprovado** caso contrário



Decisão Composta

Receba a **nota** de um aluno e a frequência dele e responda: **aprovado** caso a nota seja maior ou igual a 6,0 e a frequência seja maior ou igual a 75% e **reprovado** caso contrário



```
#include <iostream>
using namespace std;
main()
{
    float N, F;
    cout << "Digite a nota: ";
    cin >> N;
    cout << "Digite a freq.: ";
    cin >> F;
    if ( N >= 6.0 && F >= 75 )
        cout << "Aprovado";
    else
        cout << "Reprovado";
}
```


Operadores Lógicos

- O que vem a ser **&&** ?

Operador	Símbolo	Exemplo	Significado
OU	 	$x == 2 \text{ } x == 9$	Verdadeiro se $x == 2$ OU $x == 9$
E	&&	$x > 2 \text{ \&\& } x < 9$	Verdadeiro se $x > 2$ E $x < 9$ (testa se $2 < x < 9$)
NÃO	!	$!(x == 2)$	Verdadeiro se x NÃO for igual a 2

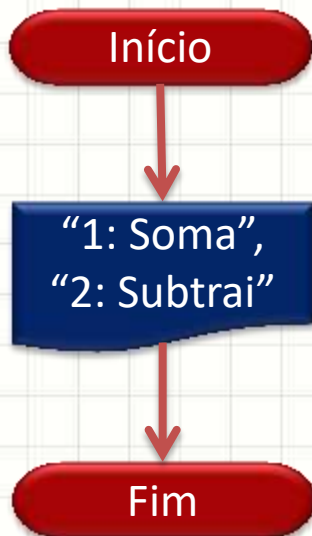
Operadores Lógicos

- O que vem a ser **&&**, **||** e **!**?

Operador		Símbolo	Exemplo		Significado	
!"@		#	\$	%	& *	
1 1		2	3	4	5	7 8
Q		W	E	R	T	Y U I O P
Tab		/	?	€		{ [Enter
Caps Lock		A	S	D	F	G H J K L Ç
Shift			Z	X	C	V B N M < > : ?
Ctrl		Win Key	Alt			Alt Gr Win Key Menu Ctrl

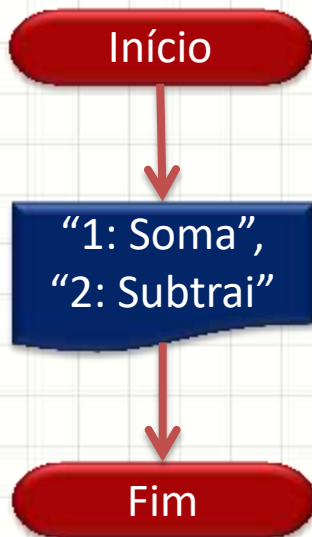
Exercício

- Faça um programa que apresente um **menu com duas opções**: 1- Soma; 2- Subtrai.



Exercício

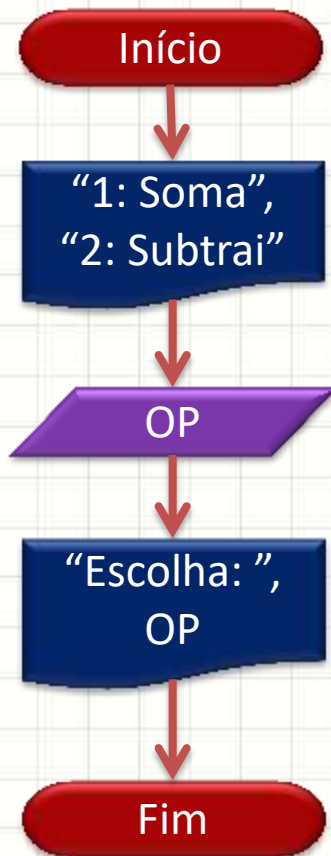
- Faça um programa que apresente um **menu com duas opções**: 1- Soma; 2- Subtrai.



```
#include <iostream>
using namespace std;
main() // Menu
{
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
}
```

