



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

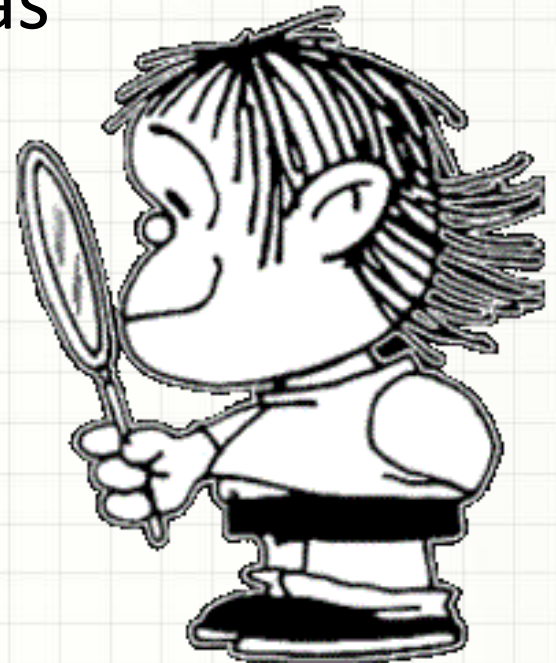
ESTRUTURA COM DECISÃO MÚLTIPLA

Prof. Dr. Daniel Caetano

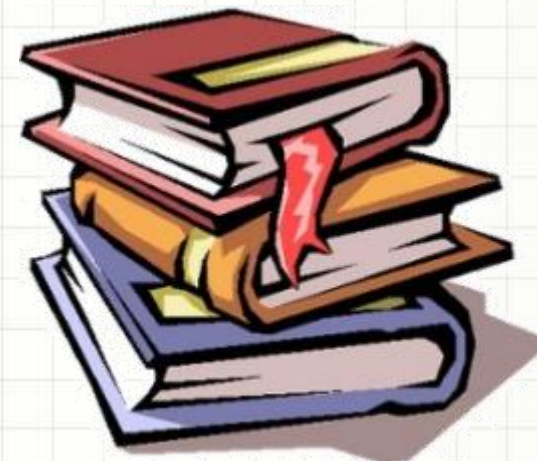
2013 - 1

Objetivos

- Entender o que são decisões múltiplas
- Compreender como implementar decisões múltiplas
- Capacitar para a criação de menus
- **PARA CASA**
 - Lista Aula 10!



Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>
(Aula 10)

