

# **INFORMÁTICA PARA ENGENHARIA**

## **ESTRUTURAS DE DECISÃO**

Prof. Dr. Daniel Caetano

2019 - 1

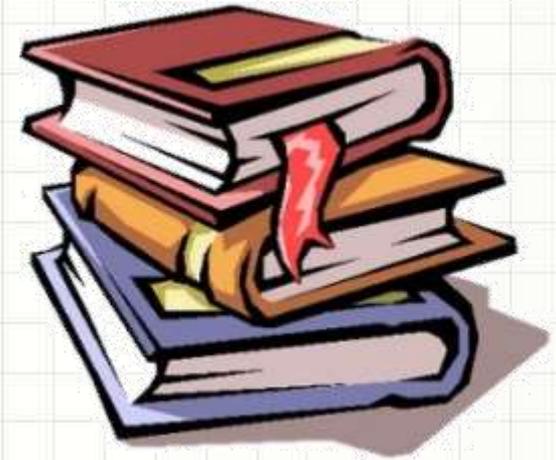
# Objetivos

- Entender a ideia de decisão
- Compreender os operadores relacionais
- Compreender como implementar decisões, decisões múltiplas, compostas e aninhadas

- **Atividades Aula 9 – SAVA!**



# Material de Estudo



---

## Material

## Acesso ao Material

Notas de Aula e  
Apresentação

<http://www.caetano.eng.br/>  
(Informática para Engenharia – Aula 9)

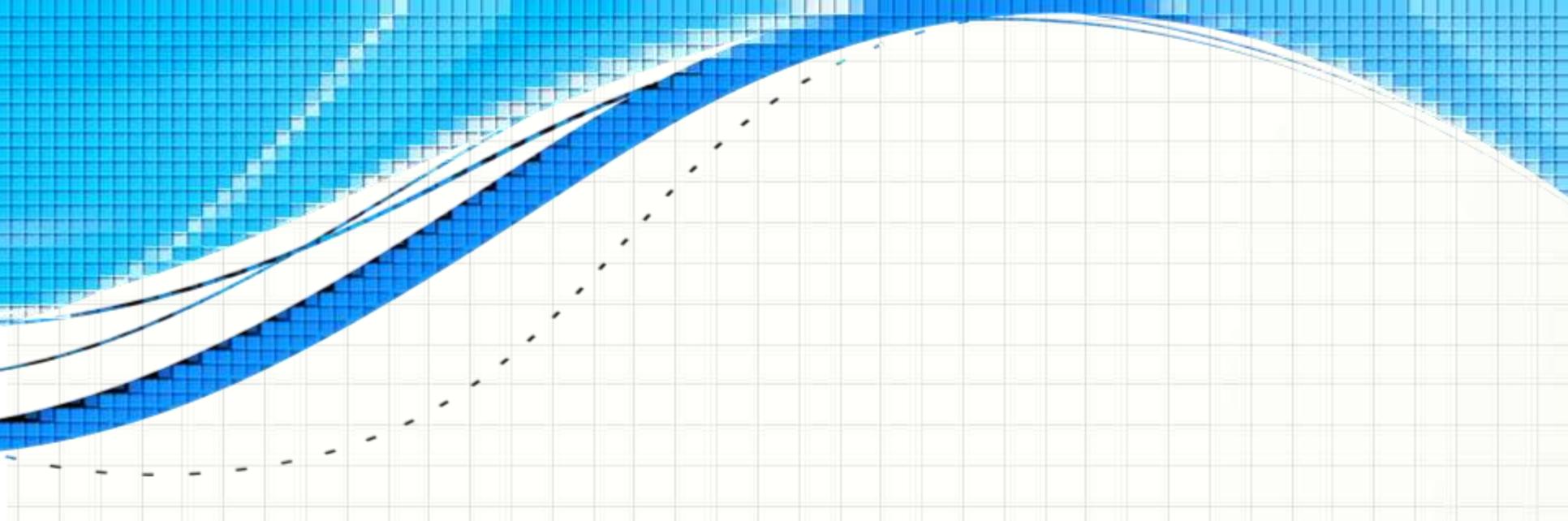
Material Didático

Lógica de Programação, págs 81 a 89.

Biblioteca Virtual

“Lógica de Programação – Fundamentos da  
Programação de Computadores”, págs 50 a 92.

**LEMBRETE: CONSULTAR O “DEPOIS” DA AULA 9 NO SAVA!**



# O QUE O COMPUTADOR PODE DECIDIR?

# Introdução

- Até agora, nossos programas...
  - Sempre com a **mesma** sequência de tarefas!



## # Um exemplo

```
print("Bem vindo!")
```

```
print("Até logo!")
```

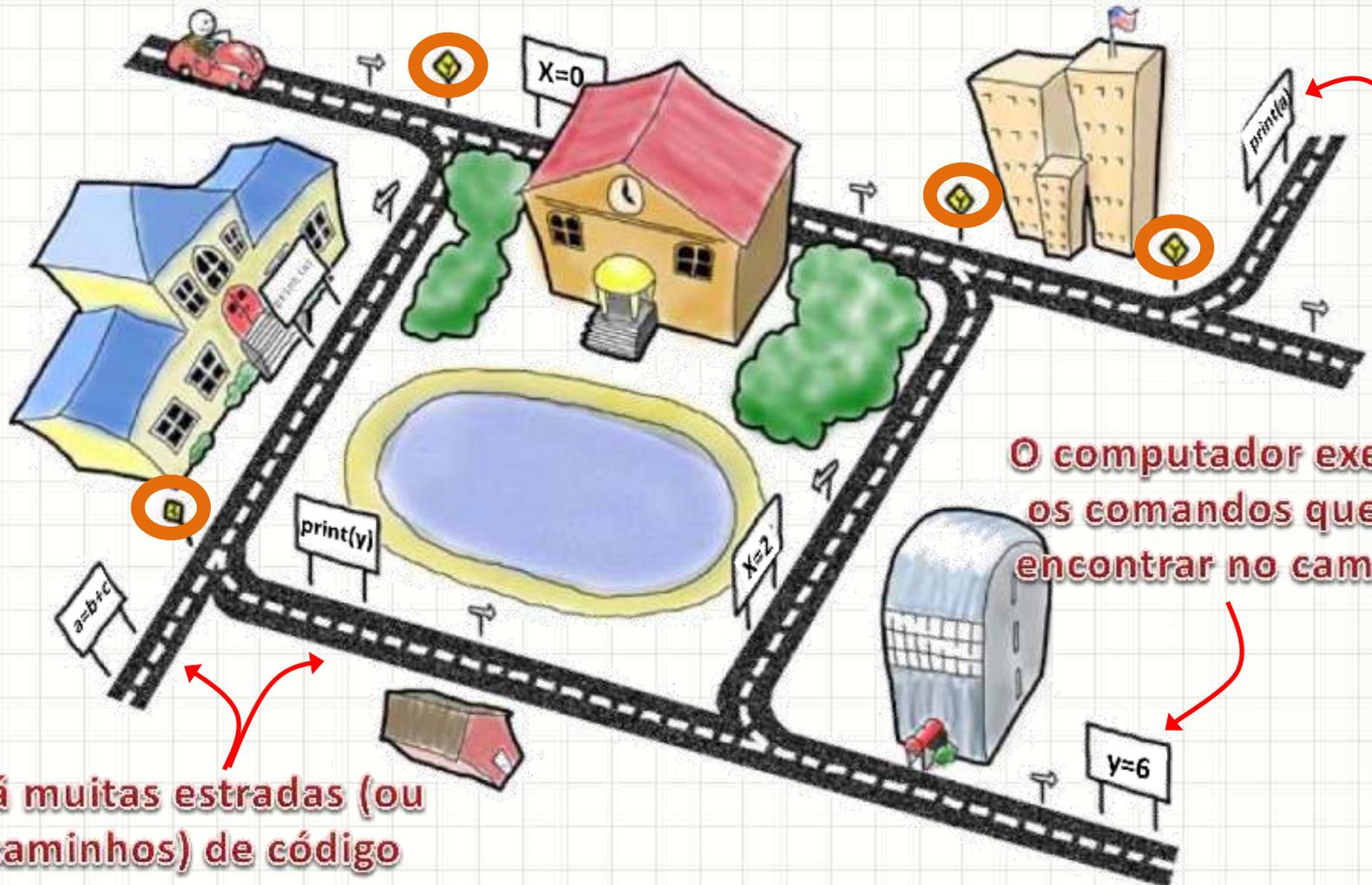
- São como estradas sem **desvios**!



# Introdução

## Como ele decide qual caminho seguir?

- O mundo real é uma estrada sem desvios?

