

# EXERCÍCIOS DE PROGRAMAÇÃO JAVA

**Samir Azzam**

**SENAC RJ**

[prof.samir.azzam@gmail.com](mailto:prof.samir.azzam@gmail.com)

# **ESTRUTURAS DE CONTROLE**

## TOMADA DE DECISÕES

# Exercício 1

---

- Escreva um programa que receba as notas de duas avaliações de um aluno, calcule a média e informe:
  - ➡ Se a média for menor que 3.0, informar que o aluno está reprovado
  - ➡ Se a média estiver entre 3.0 e 7.0, informar que o aluno estará em prova final
  - ➡ Se a média for maior ou igual a 7.0, informar que o aluno estará aprovado

## Exercício 2

---

- Problema:

Ler o ano de nascimento de uma pessoa, calcular e apresentar a sua idade. Verificar e mostrar mensagem informando se ela já pode tirar a Carteira de Habilitação (18 anos ou mais) ou quantos anos faltam para poder tirar a carteira.

# Exercício 3

---

- Aplicando os Desvios Condicionais Encadeados

"Elaborar um programa que efetue o cálculo do reajuste de salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 500. Se o salário for maior ou igual a 500, mas menor ou igual a 1000, seu reajuste será de 10%; caso seja ainda maior que 1000, o reajuste deverá ser de 5%".

# Exercício 3 - Dicas

---

- Aplicando os Desvios Condicionais Encadeados

Solução: Algoritmo

- 1) Definir uma variável para o salário reajustado: NOVO\_SALÁRIO;
- 2) Ler um valor para a variável SALÁRIO;
- 3) Verificar se o valor de SALÁRIO < SOO, se sim reajustar em 15%;
- 4) Verificar se o valor de SALÁRIO <= 1 000, se sim reajustar em 10%;
- 5) Verificar se o valor de SALÁRIO > 1000, se sim reajustar em 5%;
- 6) Apresentar o valor reajustado, implicado em NOVO\_SALÁRIO.

# Exercício 3 - Dicas

## Solução: Português Estruturado

Programa REAJUSTA\_SALARIO

var

*NOVO\_SALARIO*: inteiro

*SALARIO*: inteiro

inicio

leia *SALARIO*

se (*SALARIO* < 500) então

$NOVO\_SALARIO \leftarrow SALARIO * 1.15$

senão

se (*SALARIO* <= 1000) então

$NOVO\_SALARIO \leftarrow SALARIO * 1.10$

senão

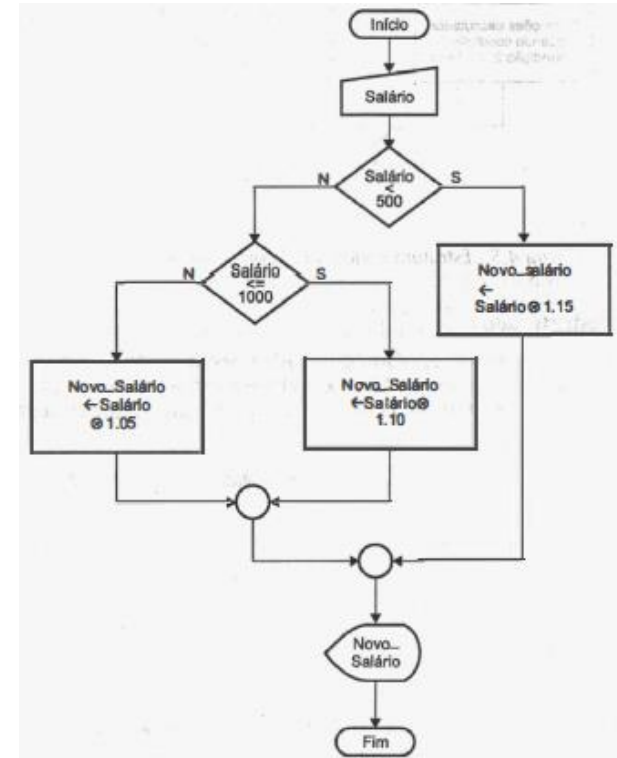
$NOVO\_SALARIO \leftarrow SALARIO * 1.15$

fim\_se

fim\_se

escreva *NOVO\_SALARIO*

fim



# Exercício 4 - Operador lógico - .e.

---

- Aplicando o operador lógico .e.

Problema: “Deseja-se codificar um trecho de programa que faça a verificação de aprovação de um candidato numa etapa de concurso. Para ser aprovado o candidato precisa marcar no mínimo 5 pontos na prova de matemática e não tirar zero na prova de português”.