Apresentação

<http://www.caetano.eng.br/>
(Aula 10) – PARCIAL / COMPLETO

Material Didático

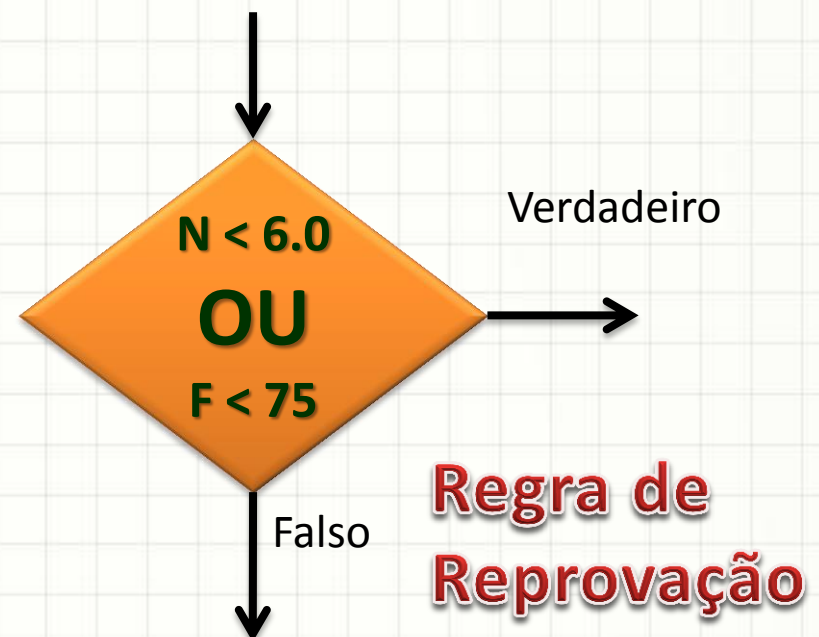
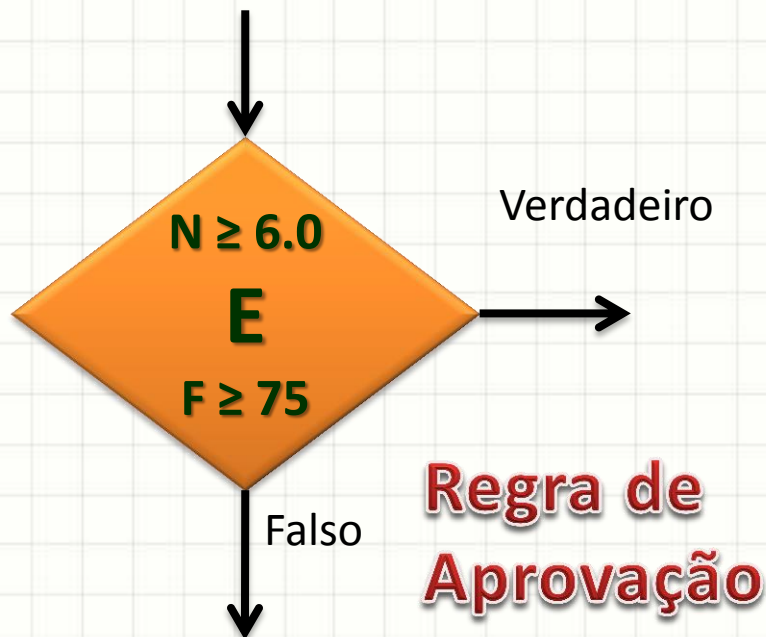
Fundamentos da Programação de Computadores –
Parte 1 – Páginas 50 a 92.



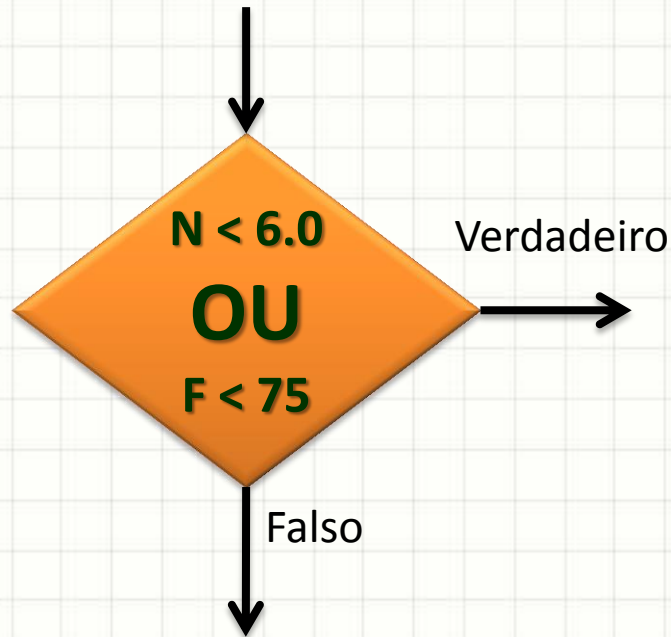
MOTIVAÇÃO

O que são decisões múltiplas?

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



O que são decisões múltiplas?



