



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

ESTRUTURA COM DECISÃO MÚLTIPLA

Prof. Dr. Daniel Caetano

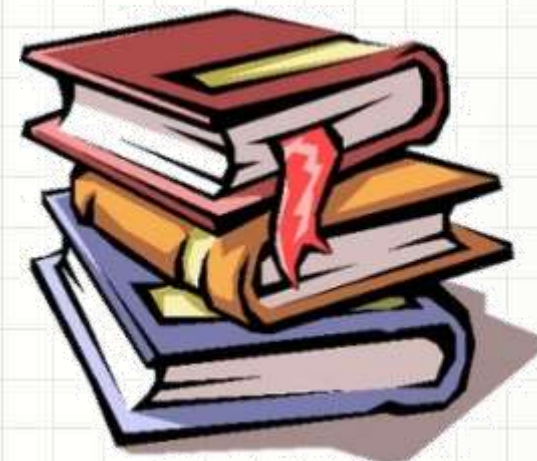
2012 - 2

Objetivos

- Entender o que são decisões múltiplas
- Compreender como implementar decisões múltiplas
- Capacitar para a criação de menus
- **PARA CASA**
 - Lista de Exercícios 2 está ONLINE!



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

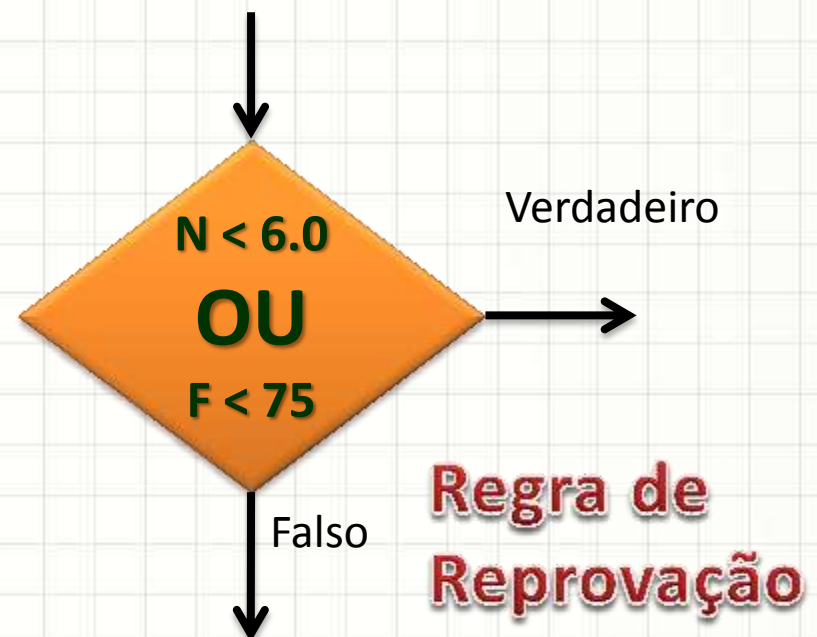
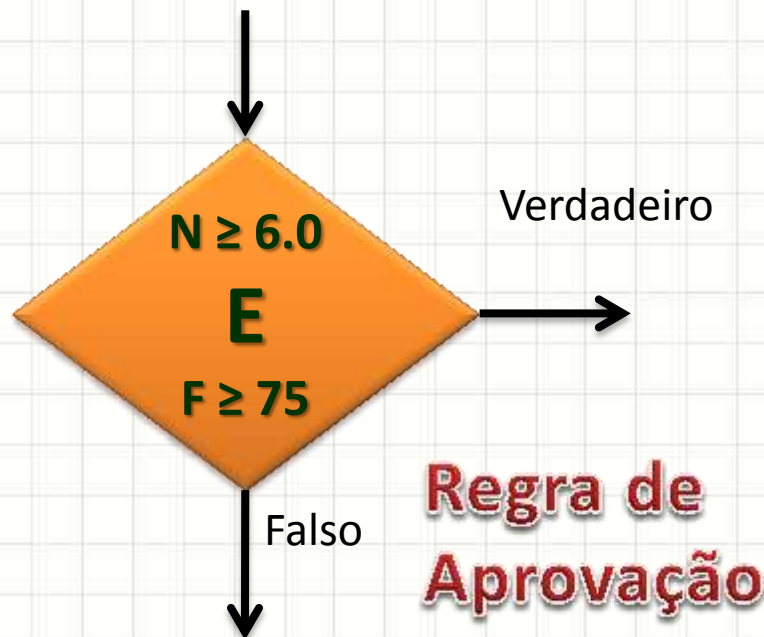
Fundamentos da Programação de Computadores –
Parte 1 – Páginas 50 a 92.



