

# Programação Aplicada à Engenharia

Professora Priscila Marques Kai  
Aula 4

# Algoritmos e suas representações

## Algoritmo:

1. matemática  
sequência finita de regras, raciocínios ou operações que, aplicada a um número finito de dados, permite solucionar classes semelhantes de problemas.
2. informática  
conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas



# Algoritmos e suas representações

## Algoritmo:



- Representa uma sequência de regras que devem ser executadas em uma ordem preestabelecida;
- Possui um conjunto finito;
- Deve seguir alguma convenção;
- Servem para uso geral.

# Algoritmos e suas representações

## Algoritmo:

- Representa uma sequência de regras que devem ser executadas em uma ordem preestabelecida;
- Possui um conjunto finito;
- Deve seguir alguma convenção;
- Servem para uso geral.

# Algoritmos e suas representações

Exemplo não computacional de um algoritmo: Sorvete de chocolate

Algumas informações sobre a receita:

Deve conter:

- 1 tablete de chocolate meio amargo;
- 1 lata de leite condensado;
- a mesma medida da lata de leite;
- raspas de chocolate ou chocolate granulado.

# Algoritmos e suas representações

Exemplo não computacional de um algoritmo: Sorvete de chocolate

## **Algoritmo** Sorvete de chocolate

### **Início**

1. Coloque o chocolate em uma tigela.
2. Deixe a tigela no micro-ondas durante um minuto em potência média.
3. Tire o chocolate do forno com cuidado e mexa-o até esfriar.
4. Bata-o no liquidificador com o leite condensado e o leite.
5. Despeje tudo em uma forma de gelo e espere congelar por três horas.
6. Distribua o sorvete em taças.
7. Sirva.

### **Fim**

# Algoritmos e suas representações

Exemplo computacional de um algoritmo: Maior valor entre dois números

Informações:

- x e y são as variáveis de entrada
- Deve-se comparar os dois números e imprimir na tela o maior entre os dois números de entrada

# Algoritmos e suas representações

Exemplo computacional de um algoritmo: Maior valor entre dois números

## **Algoritmo IMC**

### **Início**

//Pedir para o usuário fornecer os valores para x e y.

1. Se  $x > y$  então

    imprima("o maior número é o x")

2. senão

    imprima("o maior número é o y")

### **Fim**



# Algoritmos e suas representações

Exemplo computacional de um algoritmo: Maior valor entre dois números

```
def maior_entre_x_e_y(x,y):  
    if x > y:  
        print("O maior valor entre ",x ,"e ",y,"é ",x)  
    else:  
        print("O maior valor entre ",x ,"e ",y,"é ",y)  
  
if __name__ == "__main__":  
    x = input("Insira o primeiro valor:")  
    y = input("Insira o segundo valor:")  
  
    maior_entre_x_e_y(x, y)
```

# Estrutura Condicional

Estrutura de decisão SE-ENTÃO-SENÃO

A maioria dos algoritmos precisam tomar decisões ao longo de sua execução. Para isso existem as estruturas de decisão, e a mais utilizada é a estrutura:

SE-ENTÃO-SENÃO

(IF-THEN-ELSE)

## Estrutura Condicional

Com base no resultado de uma expressão booleana (VERDADEIRO ou FALSO), o fluxo do algoritmo segue para um bloco de instruções ou não.

Estrutura SE-ENTÃO-SENÃO:

