Inovação Tecnológica e Empreendedorismo

Programas Sequenciais e Estruturas de Decisão

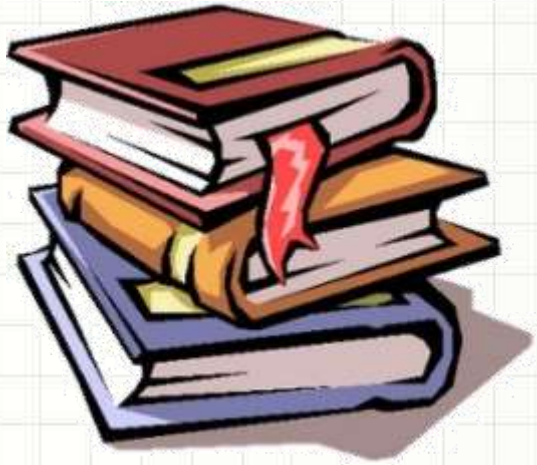
Prof. Dr. Daniel Caetano

2017 - 2

Objetivos

- Rever a estruturação básica de um programa
- Rever as estruturas de decisão
- Rever as estruturas de repetição
- Programas com decisão no Arduino

Material de Estudo



Material	Acesso ao Material
Apresentação	http://www.caetano.eng.br/ (Inovação e Empreendedorismo – Aula B)
Material Didático	da disciplina Algoritmos
Biblioteca Virtual	“algoritmos”, “programação”

CONTEXTUALIZAÇÃO

O que são Algoritmos

- Algoritmo: fabricar vinho para venda
  - Plantar a uva
  - Colher a uva
  - Amassar a uva
  - Deixar fermentar
  - Engarrafar
  - Distribuir para a venda
- Envolve
  - Tarefas/Processos
  - Decisões

Complilação

- Processo de Compilação

Programador

```
#include <io...
int main(void)
  {
  cout << “Oi”;
```

Código Fonte

Compilador

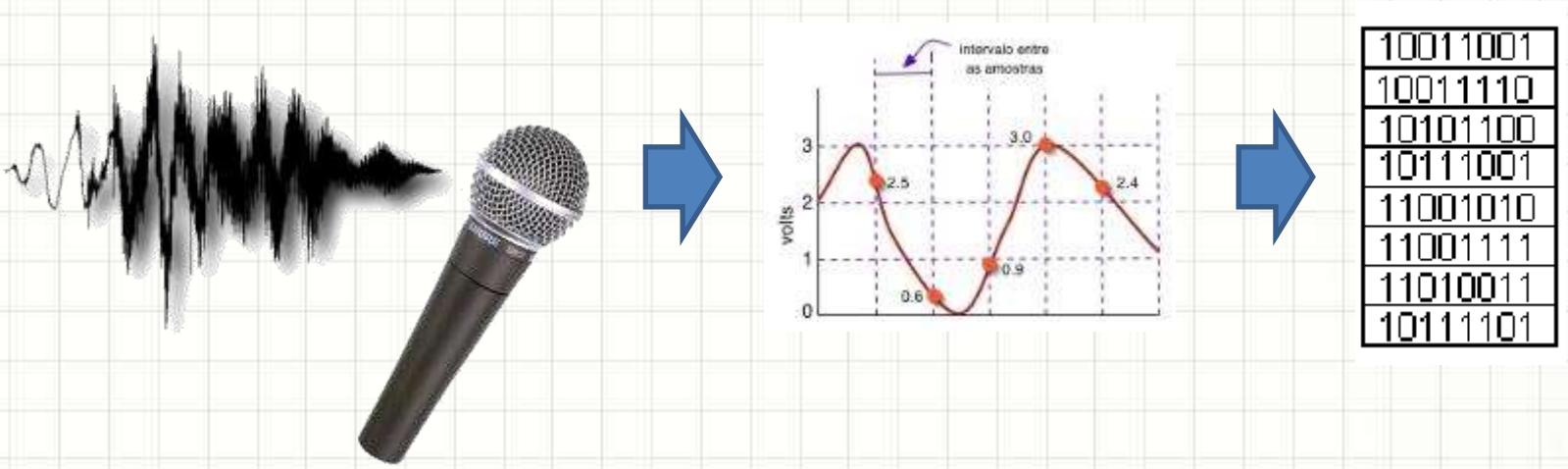
```
001010101010
101010101010
110111011011
111110010101
```

Código Binário de PC

Computador PC

Dispositivos de Entrada e Saída

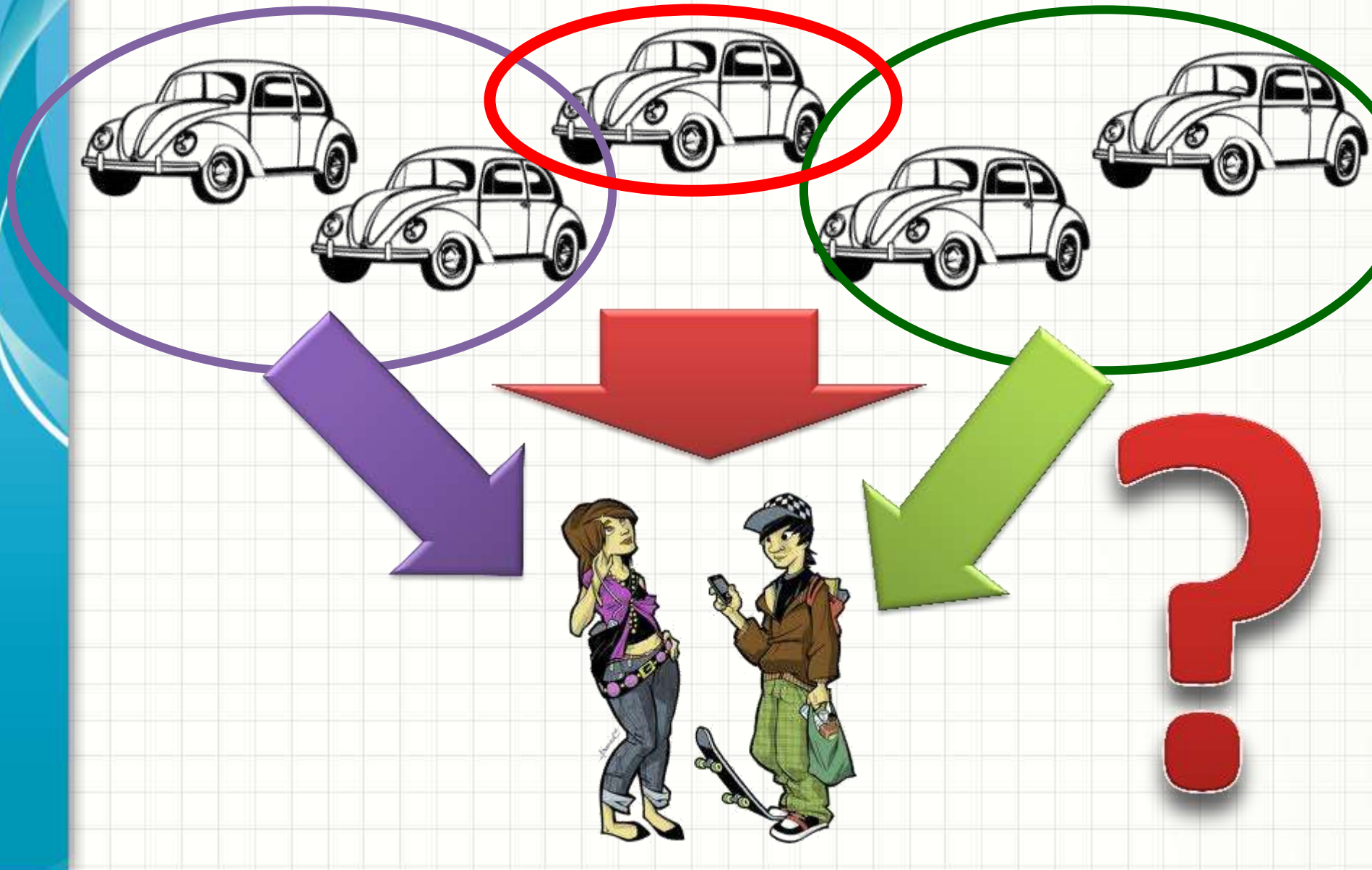
- **Dispositivos de Entrada**: convertem informações externas (usualmente fornecidas pelo usuário) em números para o computador



Exemplo de Algoritimo Sequencial:

Resto de Divisão

Resto de Divisão



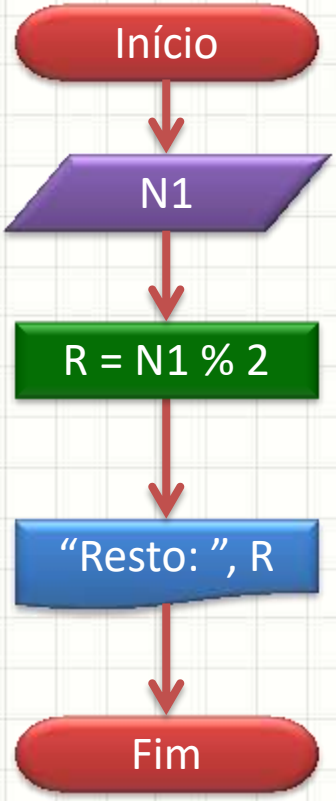
Número Par ou Ímpar?

- Como determinar se um número é par?
- Par: divisível por dois
- O que significa ser divisível por 2?
- Significa que o resto da divisão por 2 é 0!