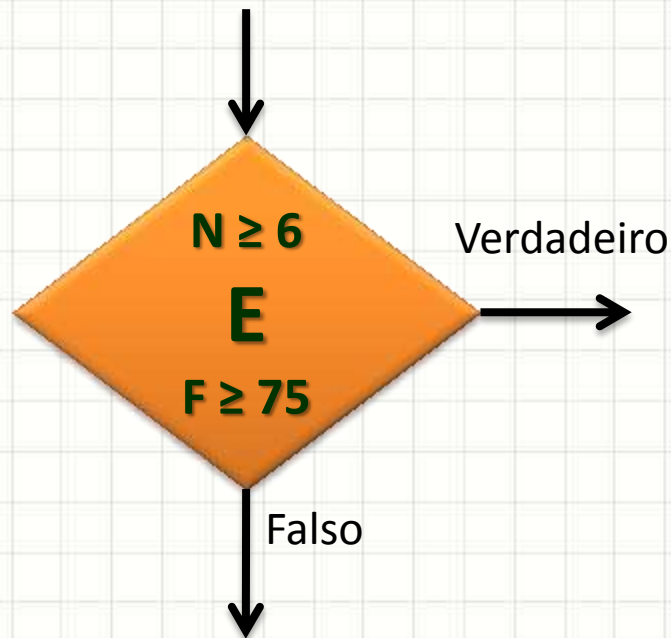
MOTIVAÇÃO

O que são decisões múltiplas?

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

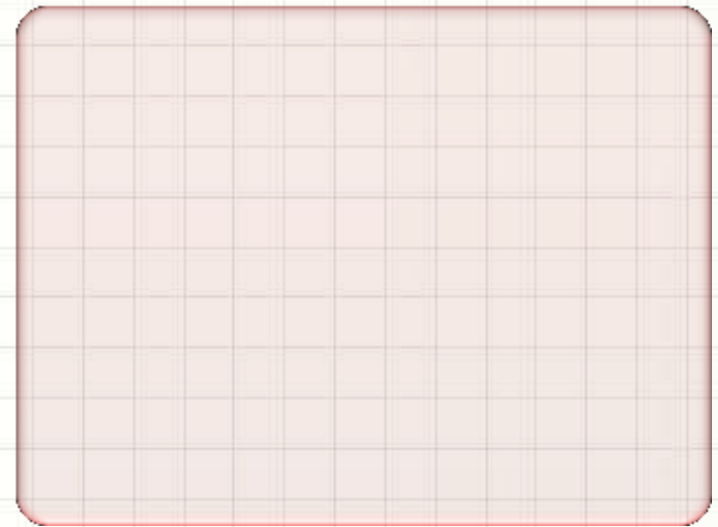


O que são decisões múltiplas?