SE <expressão booleana> ENTÃO

<instruções a serem executadas caso a expressão booleana resulte em VERDADEIRO>

SENÃO

<instruções a serem executadas caso a expressão booleana resulte em FALSO>

FIM-SE

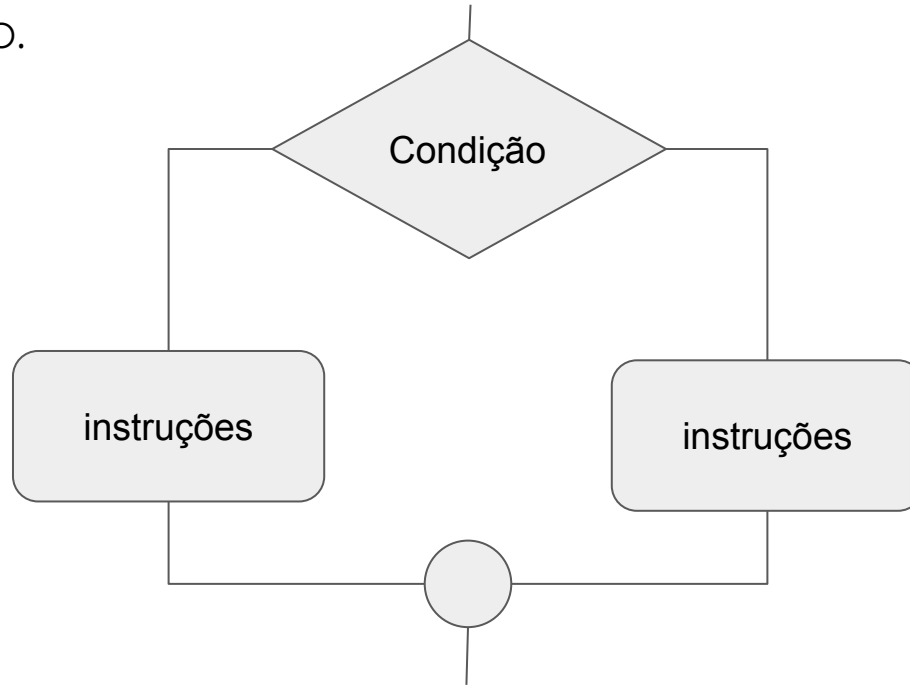
## Estrutura Condicional

Em Lógica de Programação, usamos condições para tomar decisões no decorrer da execução de um programa. Através de estruturas condicionais podemos alterar todo o fluxo de processamento.

Um exemplo simples de estrutura de decisão é uma conta corrente de um banco. Entendemos que somente será possível realizar saques na conta se houver saldo disponível, caso contrário, o saque será negado. Note que esta decisão basicamente é um VERDADEIRO ou FALSO, ou seja, SE houve saldo a condição será VERDADEIRA e o saque será permitido, SENÃO, a condição falsa será negar o saque.

## Estrutura Condicional

O bloco de código SENÃO é opcional. É comum encontrar instruções de decisão apenas com SE-ENTÃO sem o bloco SENÃO. Veja um esquema gráfico desta estrutura de decisão.



# Algoritmos e suas representações

```
ALGORITMO "SacarDinheiro"
var
    SaldoDisponivel : REAL
    ValorDoSaque : REAL
INÍCIO
    SaldoDisponivel ← 2000
    ESCREVA ("Informe o valor do Saque: ")
    LEIA (ValorDoSaque)
    SE ValorDoSaque <= SaldoDisponivel ENTÃO
        SaldoDisponivel ← SaldoDisponivel - ValorDoSaque
        ESCREVA ("Sacando R$ ", ValorDoSaque, ".")
    SENAO
        ESCREVA ("O valor solicitado é maior que o valor disponível para saque!")
    FIMSE
    ESCREVA ("Saldo disponível: R$ ", SaldoDisponivel)
FIM
```

ALGORITMO "Cálculo de raízes de uma equação de 2º grau

var

A : INTEIRO

B : INTEIRO

C : INTEIRO

INÍCIO

LEIA (A)

LEIA (B)

LEIA ©

SE A = 0 ENTÃO

IMPRIMA("Não é equação de 2º grau")

SENÃO

$D \leftarrow (B^2) - 4 \cdot A \cdot C$

SE D < 0 ENTÃO

IMPRIMA("Não existem raízes reais!")

SENÃO

$r1 \leftarrow (-B + \sqrt{D}) / (2 \cdot A)$

$r2 \leftarrow (-B - \sqrt{D}) / (2 \cdot A)$

IMPRIMA(r1)

IMPRIMA(r2)

FIMSE

FIMSE

FIM

# Exercícios

- 1) **Decisão de Maioridade:** Desenvolva um algoritmo que determine se uma pessoa é maior de idade com base em sua idade. Se a idade for igual ou superior a 18 anos, exiba a mensagem "Maior de Idade". Caso contrário, exiba "Menor de Idade".

