



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

ESTRUTURA DE DECISÃO COMPOSTA

Prof. Dr. Daniel Caetano

2011 - 2

Visão Geral

1

- Decisão Composta

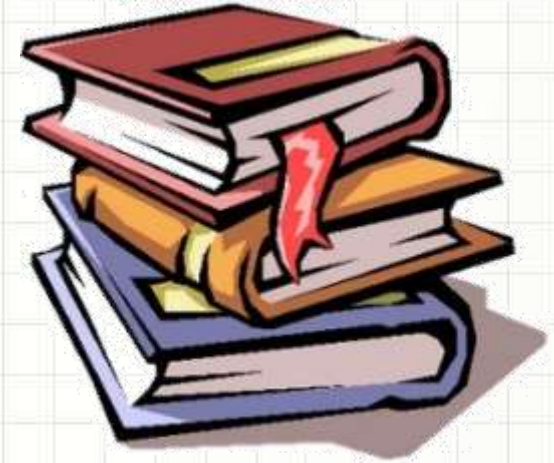
2

- Estrutura de Escolha

3

- Atividade

Material de Estudo



Material

Acesso ao Material

Notas de Aula

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

Apresentação

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

Material Didático

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

Objetivos

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

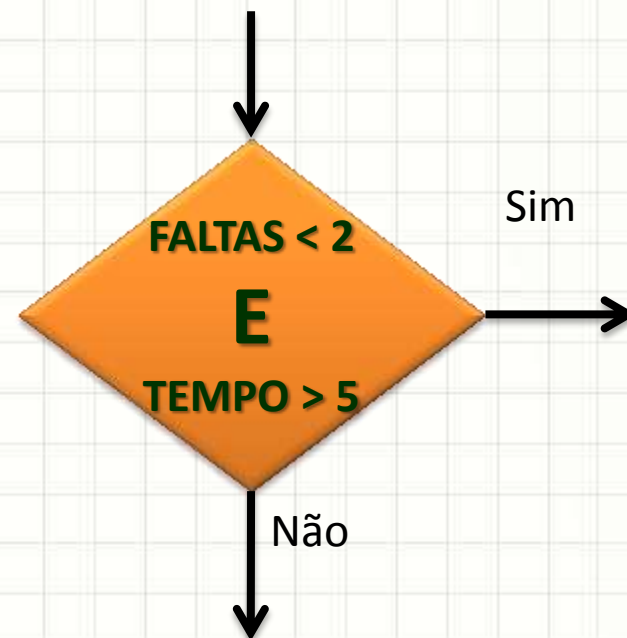
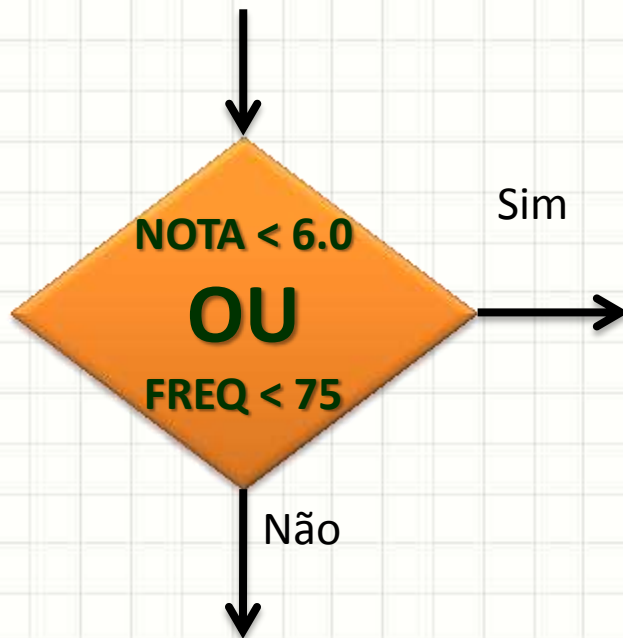




O QUE SÃO DECISÕES COMPOSTAS?