**Regra de
Aprovação**

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



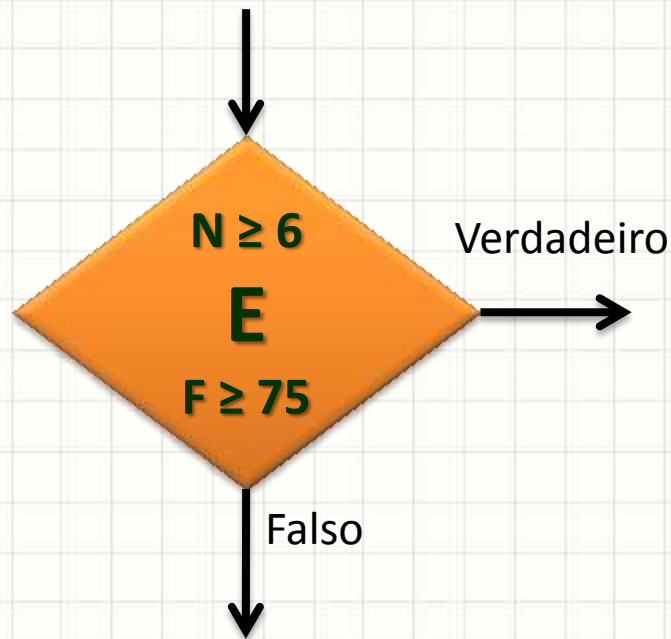
```
}
```

```
else {
```

```
    cout << "F: Reprovado!";
```

```
}
```

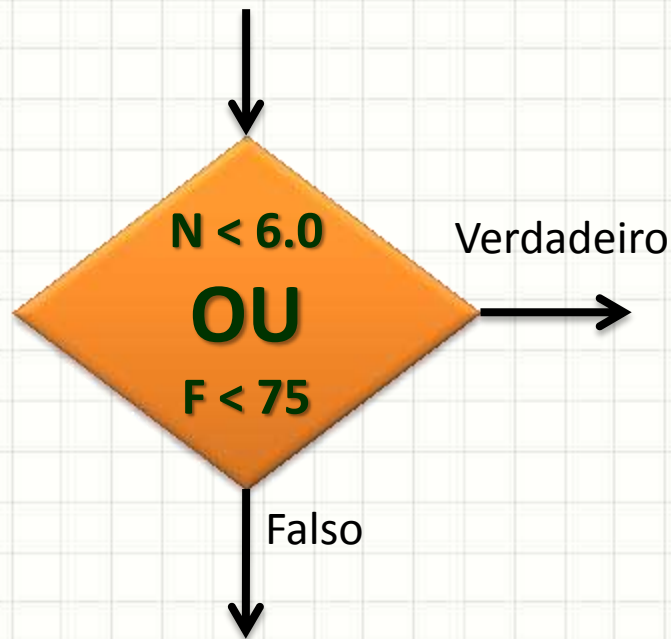
O que são decisões múltiplas?



**Regra de
Aprovação**

```
if ( NOTA >= 6.0) {  
    if ( FREQ >= 75) {  
        cout << "V: Aprovado!";  
    }  
    else {  
        cout << "F: Reprovado!";  
    }  
}  
else {  
    cout << "F: Reprovado";  
}
```

O que são decisões múltiplas?