## ALGORITMO MAIORIDADE

**var**

idade: **INTEIRO**

**INÍCIO**

// Solicitar a idade da pessoa

**ESCREVER** "Digite a idade da pessoa: "

**LER** idade

// Verificar se a idade é igual ou superior a 18 anos

**SE** idade  $\geq$  18 **ENTÃO**

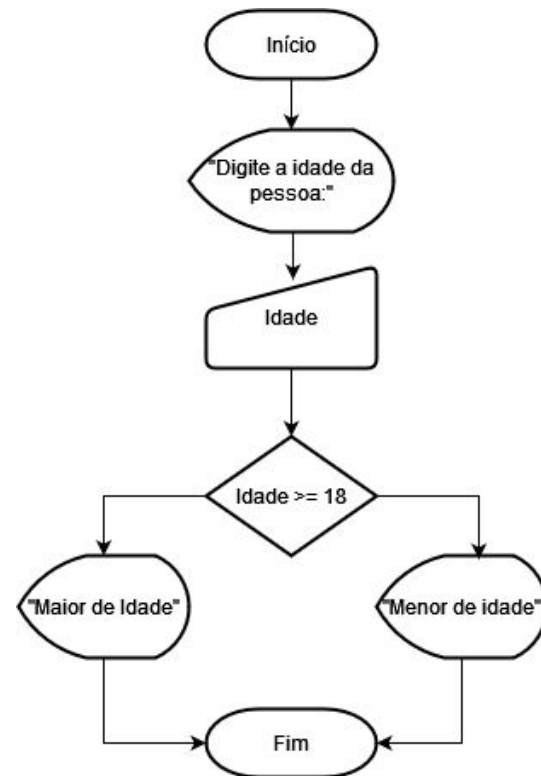
**ESCREVER** "Maior de Idade"

**SENÃO**

**ESCREVER** "Menor de Idade"

**FIM SE**

**FIM**





# Exercícios

- 1) **Decisão de Maioridade:** Desenvolva um algoritmo que determine se uma pessoa é maior de idade com base em sua idade. Se a idade for igual ou superior a 18 anos, exiba a mensagem "Maior de Idade". Caso contrário, exiba "Menor de Idade".

## ALGORITMO MAIORIDADE

**var**

idade: **INTEIRO**

**INÍCIO**

// Solicitar a idade da pessoa

**ESCREVER** "Digite a idade da pessoa: "

**LER** idade

// Verificar se a idade é igual ou superior a 18 anos

**SE** idade >= 18 **ENTÃO**

**ESCREVER** "Maior de Idade"

**SENÃO**

**ESCREVER** "Menor de Idade"

**FIM SE**

**FIM**

```
# Lê a idade da pessoa do teclado
idade = int(input("Digite a idade: "))
```

```
# Verifica se a idade é >= a 18 anos
```

```
if idade >= 18:
    print("Maior de Idade")
```

```
else:
    print("Menor de Idade")
```

# Exercícios

- 2) **Aprovação em Exame:** Escreva um algoritmo que avalie se um aluno foi aprovado em um exame com base em sua nota. Se a nota do aluno for igual ou superior a 6, exiba "Aprovado". Caso contrário, exiba "Reprovado".

## ALGORITMO EXAME

**var**

nota: **FLOAT**

**INÍCIO**

// Solicitar a nota do aluno

**ESCREVER** "Digite a nota do aluno: "

**LER** nota

// Verificar se a nota é igual ou superior a 6

**SE** nota  $\geq$  6 **ENTÃO**

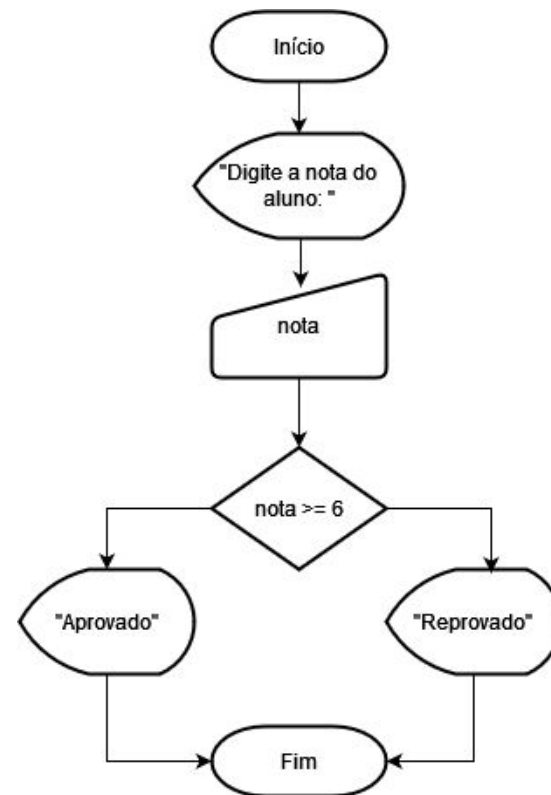
**ESCREVER** "Aprovado"

