



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

ESTRUTURAS DE DECISÃO

Prof. Dr. Daniel Caetano

2013 - 1

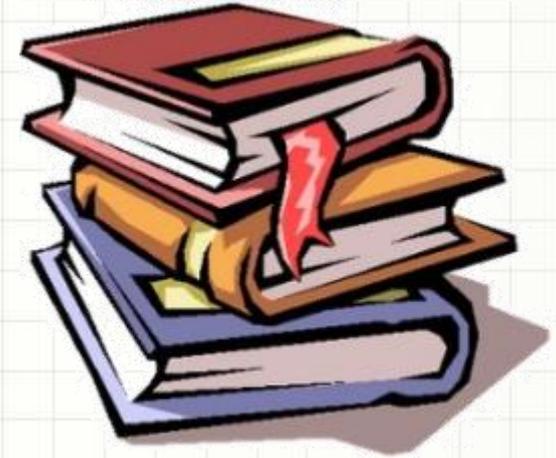
Objetivos

- Entender a ideia de decisão
- Compreender como implementar decisões
- Compreender problemas com decisões múltiplas
- Entender a representação de uma estrutura de decisão composta

- **PARA CASA**
 - Lista Aula 8!



Material de Estudo



Material

Acesso ao Material

Notas de Aula

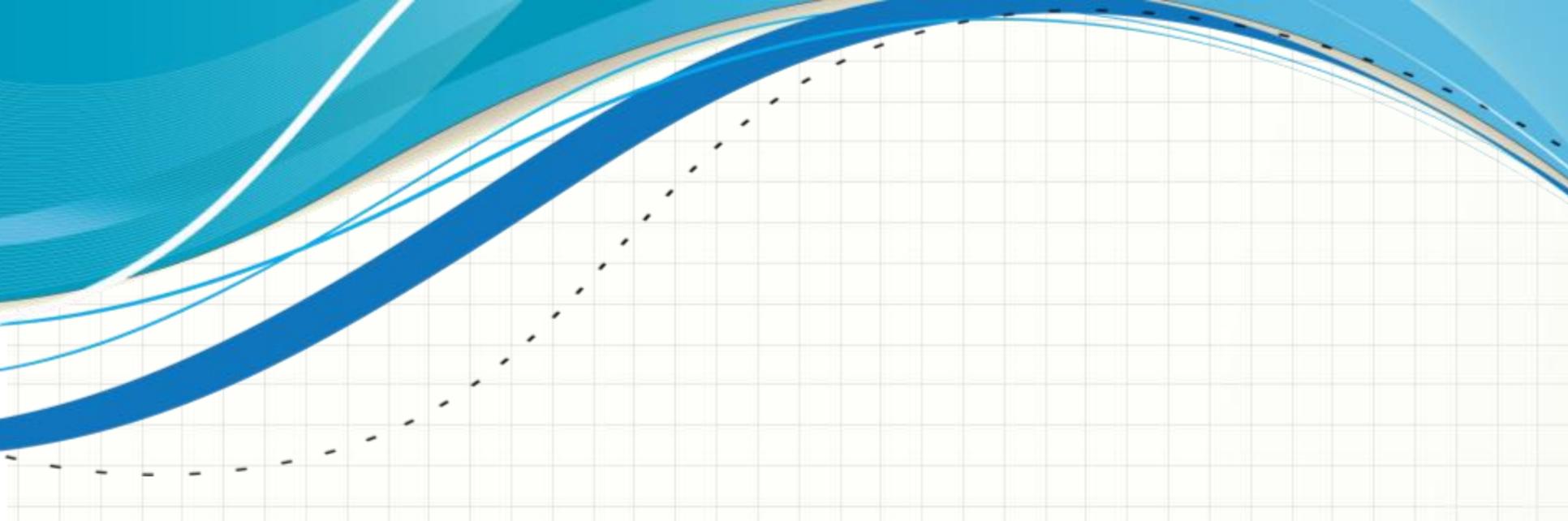
<http://www.caetano.eng.br/aulas/lpe/>
(Aula 8)

Apresentação

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 8) – PARCIAL / COMPLETO

Material Didático

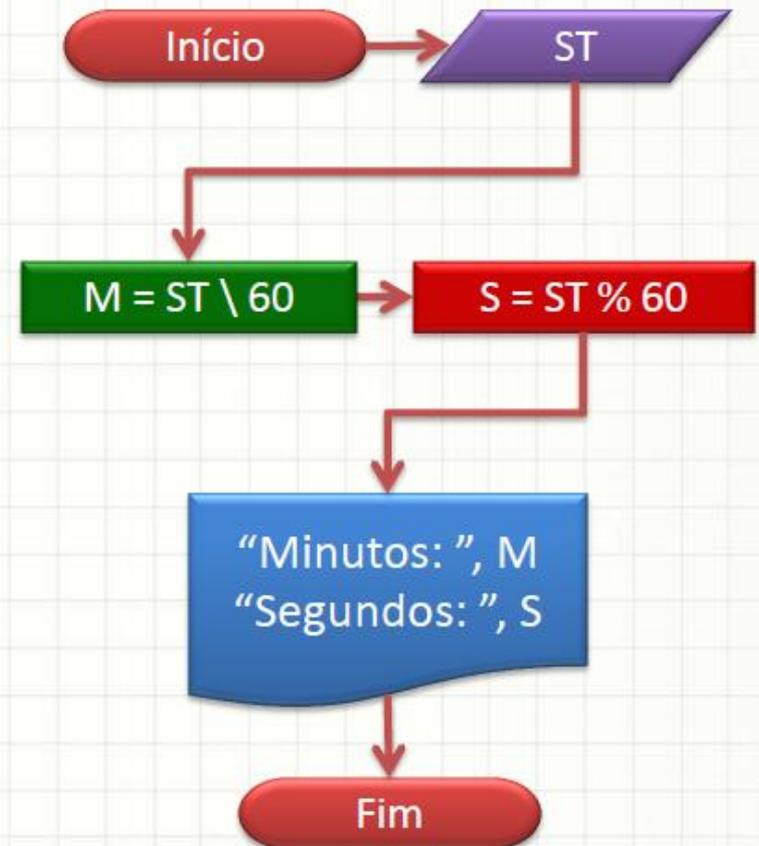
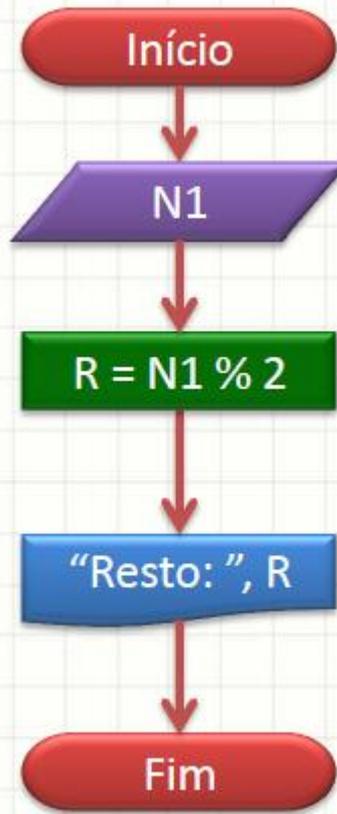
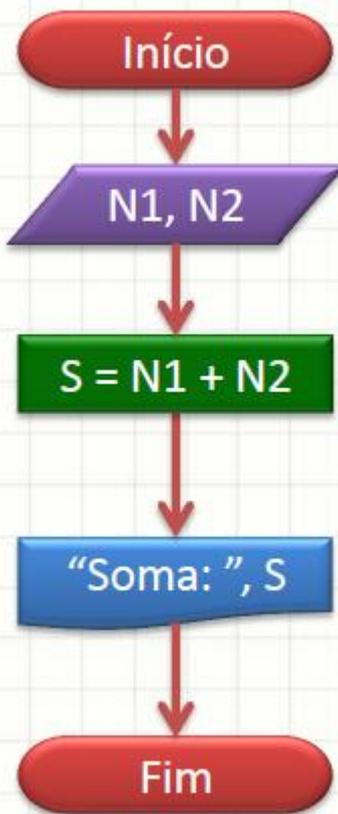
Fundamentos da Programação de Computadores –
Parte 1 – Páginas 50 a 92.



O QUE É DECISÃO?

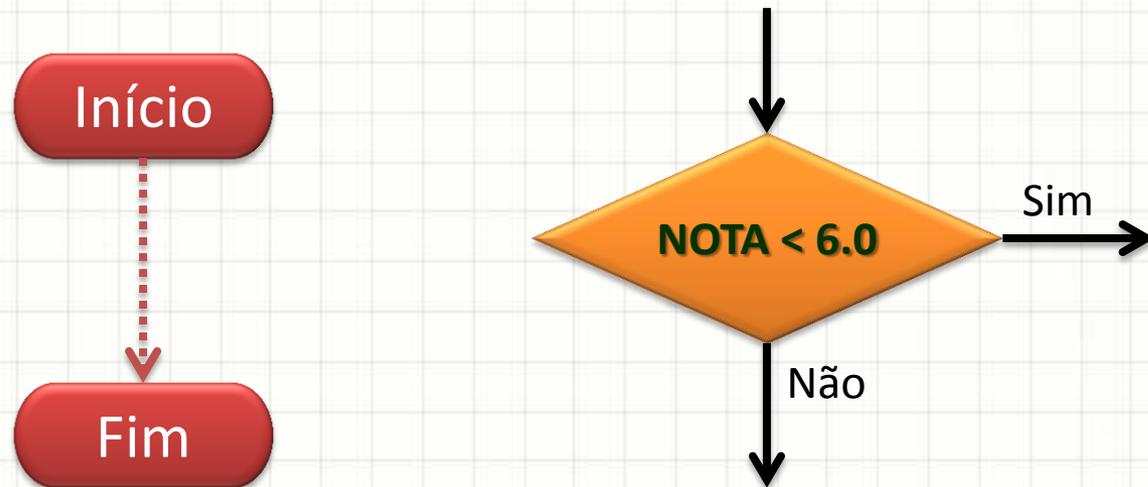
Introdução

- Até agora:
 - programas fazem sempre a **mesma** tarefa!



Introdução

- Até agora, nossos programas...
 - faziam sempre, exatamente, a **mesma** tarefa!
- Por quê?
- Porque não são capazes de tomar decisões!



O que é Decisão para o Computador?