**Regra de
Reprovação**

```
if ( NOTA < 6.0)
```

```
{
```

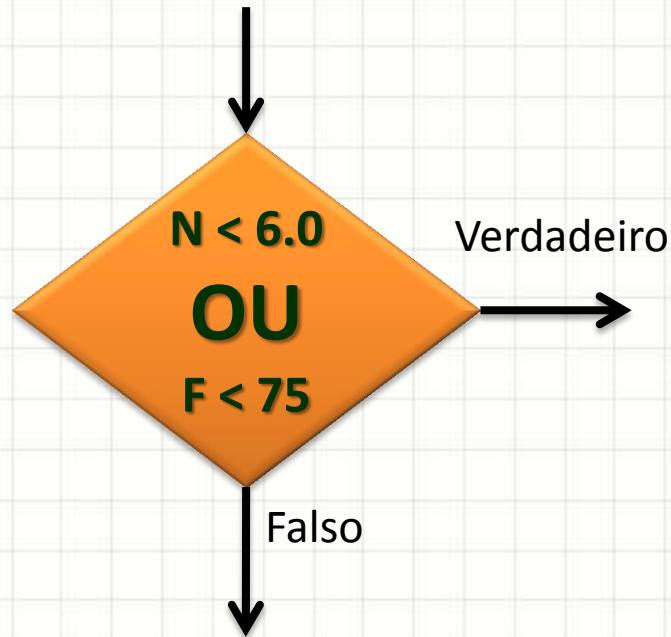
```
}
```

```
if ( FREQ < 75)
```

```
{
```

```
}
```

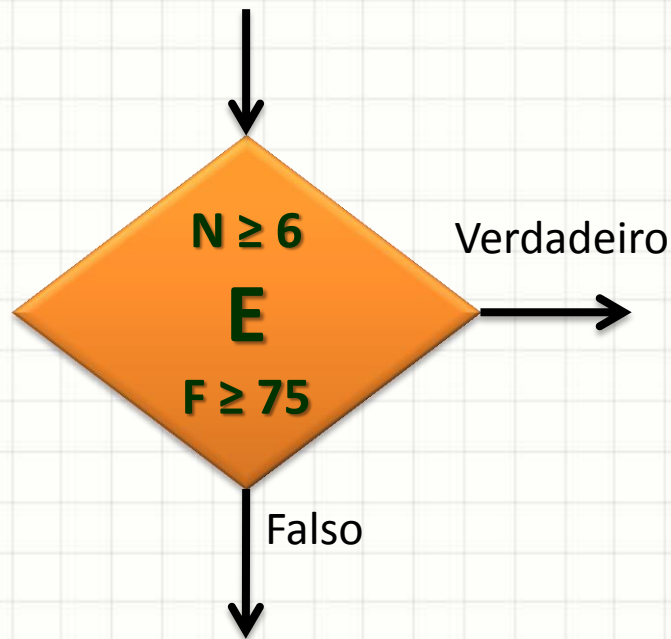
O que são decisões múltiplas?



**Regra de
Reprovação**

```
if ( NOTA < 6.0)
{
    cout << "V: Reprovado!";
}
if ( FREQ < 75)
{
    cout << "V: Reprovado!";
}
```

O que são decisões múltiplas?



**Regra de
Aprovação**

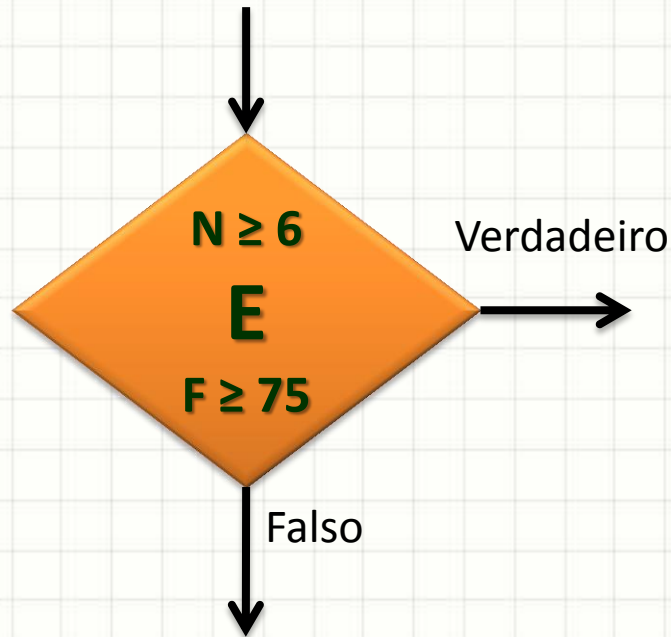
```
if ( NOTA >= 6.0)
```

```
{
```



```
}
```

O que são decisões múltiplas?