**SENÃO**

**ESCREVER** "Reprovado"

**FIM SE**

**FIM**



# Exercícios

- 2) **Aprovação em Exame:** Escreva um algoritmo que avalie se um aluno foi aprovado em um exame com base em sua nota. Se a nota do aluno for igual ou superior a 6, exiba "Aprovado". Caso contrário, exiba "Reprovado".

## ALGORITMO EXAME

**var**

nota: **FLOAT**

**INÍCIO**

// Solicitar a nota do aluno

**ESCREVER** "Digite a nota do aluno: "

**LER** nota

// Verificar se a nota é igual ou superior a 6

**SE** nota >= 6 **ENTÃO**

**ESCREVER** "Aprovado"

**SENÃO**

**ESCREVER** "Reprovado"

**FIM SE**

**FIM**

```
# Lê a nota do aluno do teclado
nota = float(input("Digite a nota do aluno: "))
```

```
# Verifica se a nota é >= a 6
```

```
if nota >= 6:
```

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

```
else:
```

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

# Exercícios

- 3) **Verificação de Número Par ou Ímpar:** Crie um algoritmo que determine se um número inteiro é par ou ímpar. Se o número for divisível por 2 (ou seja, o resto da divisão por 2 é igual a 0), exiba "Número Par". Caso contrário, exiba "Número Ímpar".

## ALGORITMO PAR\_IMPAR

**var**

numero: **INTEIRO**

**INÍCIO**

// Solicitar o número inteiro

**ESCREVER** "Digite um número inteiro: "

**LER** numero

// Verificar se o número é par ou ímpar

**SE** numero % 2 = 0 **ENTÃO**

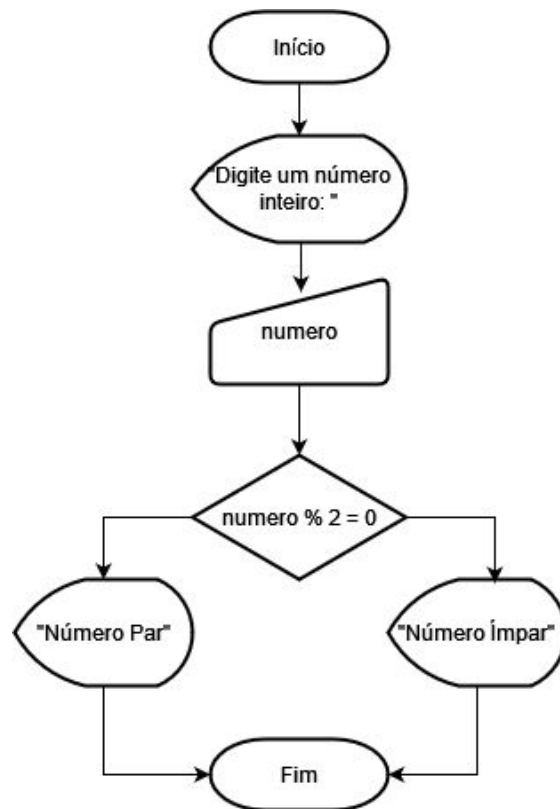
**ESCREVER** "Número Par"

**SENÃO**

**ESCREVER** "Número Ímpar"

**FIM SE**

**FIM**



# Exercícios

- 3) **Verificação de Número Par ou Ímpar:** Crie um algoritmo que determine se um número inteiro é par ou ímpar. Se o número for divisível por 2 (ou seja, o resto da divisão por 2 é igual a 0), exiba "Número Par". Caso contrário, exiba "Número Ímpar".