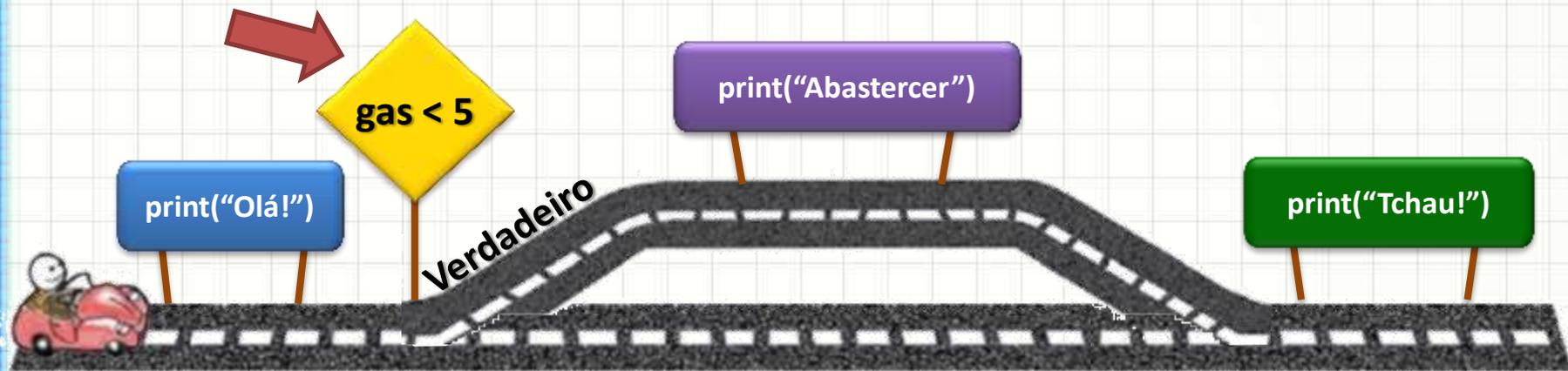


O computador executa os comandos que ele encontrar no caminho

Há muitas estradas (ou caminhos) de código

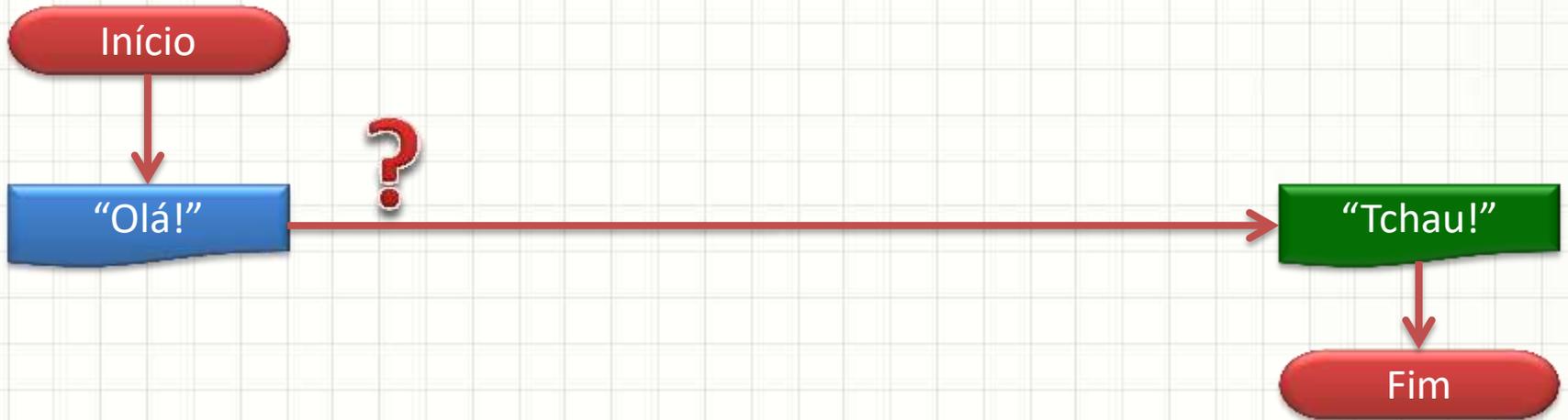
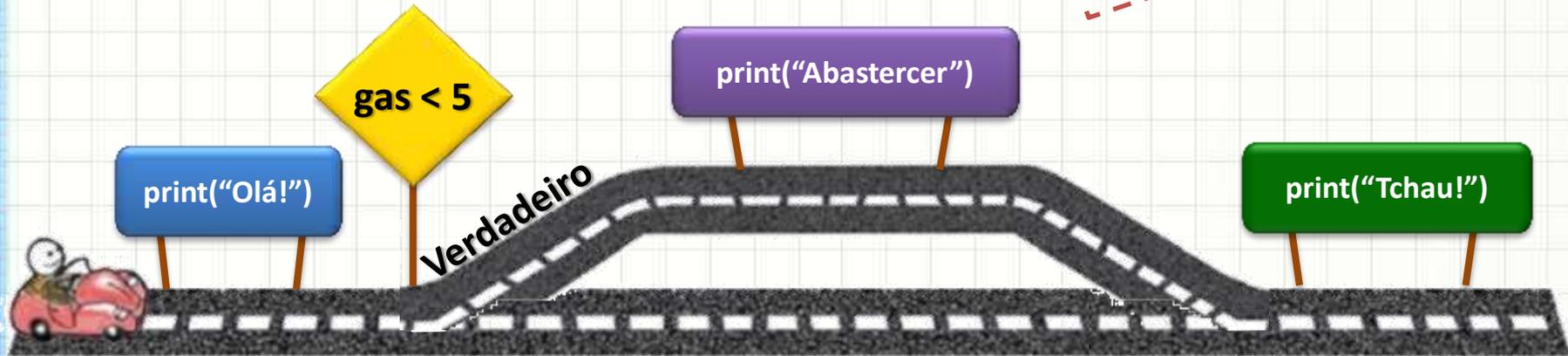
# Introdução

- Como o computador escolhe um caminho?
  - Com um desvio condicional
- Condicional?
  - Decisão com base em uma condição
    - Proposição verdadeira ou falsa



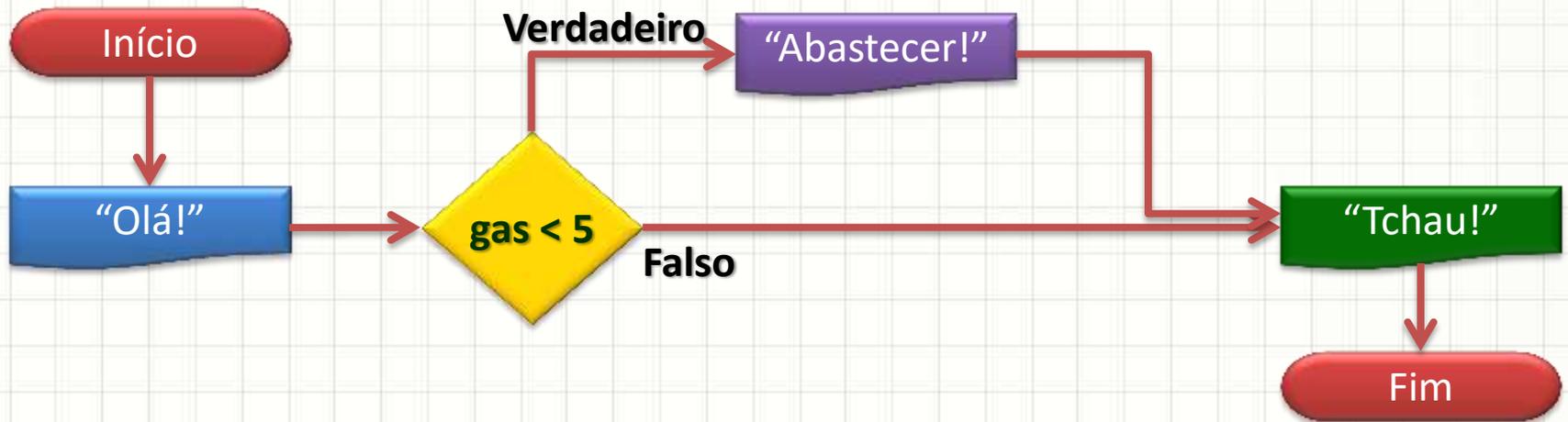
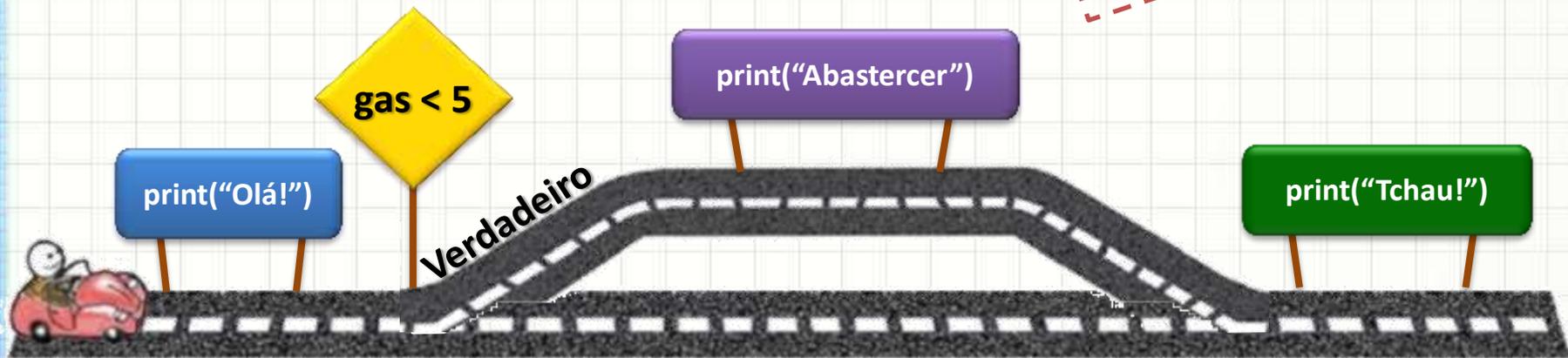
# Decisão no Fluxograma

**Recordando**



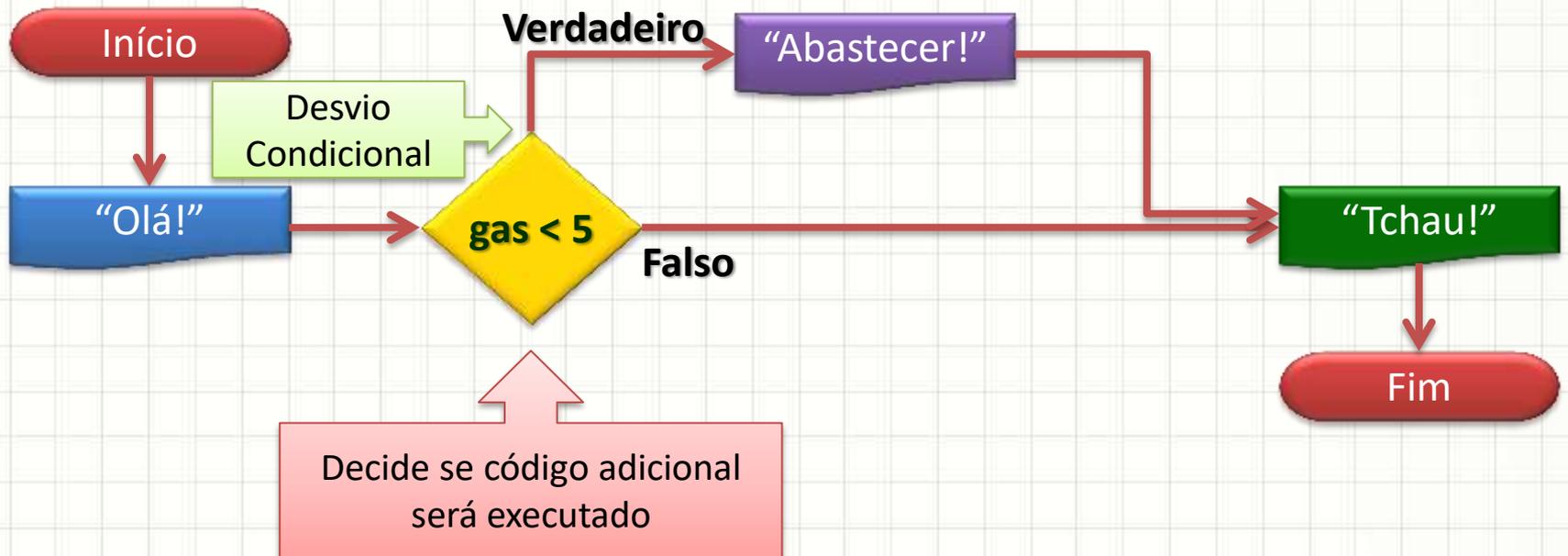
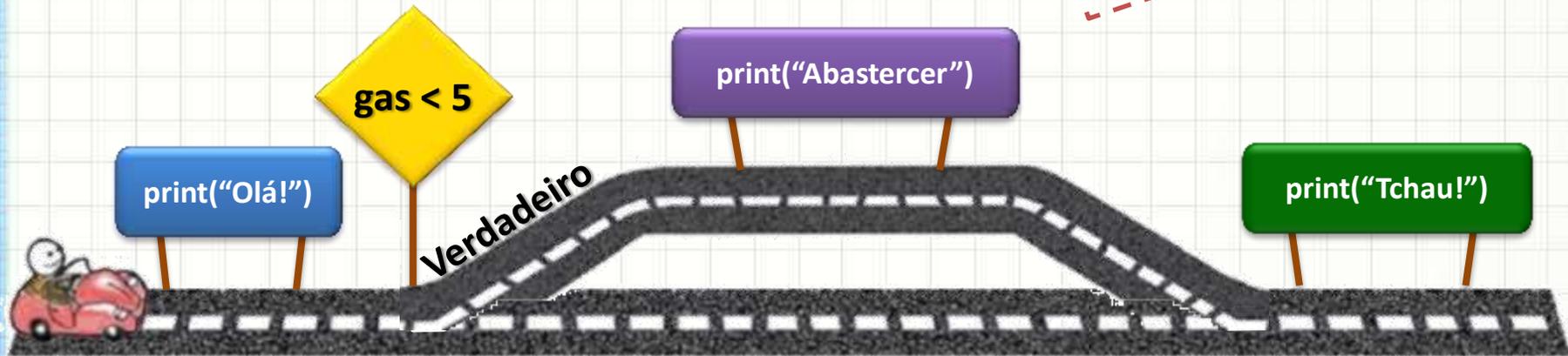
# Decisão no Fluxograma