**Regra de
Aprovação**

```
if ( NOTA >= 6.0)
```

```
{
```

```
if ( FREQ >= 75)
```

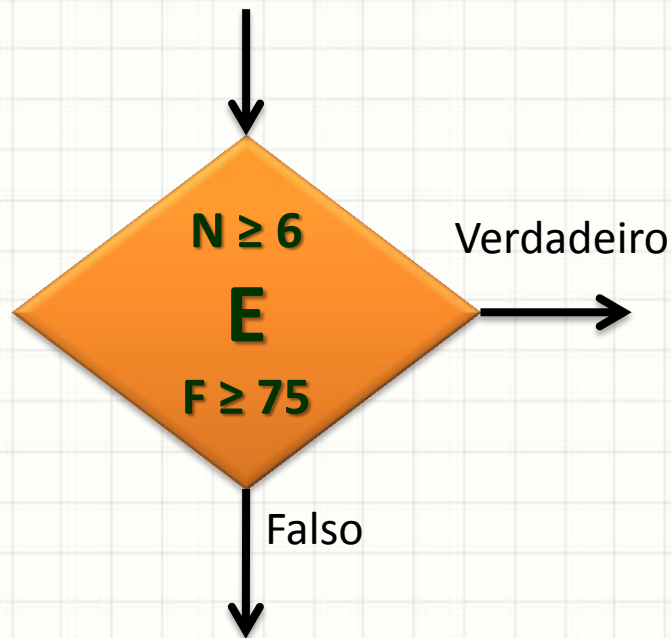
```
{
```



```
}
```

```
}
```

O que são decisões múltiplas?



**Regra de
Aprovação**

```
if ( NOTA >= 6.0)
```

```
{
```

```
if ( FREQ >= 75)
```

```
{
```

```
cout << "V: Aprovado!";
```

```
}
```

```
}
```

O que são decisões múltiplas?

```
if ( NOTA > 10 )  
{
```

```
    "Reprovado!";
```