## ALGORITMO PAR\_IMPAR

**var**

numero: **INTEIRO**

**INÍCIO**

// Solicitar o número inteiro

**ESCREVER** "Digite um número inteiro: "

**LER** numero

// Verificar se o número é par ou ímpar

**SE** numero % 2 = 0 **ENTÃO**

**ESCREVER** "Número Par"

**SENÃO**

**ESCREVER** "Número Ímpar"

**FIM SE**

**FIM**

```
# Lê um número inteiro do teclado
```

```
numero = int(input("Digite um número inteiro: "))
```

```
# Verifica se o número é divisível por 2 pelo resto  
# da divisão (resto igual a 0)
```

```
if numero % 2 == 0:
```

```
    print("Número Par")
```

```
else:
```

```
    print("Número Ímpar")
```

# Exercícios

- 4) **Verificação de Triângulo:** Desenvolva um algoritmo que determine o tipo de triângulo com base nos comprimentos de seus lados. Se todos os lados forem iguais, exiba "Triângulo Equilátero". Se dois lados forem iguais, exiba "Triângulo Isósceles". Caso contrário, exiba "Triângulo Escaleno".

## ALGORITMO TRIANGULO

**var**

lado1: **INTEIRO**

lado2: **INTEIRO**

lado3: **INTEIRO**

**INÍCIO**

// Solicitar os comprimentos dos lados do triângulo

**ESCREVER** "Digite o comprimento do primeiro,segundo e terceiro lado: "

**LER** lado1, lado2, lado3

// Verificar o tipo de triângulo

**SE** lado1 = lado2 **E** lado2 = lado3 **ENTÃO**

**ESCREVER** "Triângulo Equilátero"

**SENÃO**

**SE** (lado1 = lado2) **OU** (lado1 = lado3) **OU** (lado2 = lado3) **ENTÃO**

**ESCREVER** "Triângulo Isósceles"

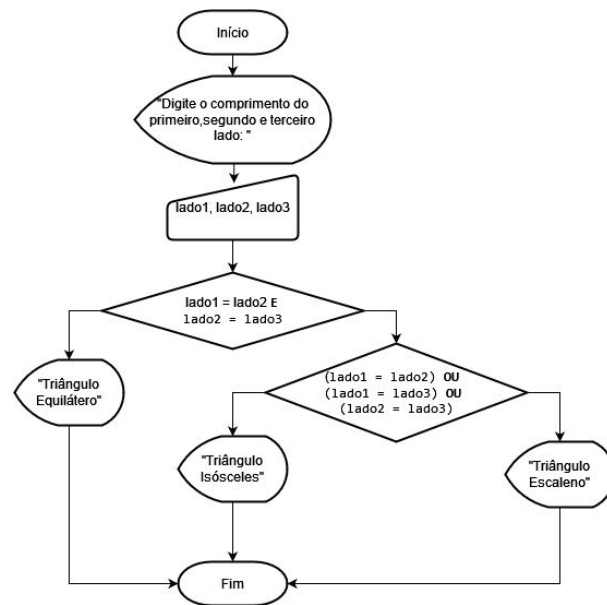
**SENÃO**

**ESCREVER** "Triângulo Escaleno"

**FIM SE**

**FIMSE**

**FIM**



# Exercícios

- 4) **Verificação de Triângulo:** Desenvolva um algoritmo que determine o tipo de triângulo com base nos comprimentos de seus lados. Se todos os lados forem iguais, exiba "Triângulo Equilátero". Se dois lados forem iguais, exiba "Triângulo Isósceles". Caso contrário, exiba "Triângulo Escaleno".

## ALGORITMO TRIANGULO

```
var
    lado1: INTEIRO
    lado2: INTEIRO
    lado3: INTEIRO
INÍCIO
    // Solicitar os comprimentos dos lados do triângulo
    ESCREVER "Digite o comprimento do primeiro, segundo e terceiro lado: "
    LER lado1, lado2, lado3

    // Verificar o tipo de triângulo
    SE lado1 == lado2 E lado2 == lado3 ENTÃO
        ESCREVER "Triângulo Equilátero"
    SENÃO
        SE (lado1 == lado2) OU (lado1 == lado3) OU (lado2 == lado3) ENTÃO
            ESCREVER "Triângulo Isósceles"
        SENÃO
            ESCREVER "Triângulo Escaleno"
        FIM SE
    FIMSE
FIM
```

```
# Lê os comprimentos dos lados do triângulo
lado1 = float(input("Digite o comprimento do primeiro lado: "))
lado2 = float(input("Digite o comprimento do segundo lado: "))
lado3 = float(input("Digite o comprimento do terceiro lado: "))

# Verifica o tipo de triângulo
if lado1 == lado2 and lado2 == lado3:
    print("Triângulo Equilátero")
else:
    if lado1 == lado2 or lado1 == lado3 or lado2 == lado3:
        print("Triângulo Isósceles")
    else:
        print("Triângulo Escaleno")
```

```
...
# Verifica o tipo de triângulo
if lado1 == lado2 and lado2 == lado3:
    print("Triângulo Equilátero")
elif lado1 == lado2 or lado1 == lado3 or lado2 == lado3:
    print("Triângulo Isósceles")
else:
    print("Triângulo Escaleno")
```