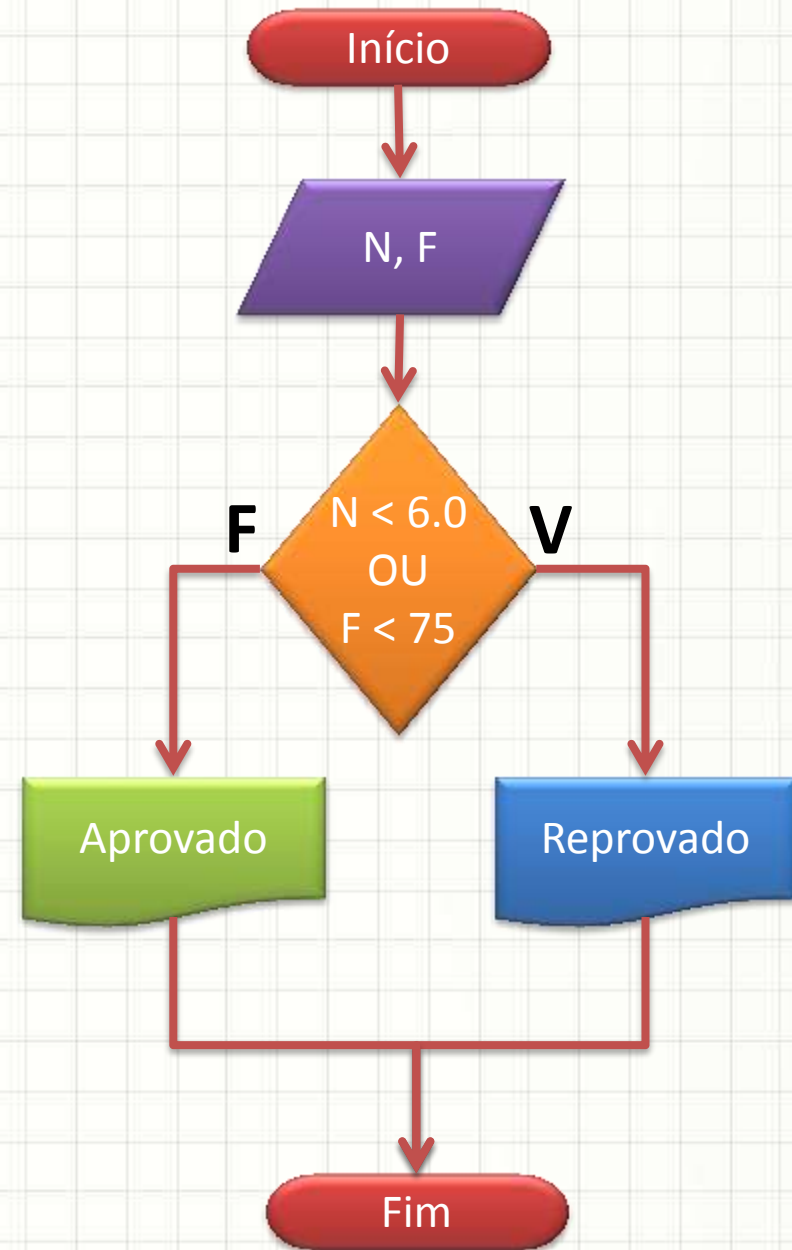
O que são decisões compostas?

- Decisão Simples: uma única comparação
- Decisão Composta: múltiplas regras
- Como?



