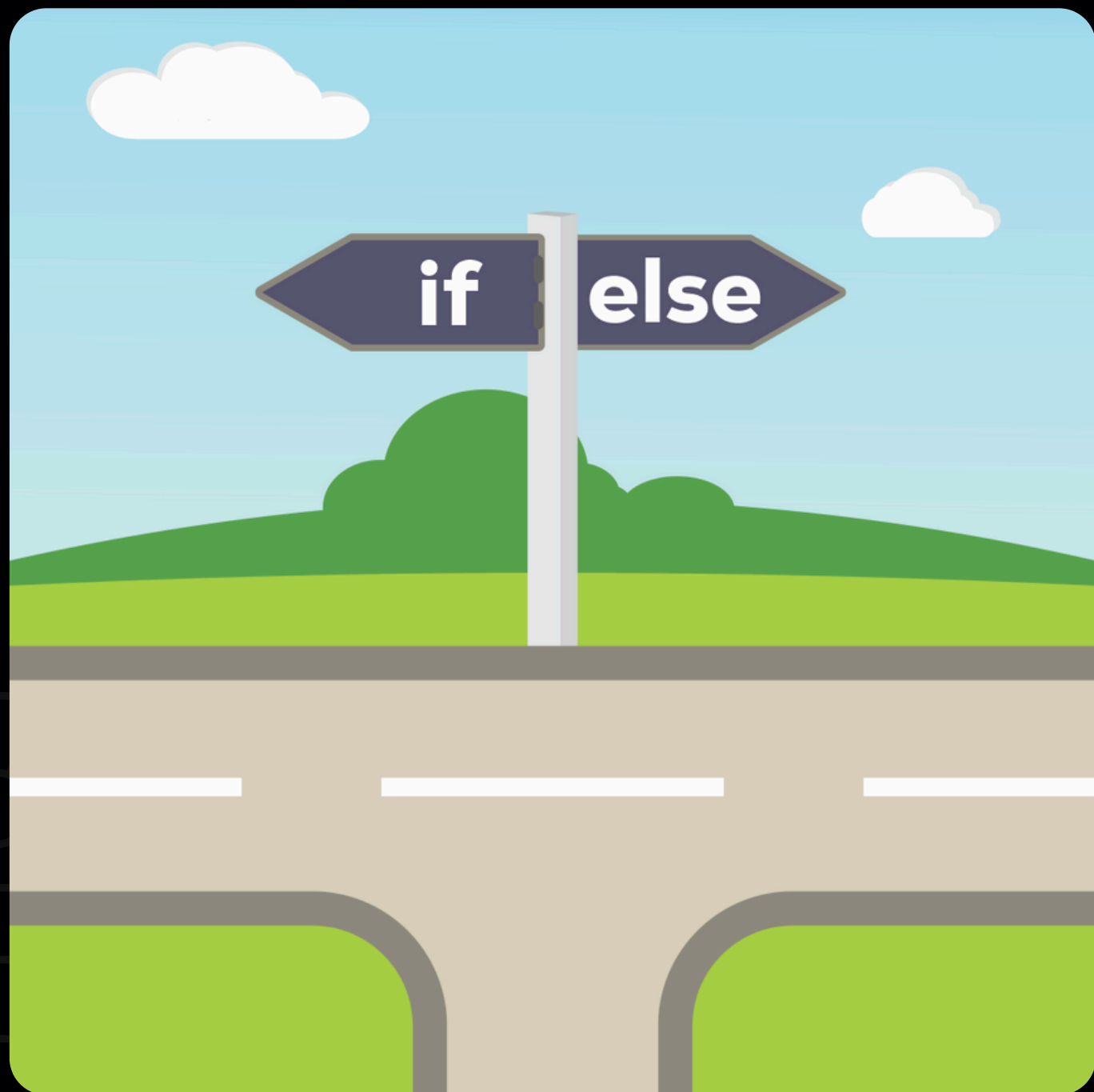


# Estrutura de Decisão

Conceito e Sintaxe

# A IMPORTÂNCIA DE ESTRUTURAS DE DECISÃO



## 01 Capacidade de tomar caminhos com o código

Fazer isso. **(if)**

ou

Fazer aquilo **(else)**

## 02 Criação de códigos não lineares

Estruturas de decisão possibilitam o código em ser dinâmico, fazer algo ou não.

# NECESSIDADE DA TOMADA DE DECISÃO

Avaliação de condições para determinar se, o que e como algo deve ou não ser feito.

## 01 Sistema de estoque

Algoritmo que **decide** se um produto deve ou não ser recarregado ao estoque com base em seu estado atual.

## 02 Semáforos

**Decisões** são tomadas pra definir se ou quando ele deve mudar de cor.

## 03 Jogos

Quando você faz algo, **decisões** são tomados no código do jogo para calcular o que deve acontecer a seguir.

## 04 Inteligencia Artificial

Independente do que voce peça para uma I.A. fazer, decisões serão tomadas para calcular o que deve ser feito

# QUAL O PAPEL PRÁTICO DESSAS ESTRUTURAS?



01

**Resposta Dinâmica:** Caminhos de código diferentes diante de condições específicas, ajuste de comportamento ao longo da execução.

02

**Adaptabilidade:** Variação de comportamento em uma gama de cenários, não só para alterar por completo seu caminho como fazer pequenos ajustes em parâmetros a serem utilizados.

03

**Versatilidade:** Faz do código mais útil em diferentes contextos, como na área da análise de dados, Inteligencia Artificial, Automação e mais.