- Vamos experimentar:
  - Algoritmo que imprime resto da divisão por 2

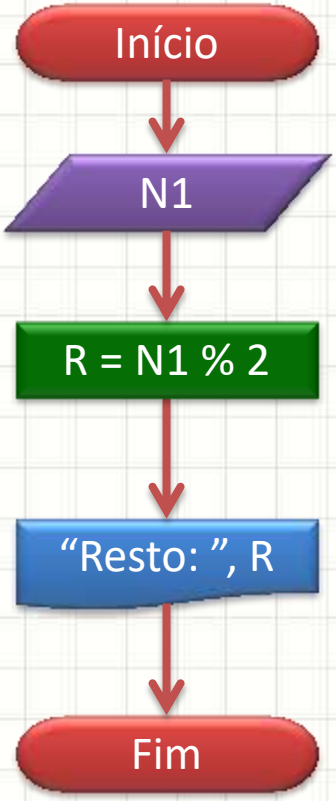
Verificando Paridade

- Linguagem Natural
  1. Leia um número
  2. Calcule o resto da divisão por 2
  3. Imprima o resto
- Fluxograma



Verificando Paridade

- Fluxograma



- Portugol

```
Algoritmo “Calcula Paridade”
Var
INTEIRO : N1, R
Inicio
Escreva(“Digite Um Número:”)
Leia(N1)
R ← N1 % 2
Escreva(“Resto:”, R)
FimAlgoritmo
```

Verificando Paridade

- Portugol

```
Algoritmo "Calcula Paridade"
Var
INTEIRO : N1, R
Inicio
Escreva("Digite Um Número:")
Leia(N1)
R ← N1 % 2
Escreva("Resto:", R)
FimAlgoritmo
```

- Linguagem C

```
#include <iostream>
using namespace std;
int main(void)  {
   int N1, R;
   cout << "Digite Um Número:";
   cin >> N1;
   R = N1 % 2;
   cout << "Resto: " << R;
}
```

Verificando Paridade

- Portugol

```
Algoritmo "Calcula Paridade"
Var
INTEIRO
Inicio
Escreva(
Leia(N1)
R ← N1 % 2
Escreva("Resto:", R)
FimAlgoritmo
```

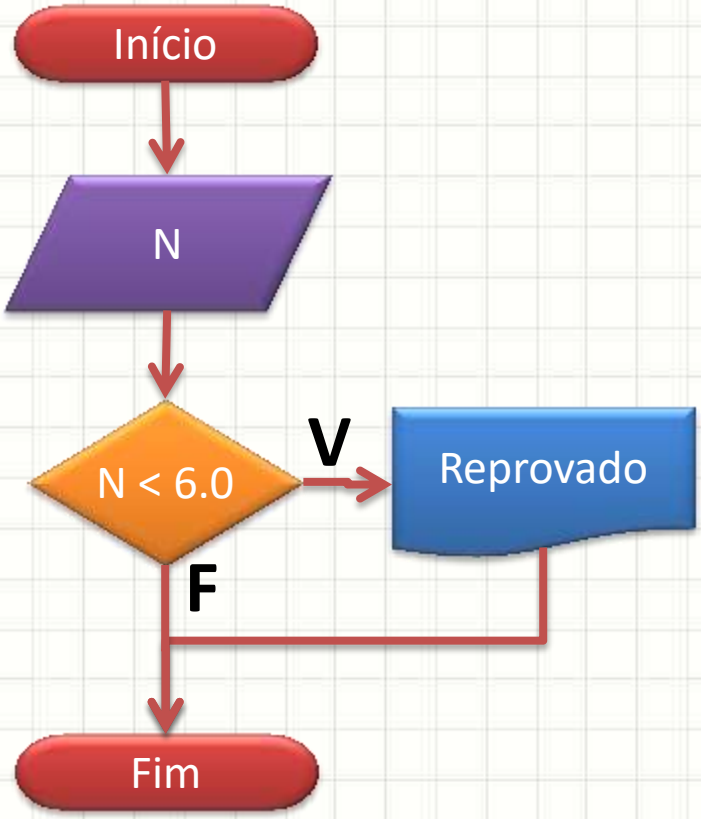
- Linguagem C

```
#include <iostream>
                                    úmero:";
    R = N1 % 2;
    cout << "Resto: " << R;
}
```

Como imprimir "Par" se o número é par e "Ímpar" se o número é ímpar?

Decisão no Código

Como Fica a Decisão no Código?

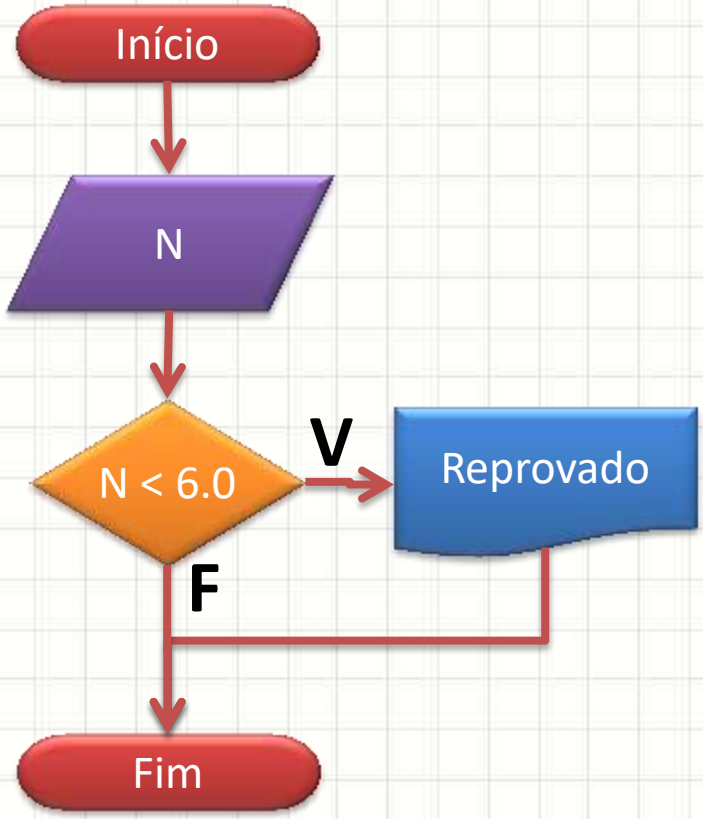


- Se Nota < 6.0 imprime que aluno está reprovado

- Português Estruturado

```
Se N < 6.0 Entao
        Inicio
        Escreva("Reprovado")
        FimSe
```

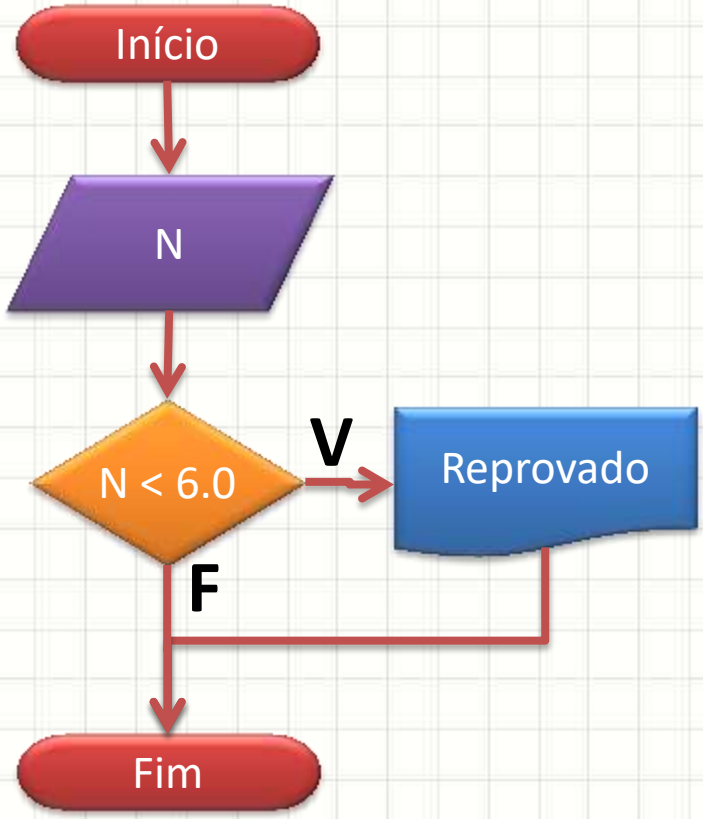
Como Fica a Decisão no Código?



- Português Estruturado

```
Algoritmo "Verifica Reprovação"
Var
 REAL : N
Inicio
Escreva("Digite a nota: ")
Leia(N)
Se N < 6.0 Entao
        Inicio
        Escreva("Reprovado")
        FimSe
FimAlgoritmo
```

Como Fica a Decisão no Código?

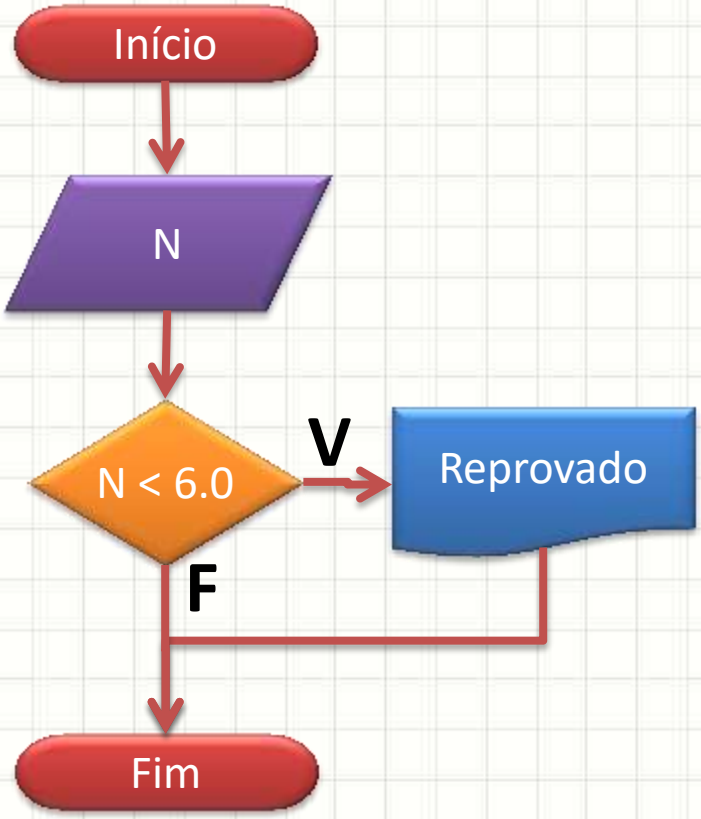


- Se Nota < 6.0 imprime que aluno está reprovado

- C / C++

```
if ( N < 6.0 ) {
        cout << "Reprovado";
        }
```

Como Fica a Decisão no Código?