# Exercícios

- 5) **Verificação de Ano Bissexto:** Crie um algoritmo que determine se um ano é bissexto. Se o ano for divisível por 4 e não for divisível por 100, ou for divisível por 400, ele é bissexto. Caso contrário, não é bissexto.

## ALGORITMO VERIFICA\_BISSEXTO

**var**

ano: **INTEIRO**

**INÍCIO**

// Solicitar o ano

**ESCREVER** "Digite um ano: "

**LER** ano

// Verificar se o ano é bissexto

**SE** (ano % 4 = 0 E ano % 100 <> 0) **OU** (ano % 400 = 0) **ENTÃO**

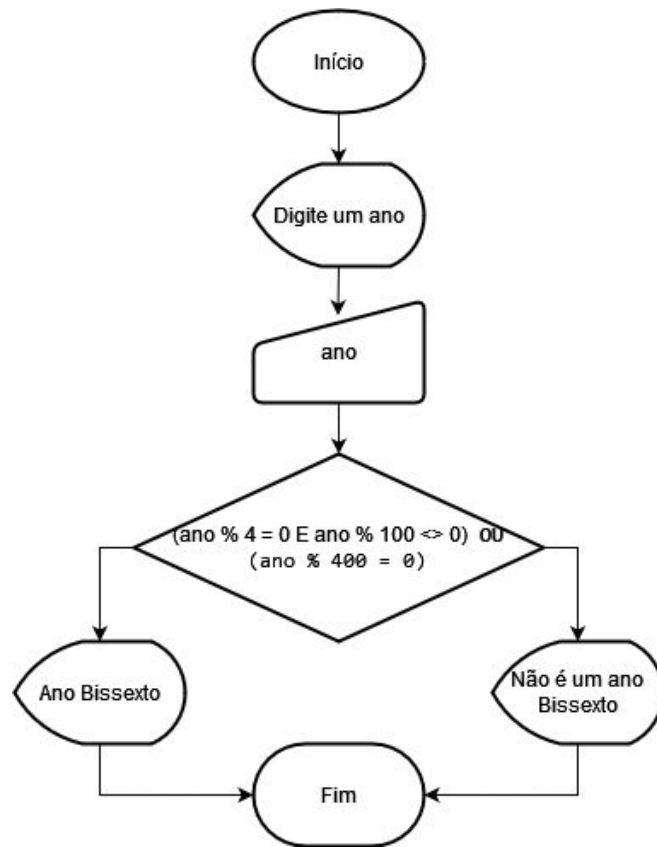
**ESCREVER** "Ano Bissexto"

**SENÃO**

**ESCREVER** "Não é um Ano Bissexto"

**FIM SE**

**FIM**





# Exercícios

- 5) **Verificação de Ano Bissexto:** Crie um algoritmo que determine se um ano é bissexto. Se o ano for divisível por 4 e não for divisível por 100, ou for divisível por 400, ele é bissexto. Caso contrário, não é bissexto.

## ALGORITMO VERIFICA\_BISSEXTO

**var**

    ano: **INTEIRO**

**INÍCIO**

    // Solicitar o ano

**ESCREVER** "Digite um ano: "

**LER** ano

    // Verificar se o ano é bissexto

**SE** (ano % 4 = 0 E ano % 100 <> 0) **OU** (ano % 400 = 0) **ENTÃO**

**ESCREVER** "Ano Bissexto"

**SENÃO**

**ESCREVER** "Não é um Ano Bissexto"

**FIM SE**

**FIM**

```
# Lê ano do teclado
```

```
ano = int(input("Digite um ano: "))
```

```
# Verifica se o ano é bissexto
```

```
if (ano % 4 == 0 and ano % 100 != 0) or (ano % 400 == 0):
```

```
    print("Ano Bissexto")
```

```
else:
```

```
    print("Ano Não Bissexto")
```