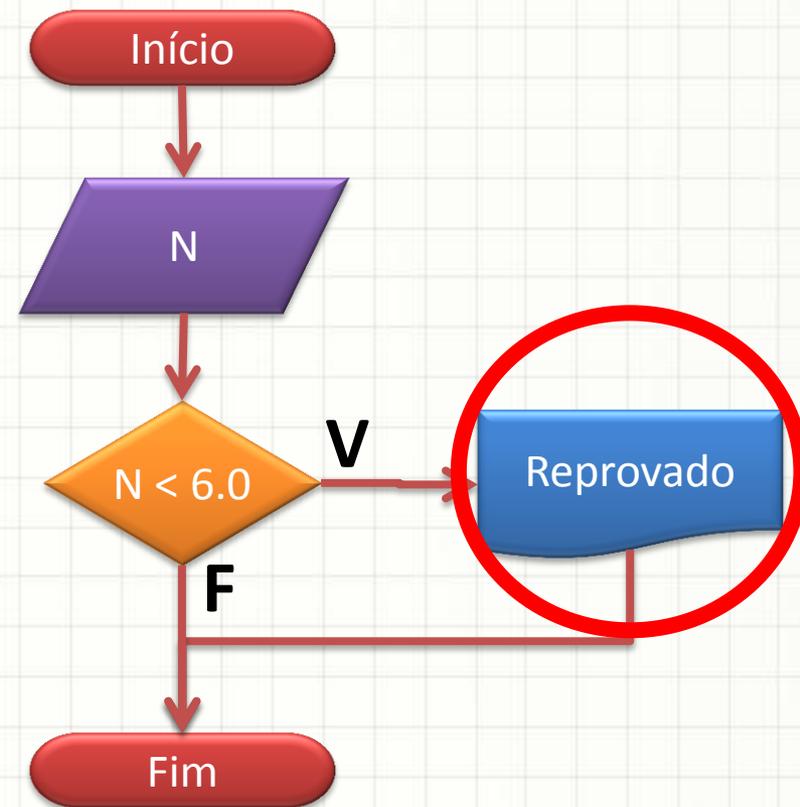
- Decidir: escolher se código será executado

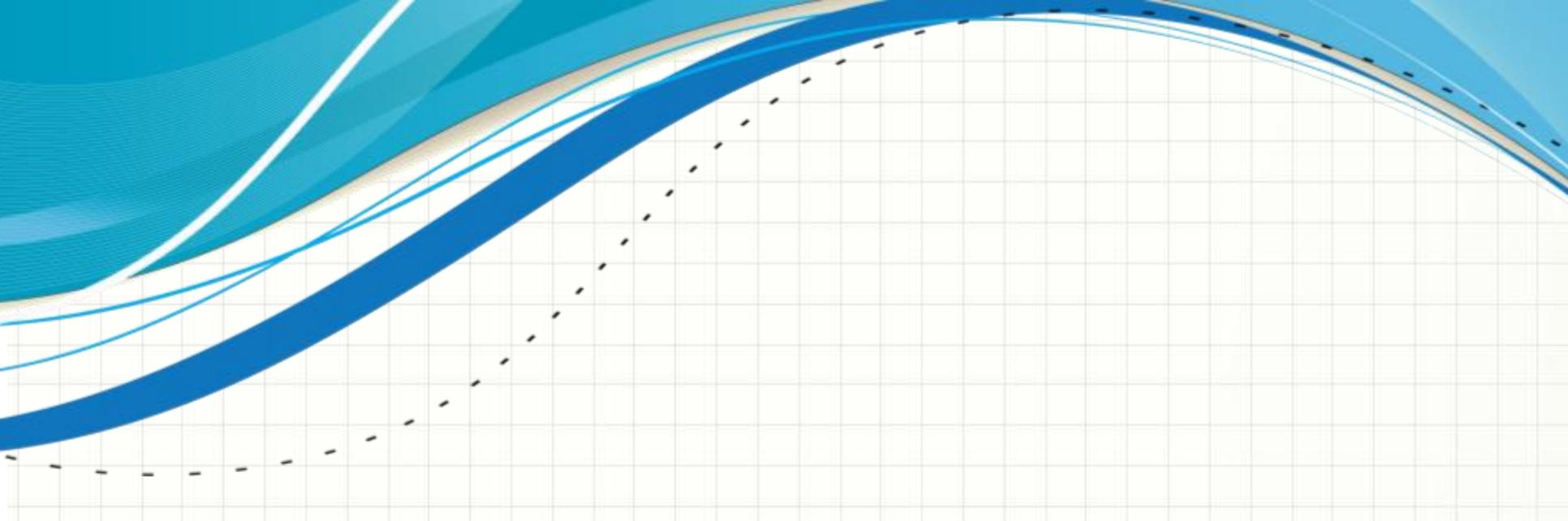


O que é Decisão para o Computador?

- Decidir: escolher se código será executado
- Com base em quê?
- Regra = proposição:
 - verdadeiro → executa
 - falso → não executa
- Exemplo:
 - Imprimir “Reprovado”

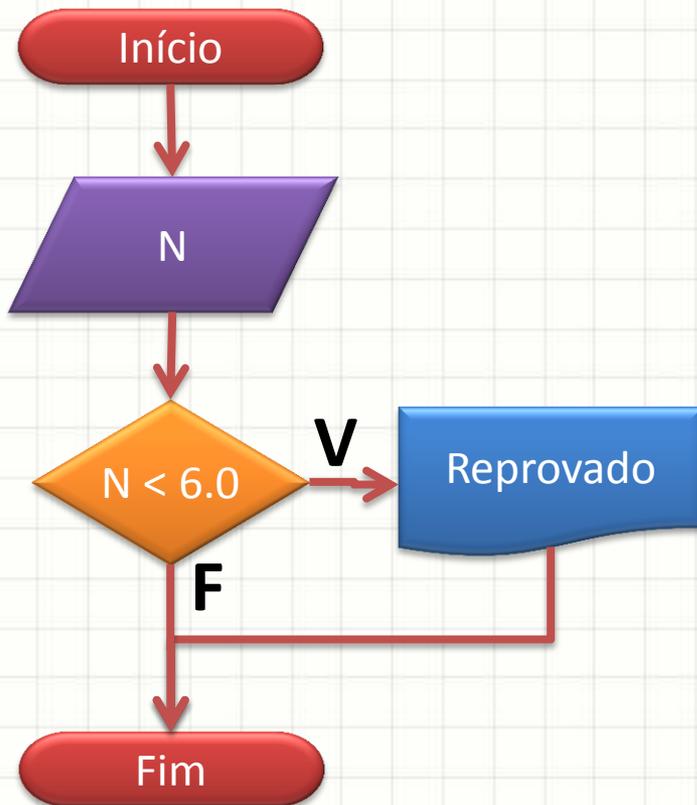
Somente Se $N < 6.0$





DECISÃO NO CÓDIGO

Como Fica a Decisão no Código?



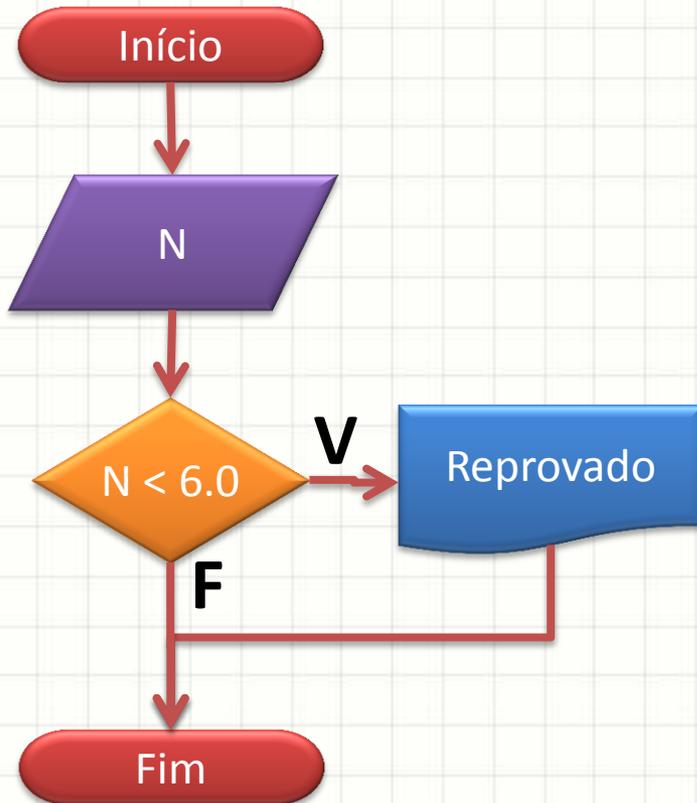
- Se Nota < 6.0 imprime que aluno está reprovado

- Português Estruturado

Se $N < 6.0$ Entao

Escreva "Reprovado"

Como Fica a Decisão no Código?



- Se Nota < 6.0 imprime que aluno está reprovado e a sua nota

- Português Estruturado

Se N < 6.0 Entao

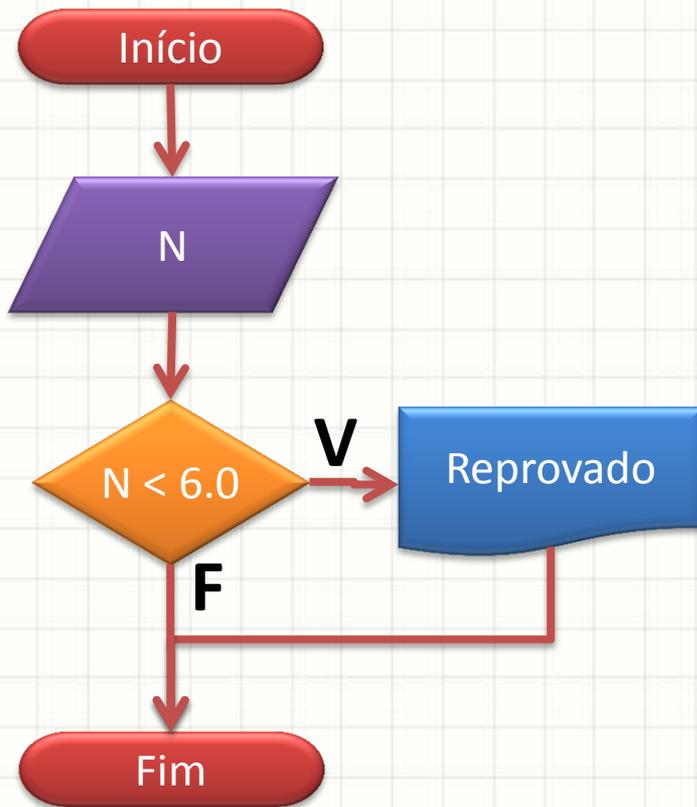
Inicio

Escreva "Reprovado:"

Escreva N

FimSe

Como Fica a Decisão no Código?



- Português Estruturado

Algoritmo “Verifica Reprovação”

Inicio

Real N

Escreva “Digite a nota: ”

Leia N

Se $N < 6.0$ **Entao**

Inicio

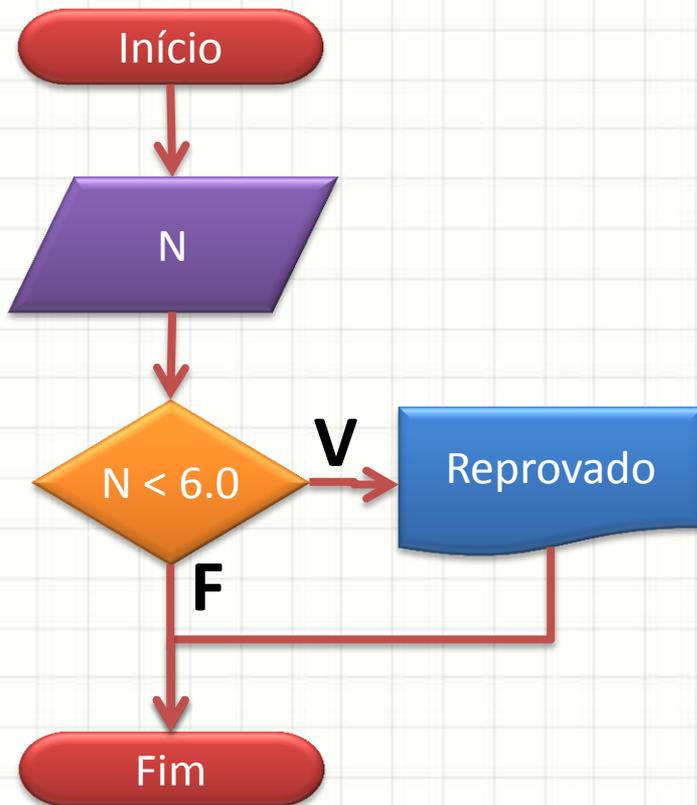
Escreva “Reprovado: ”

Escreva N

FimSe

FimAlgoritmo

Como Fica a Decisão no Código?