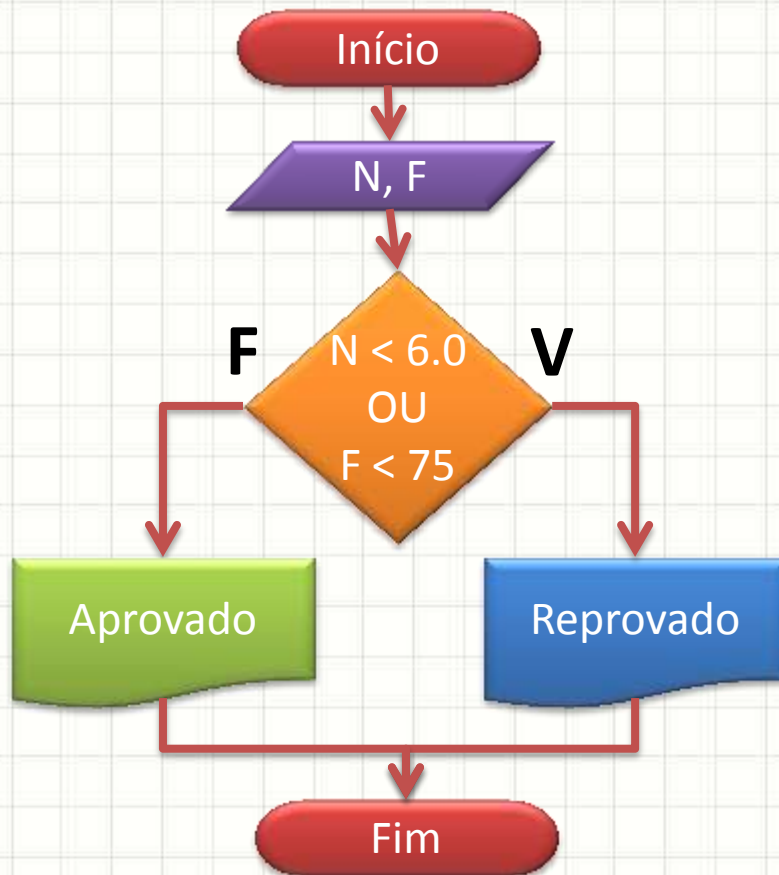
Decisão Composta

- Receba a **nota** de um aluno e a frequência dele e responda: **reprovado** caso a nota seja menor que 6,0 **ou** a frequência seja inferior a 75%



Decisão Composta

- Receba a **nota** de um aluno e a frequência dele e responda se ele está reprovado caso a nota seja menor que 6,0 **ou** a frequência seja inferior a 75%



```
#include <stdio>
#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 << "Reprovado";
    }
    else {
        cout << "Aprovado";
    }
    getchar();
}
```


Operadores Lógicos

- O que vem a ser `||` ?

Operador	Símbolo	Exemplo	Significado
OU	<code> </code>	<code>x == 2 x == 9</code>	Verdadeiro se <code>x == 2</code> OU <code>x == 9</code>
E	<code>&&</code>	<code>x > 2 && x < 9</code>	Verdadeiro se <code>x > 2</code> E <code>x < 9</code> (testa se <code>2 < x < 9</code>)
NÃO	<code>!</code>	<code>!(x == 2)</code>	Verdadeiro se <code>x</code> NÃO for igual a 2



Operadores Lógicos

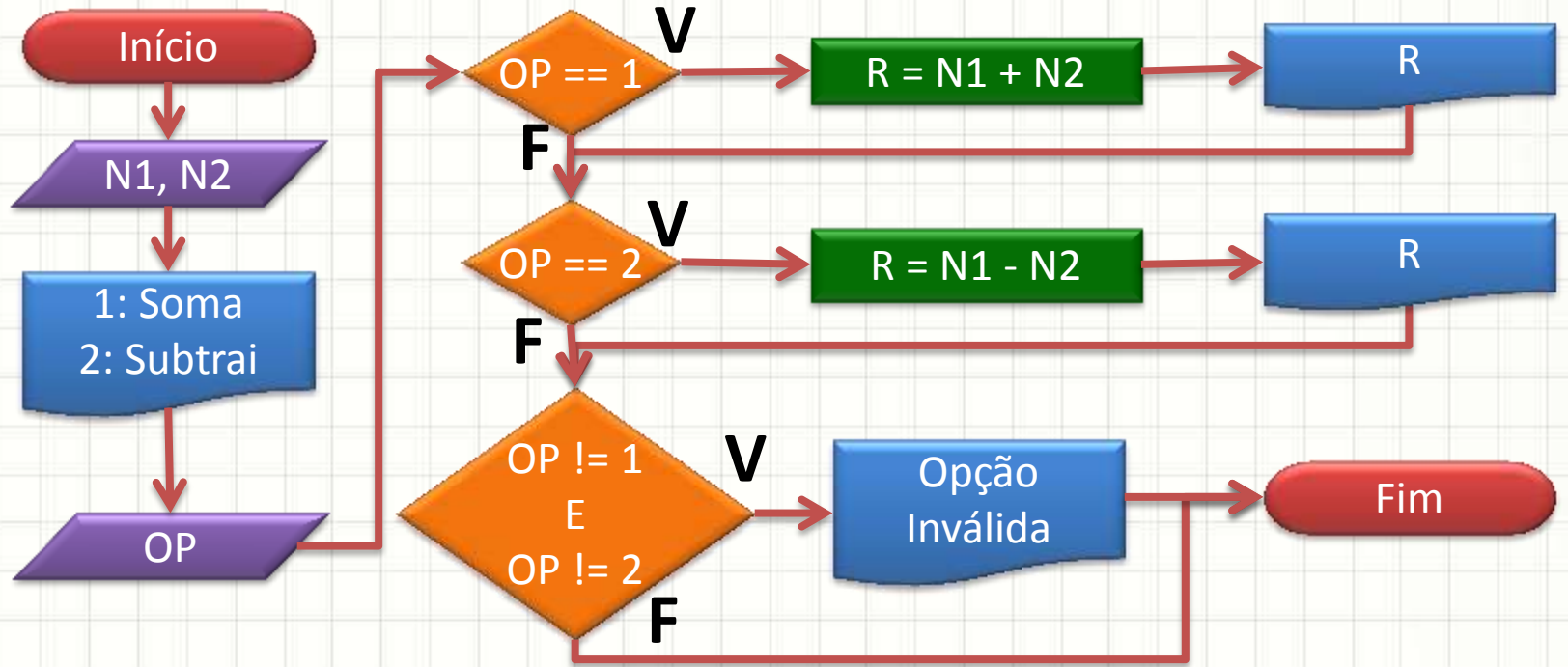
- O que vem a ser `||` ?