Exercício

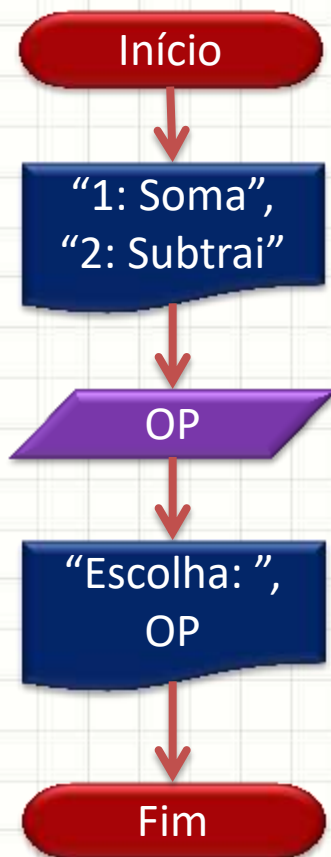
- Modifique o programa para que o usuário possa escolher uma opção e mostre a opção escolhida



```
#include <iostream>
using namespace std;
main() // Menu
{
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
}
```


Exercício

- Modifique o programa para que o usuário possa escolher uma opção e mostre a opção escolhida



```
#include <iostream>
using namespace std;
main() // Menu
```

```
{
```

```
    int OP;
```

```
    cout << "1: Soma" << endl;
```

```
    cout << "2: Subtrai" << endl;
```

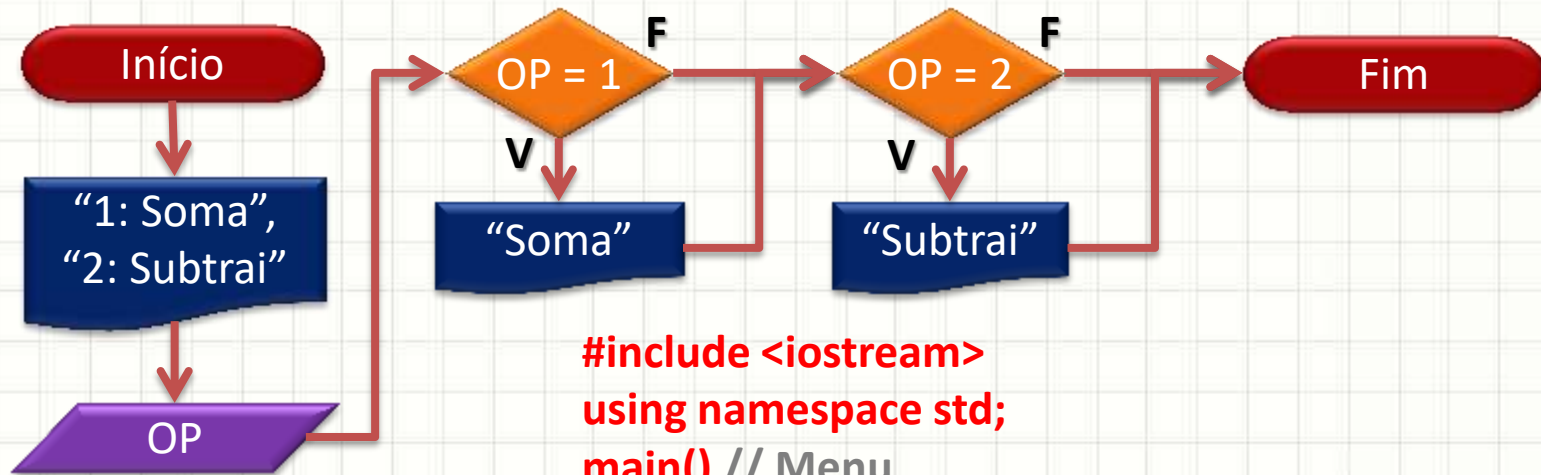
```
    cin >> OP;
```

```
    cout << "Escolha: " << OP;
```

```
}
```