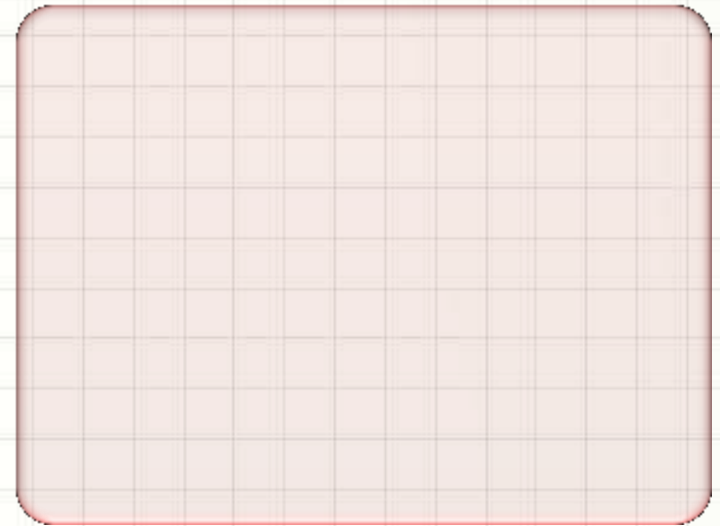
**Regra de
Reprovação**

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

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

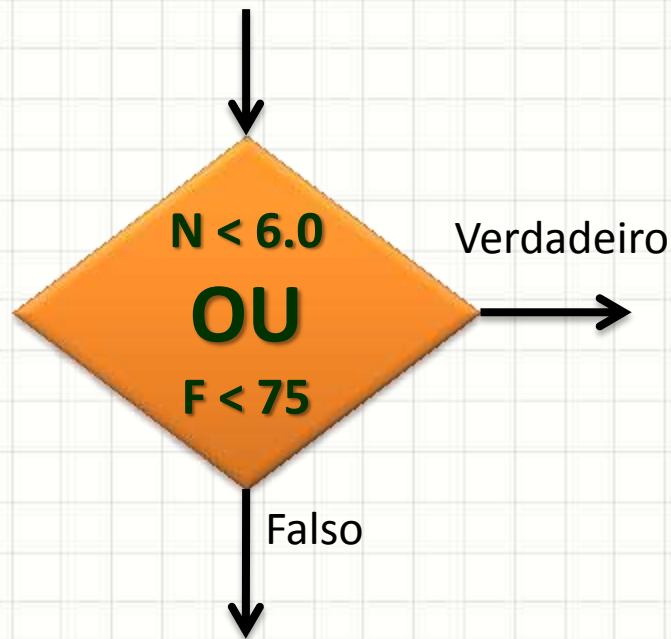
```
}
```

```
else {
```



```
}
```


O que são decisões múltiplas?



**Regra de
Reprovação**

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

O que são decisões múltiplas?

```
if ( NOTA < 5 ) {  
    cout << "V: Reprovado!";  
}
```