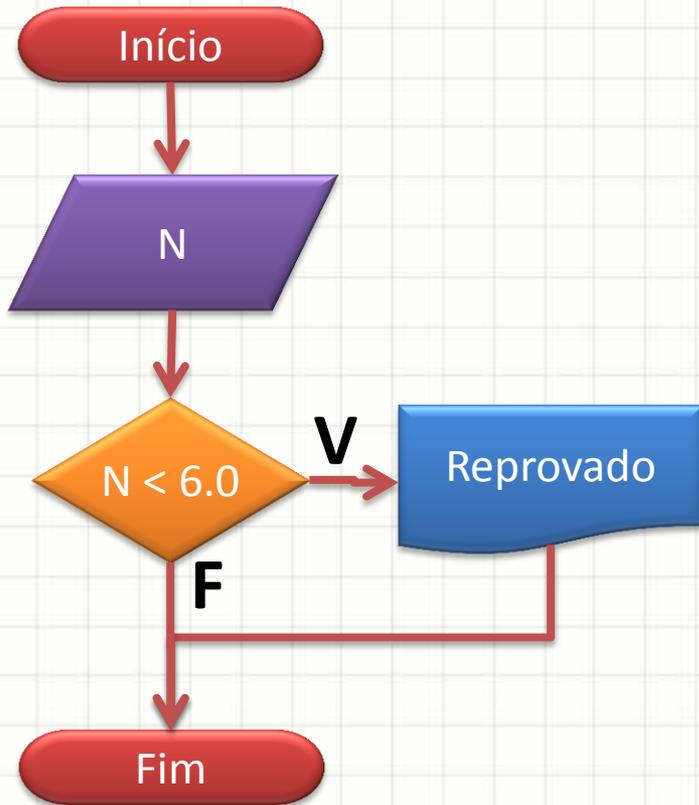
- Se Nota < 6.0 imprime que aluno está reprovado

- C / C++

```
if ( N < 6.0 )
```

```
    cout << "Reprovado";
```

Como Fica a Decisão no Código?



- Se $Nota < 6.0$ imprime que aluno está reprovado e a sua nota

- C / C++

```
if ( N < 6.0 )
```

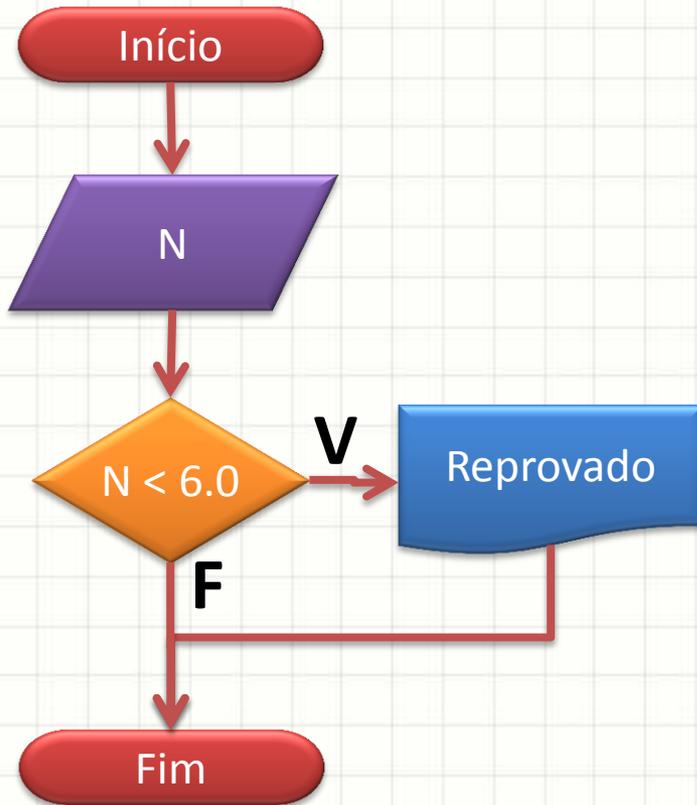
```
{
```

```
    cout << "Reprovado:";
```

```
    cout << N;
```

```
}
```

Como Fica a Decisão no Código?



- C/C++

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

```
{
```

```
float N;
```

```
cout << "Digite a nota: ";
```

```
cin >> N;
```

```
if ( N < 6.0 )
```

```
{
```

```
cout << "Reprovado: ";
```

```
cout << N;
```

```
}
```

```
}
```

Forma Geral do **Se / If**

- Português Estruturado

Se proposição_lógica **Entao**

Inicio

código a executar para proposição verdadeira

FimSe

- C / C++

if (proposição_lógica)

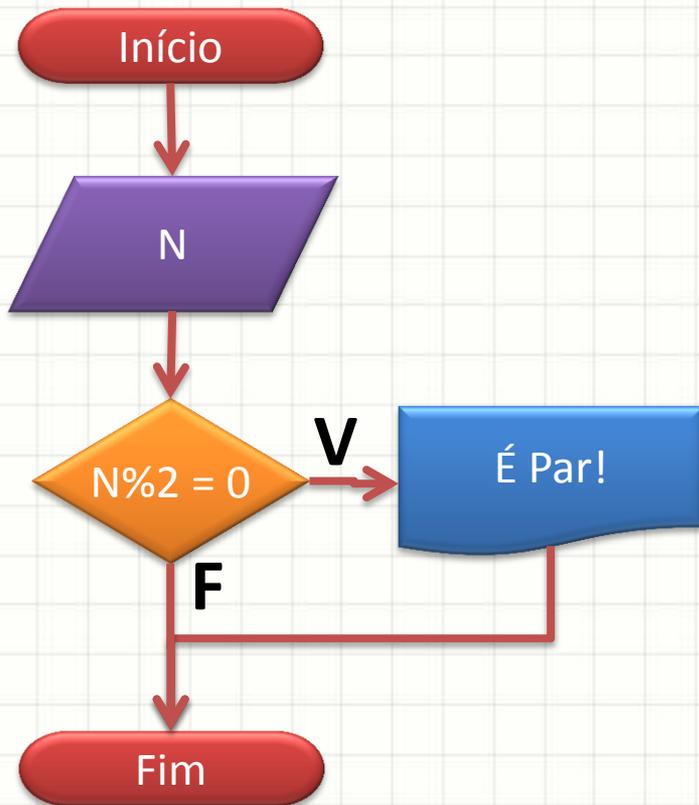
{

código a executar para proposição verdadeira

}

Outro Exemplo de Decisão

- Imprimir se número é par



- Português Estruturado

Algoritmo “Verifica Paridade”

Início

Inteiro N

Escreva “Digite um número: ”

Leia N

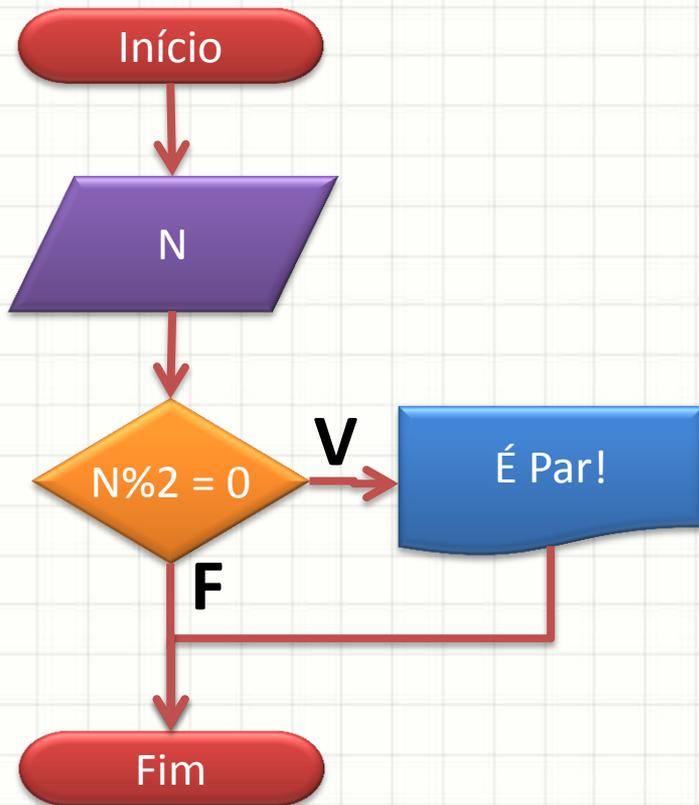
Se $N\%2 = 0$ **Entao**

Escreva “É Par!”

FimAlgoritmo

Como Fica a Decisão no Código?

- Imprimir se número é par
 - C/C++



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

```
{
```

```
int N;
```

```
cout << "Digite um número: ";
```

```
cin >> N;
```

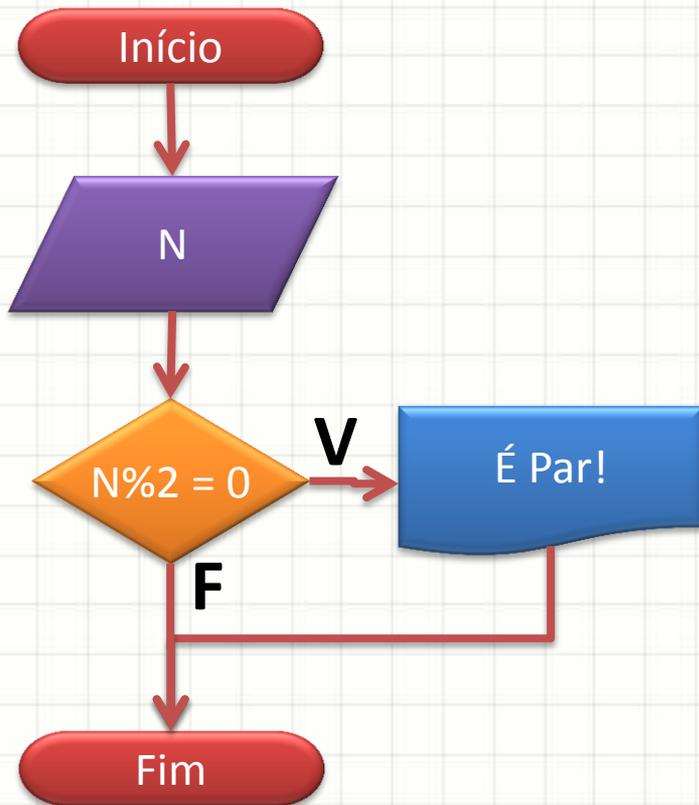
```
if ( N%2 == 0 )
```

```
    cout << "É Par!";
```

```
}
```

Como Fica a Decisão no Código?