- C/C++

```
#include <iostream>
using namespace std;

int main(void) {
    float N;
    cout << "Digite a nota: ";
    cin >> N;
    if ( N < 6.0 ) {
            cout << "Reprovado";
            }
}
```

Forma Geral do **Se** / **If**

- Português Estruturado

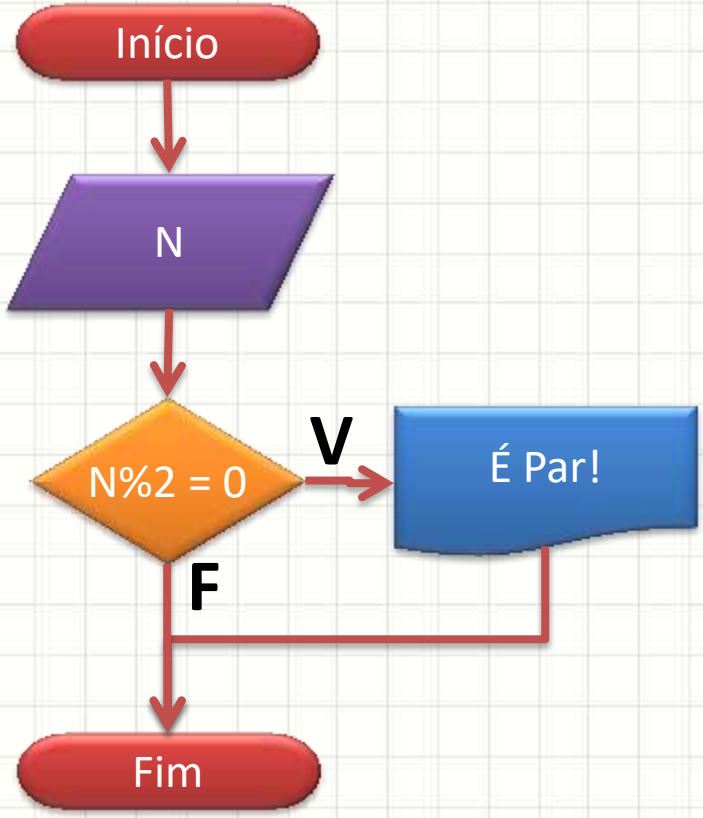
```
Se proposição_lógica Entao
        Inicio
        código a executar para proposição verdadeira
        FimSe
```

- C / C++

```
if ( proposição_lógica ) {
        código a executar para proposição verdadeira
        }
```

Outro Exemplo de Decisão

- Imprimir se número é par

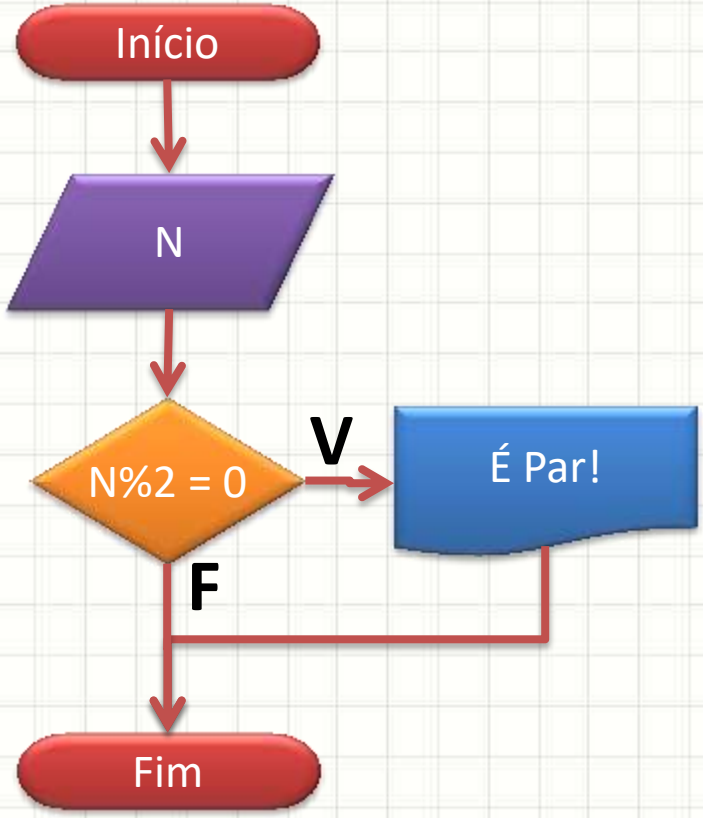


- Português Estruturado

```
Algoritmo "Verifica Paridade"
Var
INTEIRO : N
Inicio
Escreva("Digite um número: ")
Leia(N)
Se N%2 = 0 Entao
        Inicio
        Escreva("É Par!")
        FimSe
FimAlgoritmo
```

Como Fica a Decisão no Código?

- Imprimir se número é par



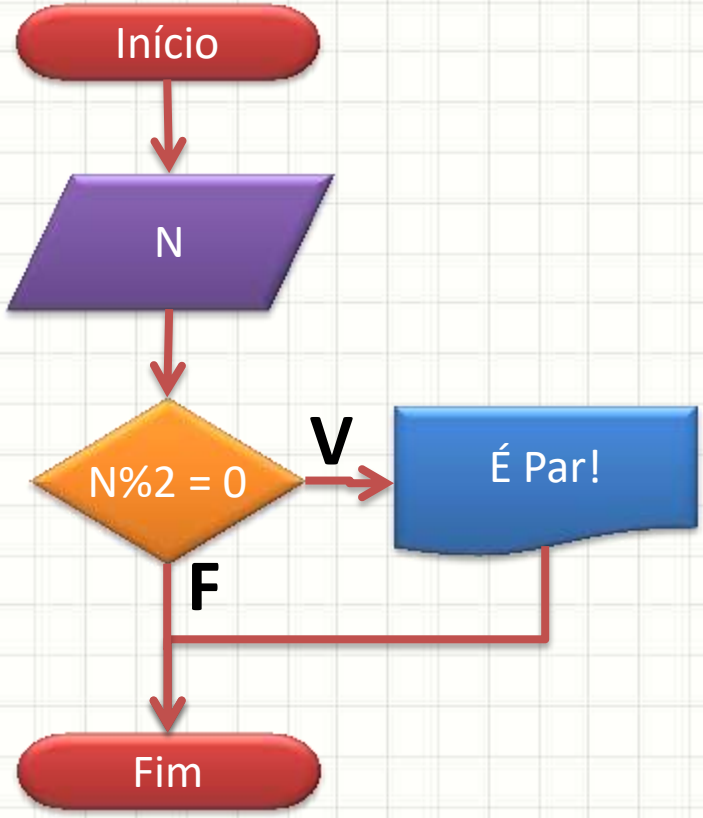
- C/C++

```
#include <iostream>
using namespace std;

int main(void) {
    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;

int main(void) {
    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 **=** ?

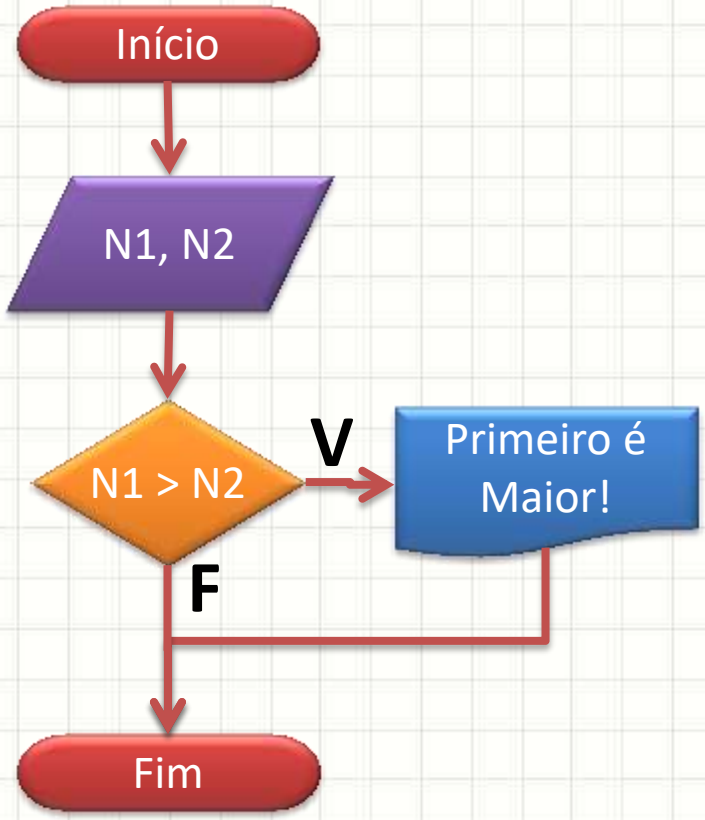
  = significa atribuição (guardar valor em variável)

  == significa comparação (resulta falso/verdadeiro)

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

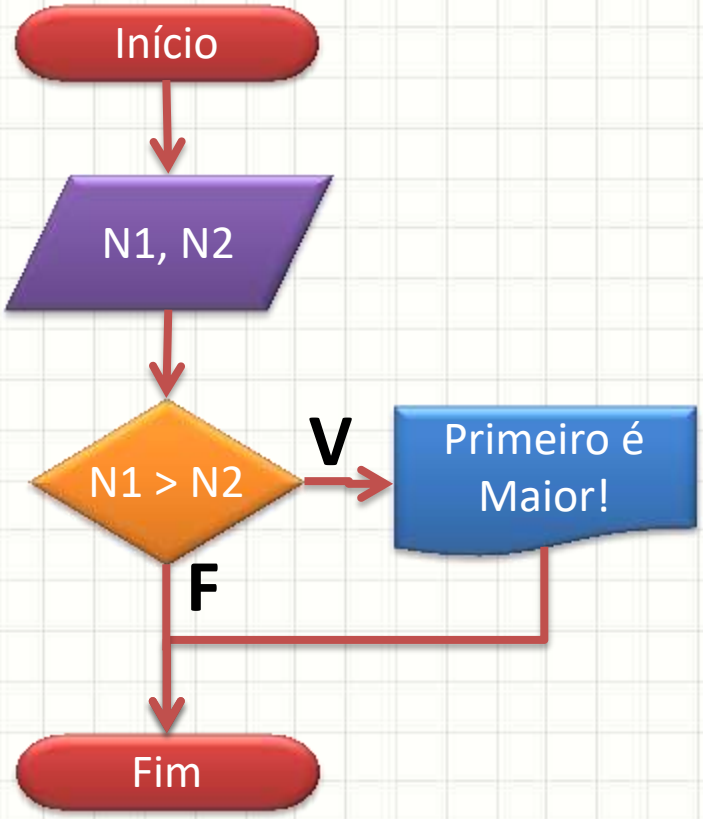
Exemplo

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



Exemplo

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

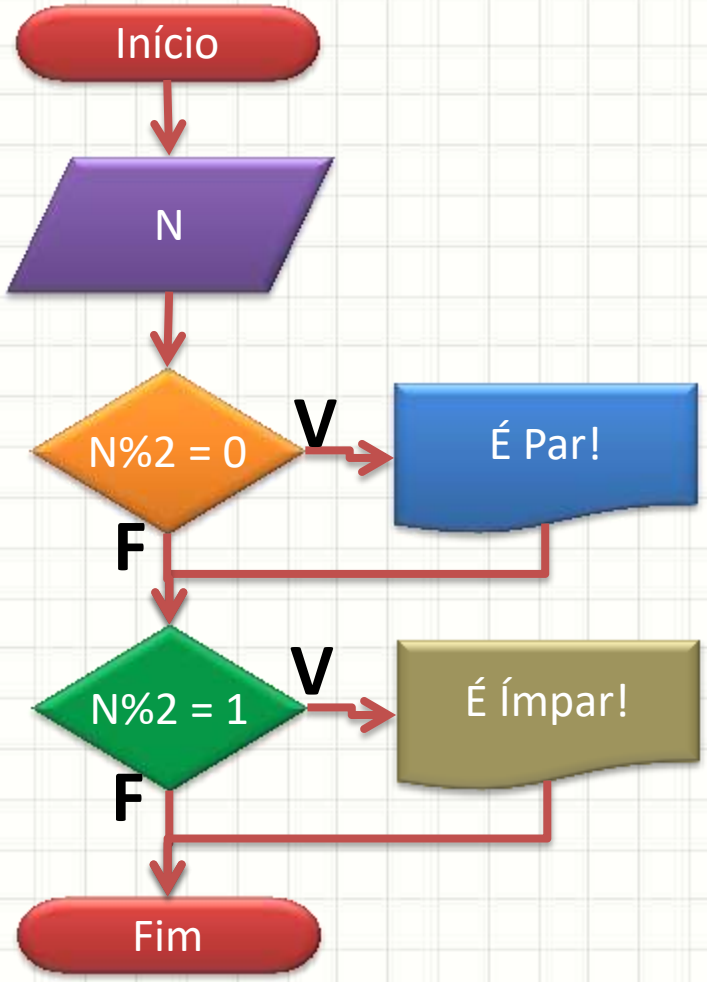



