Lógica de Programação para Engenharia

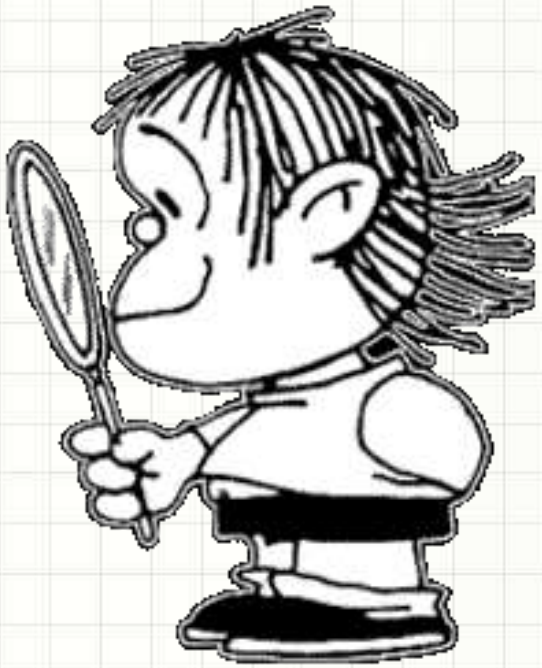
Estrutura de Repetição

Prof. Dr. Daniel Caetano

2018 - 1

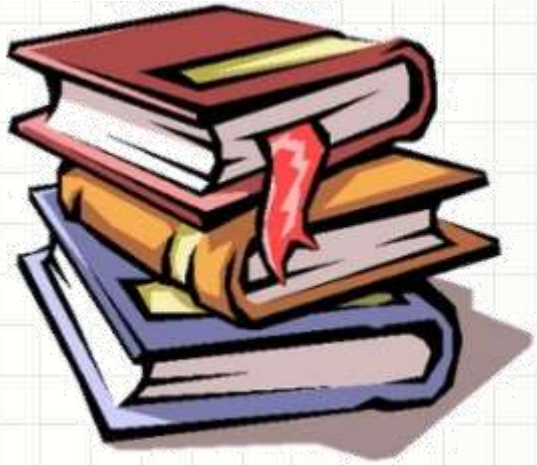
Objetivos

- Entender o que é uma estrutura de repetição
- Compreender como implementar as repetições
- Capacitar para a criação de algoritmos que envolvam repetição



- **Atividades Aula 10 – SAVA!**

Material de Estudo



Material	Acesso ao Material
Notas de Aula e Apresentação	http://www.caetano.eng.br/ (Lógica de Programação para Eng. – Aula 10)
Material Didático	Lógica de Programação, págs 119 a 124.
Aula Online	Aula 7
Biblioteca Virtual	“Lógica de Programação – Fundamentos da Programação de Computadores”, págs 93 a 144.

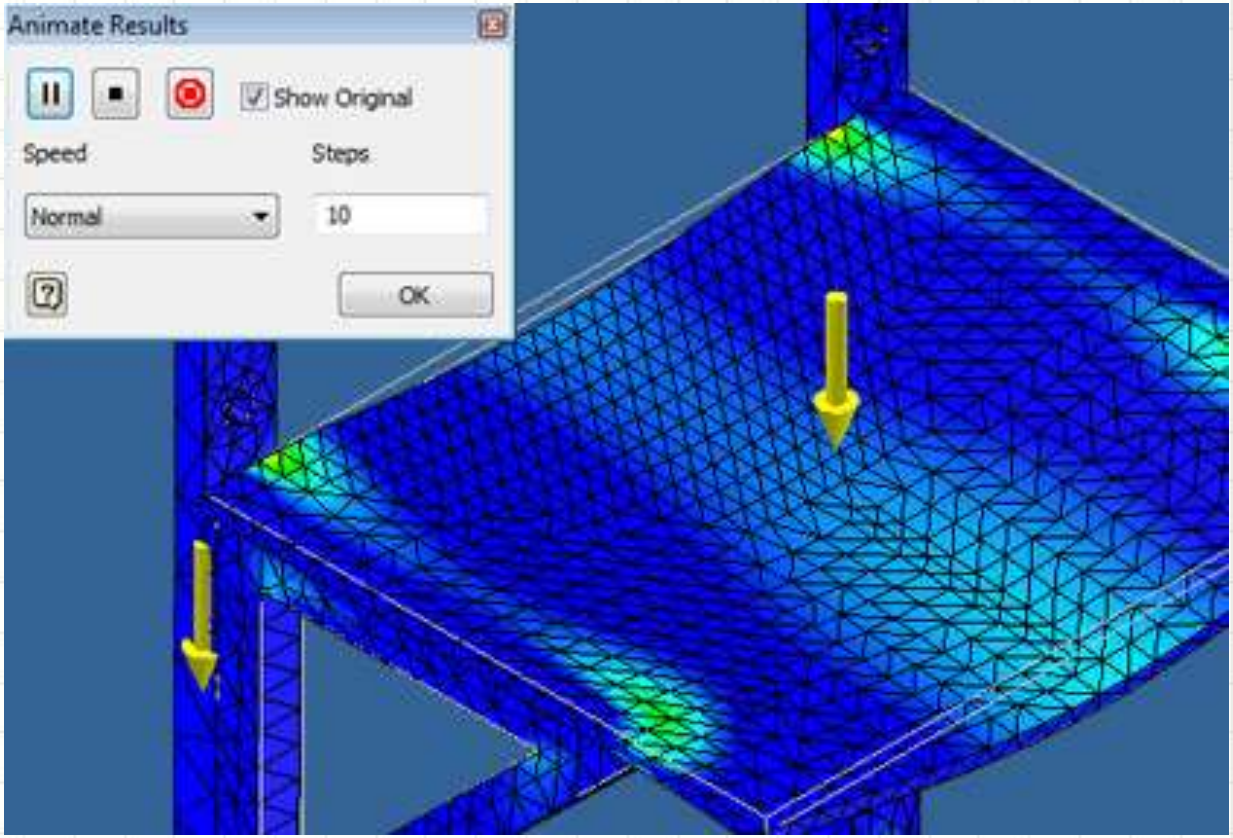
O Que é Estrutura de Repetição?