Operador	Símbolo	Exemplo	Significado										
"	!	@	#	\$	%	&	*	()	-	+	←	
1	1	2	3	4	5	7	8	9	0	-	=	Backspace	
Tab	Q	W	E	R	T	Y	U	I	O	P	{	Enter	
	/	?	€								[↵	
Caps Lock	A	S	D	F	G	H	J	K	L	Ç	^	}	
	↑										~]	
Shift		Z	X	C	V	B	N	M	<	>	:	?	Shift
	↵	\		€					,	.	:	/	↵
Ctrl	Win Key	Alt								Alt Gr	Win Key	Menu	Ctrl



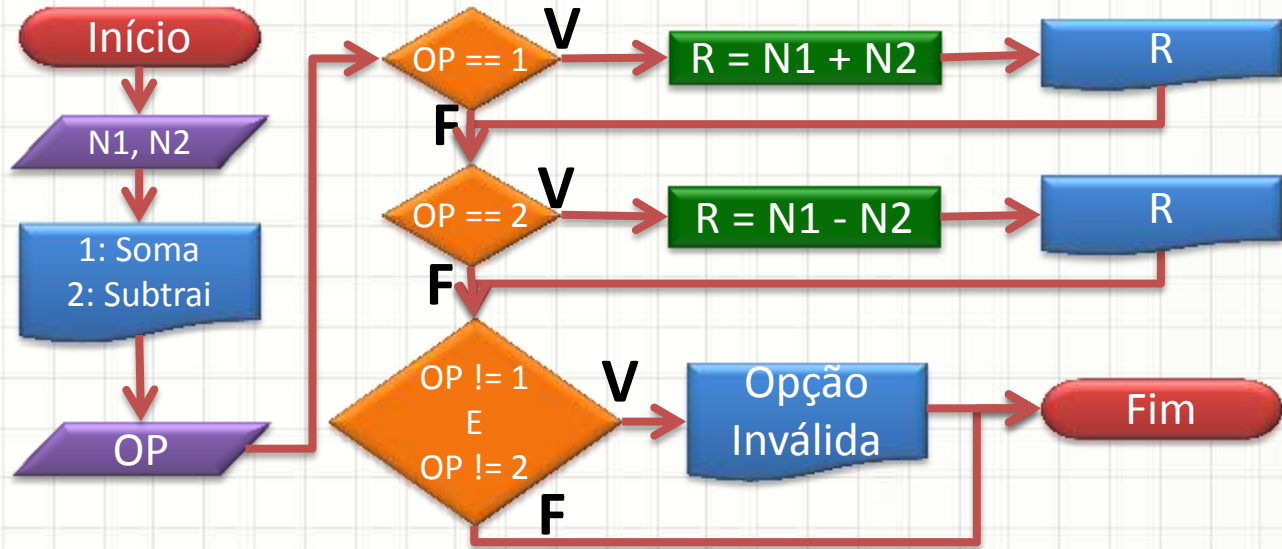
Exercício

- 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 <stdio>
#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!";
}
getchar();
}
```

Exercício – Solução (zoom)

```
#include <stdio>
#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!";
    }
    getchar();
}
```



ESTRUTURA DE ESCOLHA

Estrutura de Escolha

- E se um menu tiver 200 opções...
- Como ficaria o **if** final do exemplo anterior?
- ```
if (OP!=1 && OP!=2 && OP!=3 && ... && OP!=200) {
 ...
}
```
- Inviável!
- 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 <stdio>
#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";
 }

 getch();
}
```

# Exercício: Converta para Switch

```
#include <stdio>
#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!";
 }
 getchar();
}
```

# Exercício: Solução

```
#include <stdio>
#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!";
}
getchar();
}
```

# Exercício: Solução

```
#include <stdio>
#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 << "s:" << R;
```

**ATENÇÃO: SWITCH só funciona com variáveis INTEIRAS**

```
cout << R;
```

```
cout << "Opção Inválida!";
```

```
}
```

```
getchar();
```

```
}
```



**ATIVIDADE**

# Exercício

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



**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!**