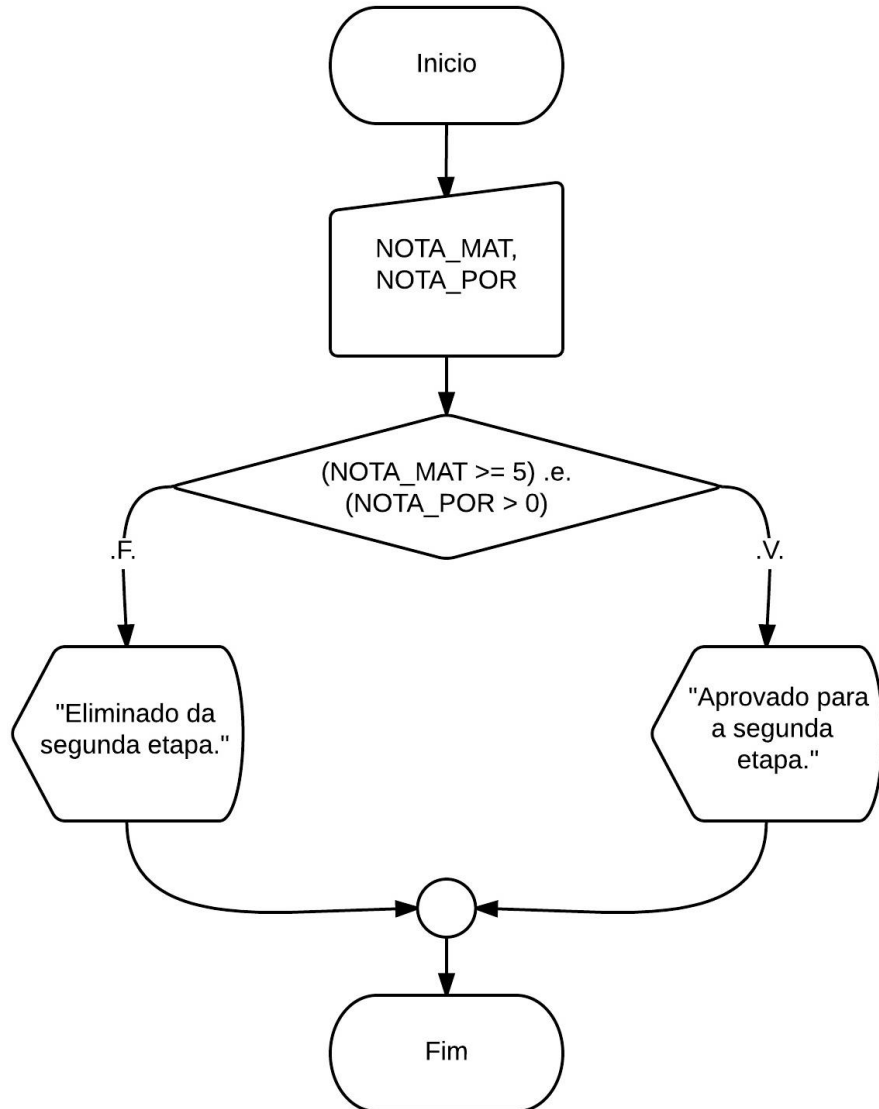


# Exercício 4 - Operador lógico - .e. (Dicas)

---

- Proposta de solução.
- Algoritmo:
  1. Receber a nota da prova de matemática
  2. Receber a nota da prova de português
  3. Verificar se a nota de matemática é igual ou maior que 5 e se a nota de português é maior que 0
  4. Se as condições forem verdadeiras informar que o candidato foi aprovado
  5. Se alguma condição for falsa informar que o candidato foi reprovado

# Exercício 4 - Operador lógico - .e. (Dicas)



Solução Diagrama de Blocos

Software utilizado: Lucidchart  
[www.lucidchart.com](http://www.lucidchart.com)

# Exercício 4 - Operador lógico - .e. (Dicas)

---

## Solução: Português Estruturado

Programa APROVAÇÃO\_CONCURSO

var

*NOTA\_MAT*: inteiro

*NOTA\_POR*: inteiro

inicio

    leia *NOTA\_MAT*, *NOTA\_POR*

    se (*NOTA\_MAT* >= 5) .e. (*NOTA\_POR* > 0) então

        escreva "Aprovado para a segunda etapa."

    senão

        escreva "Eliminado da segunda etapa."

    fim\_se

fim

# Exercício 5 - Operador lógico - .ou.

---

- Aplicando o operador lógico .ou.