O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Solicitação de entradas do usuário

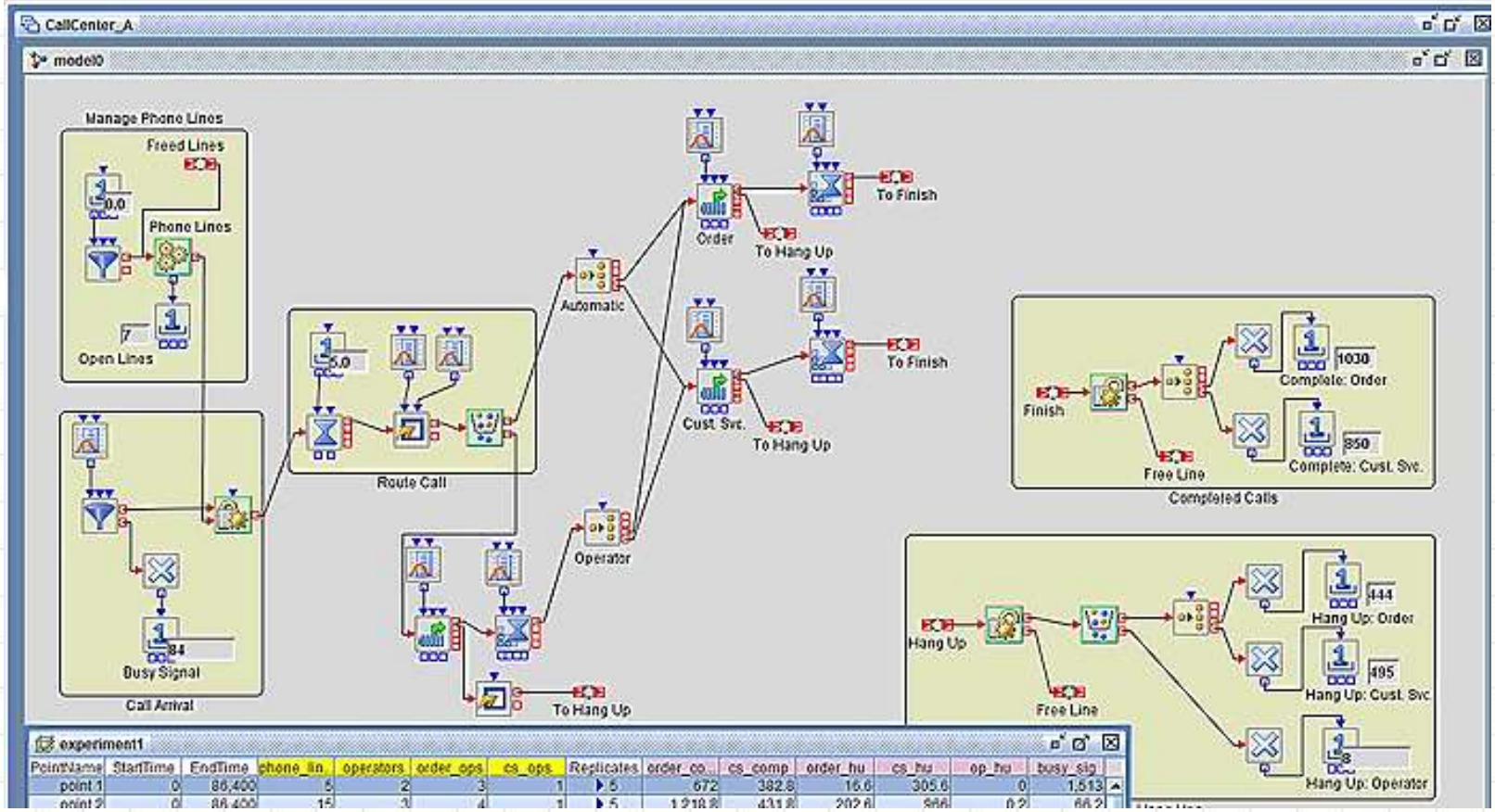
O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Procedimentos repetitivos



O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Procedimentos repetitivos



O que são Estruturas de Repetição?

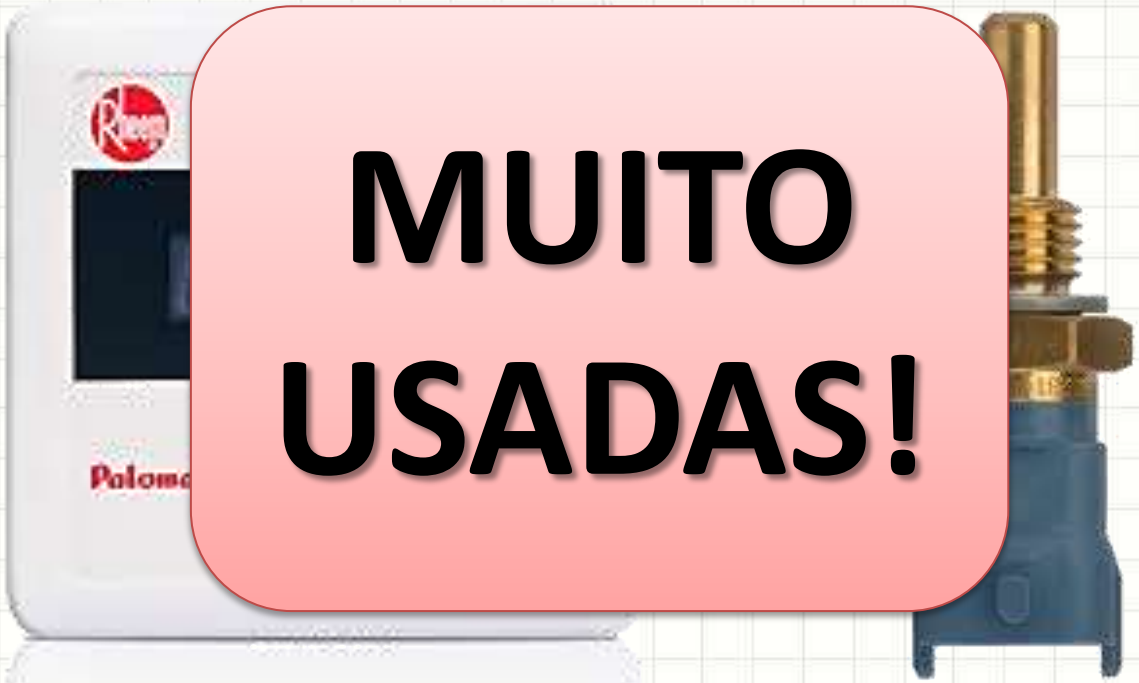
- Repetir continuamente um código
  - Esperar que alguma coisa ocorra

O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Esperar que alguma coisa ocorra

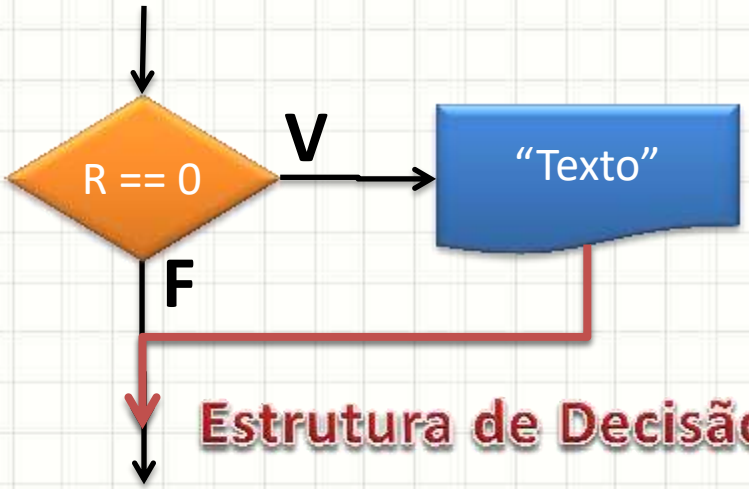
O que são Estruturas de Repetição?