# SINAIS LÓGICOS

Parametros que indicam ao código o teste que deve ser feito com determinada variável.

- Menor que:  **$a < b$**
- Menor ou igual a:  **$a \leq b$**
- Maior que:  **$a > b$**
- Maior ou igual a:  **$a \geq b$**
- igual a:  **$a == b$**
- Diferente de:  **$a != b$**



# EXEMPLOS TEÓRICOS

01

Verificação de Idade: (se é ou não adulto)

??



if idade > 17

else

É adulto.

não é adulto.

02

Teste de nota: (se está Reprovado ou Aprovado)

nota = X.X

if nota < 7.0

Reprovado.

else

Aprovado.



03

## Teste numérico: (se é impar ou par)

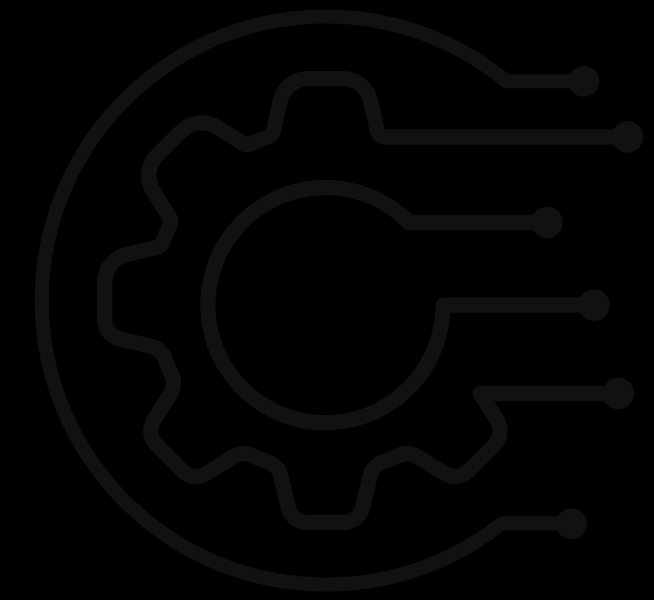
num = **X**

if num % 2 != 0

**X** é Impar.

else

**X** é Par.





# SINTAXE DE ESTRUTURA DE DECISÃO SIMPLES

```
1  if (condicao) {  
2      // Executado se a condição1 for verdadeira  
3  }  
4  //caso seja falsa, nada ocorre.
```

# SINTAXE DE ESTRUTURA DE DECISÃO COMPOSTA

```
1  if (condicao) {  
2      // Código se a condição for verdadeira  
3  } else {  
4      // Código se a condição for falsa  
5  }  
6
```

# EXEMPLOS PRÁTICOS:

01

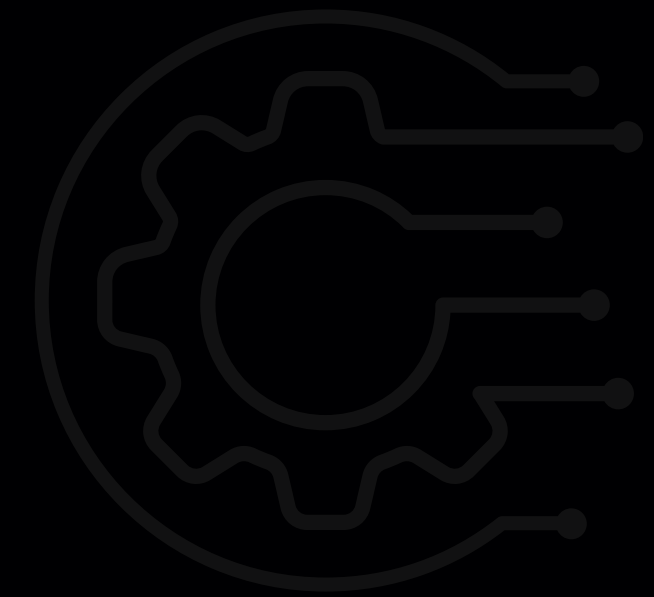
## Verificação de Idade: (se é ou não adulto)

```
1  #include <stdio.h>
2
3  int main() {
4      int idade;
5
6      printf("Digite a sua idade: ");
7      scanf("%d", &idade);
8
9      if (idade >= 18) {
10         printf("Você é maior de idade.\n");
11     } else {
12         printf("Você é menor de idade.\n");
13     }
14
15     return 0;
16 }
```

## Teste de Nota: (se está Reprovado ou Aprovado)

```
1  #include <stdio.h>
2
3  int main() {
4      float nota;
5
6      printf("Digite a sua nota na disciplina: ");
7      scanf("%f", &nota);
8
9      if (nota < 7.0) {
10         printf("Reprovado.\n");
11     } else {
12         printf("Aprovado.\n");
13     }
14
15     return 0;
16 }
17
```

## Teste numérico: (se é ímpar ou par)



```
1  #include <stdio.h>
2
3  int main() {
4      int numero;
5
6      printf("Digite um número inteiro: ");
7      scanf("%d", &numero);
8
9      if (numero % 2 != 0) {
10         printf("%d é um número ímpar.\n", numero);
11     } else {
12         printf("%d não é um número ímpar.\n", numero);
13     }
14
15     return 0;
16 }
17
```

# DESAFIO

Peça ao usuário que digite um ano, exemplo: "2016" determine e retorne se o ano digitado é ou não bissexto.

**Dica:** Um ano é bissexto se for divisível por 4, mas não por 100, exceto se também for divisível por 400.