**Recordando**



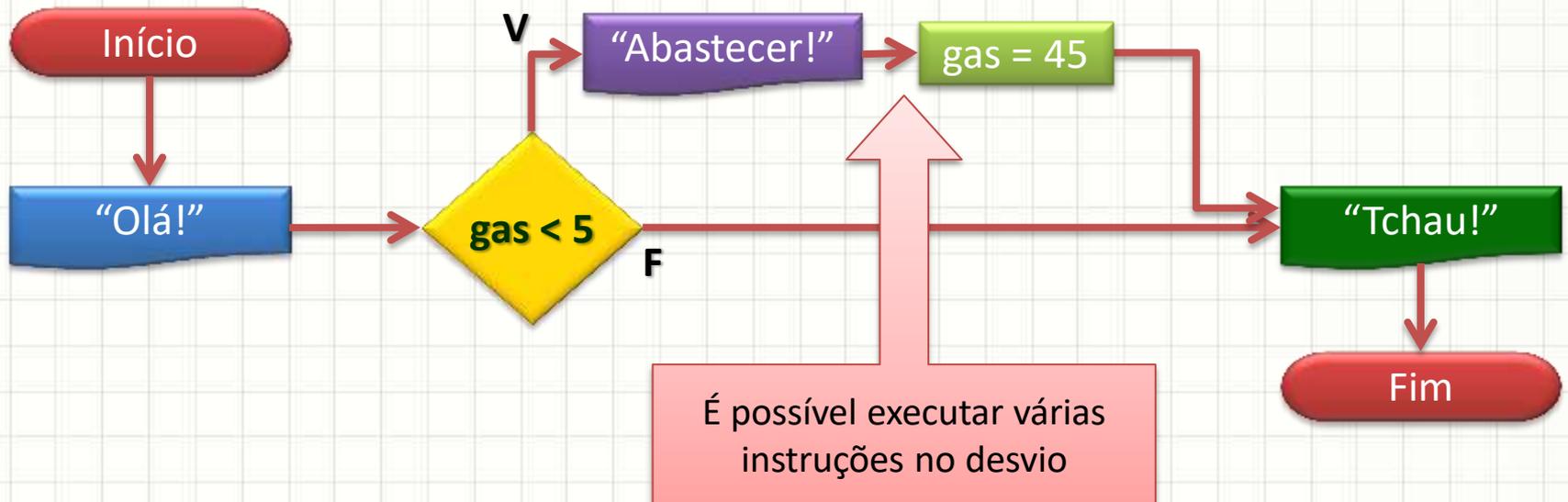
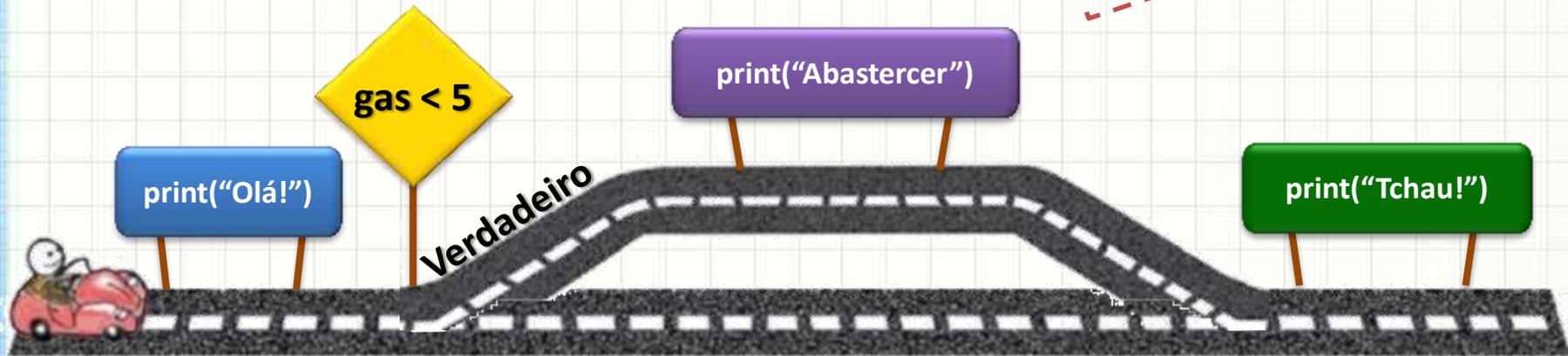
# Decisão no Fluxograma

**Recordando**



# Decisão no Fluxograma

Recordando

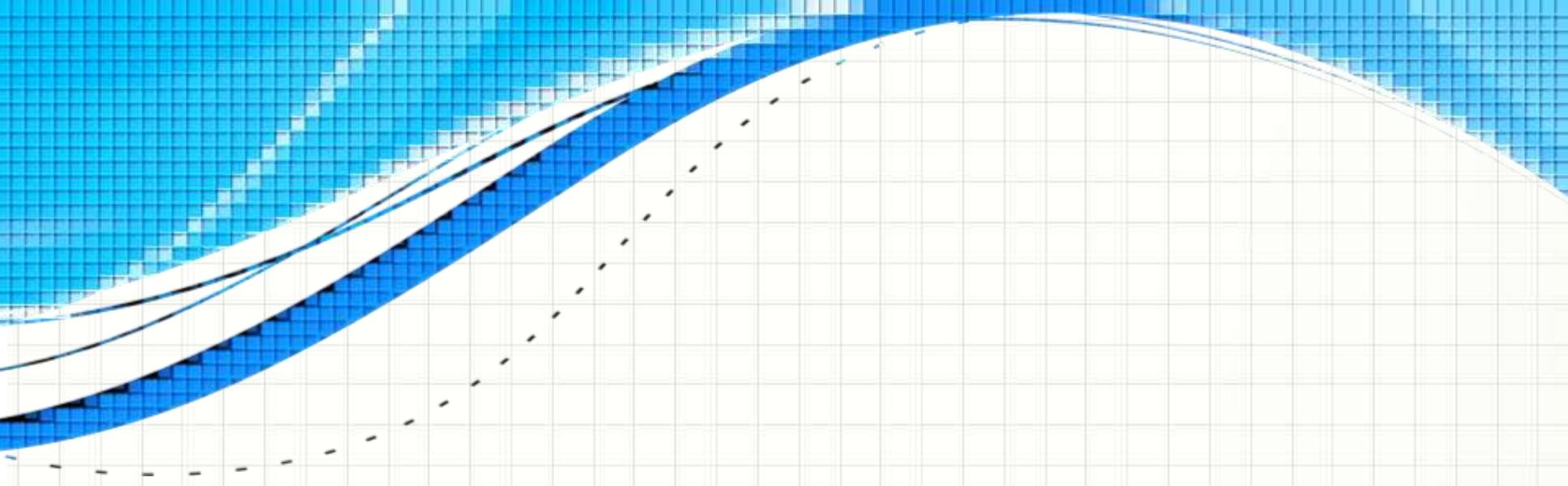


# Decisão no Fluxograma

**Recordando**

- Verificar se aluno está reprovado por nota





# **OPERADORES RELACIONAIS E LÓGICOS**

# Operadores Relacionais

- Operadores Relacionais: comparadores
  - Quais são?

Operador Relacional	Python	Exemplo	Significado
=	==	x == 2	X é igual a 2?
≠	!=	x != 2	X é diferente de 2?
>	>	x > 2	X é maior que 2?
≥	>=	x >= 2	X é maior ou igual a 2?
<	<	x < 2	X é menor que 2?
≤	<=	x <= 2	X é menor ou igual a 2?

# Testando os Comparadores

- Digite cada um dos comandos abaixo
  - Qual o resultado deles?

## Console

```
In [1]: 2 == 2
```

```
In [2]: 3 == 2
```

```
In [3]: 3 > 2
```

```
In [4]: 3 != 2
```

```
In [5]: 3 <= 2
```

# Testando os Comparadores

- Digite cada um dos comandos abaixo
  - Qual o resultado deles?

## Console

```
In [6]: X = 5
```

```
In [7]: X >= 2
```

```
In [8]: X != 2
```

```
In [9]: X < 0
```

```
In [10]: X == 3
```

# Atenção!

- Por que em Python usamos `==` e não `=` ?  
= significa atribuição (guardar valor em variável)

Muda  
X

X = 5

Guarda o valor 5 na variável X

`==` significa comparação (resulta **false/true**)

Não  
Muda  
X

X == 5

Avalia se X contém o valor 5

# Operadores Lógicos

- Operadores Lógicos: agrupam proposições
  - Quais são?

Operador Lógico	Python	Exemplo	Resulta em
<b>E</b>	<b>and</b>	true <b>and</b> false	false
<b>OU</b>	<b>or</b>	true <b>or</b> false	true
<b>NÃO</b>	<b>not</b>	<b>not</b> true	false

# Testando os Operadores Lógicos

- Digite cada um dos comandos abaixo
  - Qual o resultado deles?

## Console

```
In [11]: 2 == 2
```

```
In [12]: 3 == 2
```

```
In [13]: 2 == 2 and 3 == 2
```

```
In [14]: 2 == 2 or 3 == 2
```

```
In [15]: not 3 == 2
```

# Testando os Operadores Lógicos

- Digite cada um dos comandos abaixo
  - Qual o resultado deles?

## Console

```
In [16]: X = 5
```

```
In [17]: X >= 0 and X <= 5
```

```
In [18]: X < 0 or x > 5
```

```
In [19]: not (X >= 0 and X <= 5)
```

```
In [20]: not (X == 2)
```