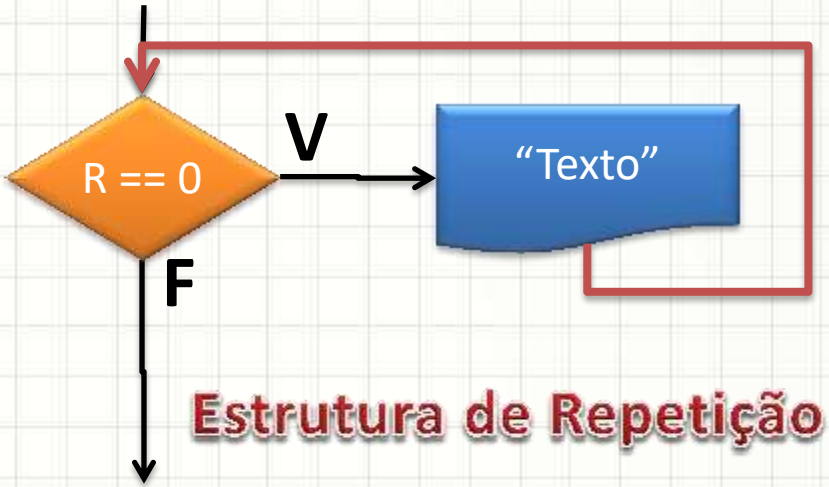
- Repetir continuamente um código
  - Esperar que alguma coisa ocorra



O que são Estruturas de Repetição?

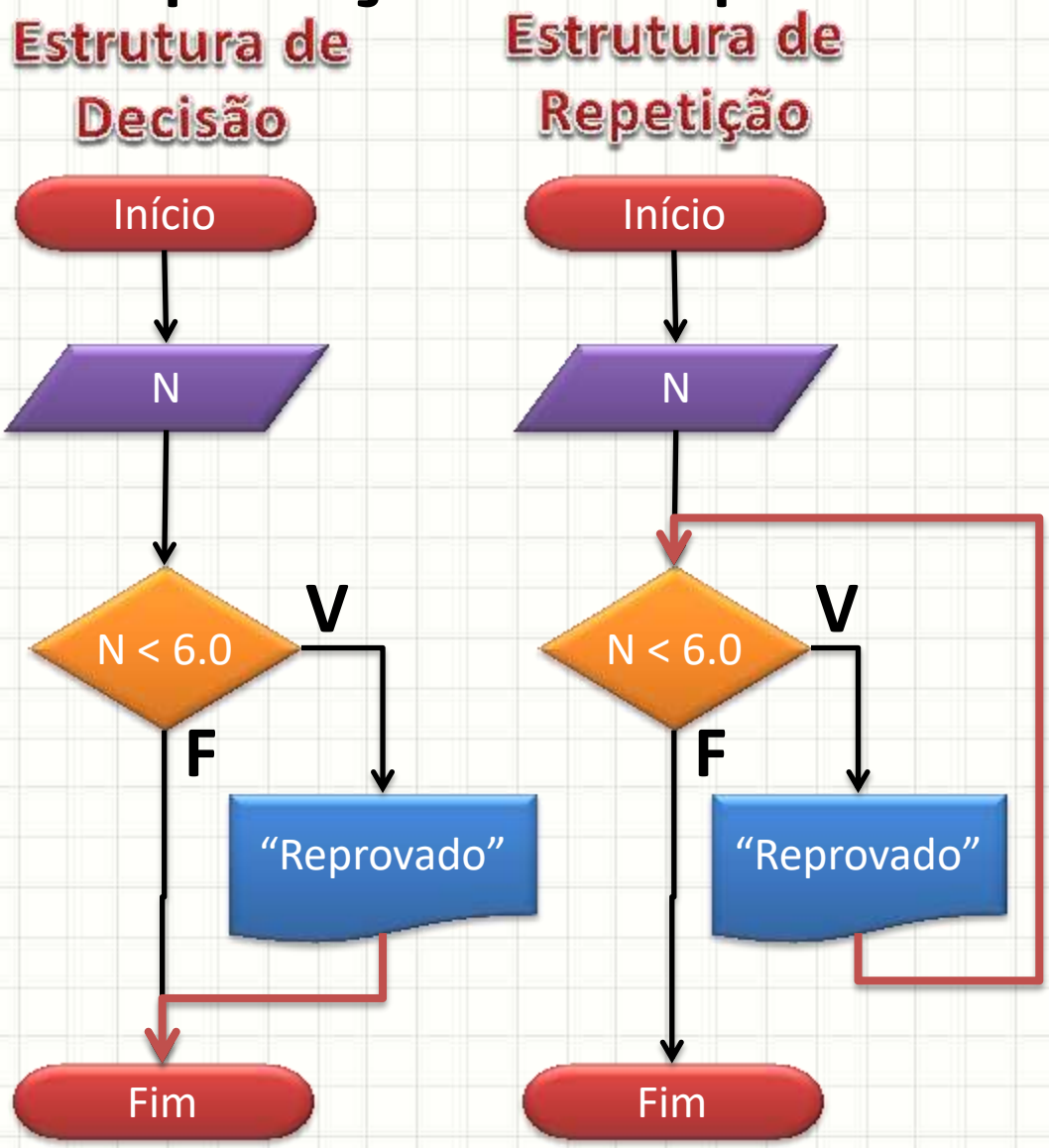
- Estrutura de Decisão: **se** executo um código
- Estrutura de Repetição é parecida...
  - "Enquanto o quê" um código será repetido





- Diferença: para onde vai a execução depois?

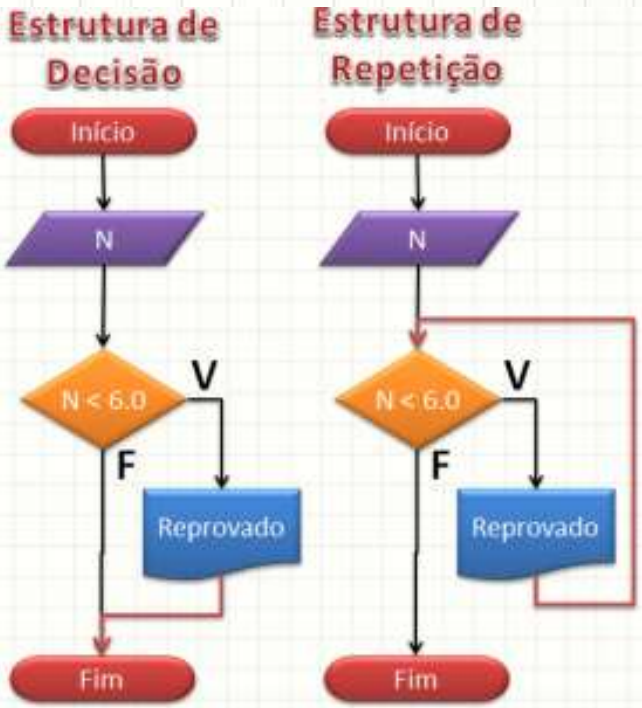
Repetição Simples na Prática



- Repetição: decisão do tipo "**enquanto isso for verdadeiro, continue repetindo!**"
- O que ocorre no código ao lado?