```
#include <iostream>
using namespace std;

int main(void) {
    int N1, N2;
    cout << "Digite um número: ";
    cin >> N1;
    cout << "Digite outro número: ";
    cin >> N2;
    if ( N1 > N2 ) {
        cout << "Primeiro é maior!";
        }
}
```

Múltiplas Decisões

- Verificar se número é par ou ímpar



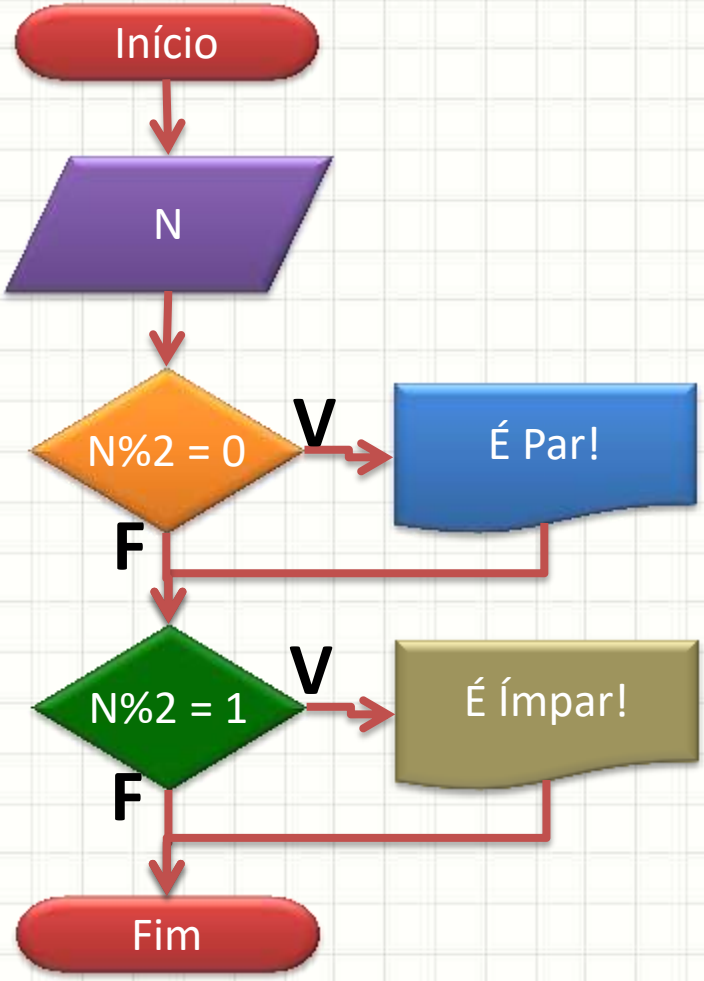

```
#include <iostream>
using namespace std;