Exercício

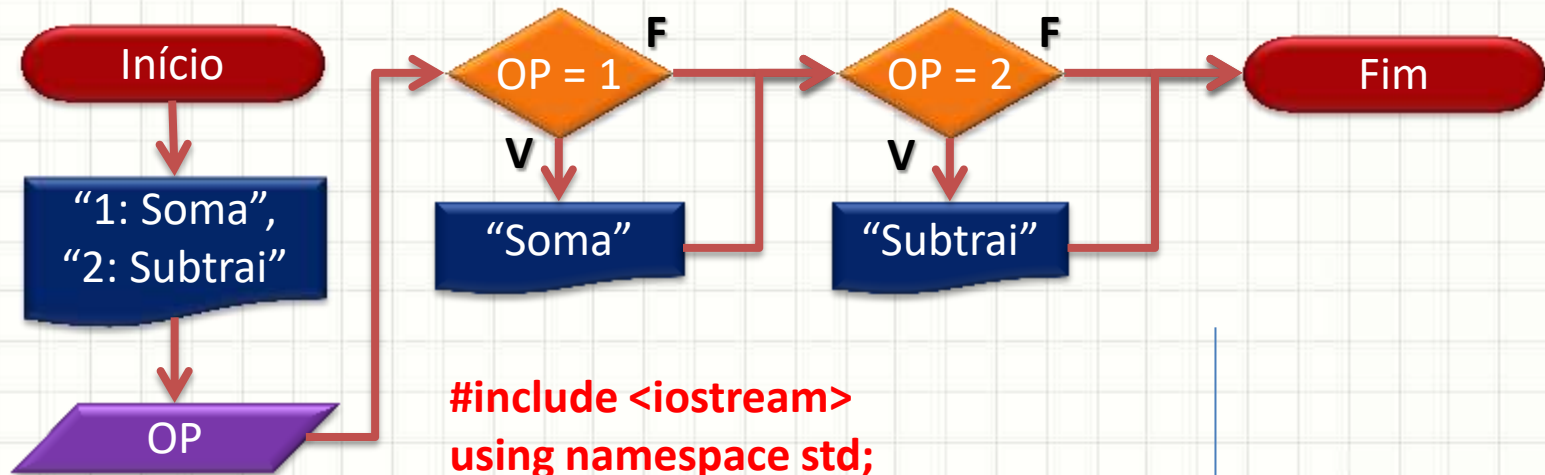
- Modifique o programa para que imprima “Soma” ou “Subtrai” de acordo com a opção feita



```
#include <iostream>
using namespace std;
main() // Menu
{
    int OP;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
    cout << "Escolha: " << OP;
}
```

Exercício

- Modifique o programa para que imprima “Soma” ou “Subtrai” de acordo com a opção feita

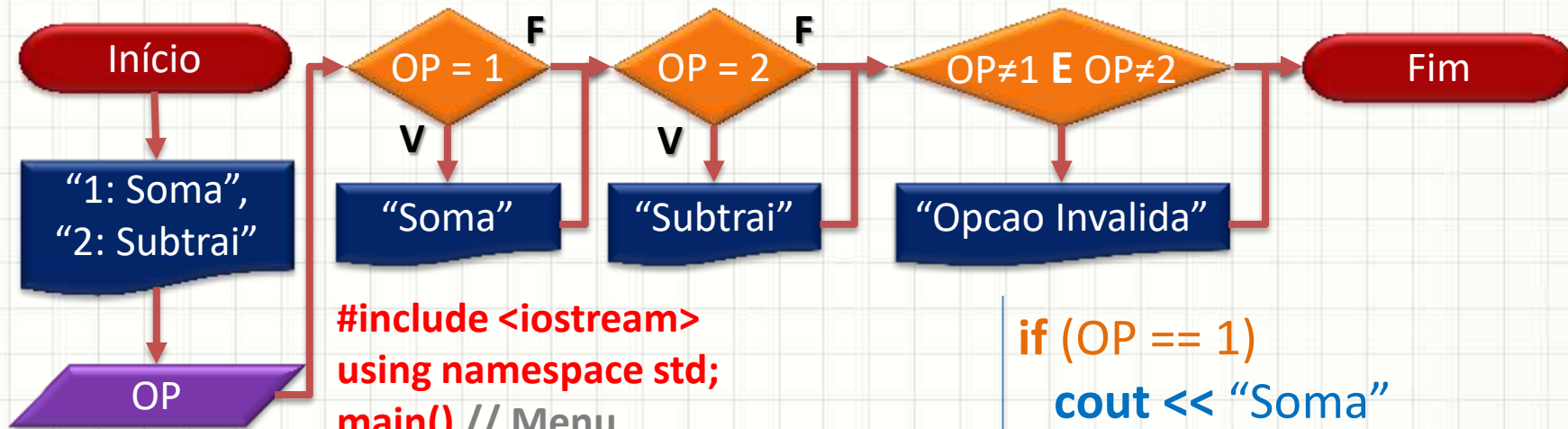


```
#include <iostream>
using namespace std;
main() // Menu
{
    int OP;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