- Imprimir se número é par
 - C/C++



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

```
{
```

```
int N;
```

```
cout << "Digite um número: ";
```

```
cin >> N;
```

```
if ( N%2 == 0 )
```

```
    cout << "É Par!";
```

```
}
```

Comparadores

- Por que em C/C++ usamos `==` ao invés de `=` ?
`=` significa atribuição (guardar valor em variável)

Muda
X

X = 5;

Guarda o valor 5 na variável X

`==` significa comparação (resulta falso/verdadeiro)

Não
Muda
X

X == 5;

Verifica se X contém o valor 5

Comparadores

- Quais são os comparadores?

Comparador	Exemplo	Significado
<code>==</code>	<code>x == 2</code>	Testa igualdade entre os elementos
<code>></code>	<code>x > 2</code>	Testa se um é maior que outro
<code>>=</code>	<code>x >= 2</code>	Testa se um é maior ou igual a outro
<code><</code>	<code>x < 2</code>	Testa se um é menor que outro
<code><=</code>	<code>x <= 2</code>	Testa se um é menor ou igual a outro
<code>!=</code>	<code>x != 2</code>	Testa se são diferentes

Exercício

- Faça um programa que **leia dois números** e responda **se o primeiro é o maior**

```
if ( proposição_lógica )  
{  
    código a executar  
}
```

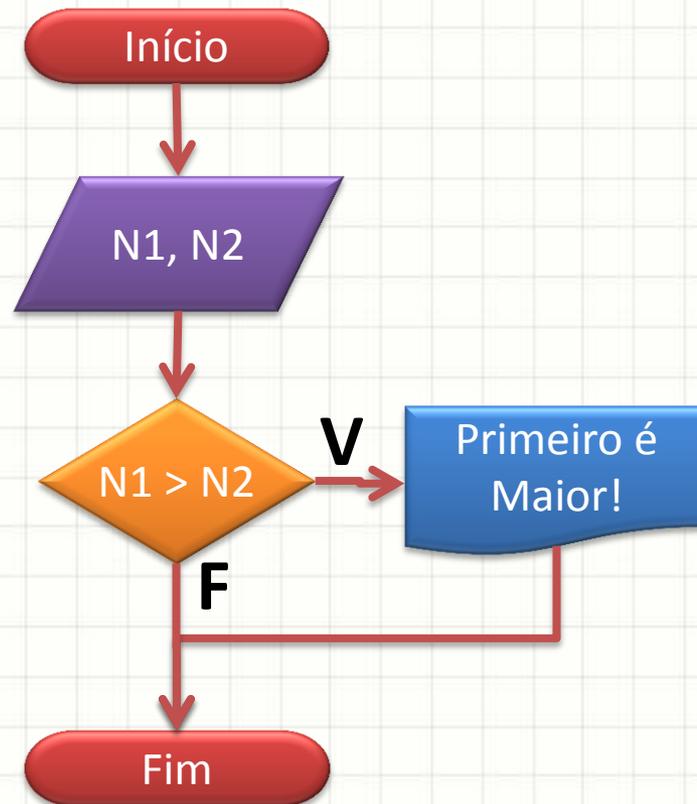
Comparador	Significado
==	Igualdade
>	Maior
>=	Maior ou Igual
<	Menor
<=	Menor ou Igual
!=	Diferente

- Verifica se é par

```
#include <iostream>  
using namespace std;  
main()  
{  
    int N;  
    cout << "Digite um número:";  
    cin >> N;  
    if ( N%2 == 0 )  
        cout << "É Par!";  
}
```

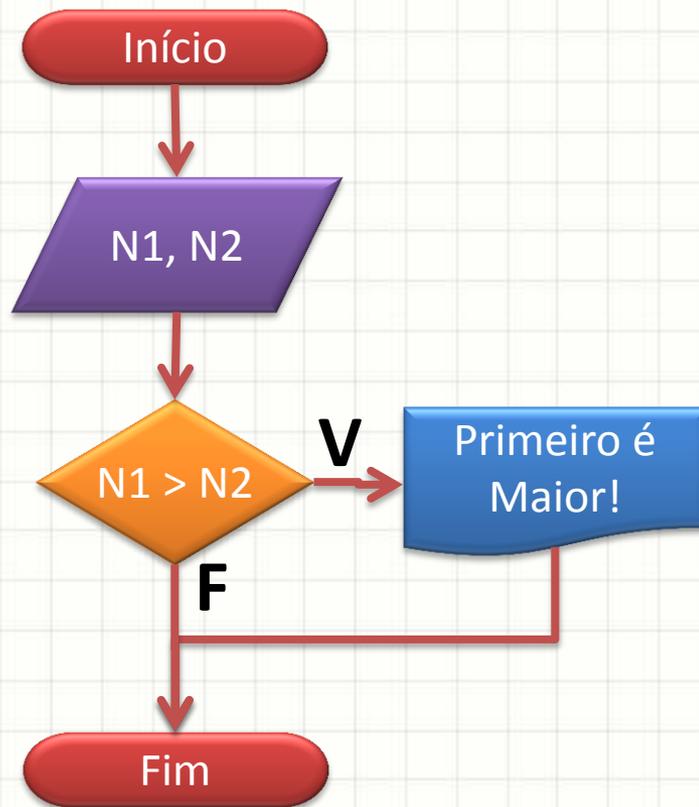
Exercício

- Faça um programa que lê dois números e responda **se o primeiro é o maior**



Exercício

- Faça um programa que lê dois números e responda se o primeiro é o maior



Algoritmo “Imprime se é Maior”

Início

Inteiro N1, N2

Escreva “Digite um número:”

Leia N1

Escreva “Digite outro número:”

Leia N2

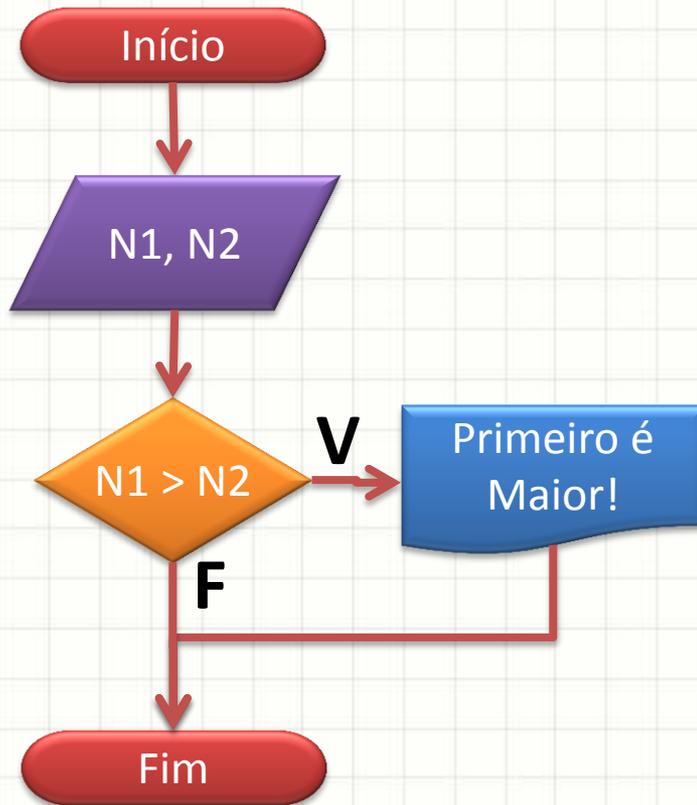
Se N1 > N2 Então

Escreva “Primeiro é maior!”

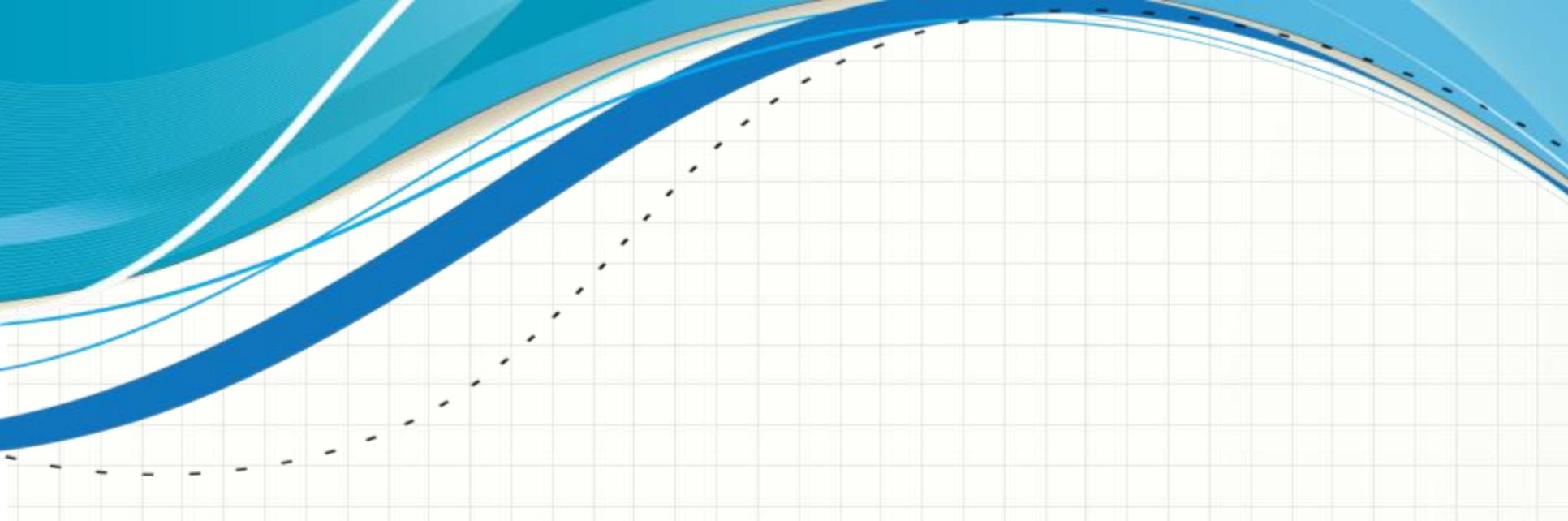
FimAlgoritmo

Exercício

- Faça um programa que lê dois números e responda se o primeiro é o maior



```
#include <iostream>  
using namespace std;  
main()  
{  
    int N1, N2;  
    cout << "Digite um número:";  
    cin >> N1;  
    cout << "Digite outro número:";  
    cin >> N2;  
    if ( N1 > N2 )  
        cout << "Primeiro é maior!";  
}
```