**Não pode ser
tão complicado!**

```
    Falso
```

Falso

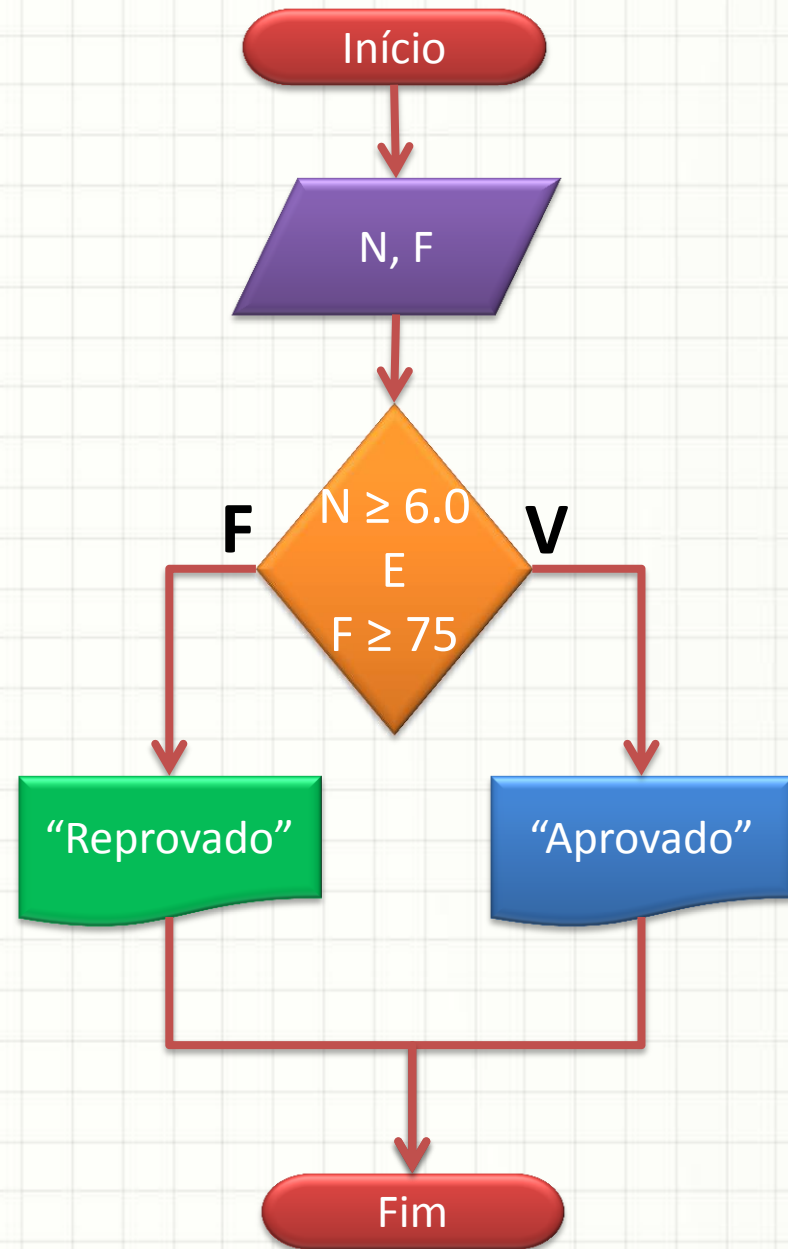
**Regra de
Aprovação**



DECISÕES MÚLTIPLAS

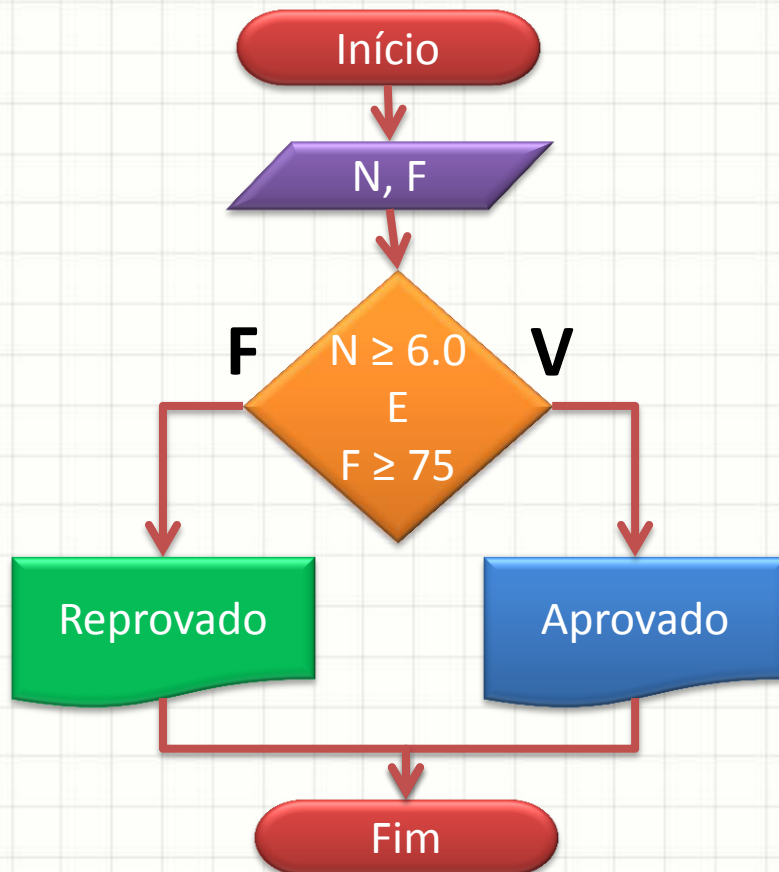
Decisão Múltipla

- 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 Múltipla

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