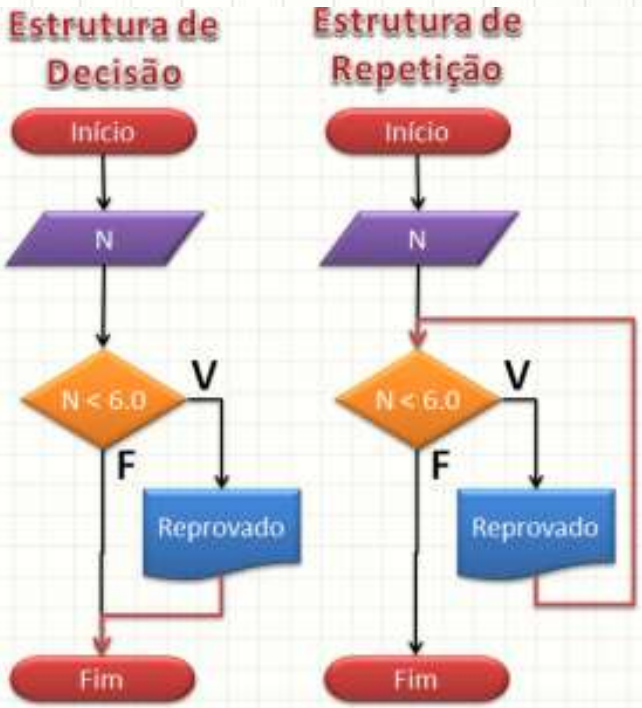
Repetição Simples na Prática

```
#include <iostream>
using namespace std;
main() // Teste
{
    float N;
    cout << "Digite a nota: ";
    cin >> N;
    if ( N < 6.0)
        cout << "Reprovado" << endl;
}
```



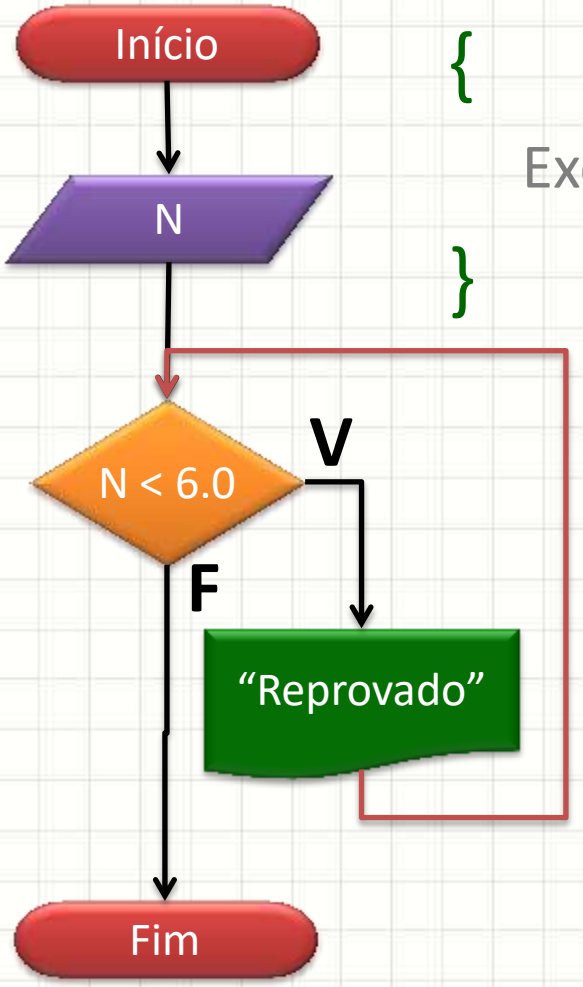
Repetição Simples na Prática

```
#include <iostream>
using namespace std;
main() // Teste
{
    float N;
    cout << "Digite a nota: ";
    cin >> N;
    while ( N < 6.0)
        cout << "Reprovado" << endl;
}
```



Forma Geral do While

```
while ( proposição_lógica )
{
    Executa enquanto a proposição for verdadeira
}
```

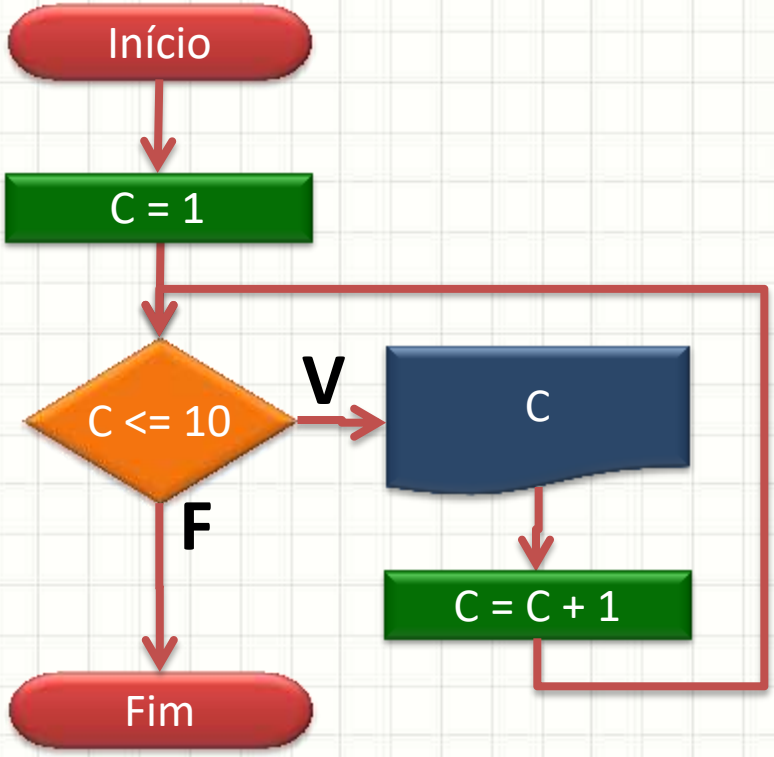


ATENÇÃO

No WHILE **não existe** ELSE!