**Não pode ser
tão complicado!**

Falso
↓
**Regra de
Reprovação**

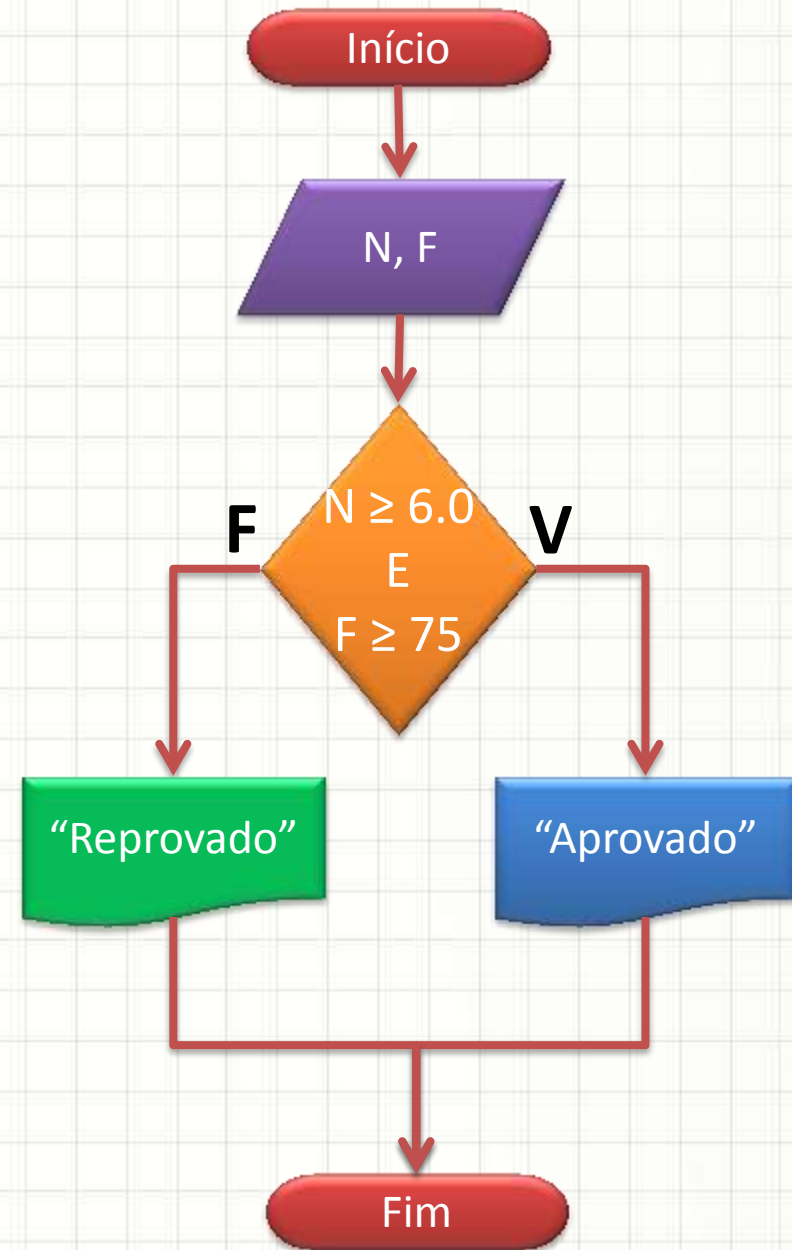
```
cout << "F: Aprovado!";  
}
```



DECISÕES MÚLTIPLAS

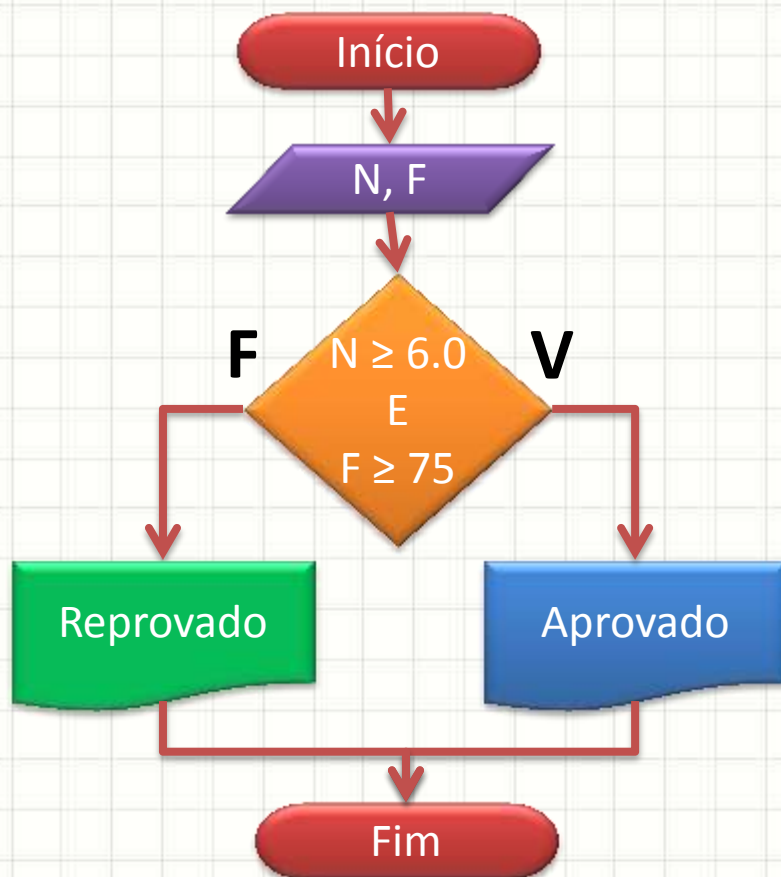
Decisão Múltipla

- 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;
int main(void) {
```

```
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 |