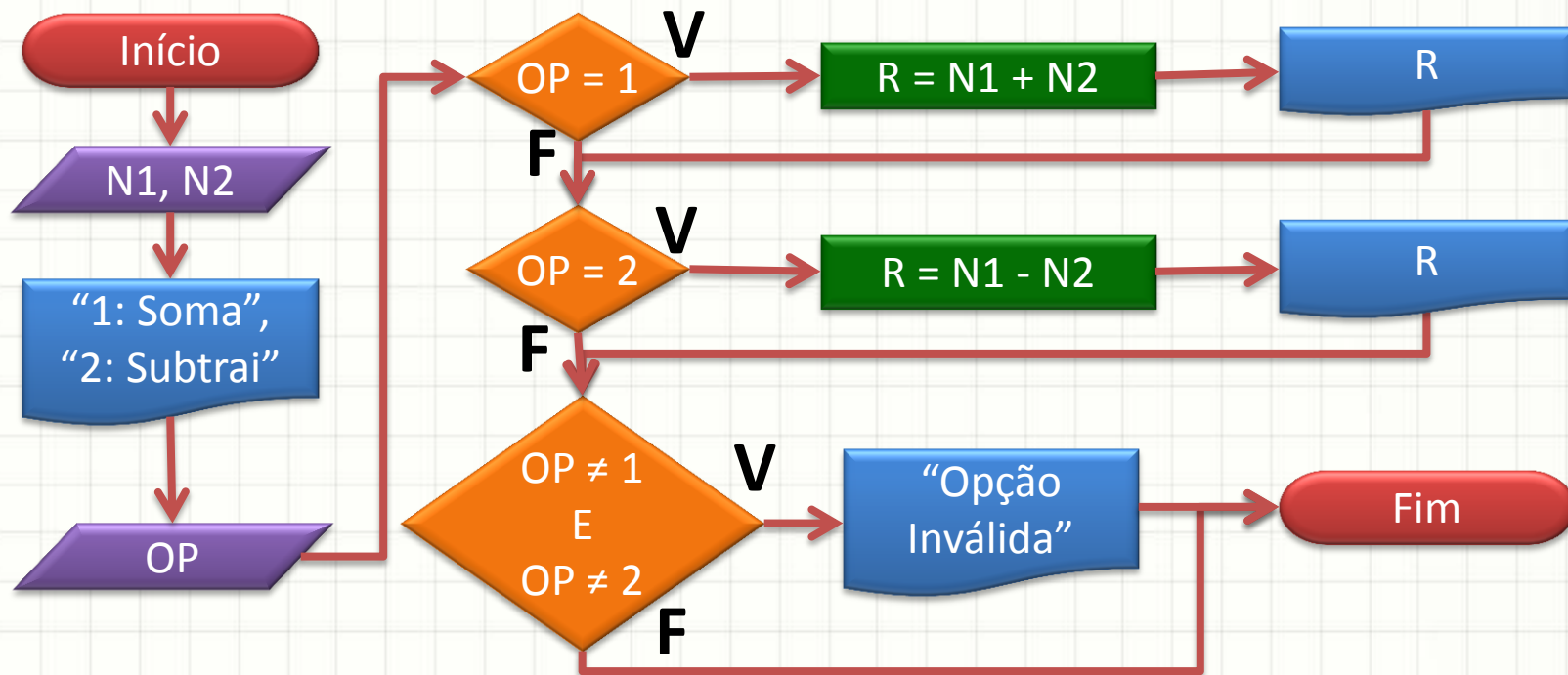
Operadores Lógicos

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

Operador		Símbolo	Exemplo		Significado	
!"@		#	\$	%	& *	
1 1 2		3 3	4 £	5 ¢	7 8	
Tab		Q	W	E	R	T
Caps Lock		A	S	D	F	G
Shift			Z	X	C	V
Ctrl		Win Key	Alt	Alt Gr		Win Key
						Menu
						Ctrl