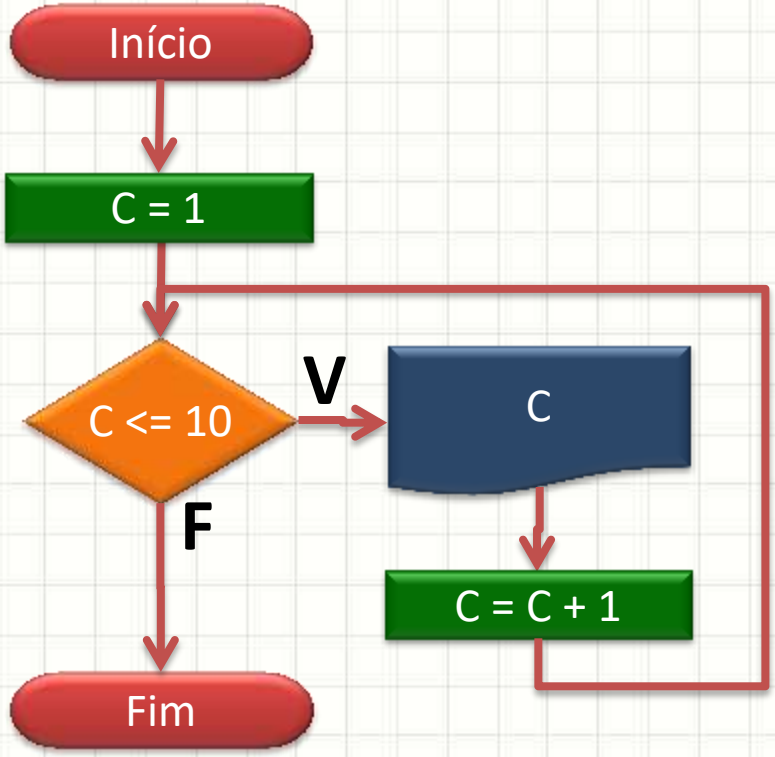
Exemplo

- Crie um algoritmo que imprime os números inteiros de 1 a 10



Exemplo

- Crie um algoritmo que imprime os números inteiros de 1 a 10

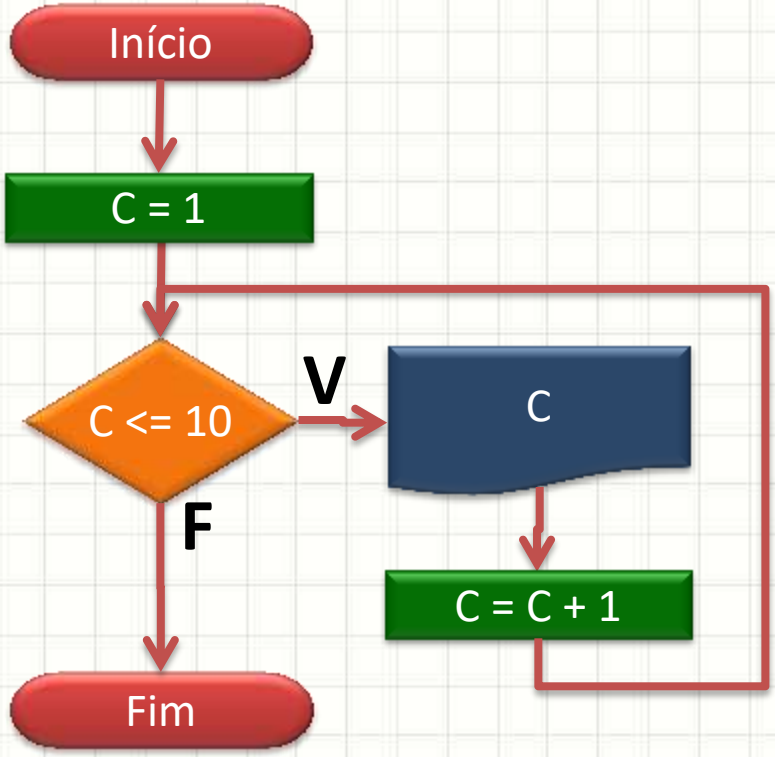



```
#include <iostream>
using namespace std;
main() // Teste de Repetição
{
    int C;
    C = 1;
    while ( C <= 10 )
        cout << C << endl;
        C = C + 1;
}
```

Está certo?

Exemplo

- Crie um algoritmo que imprime os números inteiros de 1 a 10

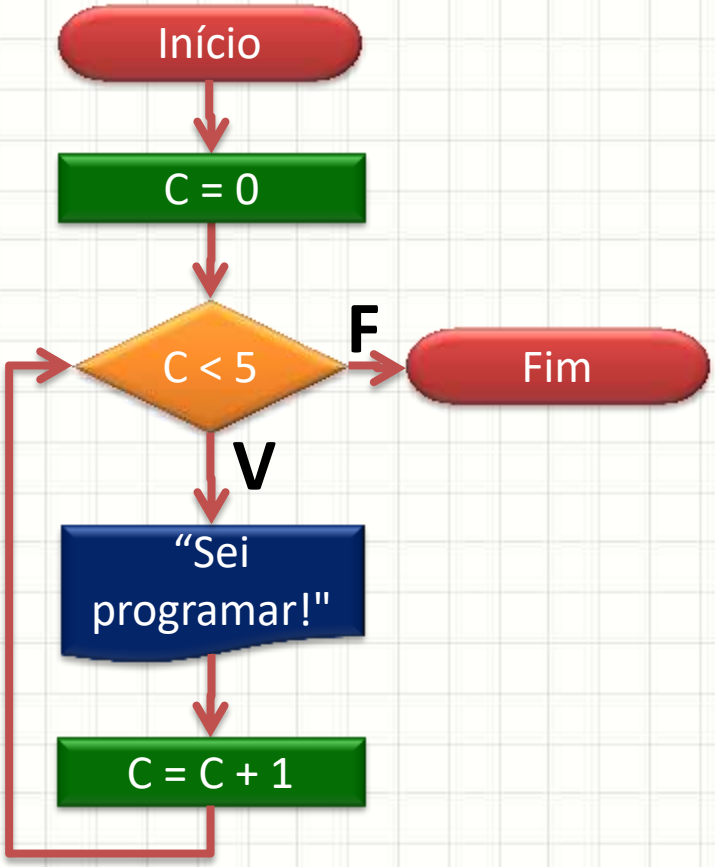



```
#include <iostream>
using namespace std;
main() // Teste de Repetição
{
    int C;
    C = 1;
    while ( C <= 10 )
    {
        cout << C << endl;
        C = C + 1;
    }
}
```

ATIVIDADE

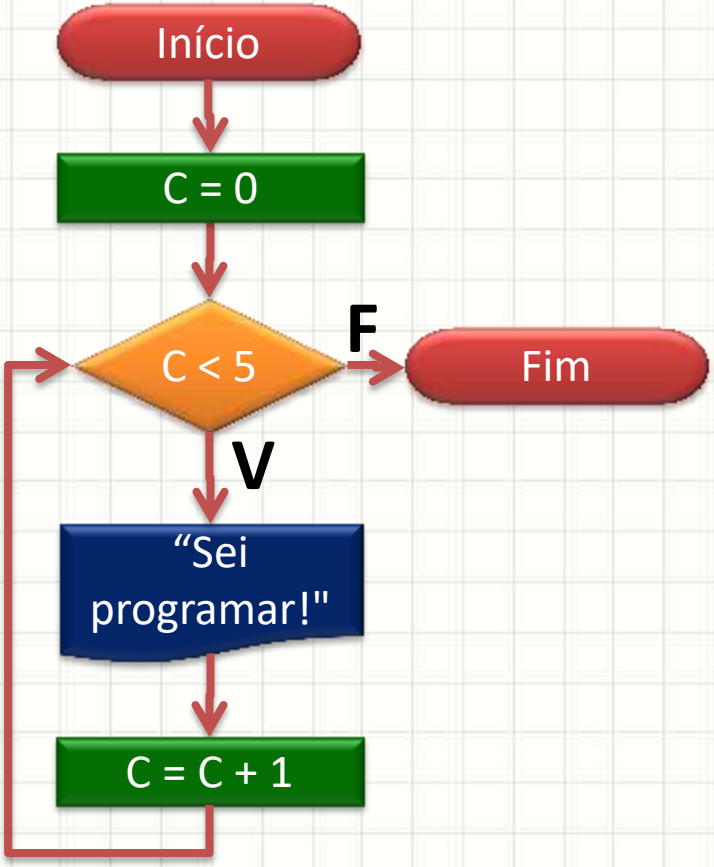
Exercício A

- Faça um programa que imprima 5 vezes a mensagem “Sei programar!”