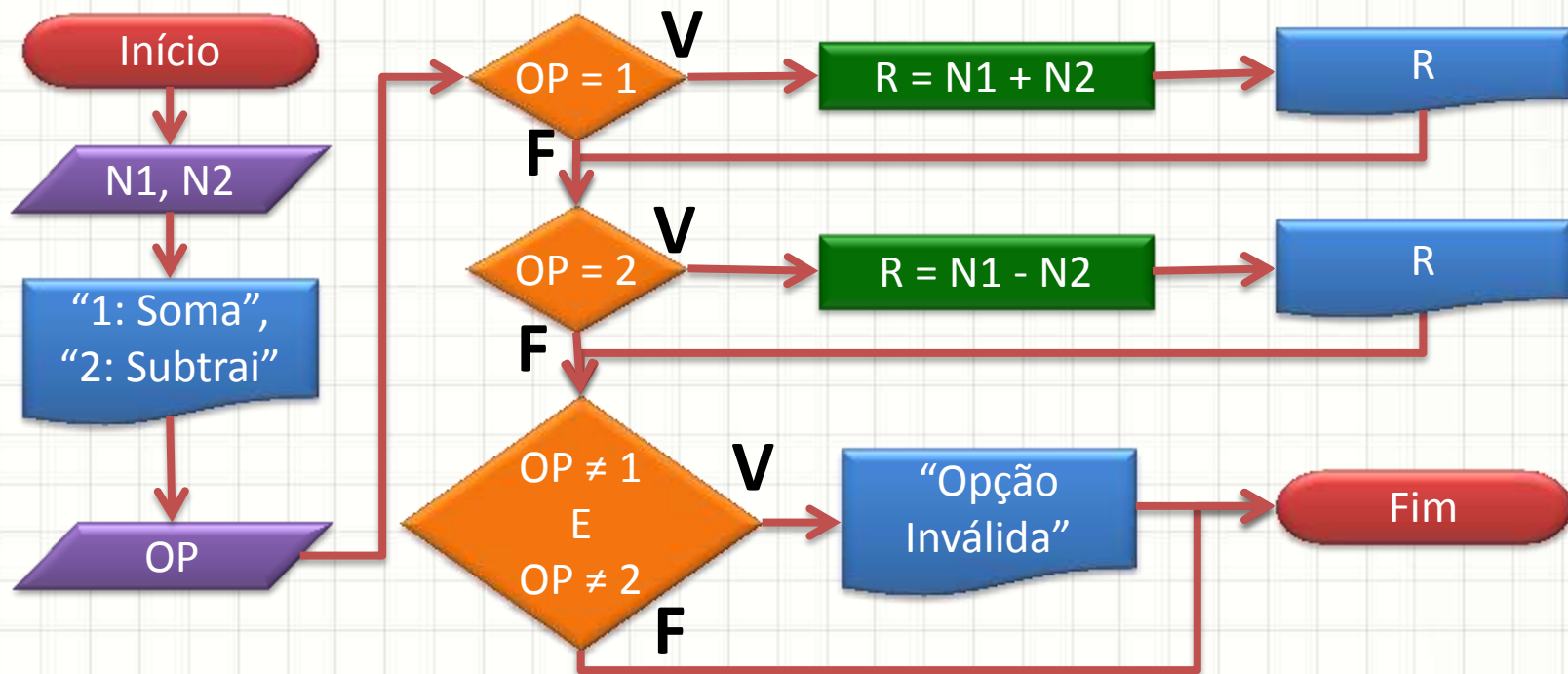
Operadores Lógicos

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

| Operador | | Símbolo | Exemplo | | Significado | |
|-----------|--|---------|---------|---|-------------|--------------------------|
| !"@ | | # | \$ | % | & * | |
| 1 1 | | 2 | 3 | 4 | 5 | 6 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 |