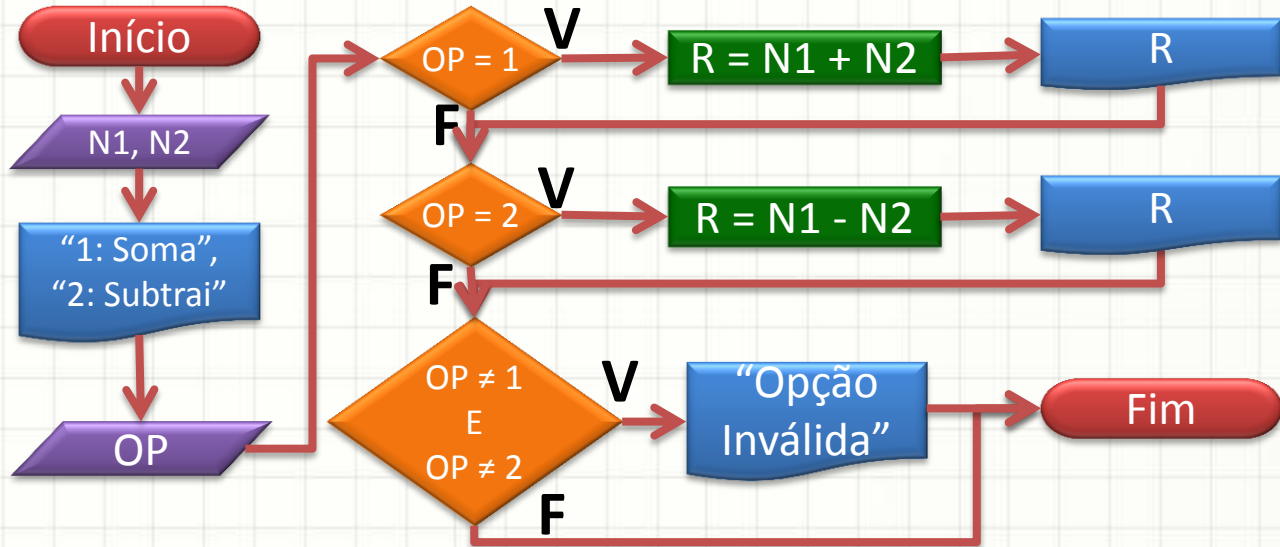
Exercício

- Faça um programa que **lê dois números** e depois apresente um **menu com duas opções**: 1- Soma; 2- Subtrai e age de acordo com a opção escolhida. Se o usuário escolher uma **opção inválida**, deve apresentar **mensagem de erro**.



Exercício

- Solução



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

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

```
    if ( OP == 1 )
    {
        R = N1 + N2;
        cout << "Res: " << R;
    }
    if ( OP == 2 )
    {
        R = N1 - N2;
        cout << "Res: " << R;
    }
    if ( OP != 1 && OP != 2 )
        cout << "Opção Inválida!";
}
```

Exercício – Solução (zoom)

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

```
    float N1, N2, R;
```

```
    int OP;
```

```
    cout << "Digite N1: ";
```

```
    cin >> N1;
```

```
    cout << "Digite N2: ";
```

```
    cin >> N2;
```

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

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

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

```
    cin >> OP;
```

```
    if ( OP == 1 )
```

```
    {
```

```
        R = N1 + N2;
```

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

```
    }
```

```
    if ( OP == 2 )
```

```
    {
```

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

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

```
    }
```

```
    if ( OP != 1 && OP != 2 )
```

```
        cout << "Opção Inválida!";
```

```
    }
```

Exercício – Solução (Outra Forma)

```
#include <iostream>
using namespace std;
main()
{
    float N1, N2, R;
    int OP;

    cout << "Digite N1: ";
    cin >> N1;
    cout << "Digite N2: ";
    cin >> N2;
    cout << "Menu" << endl;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;

    if ( OP == 1 )
    {
        R = N1 + N2;
        cout << "Res: " << R;
    }
    else if ( OP == 2 )
    {
        R = N1 - N2;
        cout << "Res: " << R;
    }
    else
        cout << "Opção Inválida!";
}
```

Exercício – Solução (Outra Forma)

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

```
float N1, N2, R;
int OP;
```

**Encadeamento
de IFs**

```
cout << "Menu" << endl;
cin >> N2;
cout << "1: Soma" << endl;
cout << "2: Subtrai" << endl;
cin >> OP;
```

```
if ( OP == 1 )
{
    R = N1 + N2;
    cout << "Res: " << R;
}
else if ( OP == 2 )
{
    R = N1 - N2;
    cout << "Res: " << R;
}
else
    cout << "Opção Inválida!";
}
```

Exercício – Solução (Outra Forma)

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

Se $OP == 1...$

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



```
    if ( OP == 1 )
    {
        R = N1 + N2;
        cout << "Res: " << R;
    }
    else if ( OP == 2 )
    {
        R = N1 - N2;
        cout << "Res: " << R;
    }
    else
        cout << "Opção Inválida!";
}
```