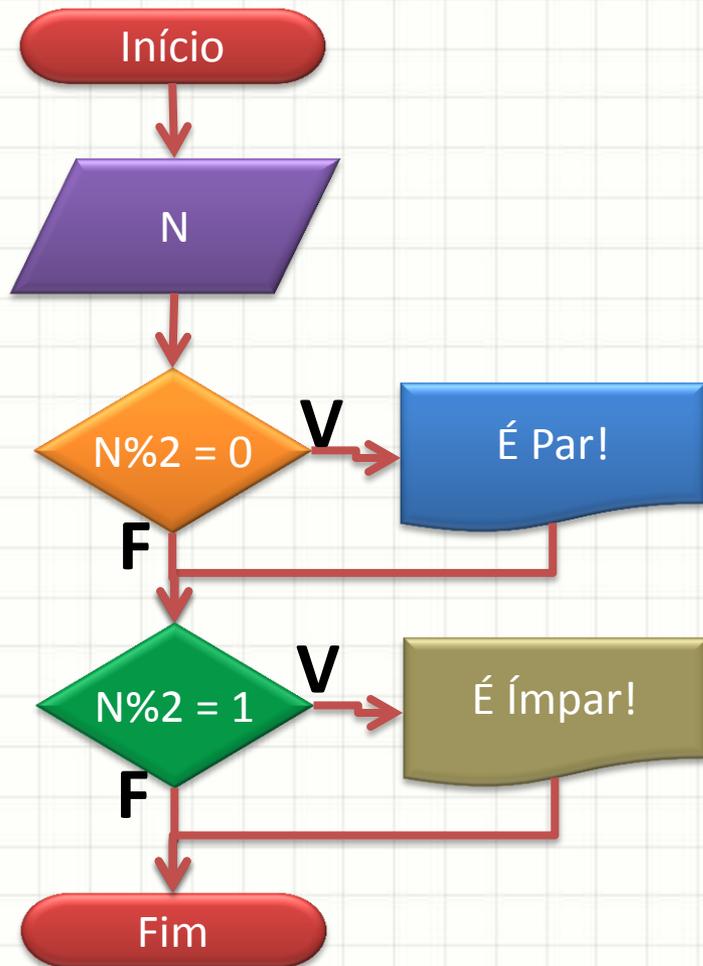
DECISÕES MÚLTIPLAS

Múltiplas Decisões

- Já implementamos uma decisão...
- Tudo pode ser feito com **uma** única decisão?
- E se quisermos:
 - a) Imprime “É Par!” se o número for par
 - b) Imprime “É Ímpar!” se o número for ímpar
- Podemos fazer isso com **duas** decisões!

Múltiplas Decisões

- Verificar se número é par ou ímpar



```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

```
int N;
```

```
cout << "Digite um número: ";
```

```
cin >> N;
```

```
if ( N%2 == 0 )
```

```
    cout << "É Par!";
```

```
if ( N%2 == 1 )
```

```
    cout << "É Ímpar!";
```

```
}
```

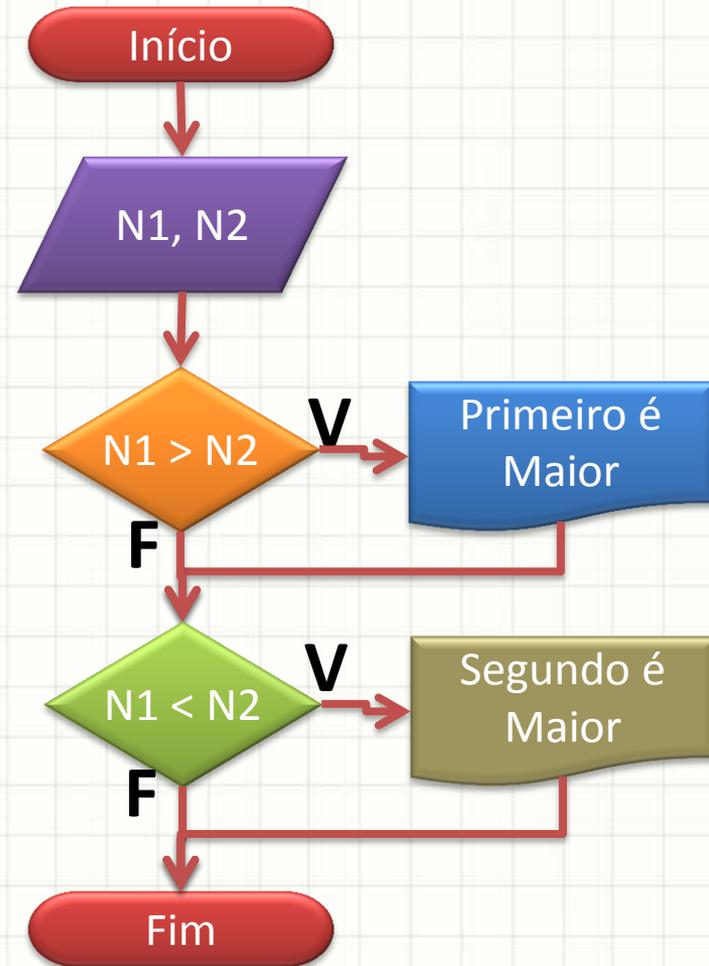
Exercício

- Modifique o programa do exercício anterior para que leia dois números e responda se **o primeiro é maior** ou se **o segundo é o maior**

```
#include <iostream>
using namespace std;
main()
{
    int N1, N2;
    cout << "Digite um número: ";
    cin >> N1;
    cout << "Digite outro número: ";
    cin >> N2;
    if ( N1 > N2 )
        cout << "Primeiro é maior!";
}
```

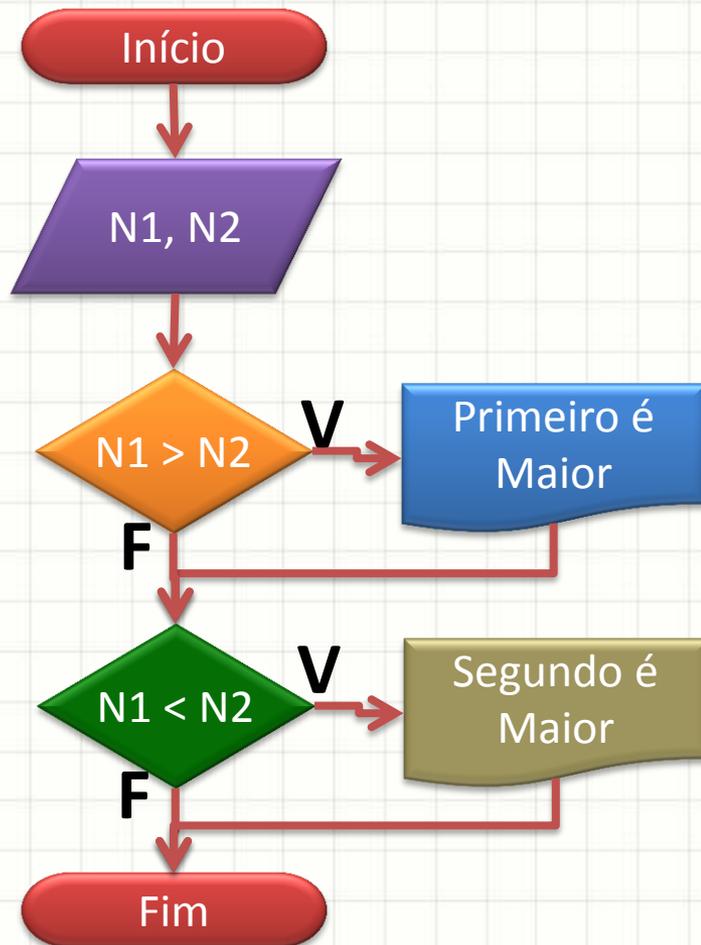
Exercício

- Modifique o programa do exercício anterior para que leia dois números e responda **se o primeiro ou o segundo é o maior**



Exercício

- Modifique o programa para que leia 2 números e responda se o primeiro ou o segundo é o maior



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

```
{
```

```
int N1, N2;
```

```
cout << "Digite um número:";
```

```
cin >> N1;
```

```
cout << "Digite outro número:";
```

```
cin >> N2;
```

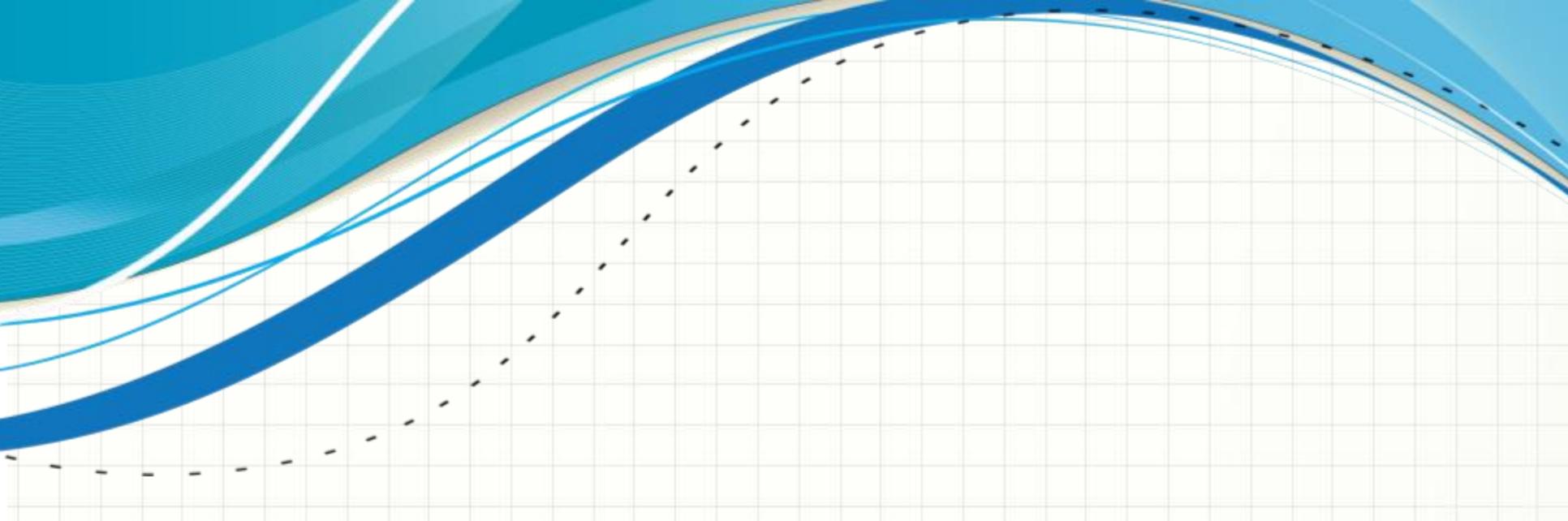
```
if ( N1 > N2 )
```