int main(void) {
    int N;
    cout << "Digite um número: ";
    cin >> N;
    if ( N%2 == 0 ) {
            cout << "É Par!";
            }
    if ( N%2 == 1 ) {
            cout << "É Ímpar!";
            }
}
```

Estrutura de Decisão Composta

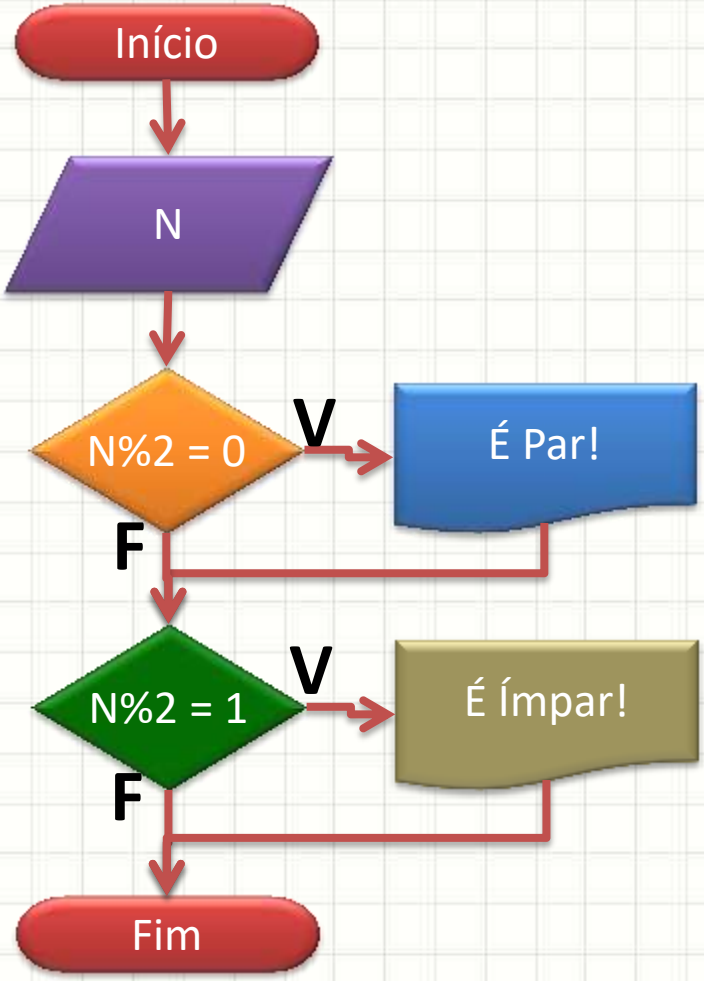
Estrutura de Decisão Composta

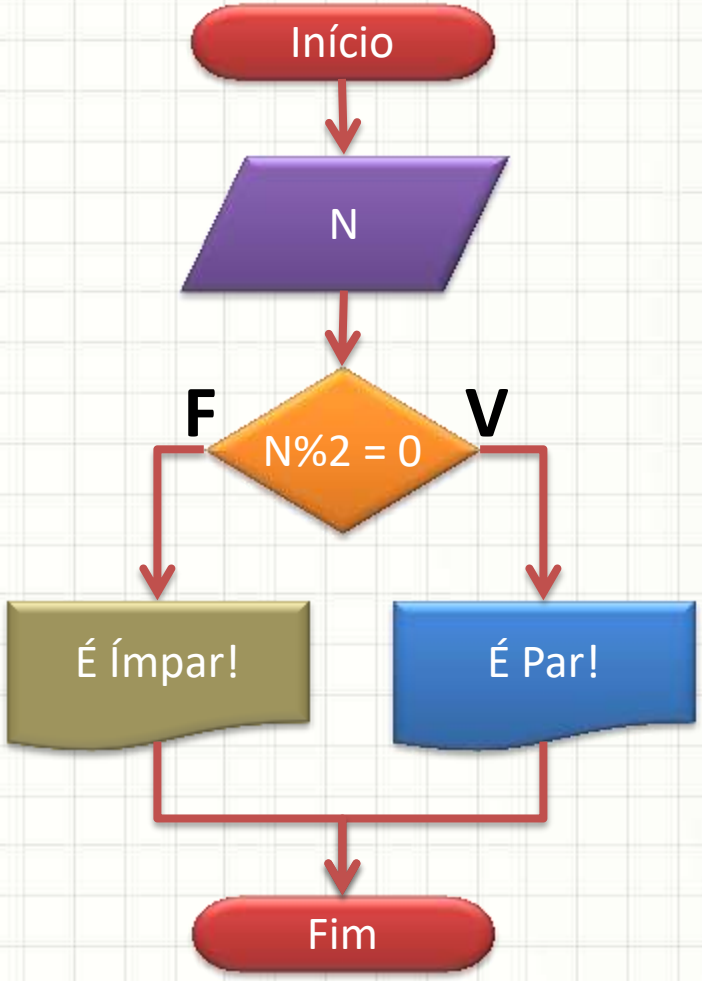
- Observe o fluxograma...



Estrutura de Decisão Composta

- Observe este outro... São iguais?



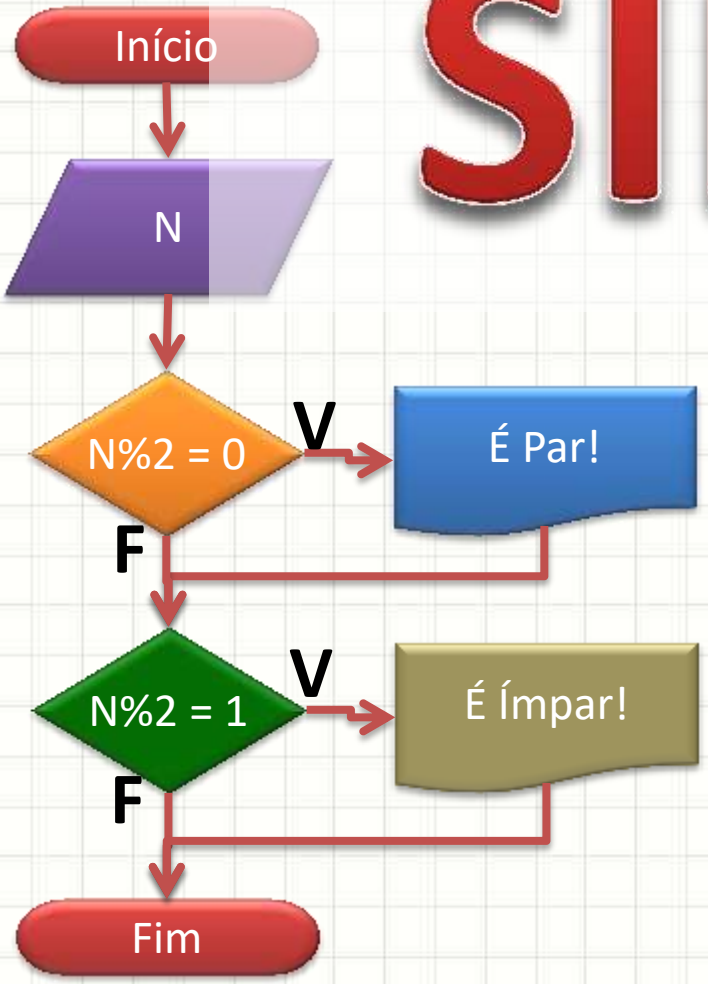


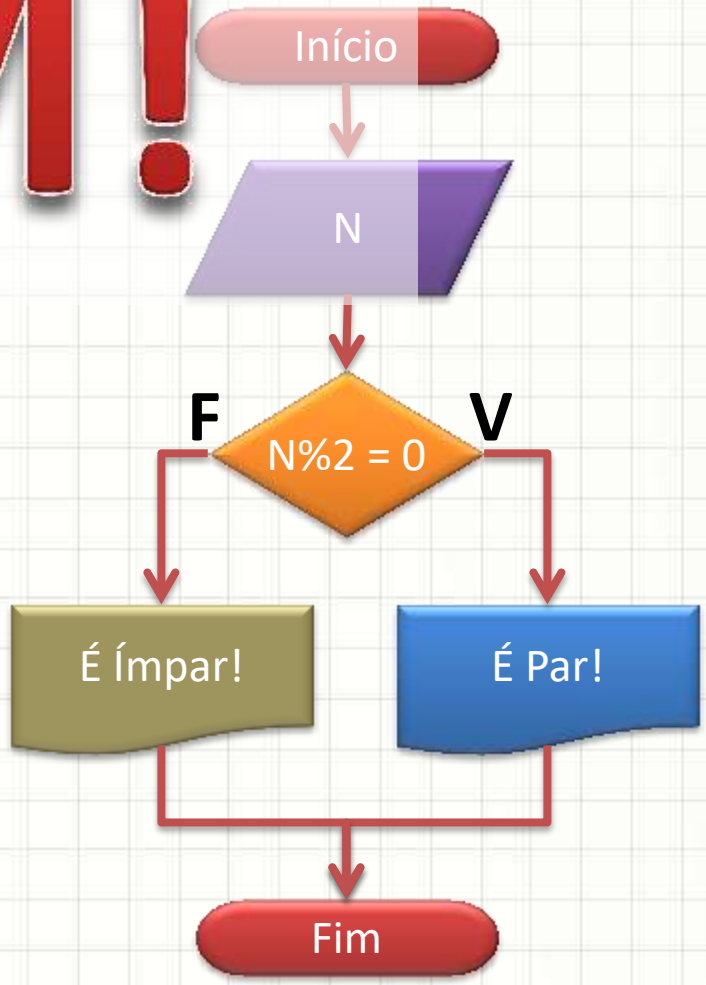
Estrutura de Decisão Composta

EM FUNÇÃO,

- Observe este outro... São iguais?

SIM!

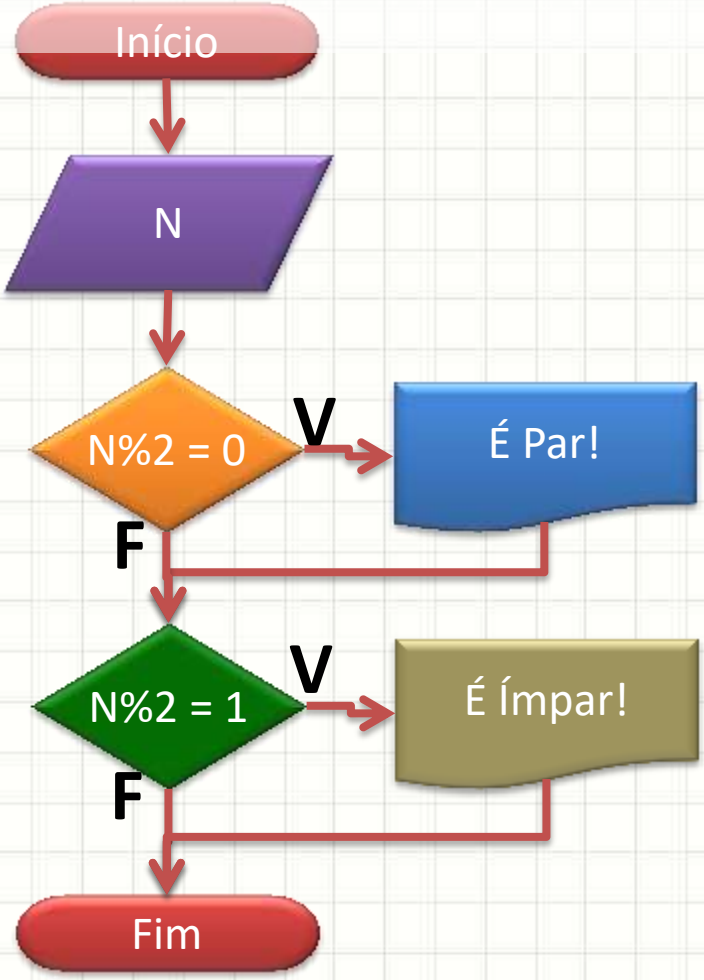


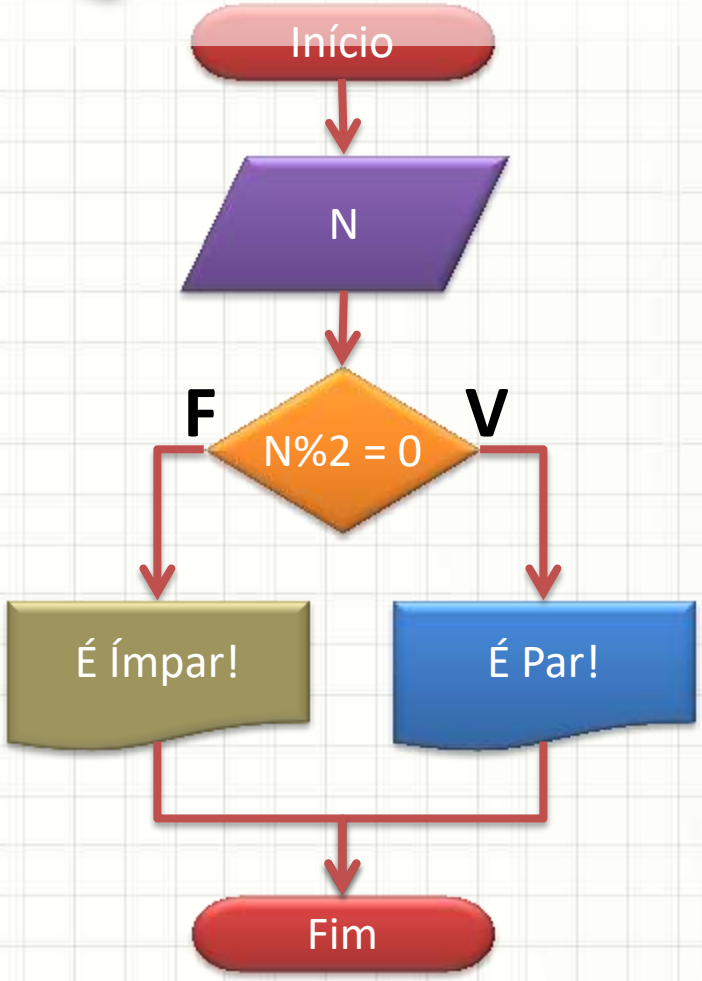


Estrutura de Decisão Composta

- Observe este outro... Alguém...

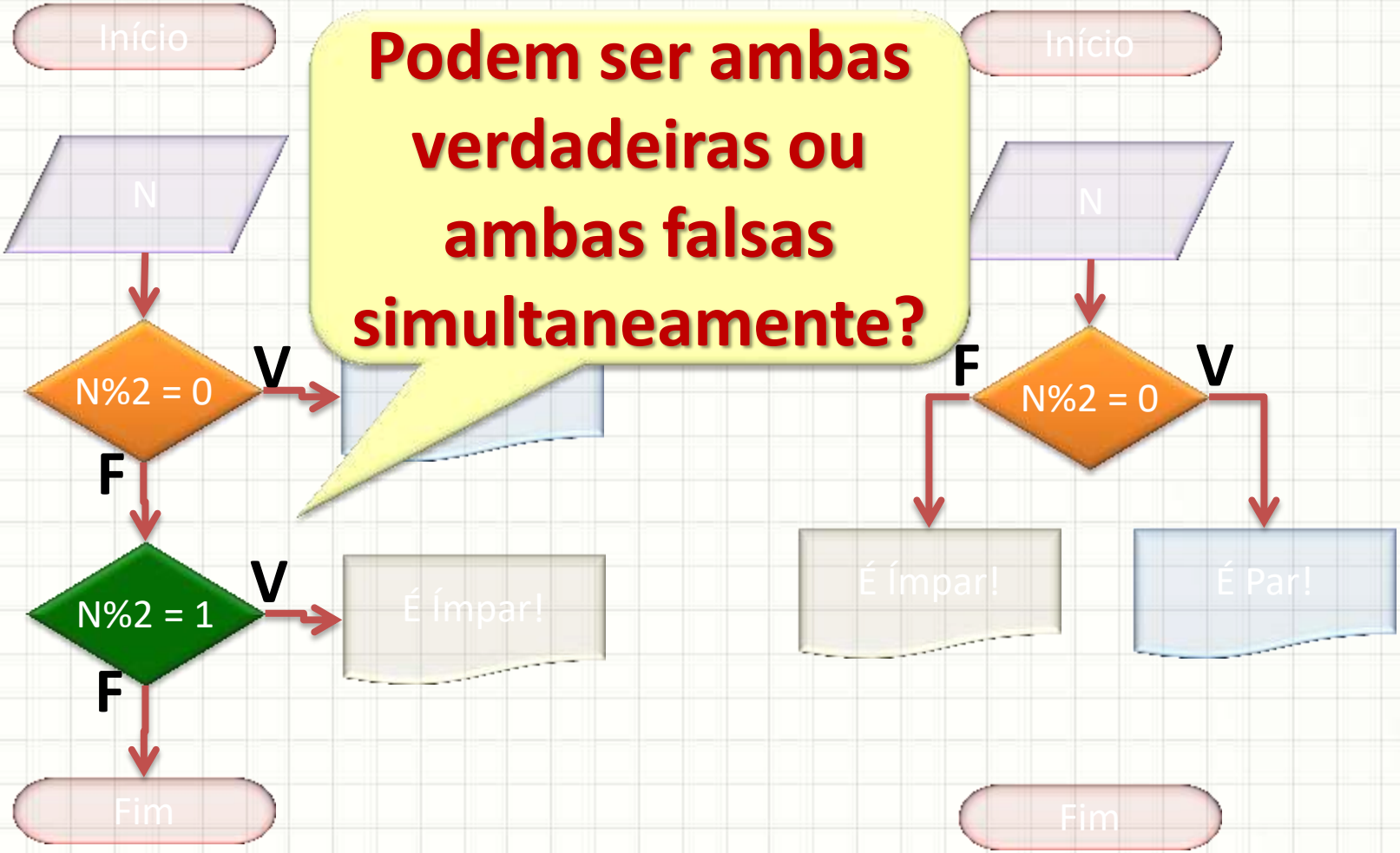
POR QUÊ?





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

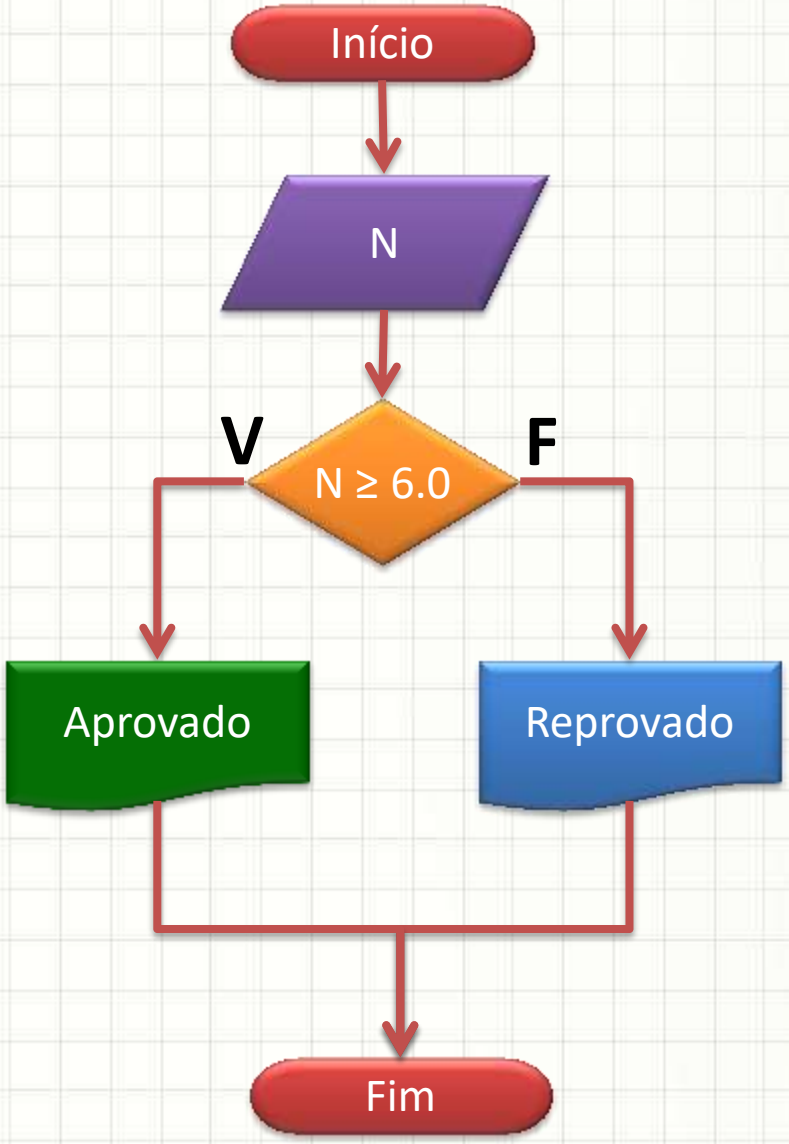
Forma do If~else

- C / C++