Exercício A

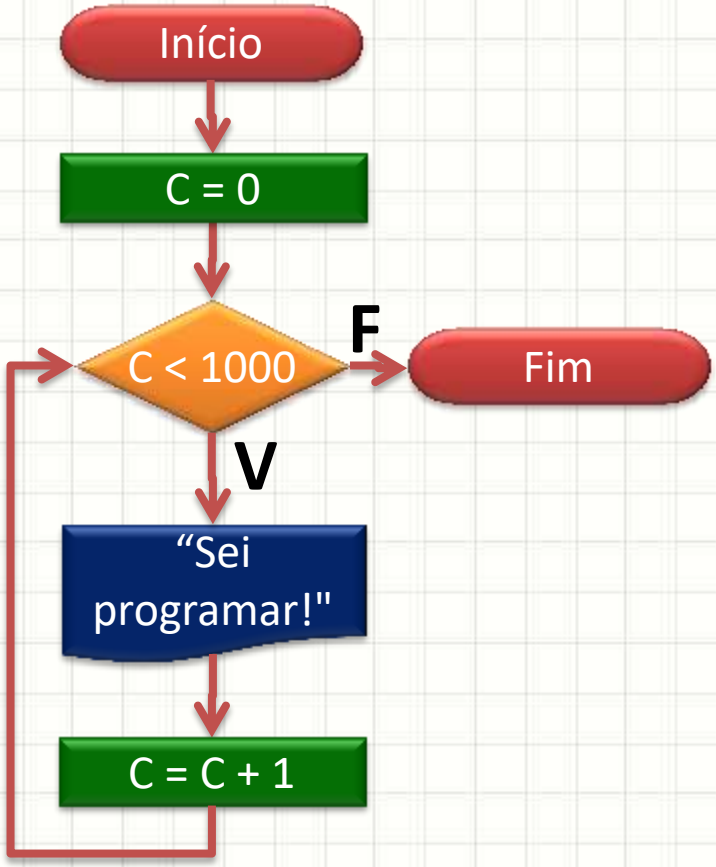
- Faça um programa que imprima 5 vezes a mensagem "Eu sei programar!"




```
#include <iostream>
using namespace std;
main() // Imprime 5 vezes
{
    int C;
    C = 0;
    while ( C < 5 )
    {
        cout << "Sei programar!";
        cout << endl;
        C = C + 1;
    }
}
```

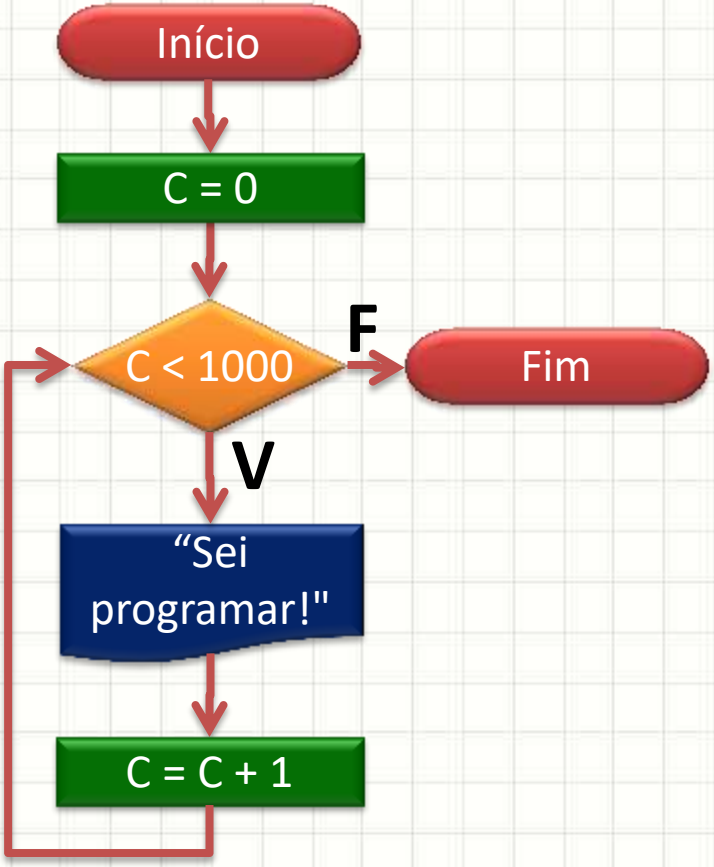
Exercício B

- Faça um programa que imprima 1000 vezes a mensagem “Sei programar!”



Exercício B

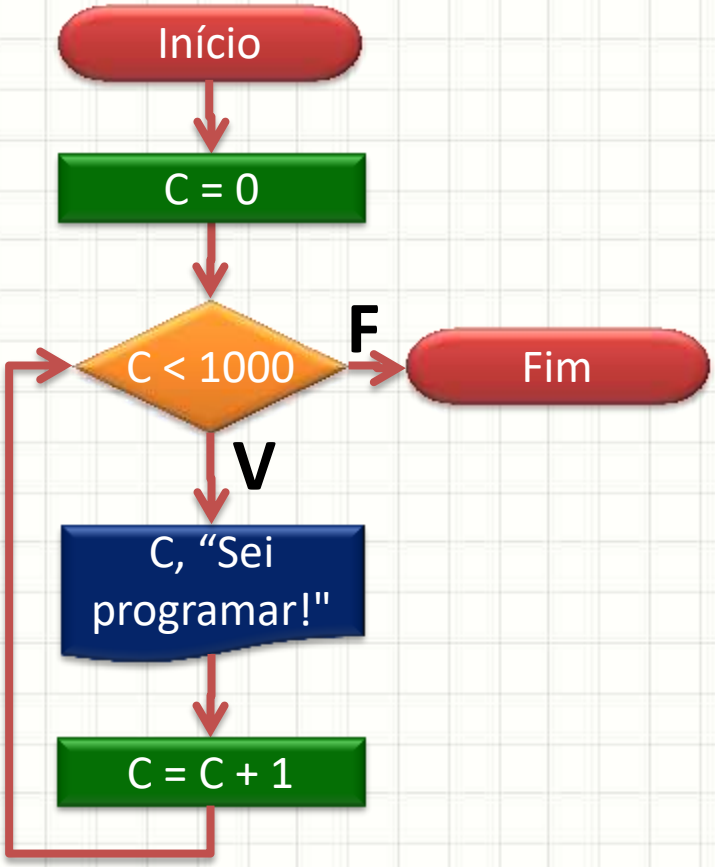
- Faça um programa que imprima 5 vezes a mensagem "Eu sei programar!"




```
#include <iostream>
using namespace std;
main() // Imprime 1000 vezes
{
    int C;
    C = 0;
    while ( C < 1000 )
    {
        cout << "Sei programar!";
        cout << endl;
        C = C + 1;
    }
}
```