```
    cout << "Primeiro é maior!";
```

```
if ( N1 < N2 )
```

```
    cout << "Segundo é maior!";
```

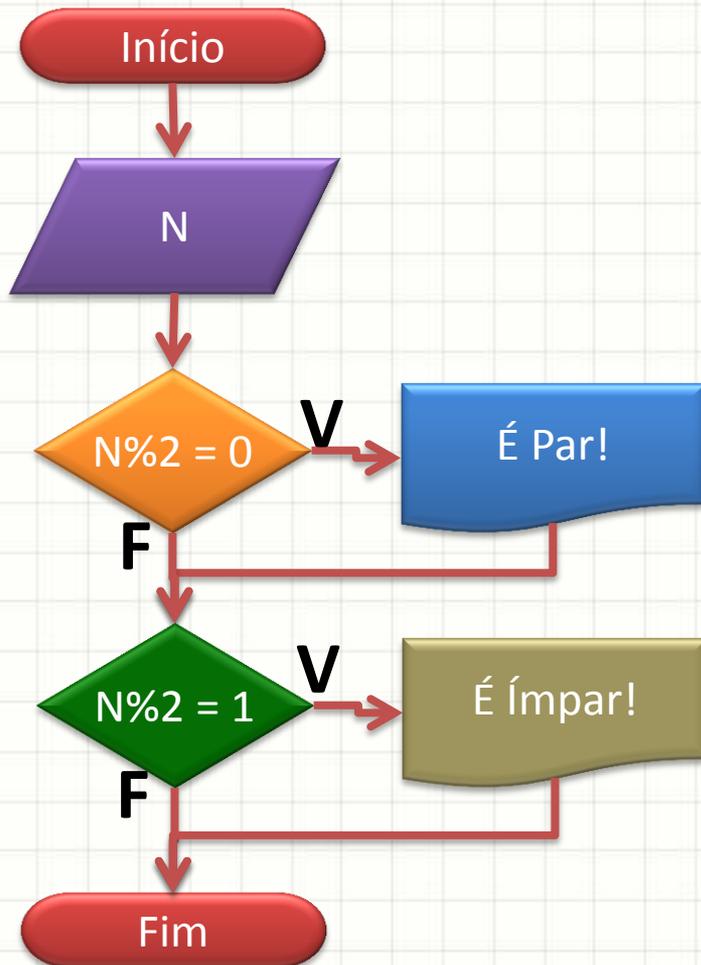
```
}
```



ESTRUTURA DE DECISÃO COMPOSTA

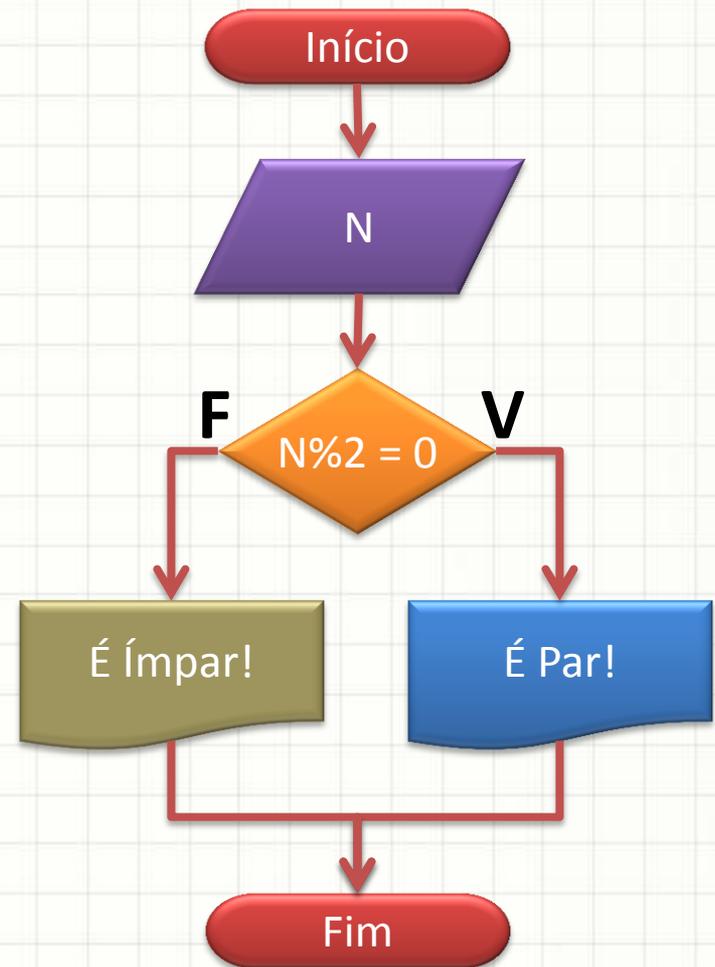
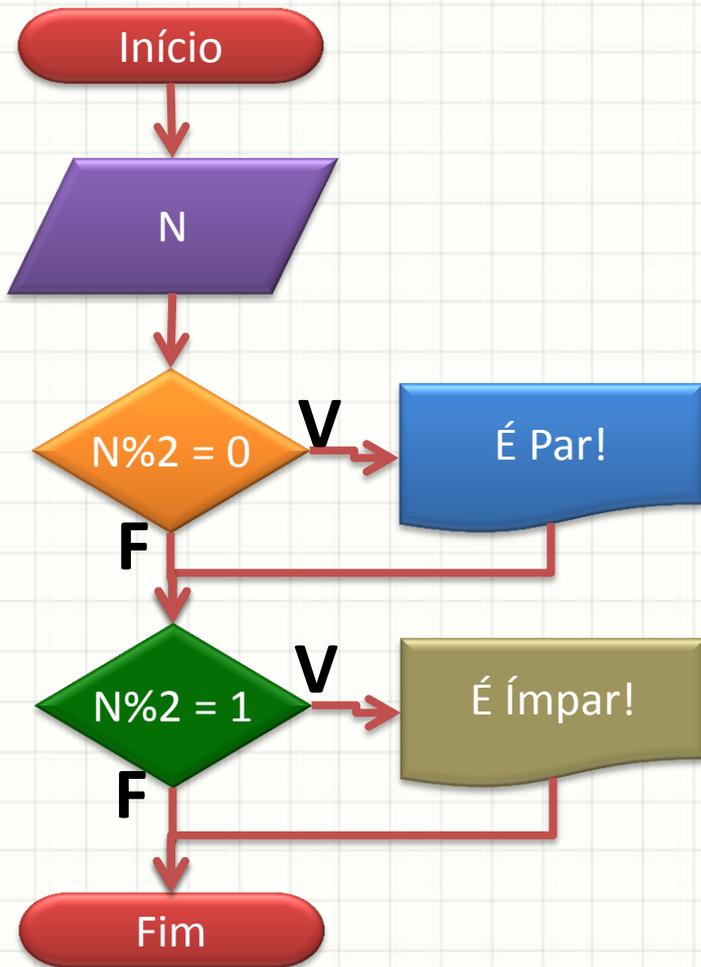
Estrutura de Decisão Composta

- Observe o fluxograma...



Estrutura de Decisão Composta

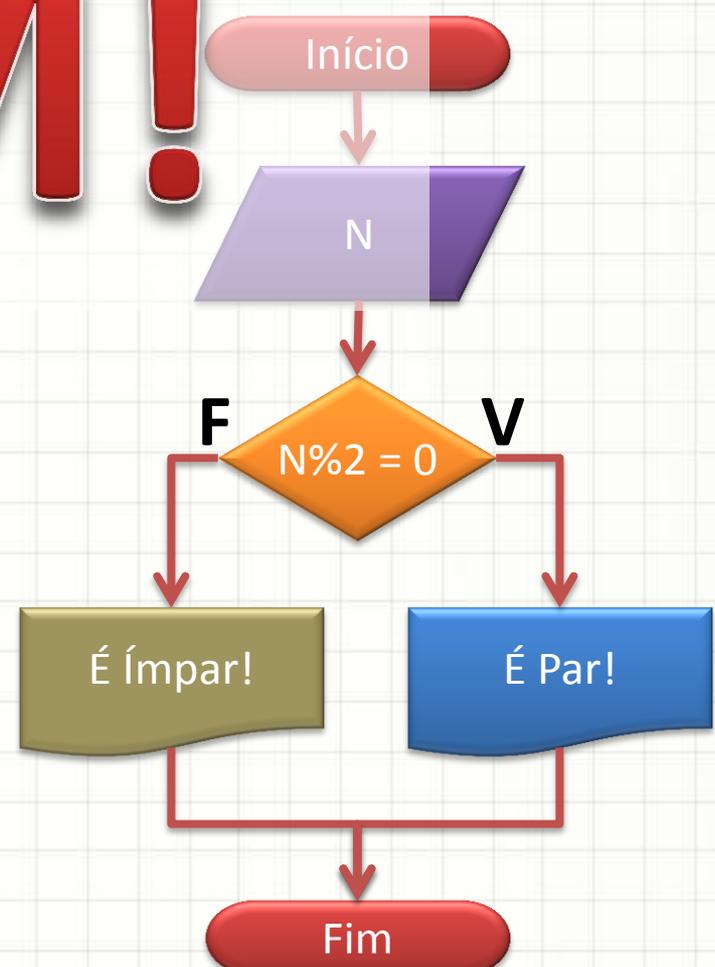
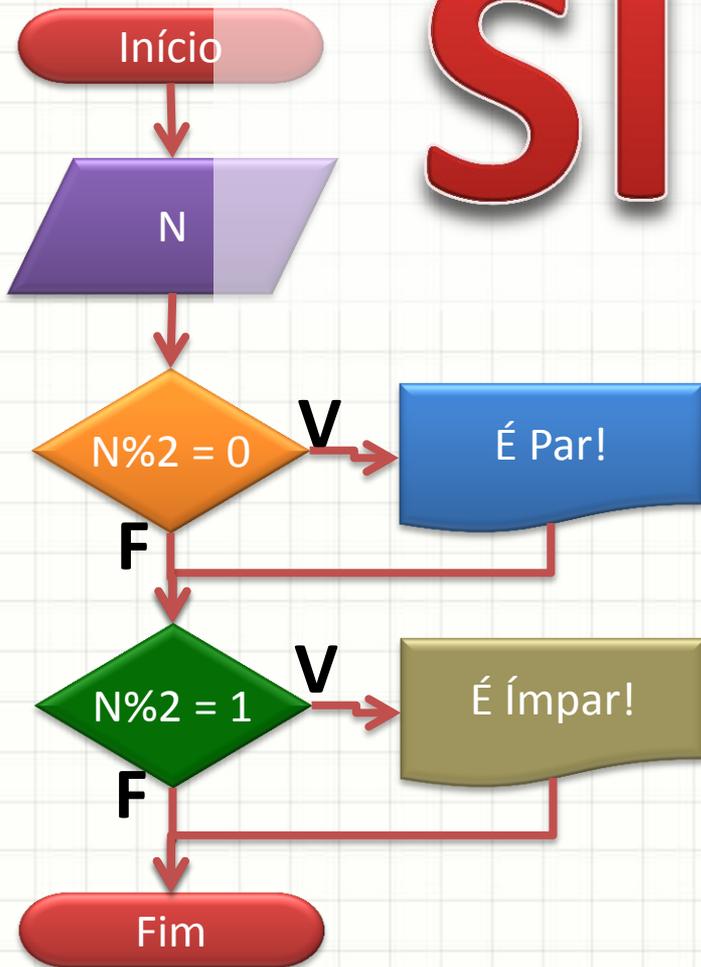
- Observe este outro... São iguais?



Estrutura de Seleção Composta, **EM FUNÇÃO,**

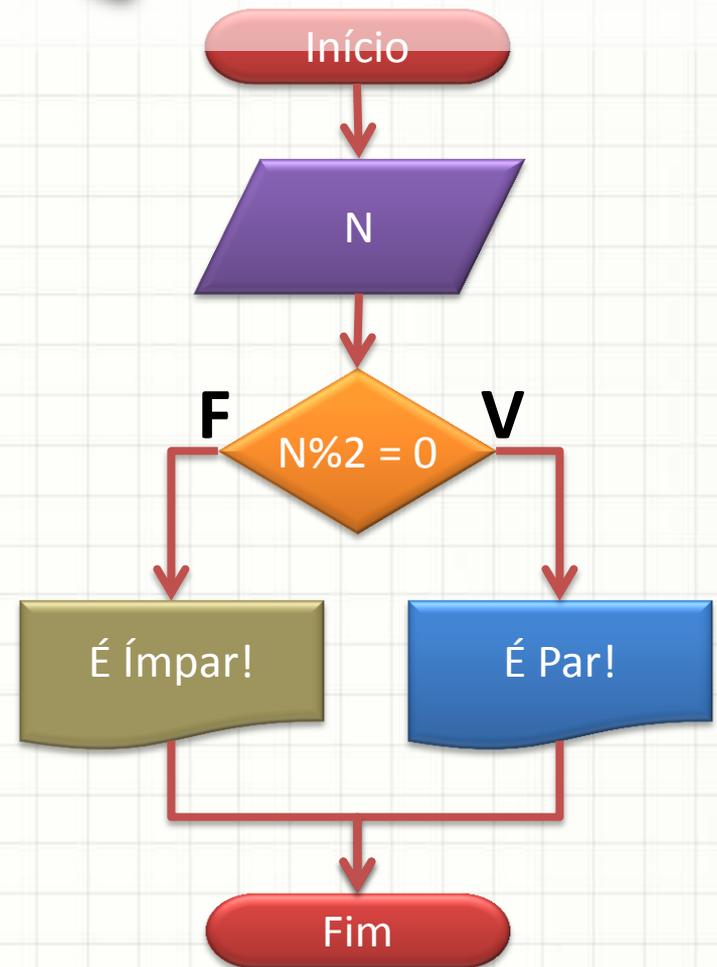
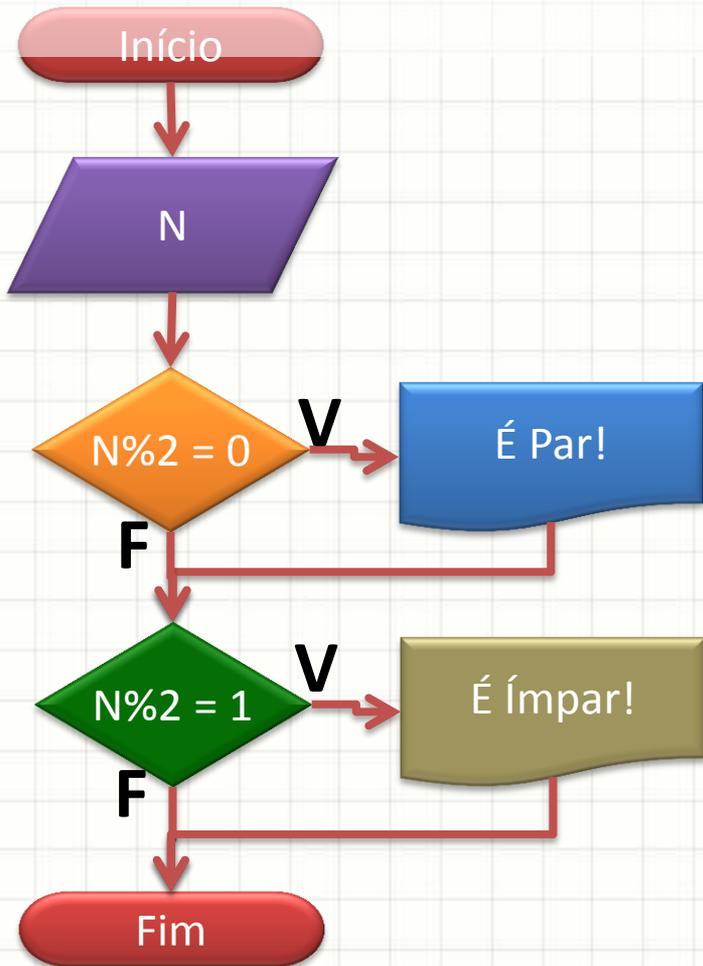
- Observe este outro... São iguais?

SIM!



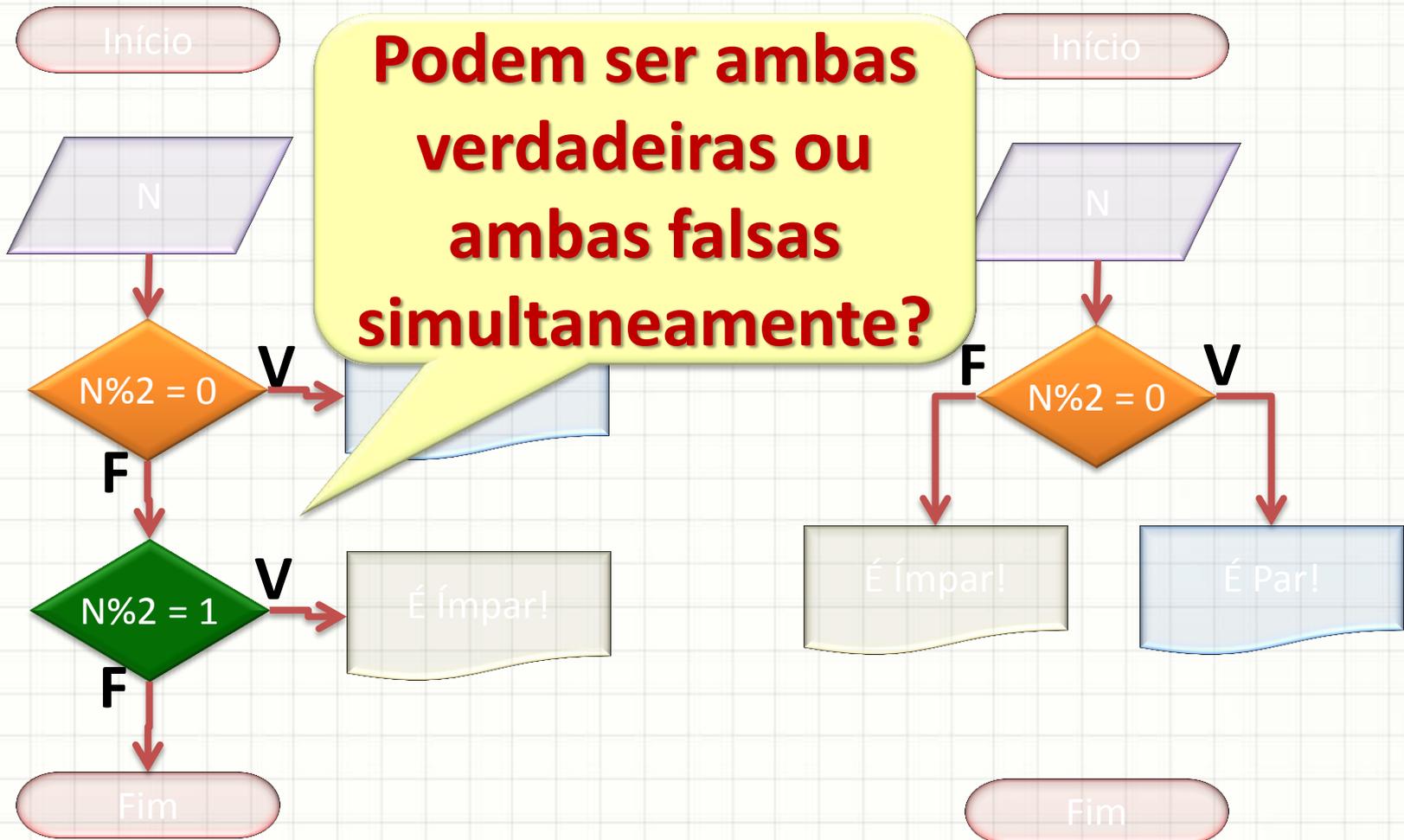
Porquê?

- Observa este outro... *algoritmo*



Estrutura de Decisão Composta

- Observe este outro... São iguais?



Forma do Se~Senao

- Português Estruturado

Se proposição_lógica **Entao**

Inicio

código a executar para proposição verdadeira

FimSe

Senao

Inicio

código a executar para proposição falsa

FimSenao

Forma do If~else

- C / C++