```
if ( proposição_lógica ) {
        código a executar para proposição verdadeira
      }
 else {
        código a executar para proposição falsa
      }
```

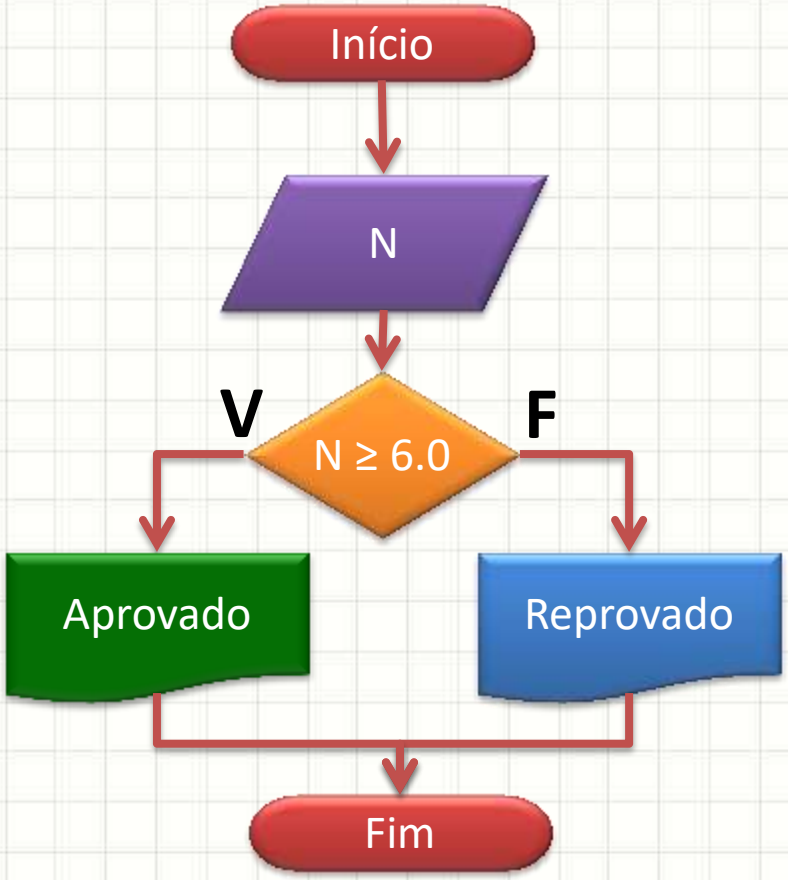
Exemplo

- 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



Exemplo

- 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;

int main(void) {
    float N;
    cout << "Digite a nota: ";
    cin >> N;
    if ( N >= 6.0 ) {
            cout << "Aprovado";
           }
    else {
            cout << "Reprovado";
           }
}
```

MOMENTO LÚDICO:

REPETIÇÃO

Momento Lúdico

- É legal ficar repetindo uma tarefa?

Momento Lúdico

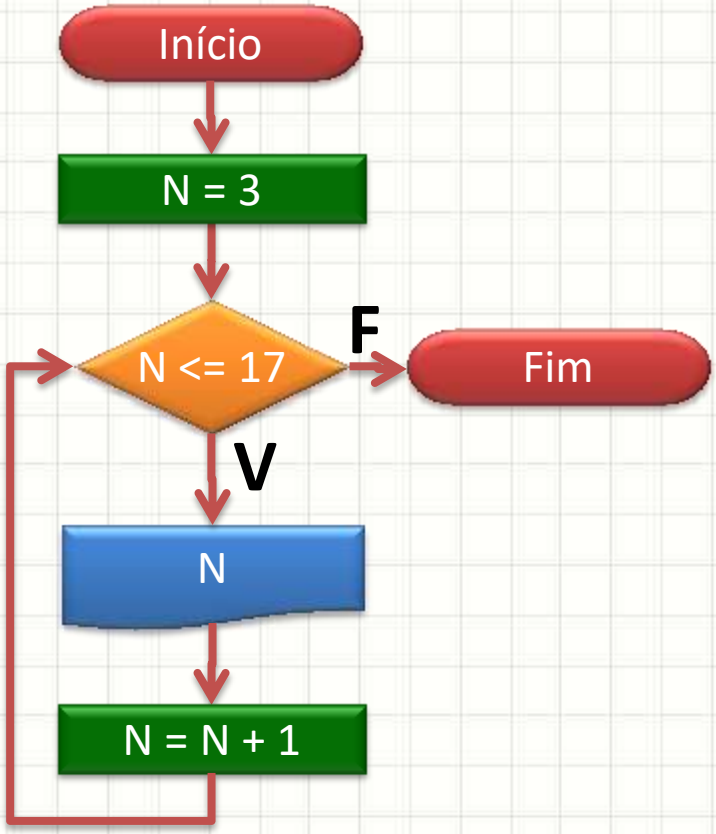
- Em geral... o computador não se importa!