```
if (OP == 1)
    cout << "Soma"
if (OP == 2)
    cout << "Subtrai";
}
```

Exercício

- Modifique para que imprima "Opcao Invalida" caso o usuário não escolha 1 ou 2



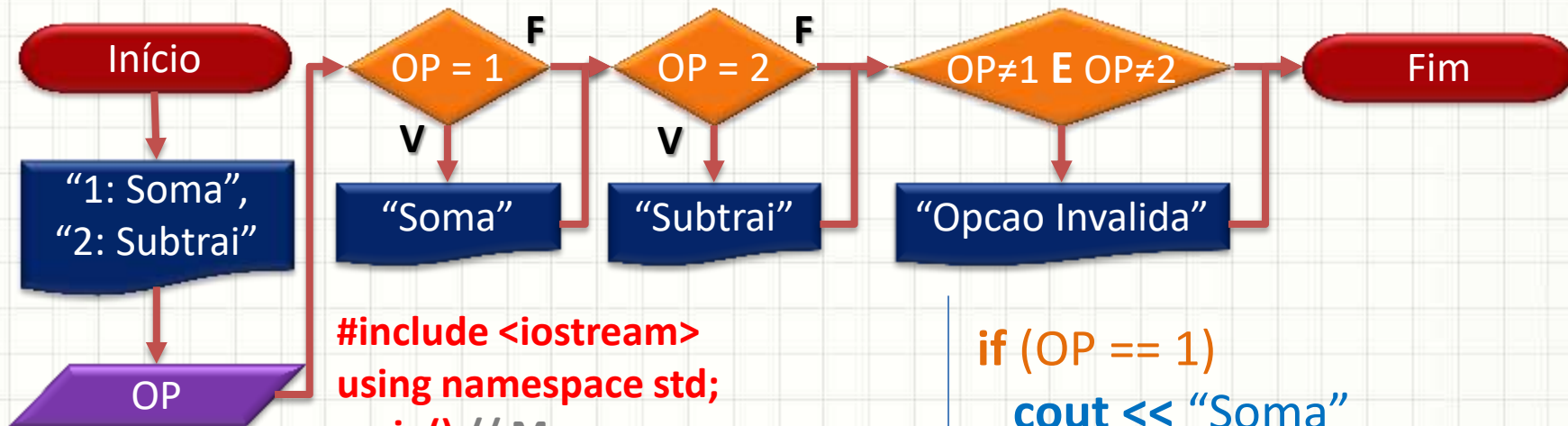
```
#include <iostream>
using namespace std;
main() // Menu
{
```

```
    int OP;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

```
    if (OP == 1)
        cout << "Soma"
    if (OP == 2)
        cout << "Subtrai";
}
```