# Exercício

- Analisemos as proposições abaixo, para  **$X = 5$**

$$X > 5 \rightarrow$$

$$X < 2 \rightarrow$$

$$X < 3 \text{ E } 3 < X \rightarrow$$

$$2 < X \text{ E } 2+X < 3+4 \rightarrow$$

$$X > 18 \text{ OU } 2*X < 17 \rightarrow$$

$$8 > X \text{ OU } 7 < 12 \text{ E } X > 9 \rightarrow$$

$$\text{NÃO } X < 5 \text{ OU } 11 < X \rightarrow$$

# Exercício

- Analisemos as proposições abaixo, para  $X = 5$

$$X > 5 \rightarrow \text{falso}$$

$$X < 2 \rightarrow \text{falso}$$

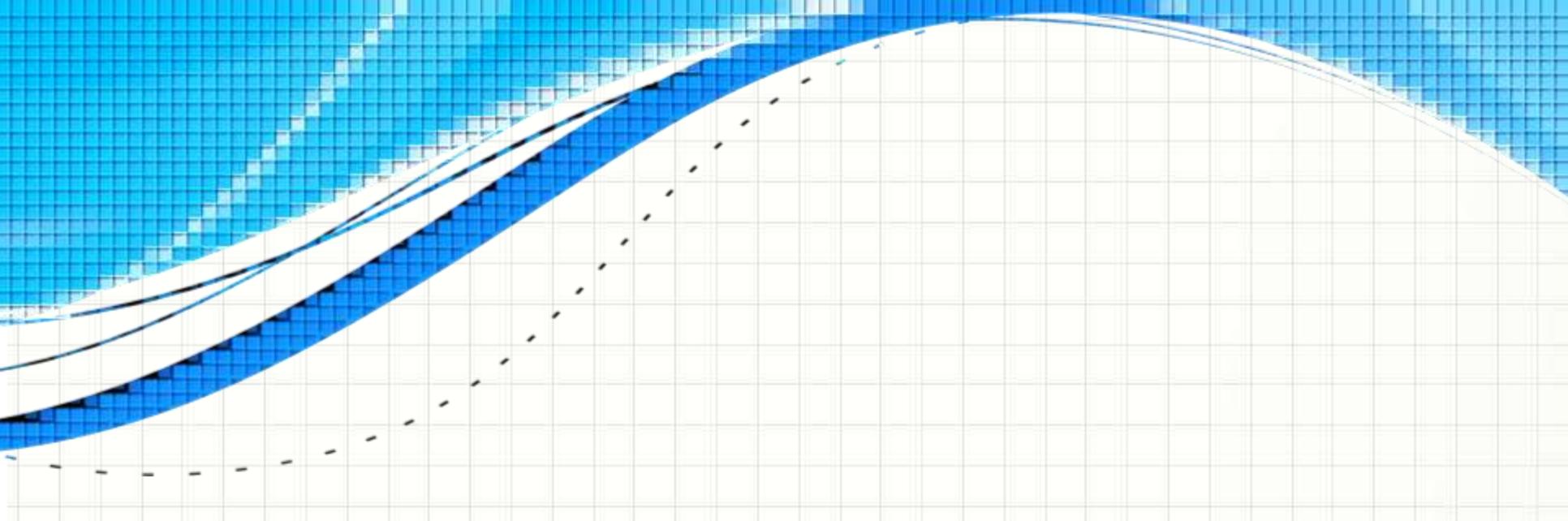
$$X < 3 \text{ E } 3 < X \rightarrow \text{falso}$$

$$2 < X \text{ E } 2 + X < 3 + 4 \rightarrow \text{falso}$$

$$X > 18 \text{ OU } 2 * X < 17 \rightarrow \text{verdadeiro}$$

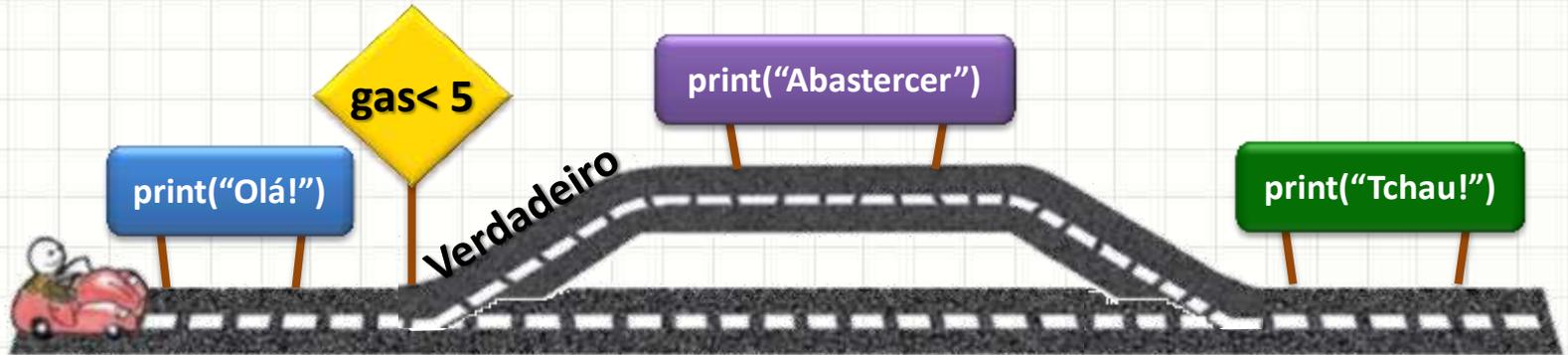
$$8 > X \text{ OU } 7 < 12 \text{ E } X > 9 \rightarrow \text{verdadeiro}$$

$$\text{NÃO } X < 5 \text{ OU } 11 < X \rightarrow \text{verdadeiro}$$



# DECISÃO NO CÓDIGO

# Decisão no Código



- Portugol

Algoritmo "Abastecimento"

Início

```
escreva("Olá!")
```

```
se gas < 5:
```

```
    escreva("Abastecer")
```

```
escreva("Tchau!")
```

Fim

- Python

**Funciona?**

```
# Abastecimento
```

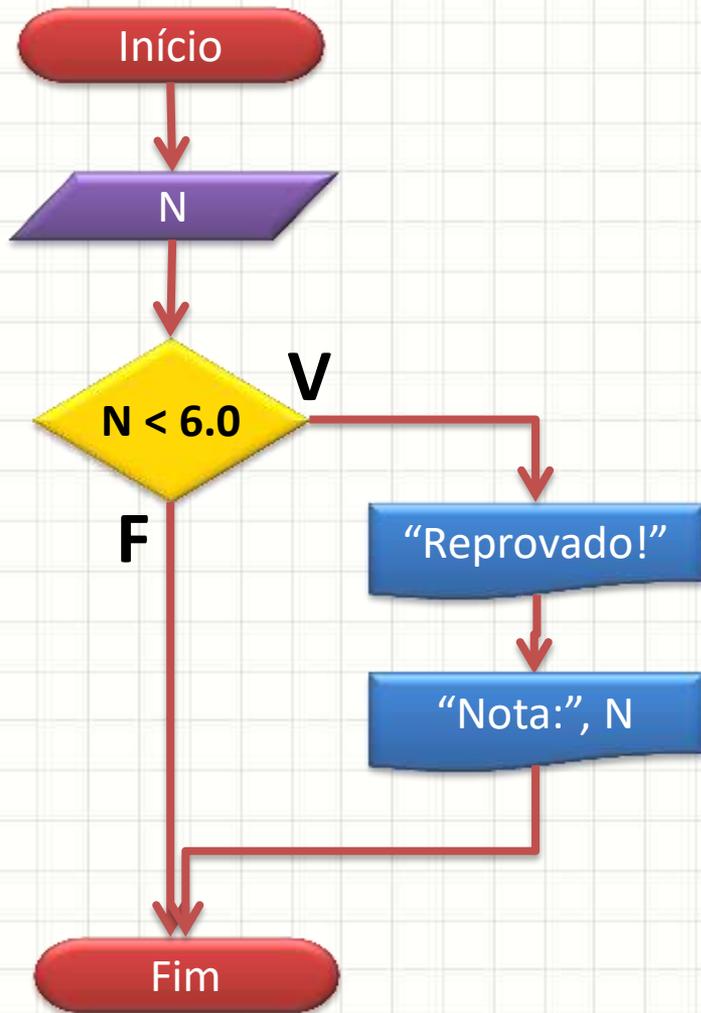
```
print("Olá!")
```

```
if gas < 5:
```

```
    print("Abastecer")
```

```
print("Tchau!")
```

# Como Fica a Decisão no Código?



- Portugol

**Algoritmo** “Verifica Reprovação”

**Início**

real N

escreva(“Digite a nota: ”)

leia(N)

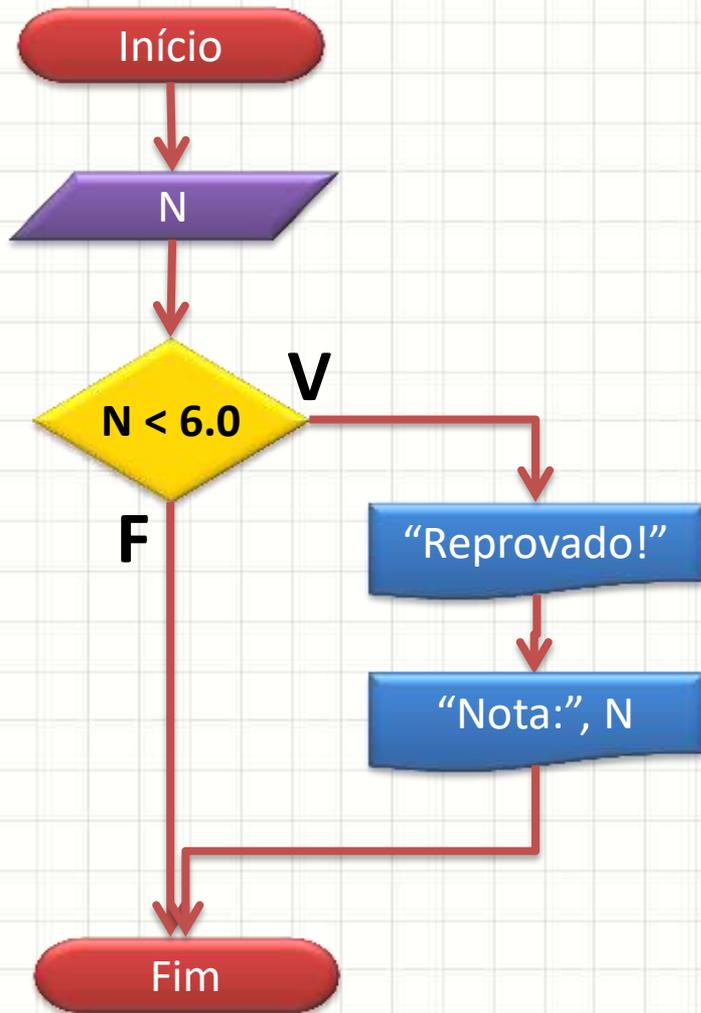
se  $N < 6.0$  :

    escreva(“Reprovado!”)

    escreva(“Nota:”, N)

**Fim**

# Como Fica a Decisão no Código?



- Python

**# Verifica Reprovação**

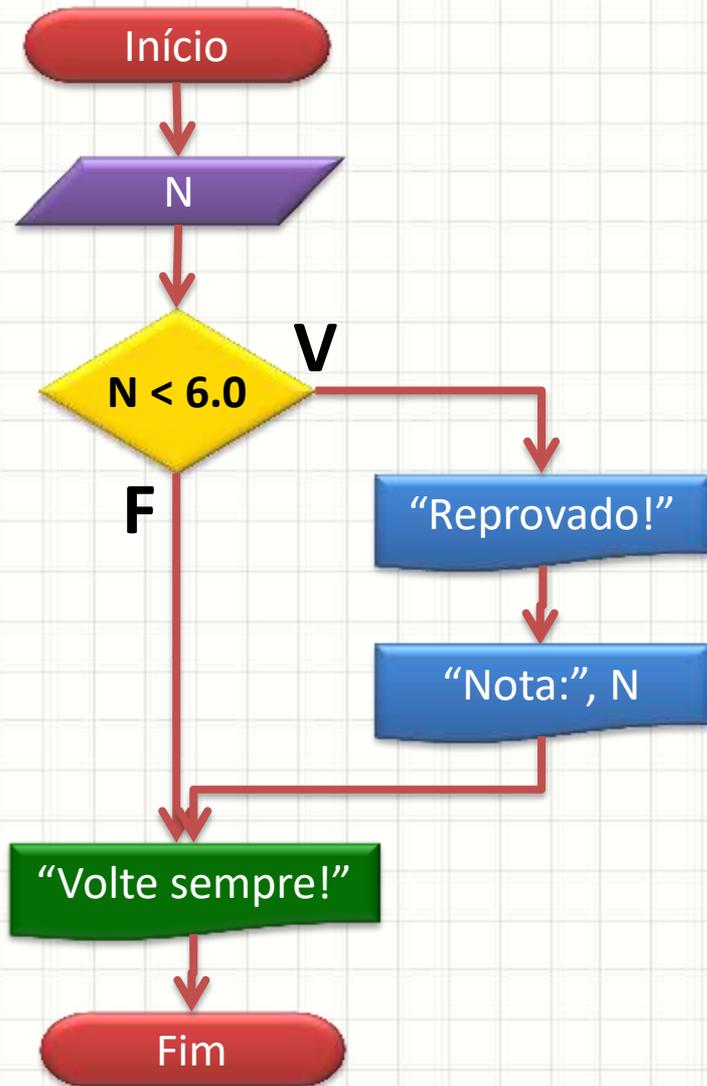
```
N = float(input("Digite a nota: "))
```

```
if N < 6.0 :
```

```
    print("Reprovado!")
```

```
    print("Nota:", N)
```