Recordando:

Repetição com While

Recordando o While

- Estrutura de repetição **while**
  - **O que faz?**

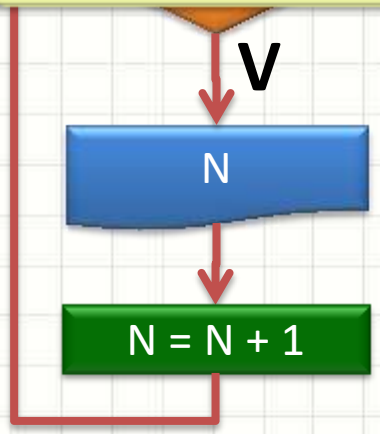



```
#include <iostream>
using namespace std;
main()
{
    int N;
    N = 3;
    while ( N <= 17 )
    {
      cout << N << endl;
      N = N + 1;
    }
}
```

Recordando o While

- Estrutura de repetição **while**
  - **O que faz?**

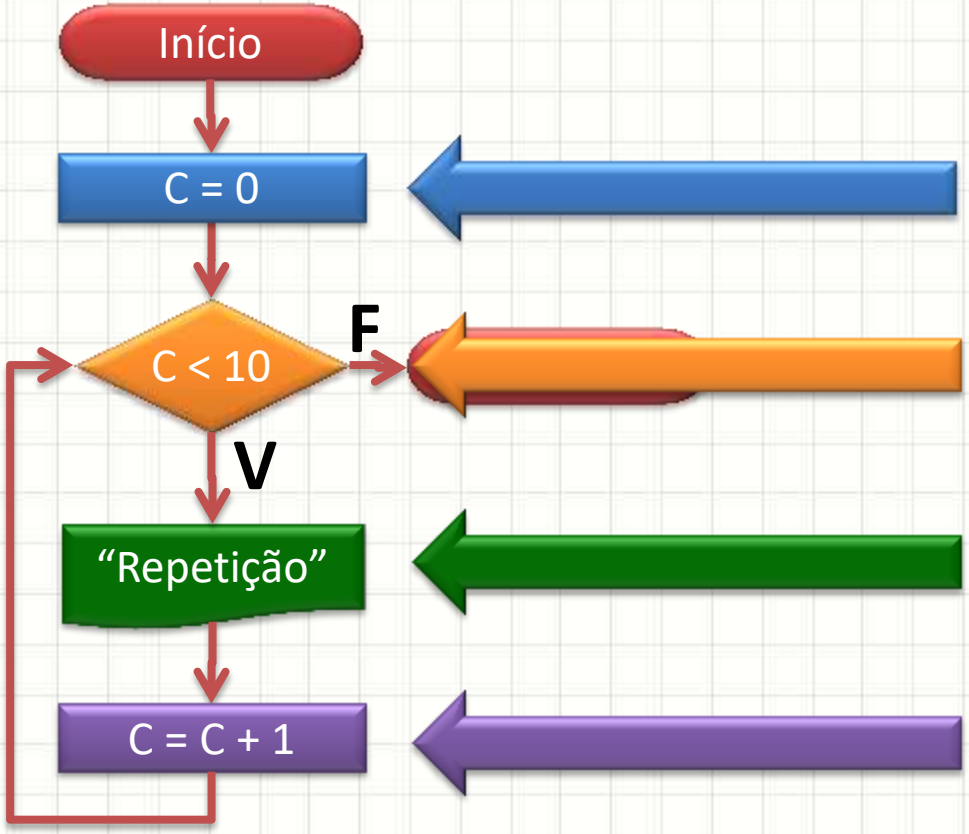
O que acontece se esquecermos essa linha?




```
#include <iostream>
using namespace std;
main()
{
    int N;
    N = 3;
    while ( N <= 17 )
    {
        cout << N << endl;
        N = N + 1;
    }
}
```

Recordando o While

- Observe:
  - O que faz?



INICIALIZAÇÃO

DECISÃO DE REPETIÇÃO

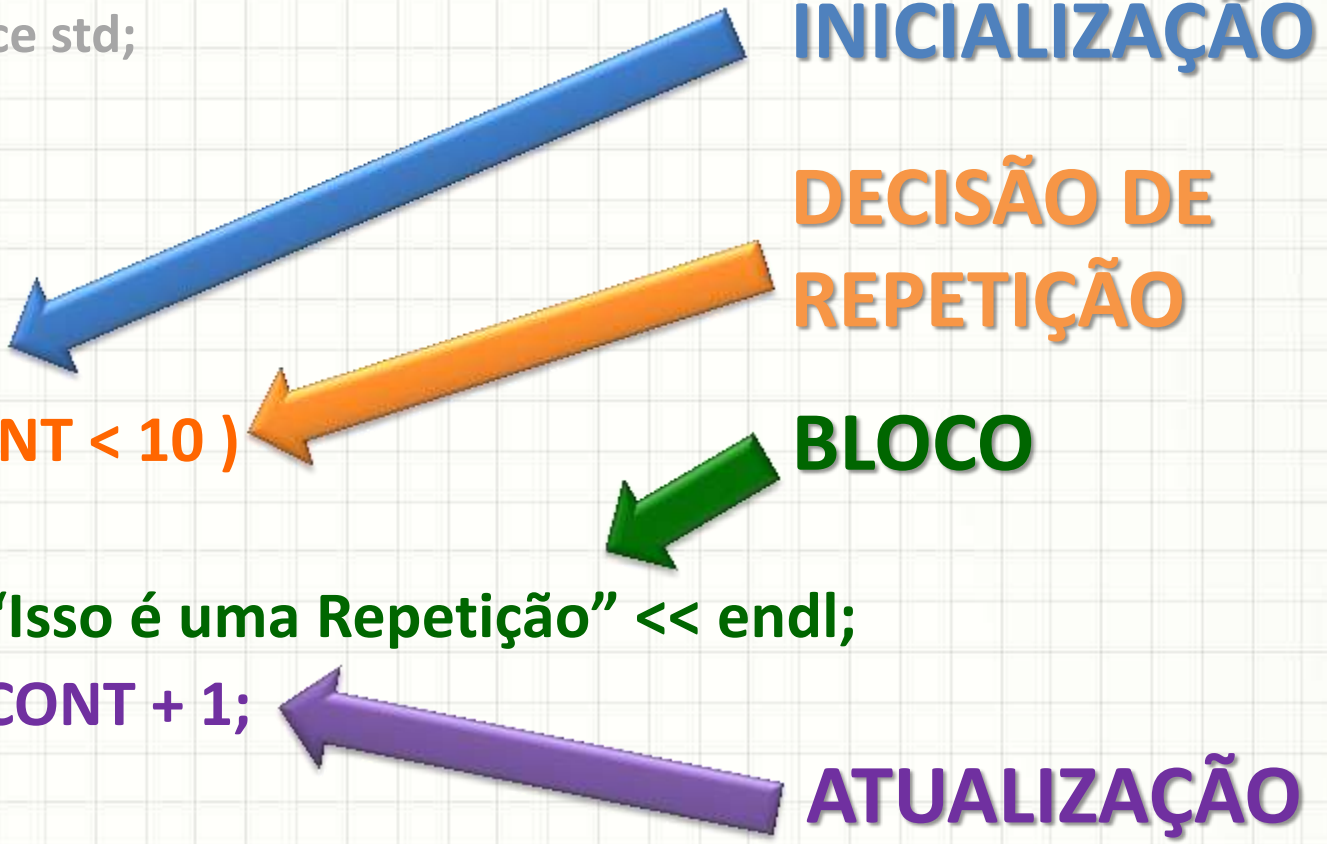
BLOCO

ATUALIZAÇÃO

Recordando o While

- No código...

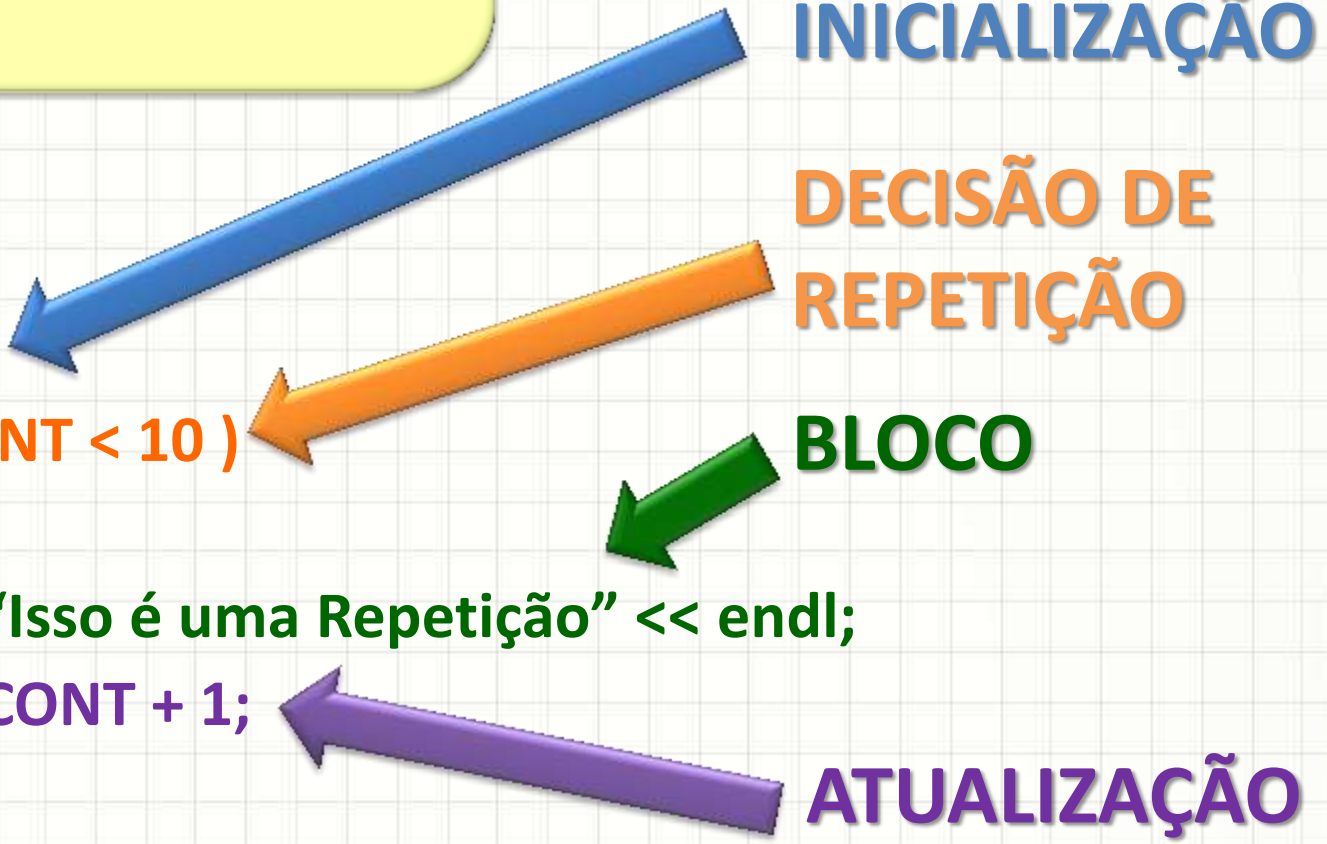
```
#include <iostream>
using namespace std;
main()
{
    int CONT;
    CONT = 0;
    while ( CONT < 10 )
    {
      cout << "Isso é uma Repetição" << endl;
      CONT = CONT + 1;
    }
}
```



le

Fácil esquecer um deles!

```
main()
{
    int CONT;
    CONT = 0;
    while ( CONT < 10 )
    {
      cout << "Isso é uma Repetição" << endl;
      CONT = CONT + 1;
    }
}
```