Exercício – Solução (Outra Forma)

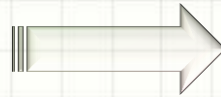
```
#include <iostream>
using namespace std;
main()
{
```

```
float N1, N2, R;
int OP;
```

```
cout << "Digite N1: ";
```

Se $OP \neq 1 \dots$

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

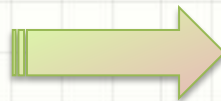


```
if ( OP == 1 )
```

```
{
```

```
    R = N1 + N2;
    cout << "Res: " << R;
```

```
}
```



```
else if ( OP == 2 )
```

```
{
```

```
    R = N1 - N2;
    cout << "Res: " << R;
```

```
}
```

```
else
```

```
    cout << "Opção Inválida!";
```

```
}
```

Exercício – Solução (Outra Forma)

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

```
float N1, N2, R;
int OP;
```

```
cout << "Digite N1: ";
```

Se $OP \neq 1 \dots$
...e $OP == 2$

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

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

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

```
cin >> OP;
```

```
if ( OP == 1 )
```

```
{
```

```
R = N1 + N2;
```

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

```
}
```

```
else if ( OP == 2 )
```

```
{
```

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

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

```
}
```

```
else
```

```
cout << "Opção Inválida!";
```

```
}
```

Exercício – Solução (Outra Forma)

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

```
float N1, N2, R;
int OP;
```

```
cout << "Digite N1: ";
```

Se $OP \neq 1 \dots$
...e $OP \neq 2$

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

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

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

```
cin >> OP;
```

```
if ( OP == 1 )
```

```
{
```

```
R = N1 + N2;
```

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

```
}
```

```
else if ( OP == 2 )
```

```
{
```

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

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

```
}
```

```
else
```

```
cout << "Opção Inválida!";
```

```
}
```



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()
```

```
{
    float N1, N2, R;
    int OP;
    cout << "Digite N1: ";
    cin >> N1;
    cout << "Digite N2: ";
    cin >> N2;
    cout << "Menu" << endl;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

```
    if ( OP == 1 )
    {
        R = N1 + N2;
        cout << "Res: " << R;
    }
    if ( OP == 2 )
    {
        R = N1 - N2;
        cout << "Res: " << R;
    }
    if ( OP != 1 && OP != 2 )
        cout << "Opção Inválida!";
}
```

Exercício: Solução

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

```
{
    float N1, N2, R;
    int OP;
    cout << "Digite N1: ";
    cin >> N1;
    cout << "Digite N2: ";
    cin >> N2;
    cout << "Menu" << endl;
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

```
    switch ( OP )
    {
        case 1:
            R = N1 + N2;
            cout << "Res: " << R;
            break;
        case 2:
            R = N1 - N2;
            cout << "Res: " << R;
            break;
        default:
            cout << "Opção Inválida!";
    }
}
```

Exercício: Solução

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

```
{
    float N1, N2;
    int OP;
    cout << "Digite o primeiro número: ";
    cin >> N1;
    cout << "Digite o segundo número: ";
    cin >> N2;
    cout << "Menu de opções: \n";
    cout << "1: Soma" << endl;
    cout << "2: Subtrai" << endl;
    cin >> OP;
```

ATENÇÃO: SWITCH só funciona com variáveis INTEIRAS

```
switch ( OP )
{
    case 1:
        R = N1 + N2;
        cout << R;
        break;
    case 2:
        R = N1 - N2;
        cout << R;
        break;
    default:
        cout << "Opção Inválida!";
        break;
}
```



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 são aquelas que atendem, simultaneamente, a vários requisitos
- As estruturas de decisão permitem que decisões compostas sejam tomadas
- Para facilitar alguns tipos de construção, existe a estrutura de escolha: **switch~case**
- Não deixe de praticar!
- **TAREFA!**
 - Lista Aula 10!

Próxima Aula



- Vamos exercitar com problemas reais?
 - Como podemos usar todos esses recursos para resolver problemas do dia-a-dia?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**