# Como Fica a Decisão no Código?



- Python

**# Verifica Reprovação**

```
N = float(input("Digite a nota: "))
```

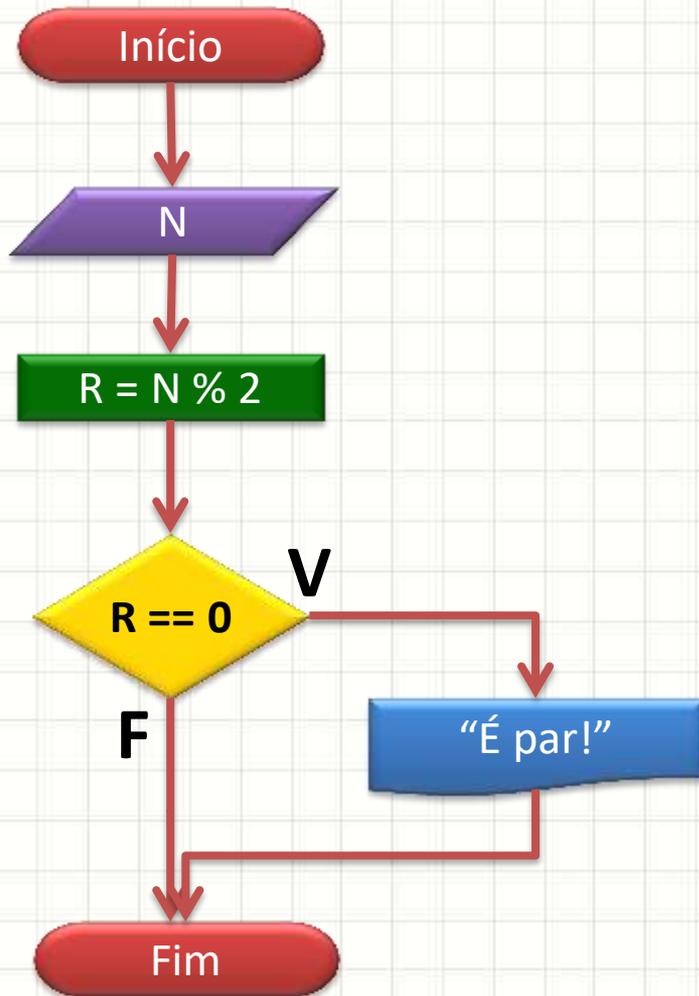
```
if N < 6.0 :
```

```
    print("Reprovado!")
```

```
    print("Nota:", N)
```

```
print("Volte sempre!")
```

# Verifica se número é par



- Python

**# Verifica se é par**

```
N = int(input("Digite um Nº: "))
```

```
R = N % 2
```

```
if R == 0:
```

```
    print("É par!")
```

# Exercício 1

- Faça um programa que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º

**if** proposição :  
código a executar

Comparador	Significado
<code>==</code>	Igualdade
<code>!=</code>	Diferença
<code>&gt;</code>	Maior
<code>&gt;=</code>	Maior ou Igual
<code>&lt;</code>	Menor
<code>&lt;=</code>	Menor ou Igual

- Python

```
# Verifica se é par
```

```
N = int(input("Digite um N°: "))
```

```
R = N % 2
```

```
if R == 0 :
```

```
    print("É par!")
```

# Exercício 1

- Faça um programa que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º

aula09ex01.py

```
# Verifica se 1º é maior que 2º  
N1 = int(input("Digite um Nº:"))  
N2 = int(input("Digite outro Nº:"))  
if N1 > N2 :  
    print("Sucesso!")
```

# Exercício 2

- Faça um programa que **leia a nota e a frequência do aluno** e informe se ele está **aprovado** (**nota  $\geq 6,0$**  e **frequência  $\geq 75\%$** )

- Exemplo

**# Verifica Reprovação**

**N = float(input("Digite a nota: "))**

**if N < 6.0 :**

**print("Reprovado!")**

# Exercício 2

- Faça um programa que **leia a nota e a frequência do aluno** e informe se ele está **aprovado** (**nota  $\geq 6,0$**  e **frequência  $\geq 75\%$** )

aula09ex02.py

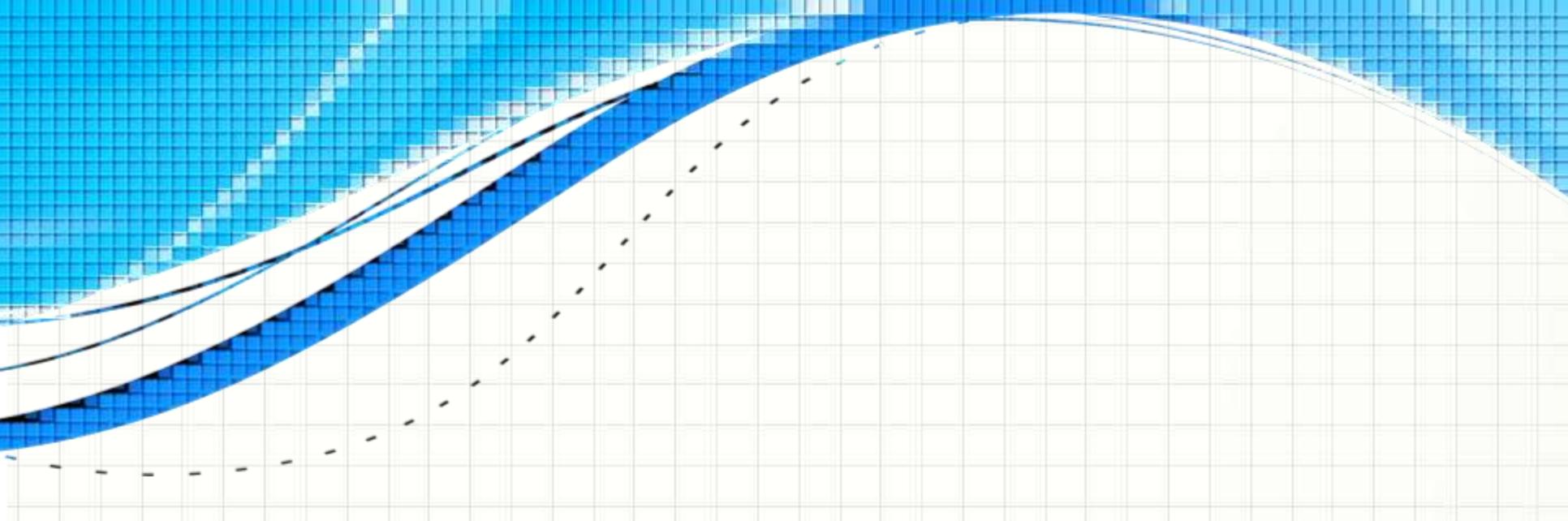
```
# Verifica Aprovação
```

```
N = float(input("Digite a nota: "))
```

```
F = float(input("Digite a frequência: "))
```

```
if N >= 6.0 and F >= 75:
```

```
    print("Aprovado!")
```



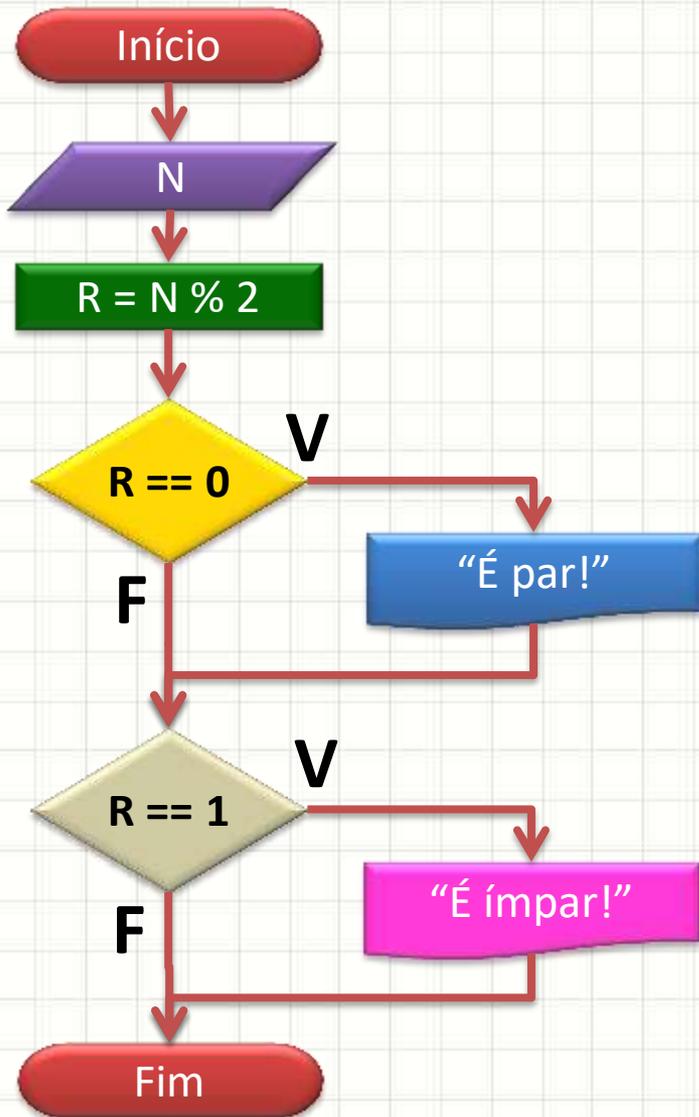
# DECISÕES MÚLTIPLAS

# Múltiplas Decisões

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



# Verifica se número é par ou ímpar



- Python

# Verifica se é par ou ímpar

```
N = int(input("Digite um N°: "))
```

```
R = N % 2
```

```
if R == 0:
```

```
    print("É par!")
```

```
if R == 1:
```

```
    print("É ímpar!")
```

R == 1

R != 0

Alternativas?

not ( R == 0 )

# Exercício 3

- Modifique o programa abaixo para que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º e **fracasso** quando o 2º for maior que o 1º

Exemplo.py

```
# Verifica se 1º é maior que 2º  
N1 = int(input("Digite um Nº:"))  
N2 = int(input("Digite outro Nº:"))  
if N1 > N2 :  
    print("Sucesso!")
```