Exercício

- Modifique para que imprima "Opcao Invalida" caso o usuário não escolha 1 ou 2



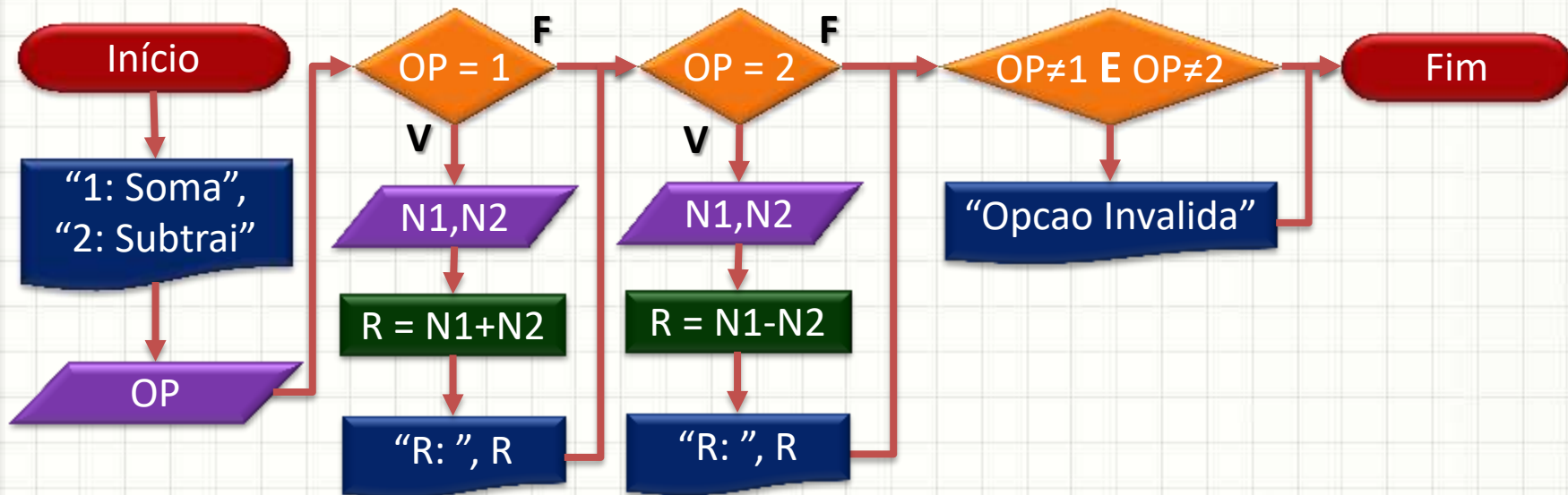
```
#include <iostream>
using namespace std;
main() // Menu
{
```

```
    int OP;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

```
    if (OP == 1)
        cout << "Soma"
    if (OP == 2)
        cout << "Subtrai";
    if (OP != 1 && OP != 2)
        cout << "Opcao Invalida";
}
```


Acompanhe o Professor

- Modifique o programa para “fazer o serviço”!



Exercício – Solução

```
#include <iostream>
using namespace std;
main() // Menu
{
    int OP, N1, N2;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
    if (OP == 1)
    {
        cout << "Digite N1: ";
        cin >> N1;
        cout << "Digite N2: ";
        cin >> N2;
        R = N1 + N2;
        cout << "R: " << R;
    }
}
```

```
if (OP == 2)
{
    cout << "Digite N1: ";
    cin >> N1;
    cout << "Digite N2: ";
    cin >> N2;
    R = N1 - N2;
    cout << "R: " << R;
}
if (OP != 1 && OP != 2)
    cout << "Opcao Invalida";
}
```

Exercício – Outra Solução

```
#include <iostream>
using namespace std;
main() // Menu
{
    int OP, N1, N2;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
    if (OP == 1)
    {
        cout << "Digite N1: ";
        cin >> N1;
        cout << "Digite N2: ";
        cin >> N2;
        R = N1 + N2;
        cout << "R: " << R;
    }
}
```

```
else if (OP == 2)
{
    cout << "Digite N1: ";
    cin >> N1;
    cout << "Digite N2: ";
    cin >> N2;
    R = N1 - N2;
    cout << "R: " << R;
}
else
    cout << "Opcao Invalida";
}
```

Encadeamento de Ifs – Compare!