A Estrutura de Repetição FOR

O que é a estrutura **for**

- Todos os elementos em uma única linha
  - Só o bloco fica “isolado”

```
#include <iostream>
using namespace std;
main()
{
    int CONT;
    CONT = 0;
    while ( CONT < 10 )
    {
      cout << “Isso é uma Repetição” << endl;
      CONT = CONT + 1;
    }
}
```

O que é a estrutura **for**

- Todos os elementos em uma única linha
  - Só o bloco fica “isolado”

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

    for ( CONT = 0 ; CONT < 10 ; CONT = CONT + 1 )
    {
      cout << “Isso é uma Repetição” << endl;
    }
}
```

O que é a estrutura **for**

- Todos os elementos em uma única linha
  - Só o bloco fica “isolado”

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

    for ( CONT = 0 ; CONT < 10 ; CONT = CONT + 1 )
    {
      cout << “Isso é uma Repetição” << endl;
    }
}
```

INICIALIZAÇÃO

DECISÃO DE REPETIÇÃO

ATUALIZAÇÃO

BLOCO

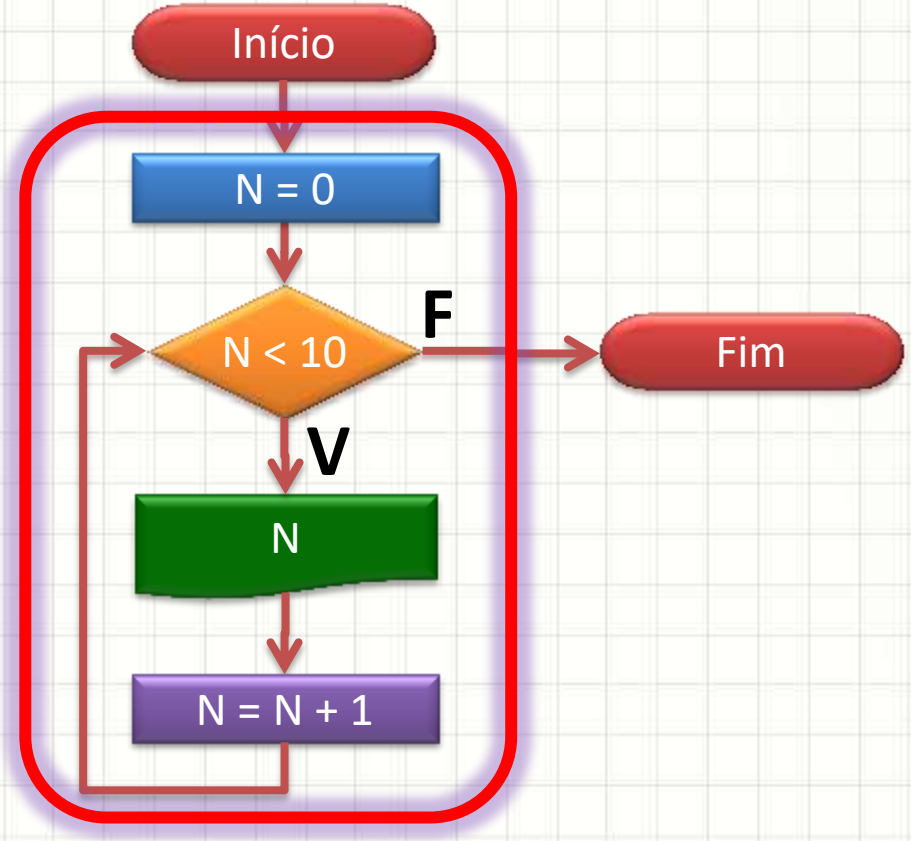
Forma Geral do **for**

for (inicialização; condição de repetição; atualização)

{

Executa enquanto a proposição for verdadeira

}



Exemplo

A) Faça um programa que apresente seu nome 10.000 vezes.

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

    for ( C = 0 ; C < 1000; C = C + 1 )
    {
      cout << “Daniel Caetano” << endl;
    }
}
```

Exemplo

B) Faça um programa que apresente os números de 52 a 75.

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

    for ( C = 52 ; C <= 75; C = C + 1 )
    {
      cout << C << endl;
    }
}
```

Usando Decisão no Arduino

Implementando um Pisca-Pisca

piscapisca.ino

```
void setup() {
   pinMode(12, INPUT);
}
void loop() {
   digitalWrite(13,HIGH);
   delay(1000);
   digitalWrite(13,LOW);
   delay(1000);
}
```

Implementando um Pisca-Pisca 2

piscapisca.ino

```
void setup() {
   pinMode(12, INPUT);
   pinMode(13, OUTPUT);
}
void loop() {
   int espera = 1000;
   if (digitalRead(12) == HIGH) espera = 250;
   digitalWrite(13,HIGH);
   delay(espera);
   digitalWrite(13,LOW);
   delay(espera);
}
```

Implementando um Pisca-Pisca 3

piscapisca.ino

```
int espera = 1000;
void setup() {
   pinMode(11, INPUT);
   pinMode(12, INPUT);
   pinMode(13, OUTPUT);
}
void loop() {
   if (espera <= 1750 && digitalRead(11) == HIGH)
      espera = espera + 250;
   if (espera >= 250 && digitalRead(12) == HIGH)
      espera = espera - 250;
   digitalWrite(13,HIGH);
   delay(espera);
   digitalWrite(13,LOW);
   delay(espera);
}
```

Implementando um Pisca-Pisca 4

piscapisca.ino

```
int espera = 1000;
int contador = 0;
void setup() {
   pinMode(11, INPUT);
   pinMode(12, INPUT);
   pinMode(13, OUTPUT);
}
void loop() {
    contador = contador + 10;
    delay(10);
    if (espera <= 1950 && digitalRead(11) == HIGH)
         espera = espera + 50;
    if (espera >= 50 && digitalRead(12) == HIGH)
         espera = espera - 50;
    if (contador == espera) digitalWrite(13,HIGH);
    if (contador >= 2*espera) {
         digitalWrite(13,LOW);
         contador = 0;
    }
}
```

Implementando um Pisca-Pisca 4

piscapisca.ino

```
int espera = 1000;
int contador = 0;
int podemudar = 1;
void setup() {
   pinMode(11, INPUT);
   pinMode(12, INPUT);
   pinMode(13, OUTPUT);
}
void loop() {
   contador = contador + 10;
   delay(10);
   if (podemudar == 1) {
       if (espera <= 1950 && digitalRead(11) == HIGH) {
           espera = espera + 50;
            podemudar = 0;
           }
       if (espera >= 50 && digitalRead(12) == HIGH) {
           espera = espera - 50;
            podemudar = 0;
           }
       }
    else { /* podemudar == 0 */
       if (digitalRead(11) == LOW && digitalRead(12) == LOW) podemudar = 1;
       if (digitalRead(11) == HIGH && digitalRead(12) == HIGH) espera = 1000;
       }
    if (contador == espera) digitalWrite(13,HIGH);
    if (contador >= 2*espera) {
       digitalWrite(13,LOW);
       contador = 0;
       }
}
```

PERGUNTAS?

Conclusões

Resumo

- Estruturação Básica de um Programa
- Estruturas de decisão
  - If / If ~ Else
- Estruturas de Repetição
  - For / While

- Matrizes?
- Funções?

Parte Prática

Parte Prática

- Por que realizar uma parte prática?

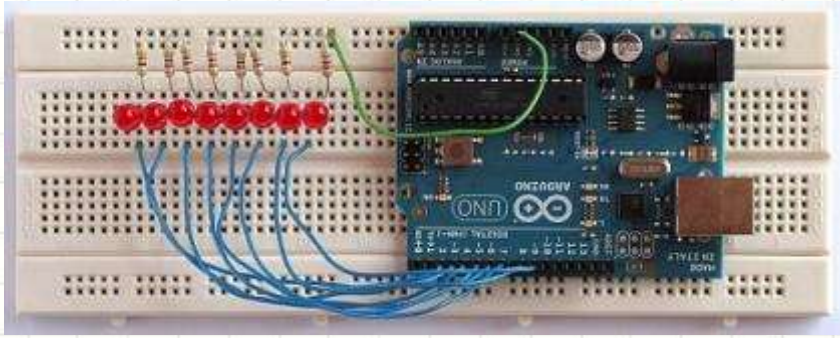
Projeto em Grupo

- Parte A - Arduíno
- Onde conseguir?
  - www.robocore.com.br
  - Mercado Livre...
- Qual comprar?
  - http://blog.filipeflop.com/arduino/tipos-de-arduino-qual-comprar.html
  - http://br-arduino.org/2014/11/arduino-nano-x-uno-repetindo-o-programa-do-semaforo-agora-no-nano.html

Projeto em Grupo

	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Esplora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v

Projeto em Grupo



- Parte A
- IDE Arduino
  - https://www.arduino.cc/en/Main/Software
- Tutoriais
  - Tutorial no site do professor (Material de Apoio)
  - http://www.comofazerascoisas.com.br/projeto-arduino-pisca-led.html
  - https://www.tutorialspoint.com/arduino/index.htm
  - https://openwebinars.net/blog/tutorial-arduino-ejemplo-semaforo/
- Avançado
  - https://www.embarcados.com.br/arduino/

Trabalho de Hoje

- Pesquise as características dos modelos de Arduíno do trabalho (UNO e Nano v3).
  - Preço, módulos, sensores, disponibilidade...
- Pense em **duas** ou **três** soluções possíveis de implementar com o Arduino.
  - Tente identificar quais elementos são necessários
  - Imagine a lógica para cada um deles
- Responda às questões