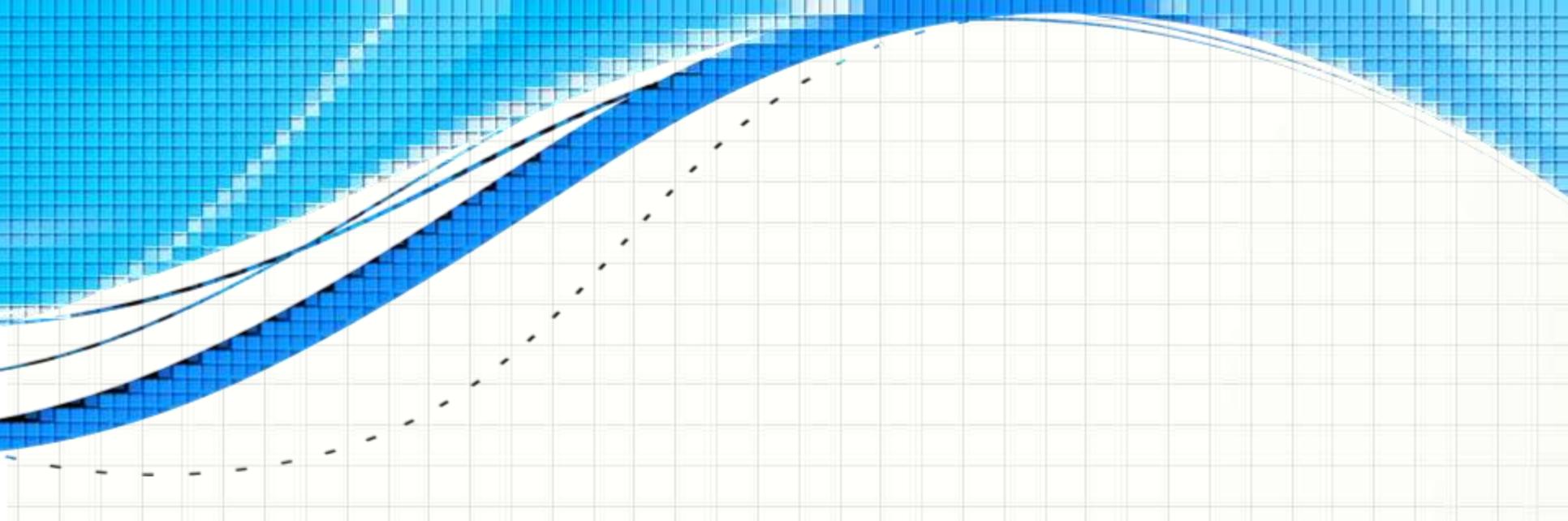
# Exercício 3

- Modifique o programa abaixo para que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º e **fracasso** quando o 2º for maior que o 1º

aula09ex03.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
if N2 > N1 :
    print("Fracasso!")
```

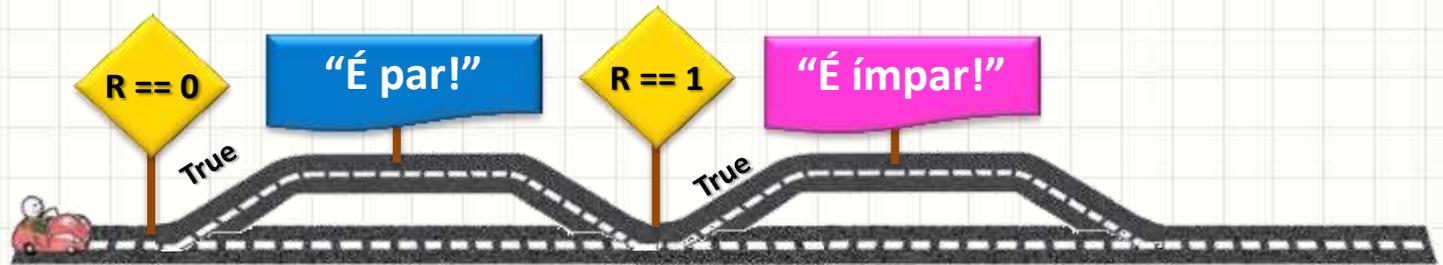
N1 < N2



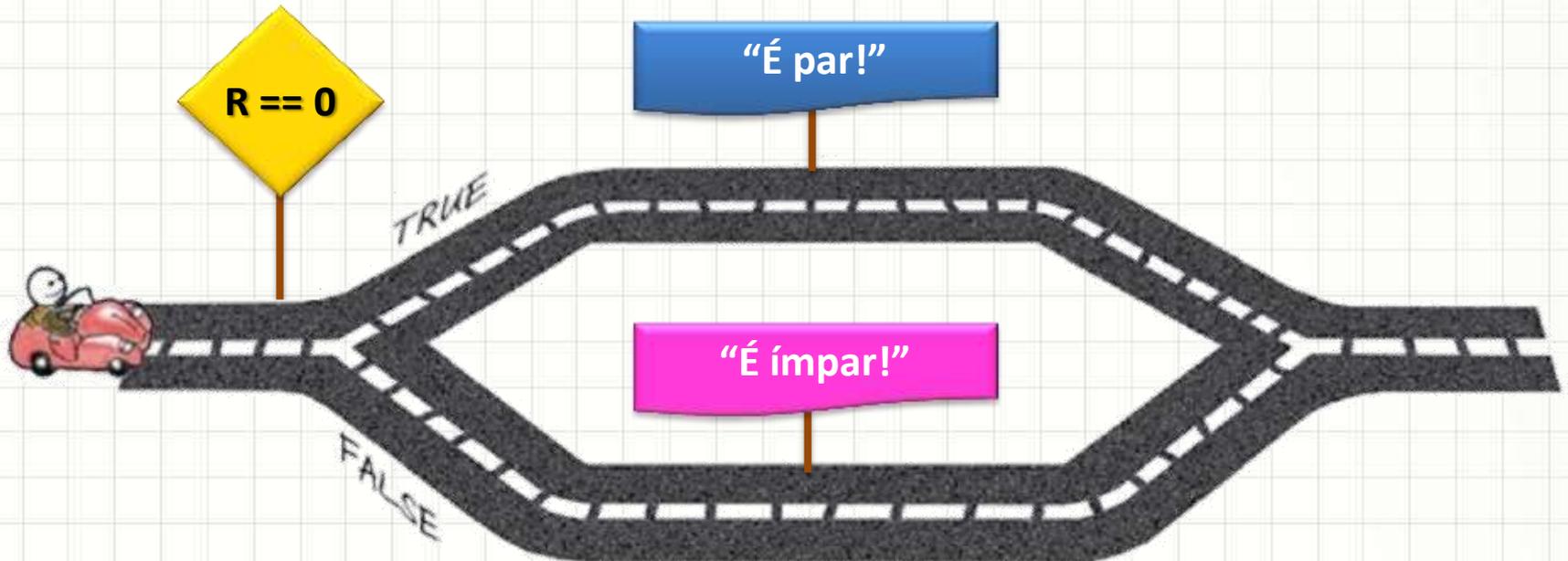
# **ESTRUTURA DE DECISÃO COMPLETA**

# Estrutura de Decisão Completa

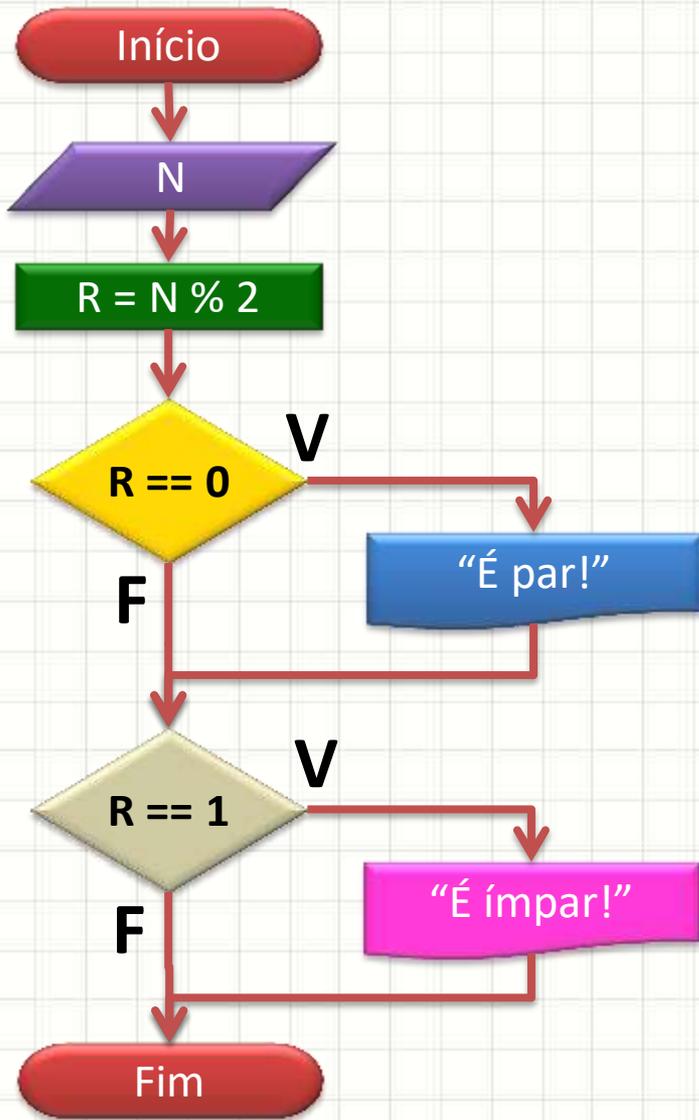
- Ao invés disso...



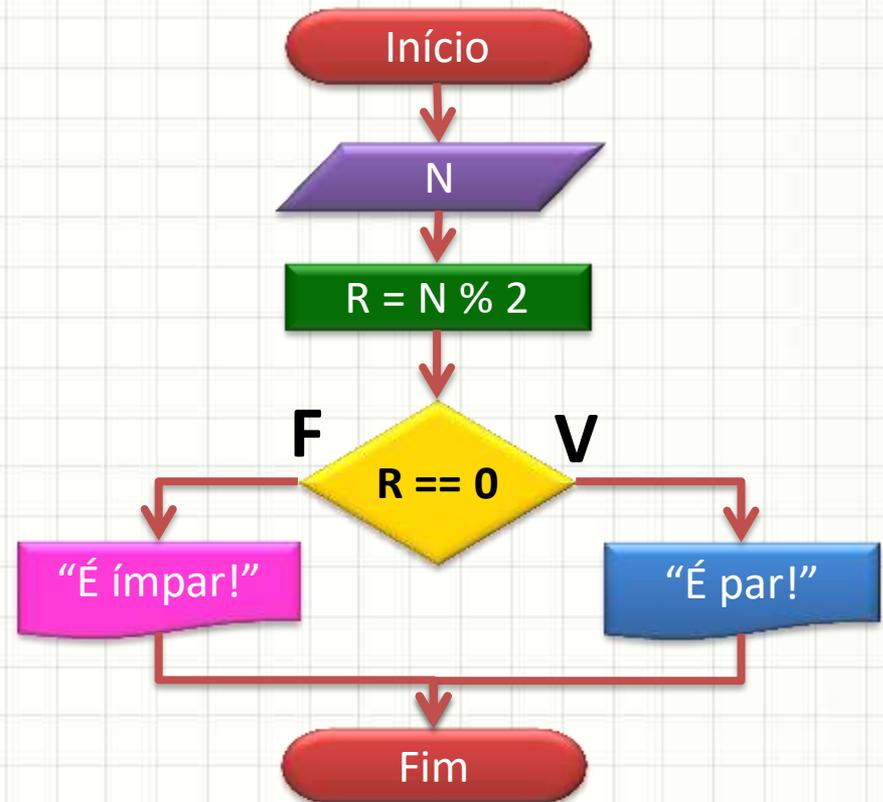
- Não seria legal poder fazer isso?



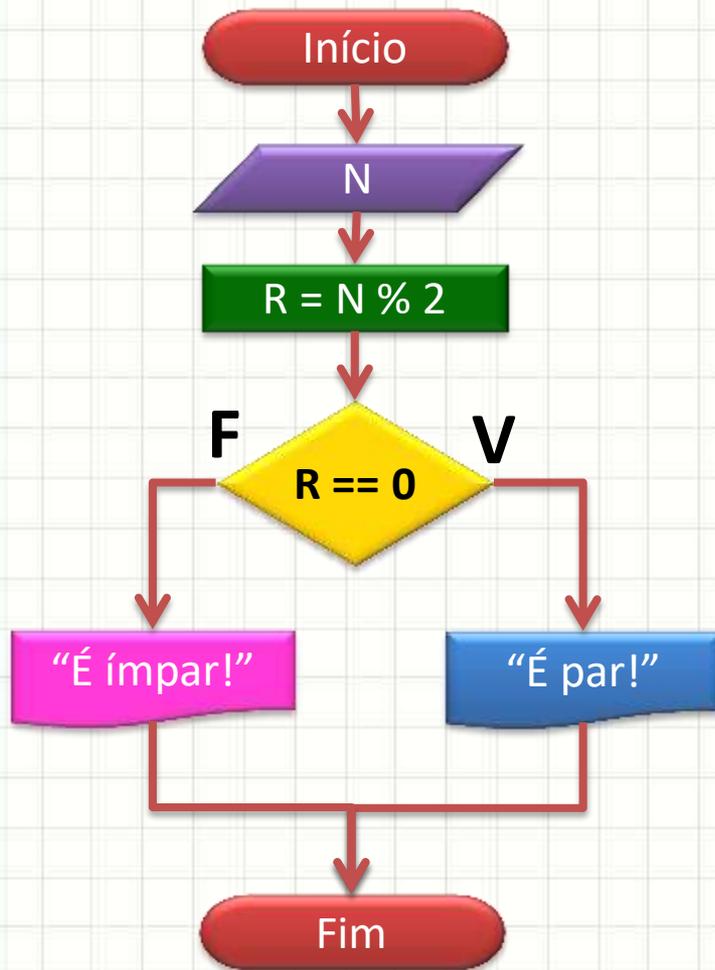
# Estrutura de Decisão Completa



- Observe...
  - A lógica não é igual?



# Verifica se número é par ou ímpar



- Portugol

**Algoritmo** “Verifica se é par ou ímpar”

**Início**

inteiro N, R

escreva(“Digite um Nº: ”)

leia(N)

$R \leftarrow N \% 2$

se  $R == 0$  :

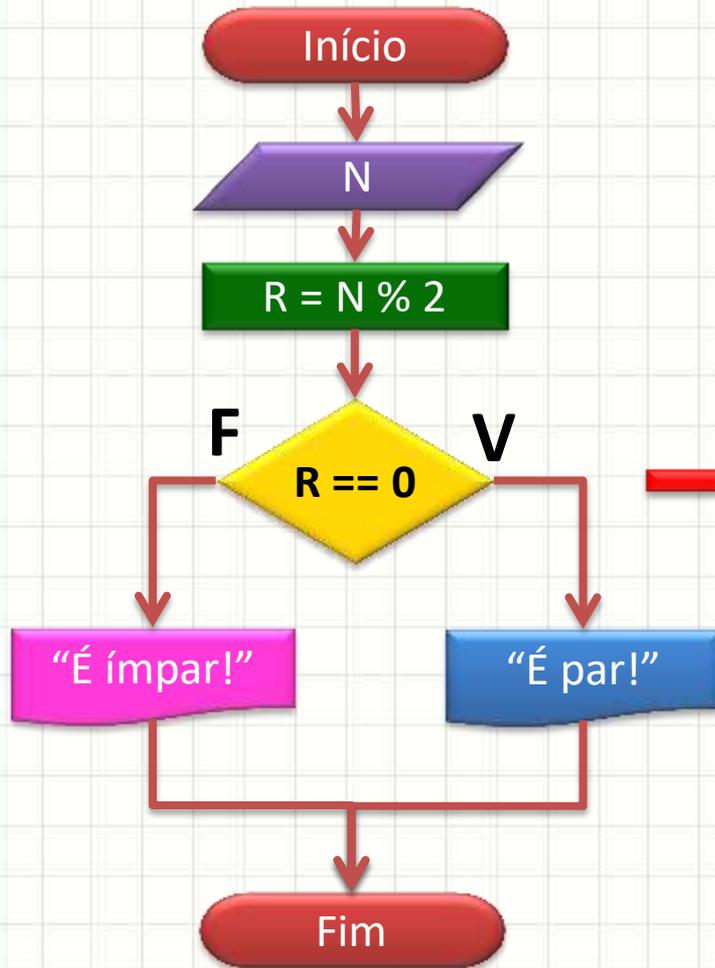
    escreva(“É par!”)

senão:

    escreva(“É ímpar!”)

**Fim**

# Verifica se número é par ou ímpar



- Python

**# Verifica se é par ou ímpar**