```
if ( proposição_lógica )
```

```
{
```

```
    código a executar para proposição verdadeira
```

```
}
```

```
else
```

```
{
```

```
    código a executar para proposição falsa
```

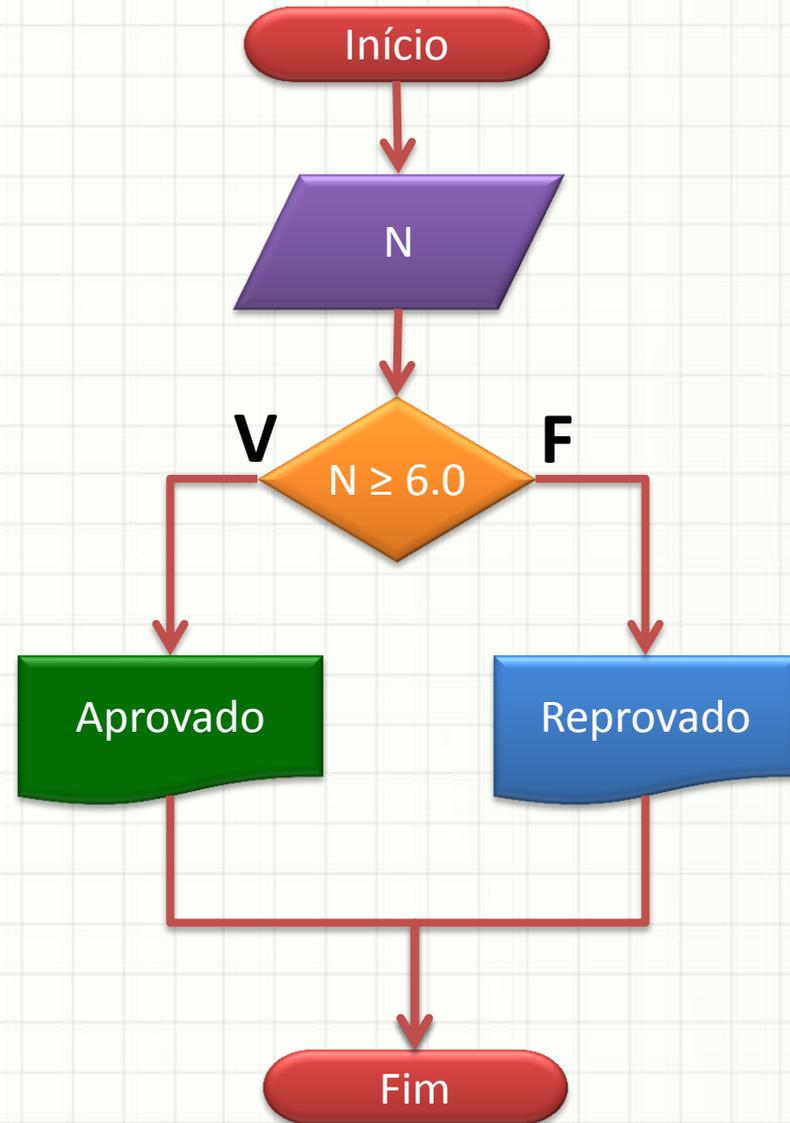
```
}
```

Exercício

- Faça um programa que receba a nota de um aluno e responda que **ele está aprovado se a nota for maior ou igual a 6,0 e reprovado caso contrário**

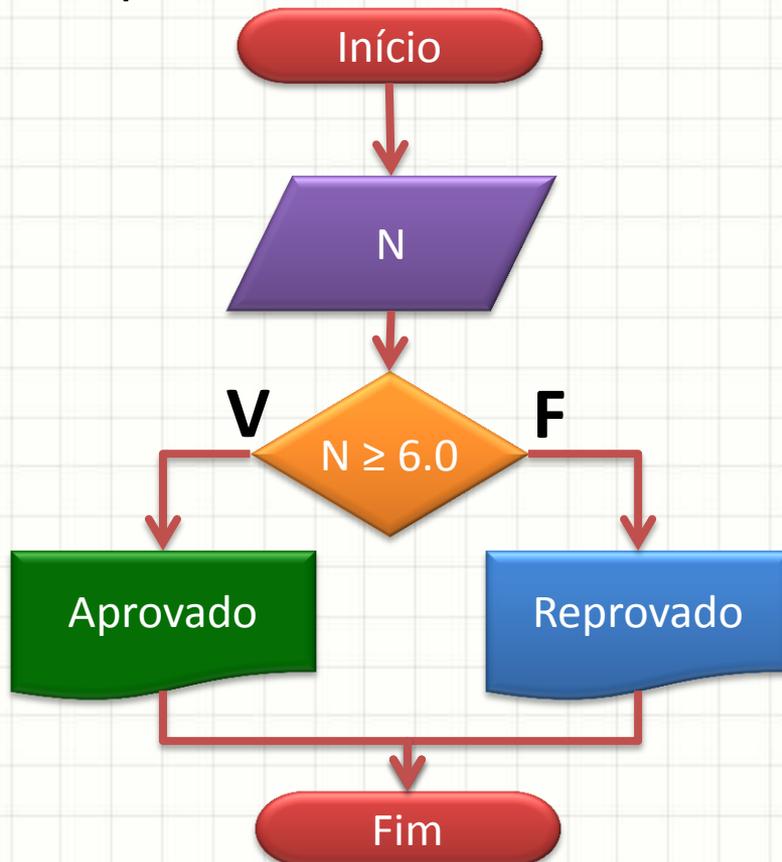
Exercício

- Faça um programa que receba a nota de um aluno e responda que ele está aprovado se a nota for maior ou igual a 6,0 e reprovado caso contrário



Exercício

- Faça um programa que receba a nota de um aluno e responda que ele está aprovado se a nota for maior ou igual a 6,0 e reprovado caso contrário



- C/C++

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