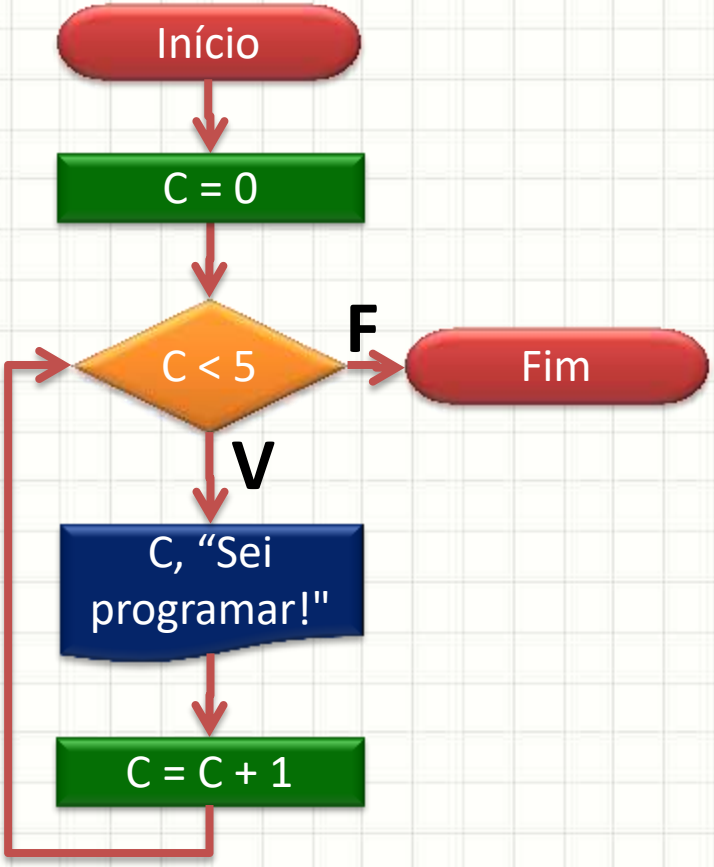
Exercício C

- Modifique o programa para imprimir o **contador** antes de imprimir o texto “Sei programar!”



Exercício C

- Modifique o programa para imprimir o contador antes de imprimir o texto “Sei programar!”



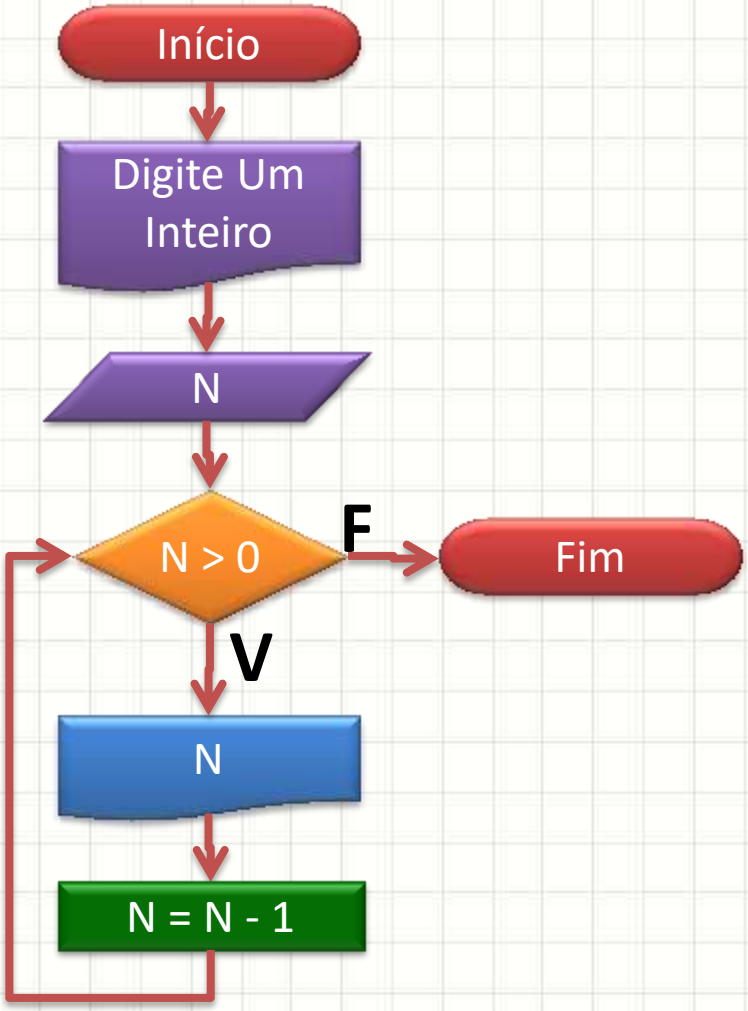

```
#include <iostream>
using namespace std;
main() // Imprime contador
{
    int C;
    C = 0;
    while ( C < 1000 )
    {
       cout << C << " – ";
       cout << "Sei programar!";
       cout << endl;
       C = C + 1;
    }
}
```

Repetindo Código N Vezes

(De trás para Frente)

Repetindo Código N Vezes

- Observe:




```
#include <iostream>
using namespace std;
main()
{
    int N;
    cout << "Digite um Inteiro: ";
    cin >> N;
    while ( N > 0 )
    {
      cout << N << endl;
      N = N - 1;
    }
}
```

Repetindo Código N Vezes

```
#include <iostream>
using namespace std;
main()
{
    int N;
    cout << "Digite um Inteiro: ";
    cin >> N;
    while ( N > 0 )
    {
      cout << N << endl;
      N = N - 1;
    }
}
```

1. Digite Este Programa

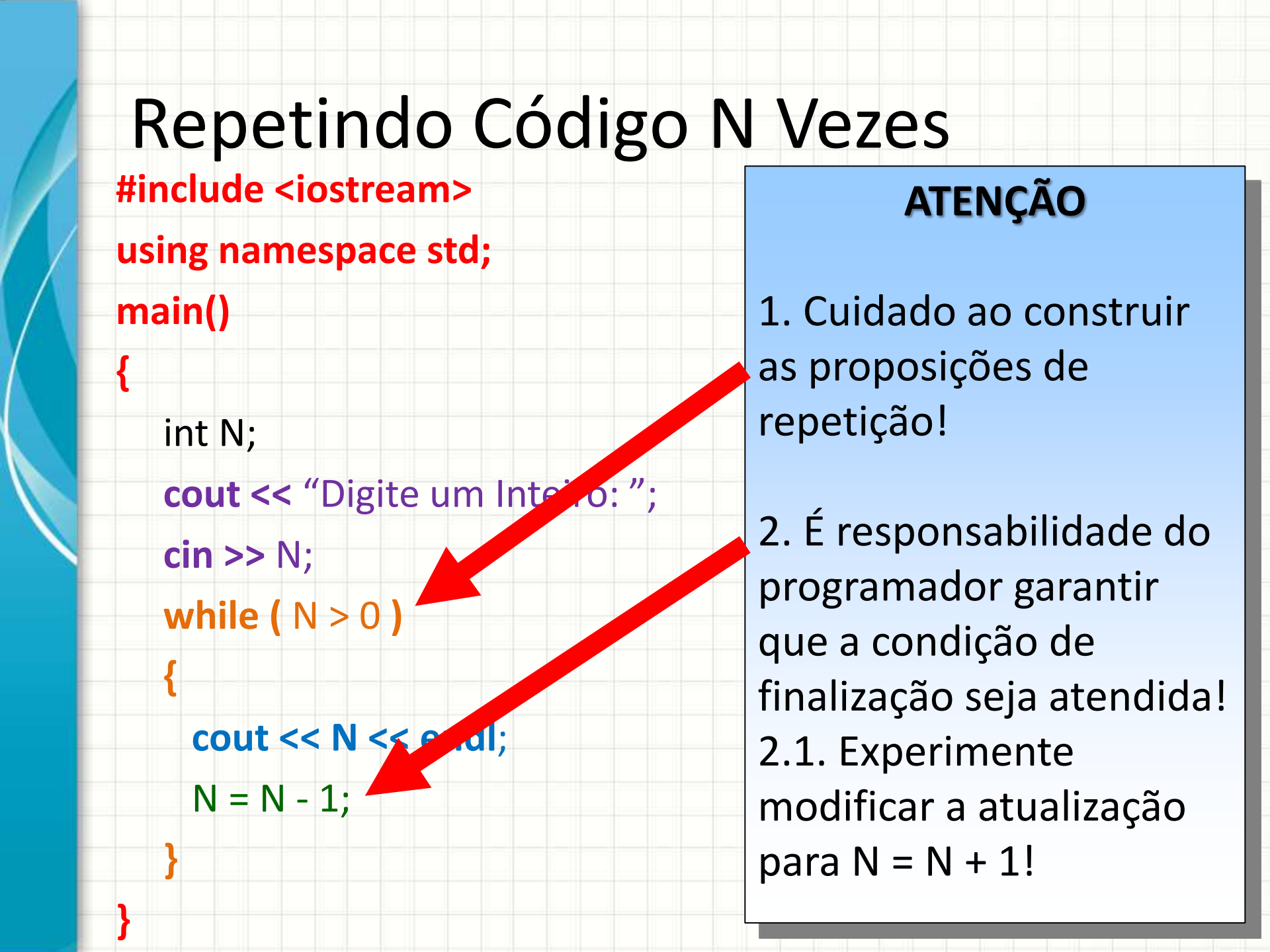
2. Expermente executá-lo com diferentes valores. Exemplo: 5, 1, 0, -10