```
if ( Prop1 )  
{  
    // ...  
}  
if ( Prop2 )  
{  
    // ...  
}  
if ( !Prop1 && !Prop2 )  
{  
    // ...  
}
```

```
if ( Prop1 )  
{  
    // ...  
}  
else if ( Prop2 )  
{  
    // ...  
}  
else  
{  
    // ...  
}
```

Qual a
Diferença?



ESTRUTURA DE ESCOLHA

Estrutura de Escolha

- É muito comum precisar criar estruturas como a indicada anteriormente...
 - Ler opções do usuário
 - Atuar em uma tecla de equipamento
 - Agir de acordo com o resultado de um processo
- Por isso existe uma estrutura **especial** para este tipo de menu!

Estrutura de Escolha

```
switch ( variável )
```

```
{
```

```
    case 1:
```

```
        Código caso variável == 1
```

```
        break;
```

```
    case 2:
```

```
        Código caso variável == 2
```

```
        break;
```

```
    default:
```

```
        Código para todos os outros casos
```

```
}
```

Estrutura de Escolha: Ex.: Menu

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

```
    int OP;
```

```
    cout << "Menu: " << endl;
```

```
    cout << "1: Opção 1" << endl;
```

```
    cout << "2: Opção 2" << endl;
```

```
    cout << "Digite uma Opção: ";
```

```
    cin >> OP;
```

```
    switch (OP)
```

```
    {
```

```
        case 1:
```

```
            cout << "Opção 1";
```

```
            break;
```

```
        case 2:
```

```
            cout << "Opção 2";
```

```
            break;
```

```
        default:
```

```
            cout << "Inválido";
```

```
    }
```

```
}
```

Exercício: Converta para Switch

```
#include <iostream>
using namespace std;
main() // Menu
{
    int OP, N1, N2;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
    if (OP == 1)
    {
        cout << "Digite N1: ";
        cin >> N1;
        cout << "Digite N2: ";
        cin >> N2;
        R = N1 + N2;
        cout << "R: " << R;
    }
}
```

```
if (OP == 2)
{
    cout << "Digite N1: ";
    cin >> N1;
    cout << "Digite N2: ";
    cin >> N2;
    R = N1 - N2;
    cout << "R: " << R;
}
if (OP != 1 && OP != 2)
    cout << "Opcao Invalida";
}
```

Exercício: Solução

```
#include <iostream>
using namespace std;
main() // Menu
{
```

```
    int OP, N1, N2;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

```
    switch (OP)
```

```
    {
```

```
        case 1:
```

```
            cout << "Digite N1: ";
            cin >> N1;
            cout << "Digite N2: ";
            cin >> N2;
            R = N1 + N2;
            cout << "R: " << R;
            break;
```

```
        case 2:
```

```
            cout << "Digite N1: ";
            cin >> N1;
            cout << "Digite N2: ";
            cin >> N2;
```

```
            R = N1 - N2;
```

```
            cout << "R: " << R;
```

```
            break;
```

```
        default:
```

```
            cout << "Opcao Invalida";
```

```
        }
```

```
    }
```


Exercício: Solução

```
#include <iostream>
using namespace std;
main() // Menu
{
    int OP, N1, N2;
    cout << "1: Soma" << endl;
    cout << "2: Sub" << endl;
    cin >> OP;
    switch (OP)
    {
        case 1:
            cout << "Digite N1: ";
            cin >> N1;
            cout << "Digite N2: ";
            cin >> N2;
            R = N1 + N2;
            cout << "R: " << R;
            break;
```

case 2:

cout << "Digite N1: ";

cin >> N1;

cout << "Digite N2: ";

R;

cout << "Opcao Invalida";

}

}

ATENÇÃO: SWITCH só funciona com variáveis INTEIRAS



ATIVIDADE

Exercício

- Modifique o programa da soma e subtração para que também tenha opções de **multiplicação** e **divisão**.
- ATENÇÃO: verifique **TUDO** que for necessário!
 - Pode dividir por zero?



CONCLUSÕES

Resumo

- Decisões Compostas: diversos critérios
- Linguagem C/C++: permite decisões compostas
- **switch~case**: facilita algumas construções
- **TAREFA: Atividade Aula 9!**

-
- Decisões são importantes... mas será que não podemos decidir repetir algo?
 - Como decidir que algo deve ser repetido?



PERGUNTAS?