```
N = int(input("Digite um N°: "))
```

```
R = N % 2
```

```
if R == 0 :
```

```
    print("É par!")
```

```
else :
```

```
    print("É ímpar!")
```

# Exercício 4

- Faça um programa que **receba a nota** de um aluno e responda que ele está **aprovado** se a **NOTA  $\geq 6,0$**  e **reprovado caso contrário**

- Exemplo

**# Verifica Reprovação**

```
N = float(input("Digite a nota: "))
```

```
if N < 6.0 :
```

```
    print("Reprovado!")
```

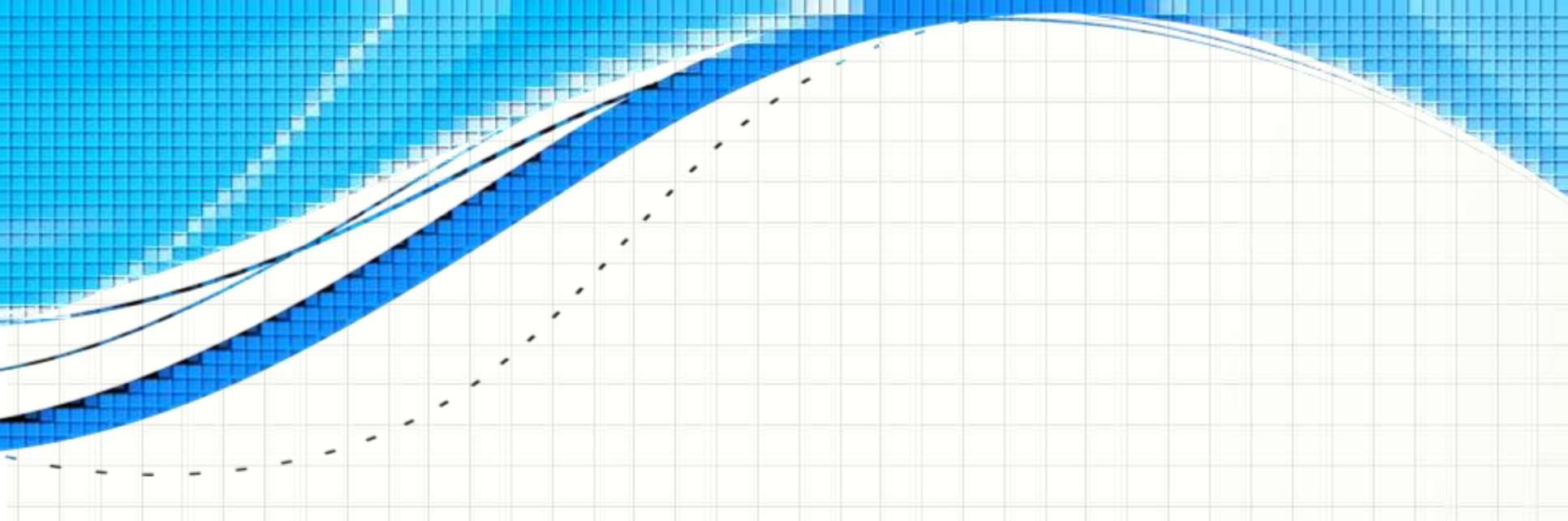
```
    print("Nota:", N)
```

# Exercício 4

- Faça um programa que **receba a nota** de um aluno e responda que ele está **aprovado** se a **NOTA  $\geq 6,0$**  e **reprovado caso contrário**

aula09ex04.py

```
# Verifica Aprovação ou Reprovação  
N = float(input("Digite a nota: "))  
if N >= 6.0 :  
    print("Aprovado!")  
else:  
    print("Reprovado!")
```



# **DECISÕES ANINHADAS OU ENCADEADAS**

# Exercício

- É possível usar **else** no programa que lê dois inteiros e mostra **sucesso** quando o 1º for maior que o 2º ou **fracasso** quando o 2º for maior que o 1º?

aula09ex02.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
if N2 > N1 : 
    print("Fracasso!") 
```

aula09ex02.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
else :
    print("Fracasso!")
```

**O que acontece se N1 == N2?**

# Exercício 5

- Modifique o programa abaixo para que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º, **fracasso** quando o 2º for maior que o 1º ou **empate** quando forem iguais.

Exemplo.py

```
# Verifica se 1º é maior que 2º  
N1 = int(input("Digite um Nº:"))  
N2 = int(input("Digite outro Nº:"))  
if N1 > N2 :  
    print("Sucesso!")  
if N2 > N1 :  
    print("Fracasso!")
```

# Exercício 5

- Modifique o programa abaixo para que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º, **fracasso** quando o 2º for maior que o 1º ou **empate** quando forem iguais.

aula09ex05.py

```
# Verifica se 1º é maior que 2º
```

```
N1 = int(input("Digite um Nº:"))
```

```
N2 = int(input("Digite outro Nº:"))
```

```
if N1 > N2 :
```

```
    print("Sucesso!")
```

```
if N2 > N1 :
```

```
    print("Fracasso!")
```

```
if N1 == N2 :
```

```
    print("Empate!")
```

Mais de uma  
pode ocorrer  
ao mesmo  
tempo?

# Decisões Encadeadas



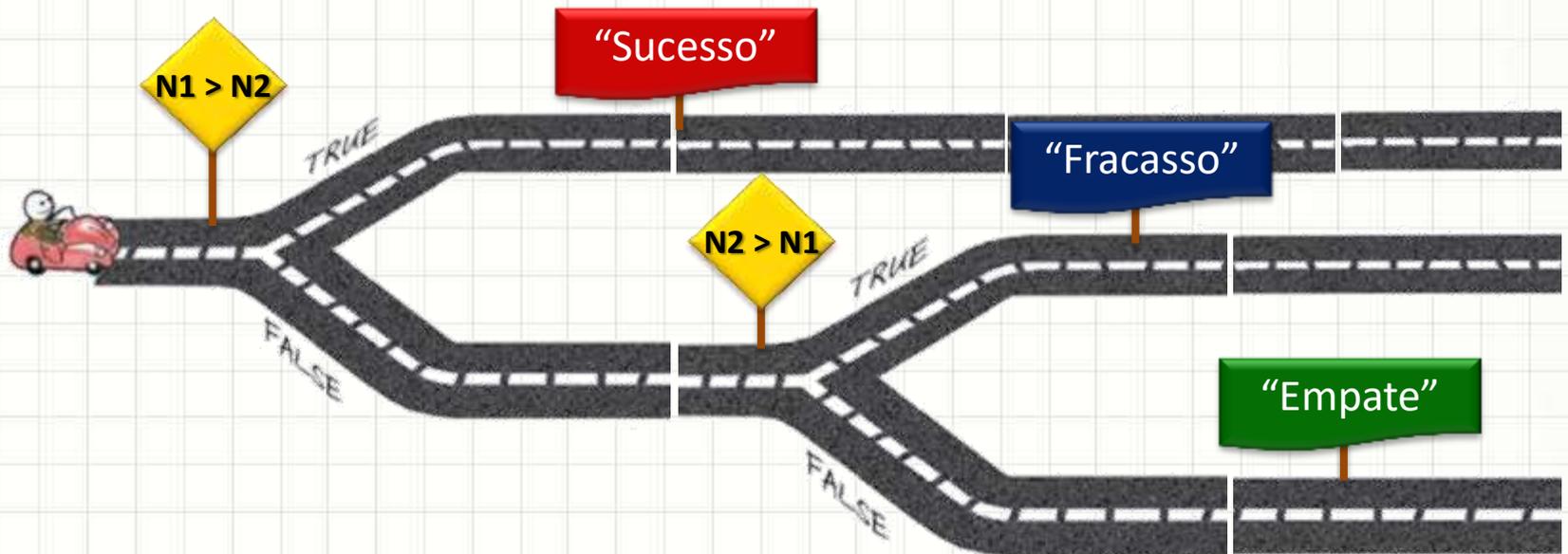
aula09ex05.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
if N2 > N1 :
    print("Fracasso!")
if N1 == N2 :
    print("Empate!")
```

# Decisões Encadeadas



- Alternativa?