3. Experimente modificar o **while** para que a condição seja **N >= 0** .

4. O que aconteceu / mudou em cada caso?

Repetindo Código N Vezes

```
#include <iostream>
using namespace std;
main()
{
    int N;
    cout << "Digite um Inteiro: ";
    cin >> N;
    while ( N > 0 )
    {
      cout << N << endl;
      N = N - 1;
    }
}
```



ATENÇÃO

1. Cuidado ao construir as proposições de repetição!

2. É responsabilidade do programador garantir que a condição de finalização seja atendida!
2.1. Experimente modificar a atualização para N = N + 1!

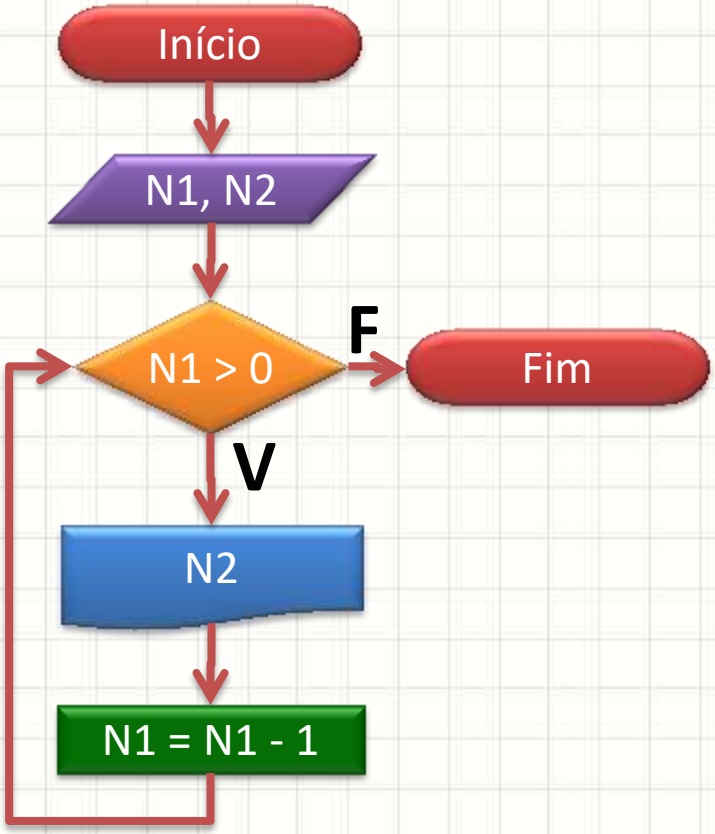
ATIVIDADE

Exercício D

- Faça um programa que

a) Leia dois números N1 e N2

b) Imprima N1 vezes o valor de N2.

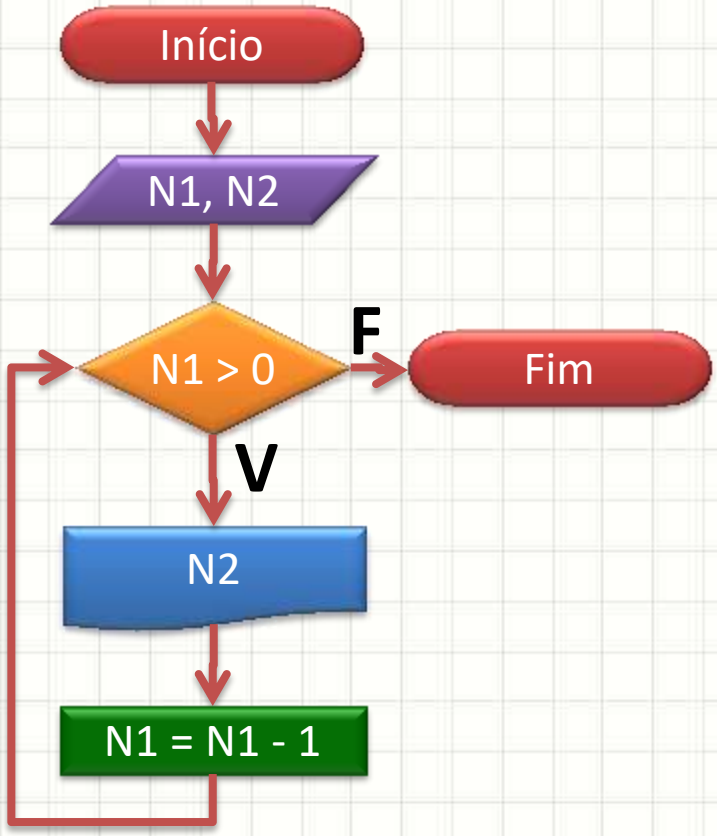


Exercício D

- Faça um programa que

a) Leia dois números N1 e N2

b) Imprima N1 vezes o valor de N2.




```
#include <iostream>
using namespace std;
main()
{
    int N1, N2;
    cout << "Digite um No.: ";
    cin >> N1;
    cout << "Digite outro No.: ";
    cin >> N2;
    while ( N1 > 0 )
    {
       cout << N2 << endl;
       N1 = N1 - 1;
    }
}
```

Conclusões

Resumo

- Repetição: amplia a utilidade do computador
- Decisão que verifica “se continua repetindo”.
- Não deixe de praticar!
- **TAREFA: Lista Aula 10!**

- Só existe um tipo de estrutura de repetição?
  - Não!
- Casos diferentes, estruturas mais adequadas!

PERGUNTAS?