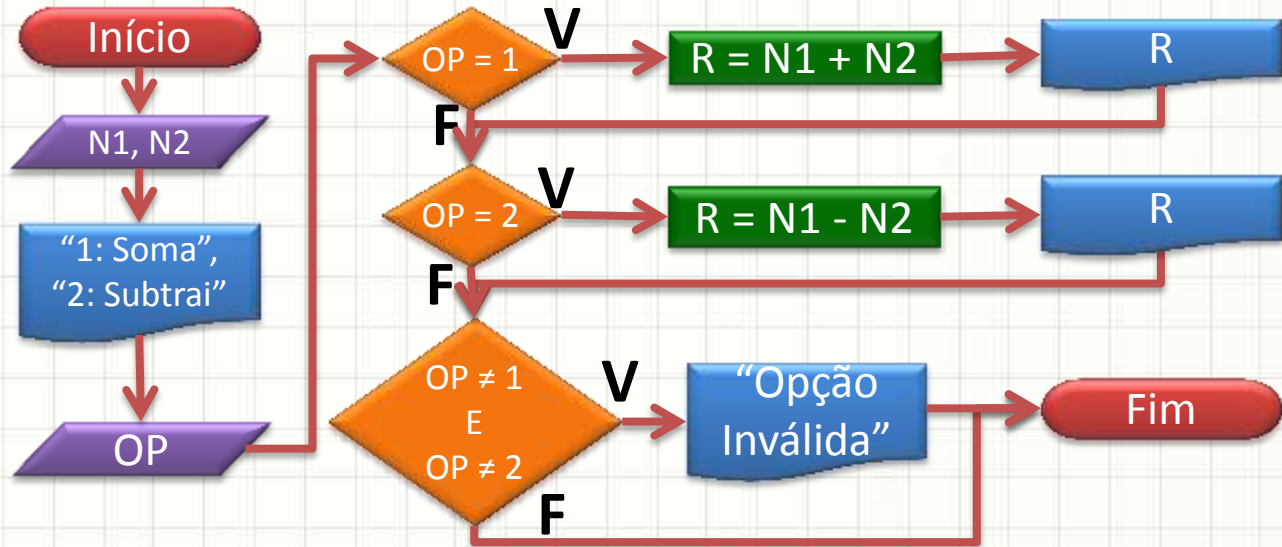
Exercício

- Faça um programa que **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;
int main(void) {
    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;
int main(void) {
    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;
int main(void) {
    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;
int main(void) {
    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;
```

**Encadeamento
de IFs**

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

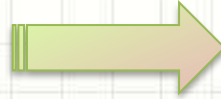
Exercício – Solução (Outra Forma)

```
#include <iostream>
```

```
using
```

```
namespace
```

Se $OP == 1...$



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

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

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

```
}
```

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

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

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

```
}
```

```
else {
```

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

```
}
```

```
}
```

```
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;
```

Exercício – Solução (Outra Forma)

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

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

Se $OP \neq 1...$

```
    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;
int main(void) {
```

Se OP == 1...

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

```
int OP;
```

Se OP != 1...

...e OP == 2

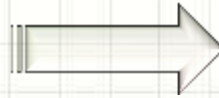
```
cin >> N2;
```

```
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;
int main(void) {
```

Se $OP == 1...$

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

```
int OP;
```

Se $OP \neq 1...$

...e $OP \neq 2$

```
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!";
```

```
}
```

```
}
```



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;
```

```
int main(void) {
```

```
    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;
int main(void) {
```

```
    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;
int main(void) {
    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;
int main(void) {
```

```
float N1, N2;
```

```
int OP;
```

```
cout << "Di
```

```
cin >> N1;
```

```
cout << "Di
```

```
cin >> N2;
```

```
cout << "Me
```

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

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

```
cin >> OP;
```

```
switch ( OP ) {
```

```
case 1:
```

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

```
cout << R;
```

```
cout << R;
```

```
cout << "Opção Inválida!";
```

```
}
```

```
}
```

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 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 de Exercícios 2!

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!**