```
{
```

```
float N;
```

```
cout << "Digite a nota: ";
```

```
cin >> N;
```

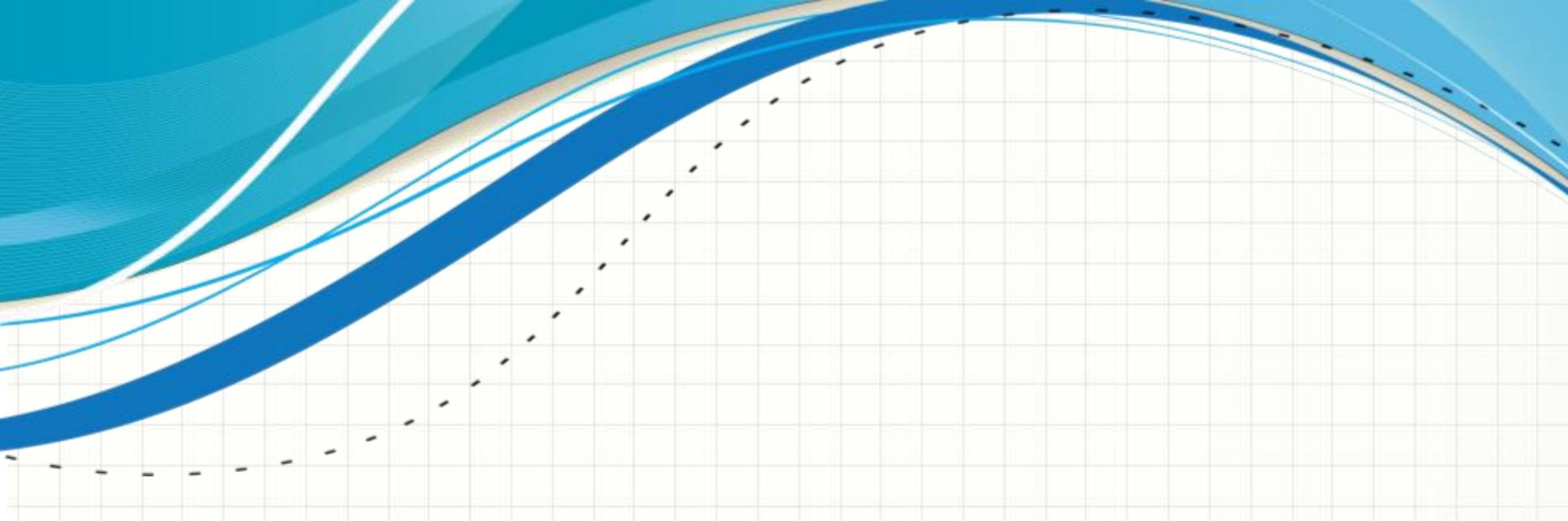
```
if ( N >= 6.0 )
```

```
    cout << "Aprovado";
```

```
else
```

```
    cout << "Reprovado";
```

```
}
```



CONCLUSÕES

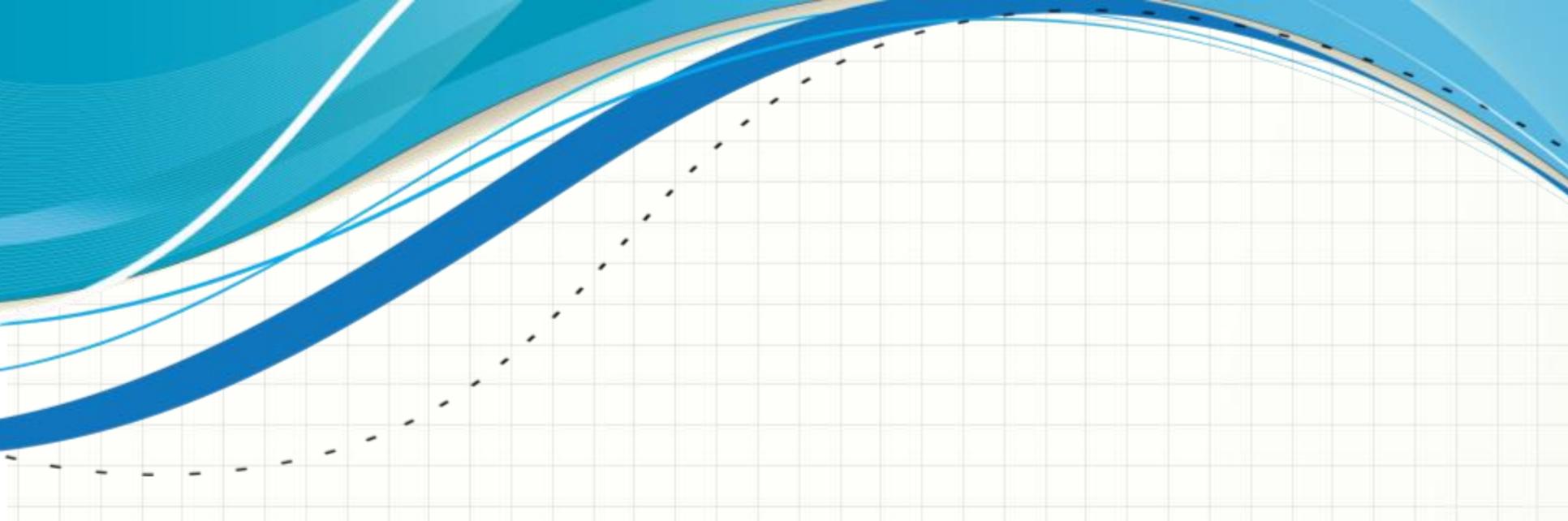
Resumo

- Estruturas de decisão: comportamento muda
 - De acordo com proposições lógicas
- Estruturas de decisão: trazem flexibilidade
 - Computador resolve sequências de cálculos sem intervenção humana
- Não deixe de praticar!
- **TAREFA!**
 - Lista Aula 8!

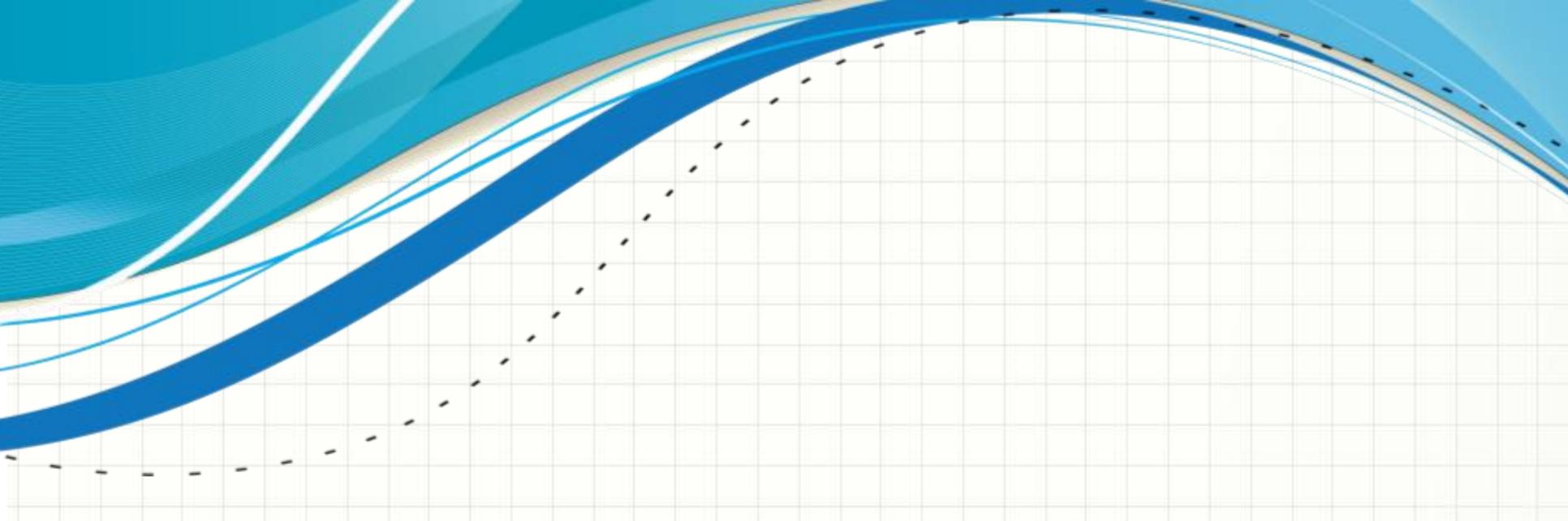
Próxima Aula



- Vamos exercitar?
 - Só se aprende a desenvolver algoritmos...
 - ...programando!



PERGUNTAS?



**BOM DESCANSO
A TODOS!**