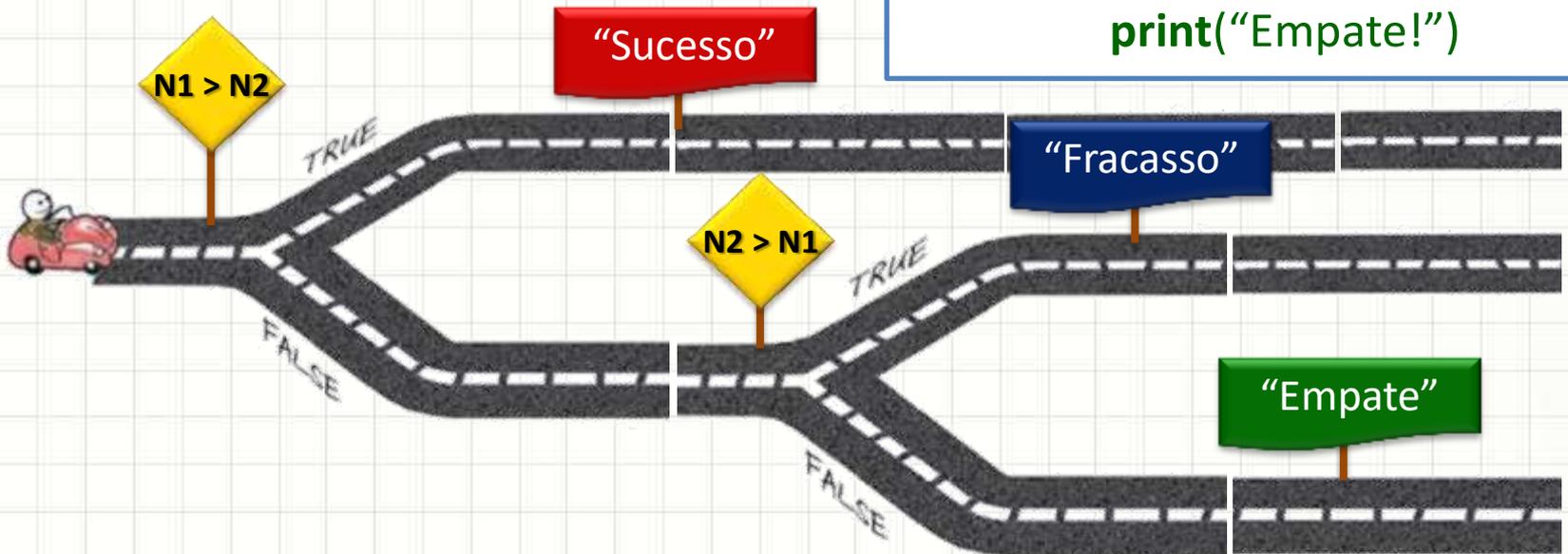


# Decisões Encadeadas

aula09ex05a.py



```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um N°:"))
N2 = int(input("Digite outro N°:"))
if N1 > N2 :
    print("Sucesso!")
else:
    if N2 > N1 :
        print("Fracasso!")
    else:
        print("Empate!")
```



# Exemplo

- Modifique o programa abaixo para que **leia dois inteiros** e mostre **sucesso** quando o 1º for maior que o 2º, **fracasso** quando o 2º for maior que o 1º ou **empate** quando forem iguais.

aula09ex05.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
if N2 > N1 :
    print("Fracasso!")
if N1 == N2 :
    print("Empate!")
```

aula09ex05a.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
else:
    if N2 > N1 :
        print("Fracasso!")
    else:
        print("Empate!")
```

# Instrução Elif

- A instrução “elif” agrega o “else if”

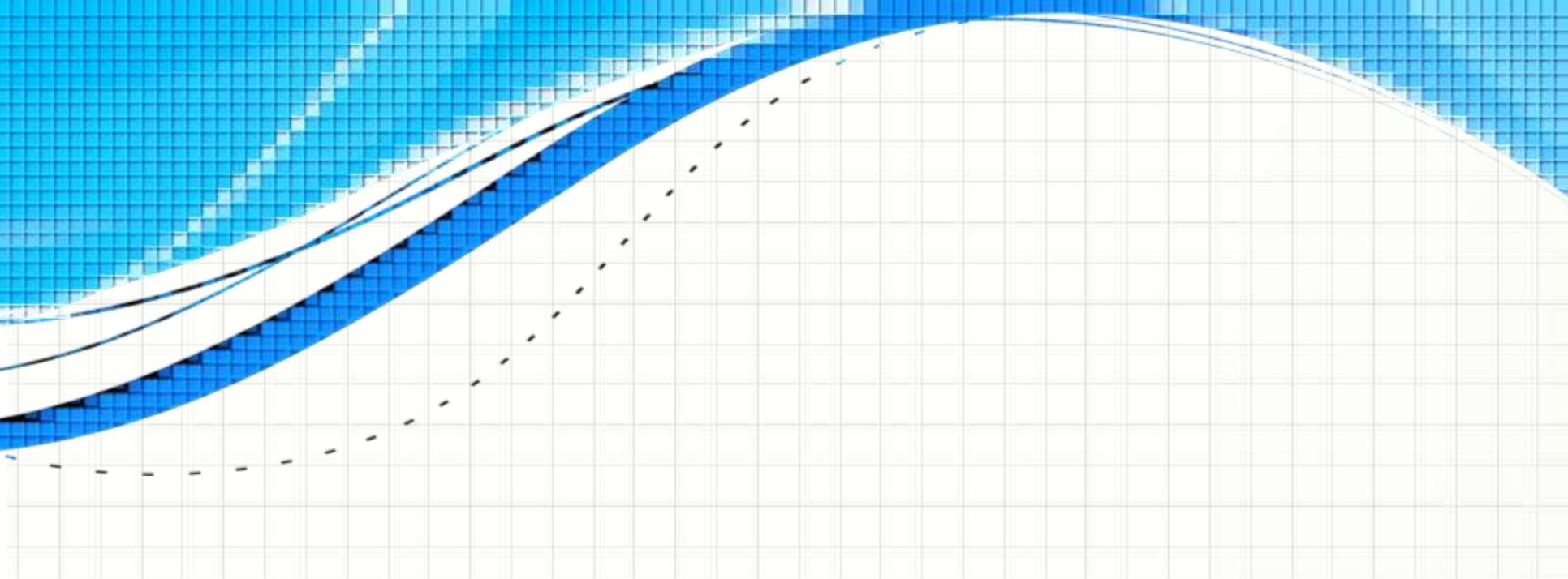
aula09ex05a.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
else:
    if N2 > N1 :
        print("Fracasso!")
    else:
        print("Empate!")
```

aula09ex05b.py

```
# Verifica se 1º é maior que 2º
N1 = int(input("Digite um Nº:"))
N2 = int(input("Digite outro Nº:"))
if N1 > N2 :
    print("Sucesso!")
elif N2 > N1 :
    print("Fracasso!")
else:
    print("Empate!")
```

**Na próxima aula  
faremos exercícios!**



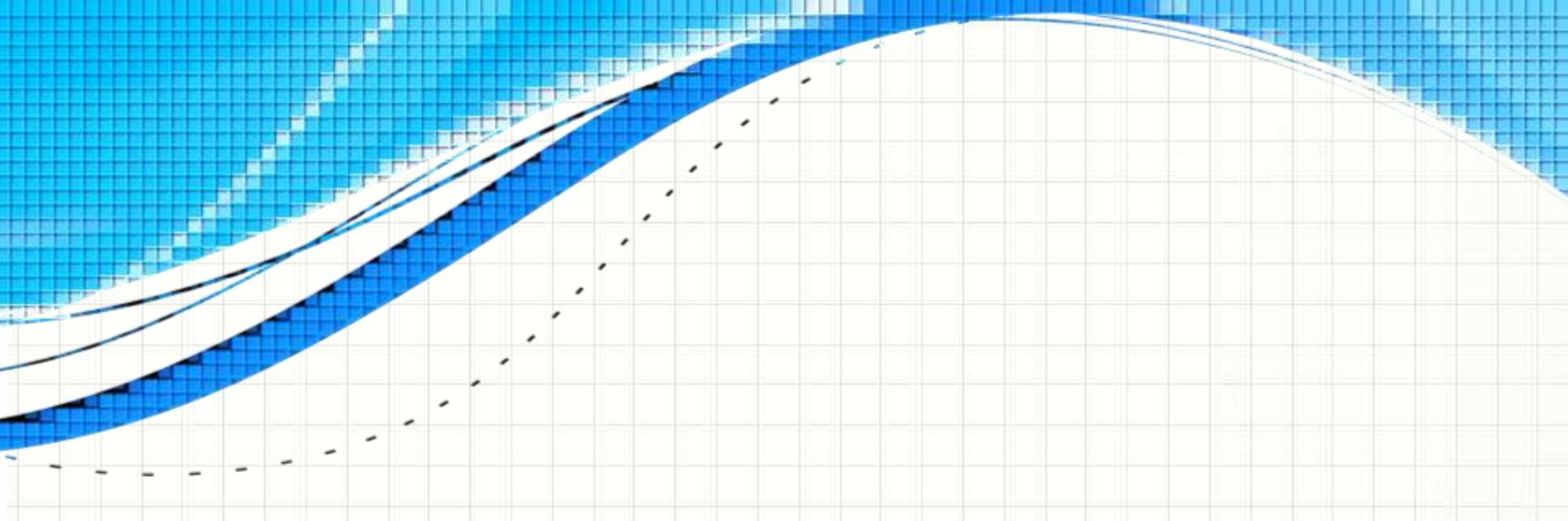
**CONCLUSÕES**

# Resumo

- Estruturas de decisão: comportamento muda
  - De acordo com proposições lógicas
- Estruturas de decisão: trazem flexibilidade
  - Computador pode tomar decisões em cascata
- **TAREFA: Atividade Aula 9!**

**SAVA9!**

- Vídeos: estruturas de decisão!
- Exercitando decisões complexas!
  - Decidindo a melhor estrutura para a decisão!



**PERGUNTAS?**