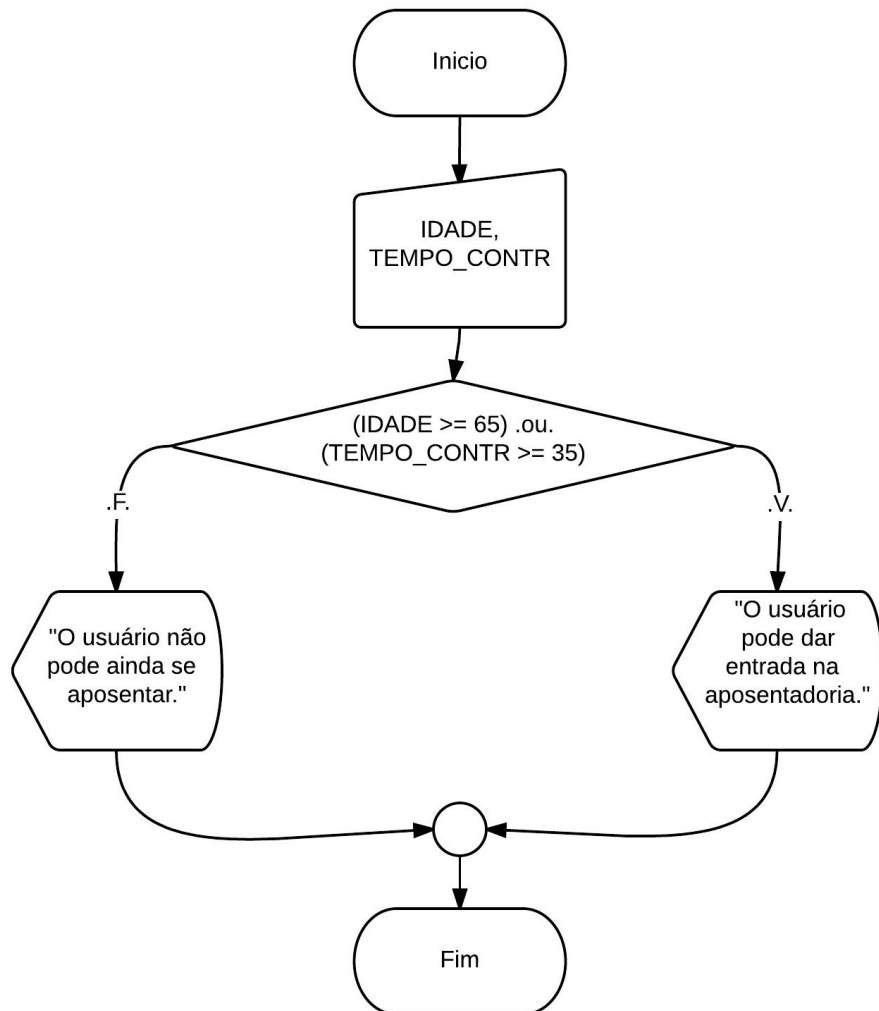
Problema: “Um pessoa pode se aposentar em duas situações: atingir 65 anos de idade ou ter contribuído por 35 anos para a previdência. Escreva um trecho de programa que verifique se uma pessoa pode se aposentar ou não”.

# Exercício 5 - Operador lógico - .ou. (Dicas)

---

- Proposta de solução.
- Algoritmo:
  1. Receber o valor da idade da pessoa
  2. Receber a quantidade de anos como contribuinte
  3. Verificar se a idade é igual ou maior que 65 ou se a quantidade de anos de contribuição é maior ou igual a 35
  4. Se uma das condições for verdadeira, então exibir a mensagem de que a pessoa pode dar entrada na aposentadoria
  5. Senão, informar que ainda não pode se aposentar

# Exercício 5 - Operador lógico - .ou. (Dicas)



Solução Diagrama de Blocos

Software utilizado: Lucidchart  
[www.lucidchart.com](http://www.lucidchart.com)

# Exercício 5 - Operador lógico - .ou. (Dicas)

---

## Solução: Português Estruturado

Programa VERIFICA\_APOSENTADORIA

var

*IDADE*: inteiro

*TEMPO\_CONTR*: inteiro

inicio

leia *IDADE*, *TEMPO\_CONTR*

se (*IDADE* >= 65) .ou. (*TEMPO\_CONTR* >= 35) então

    escreva "O usuário já pode se aposentar."

senão

    escreva "O usuário não pode ainda se aposentar."

fim\_se

fim

# Desafio 1

---

- Vamos exercitar?

Desenvolva o algoritmo, diagrama de blocos e o pseudocódigo para o seguinte problema:

Ler três valores para os lados de um triângulo, considerando lados como: A, B e C. Verificar se os lados fornecidos formam realmente um triângulo, e se for esta condição verdadeira, deverá ser indicado qual tipo de triângulo foi formado: isósceles, escaleno ou equilátero. Veja o algoritmo, diagrama de blocos e a codificação em português estruturado, prestando atenção na utilização dos operadores lógicos.



# Desafio 1 (Dicas)

---

- Vamos exercitar?

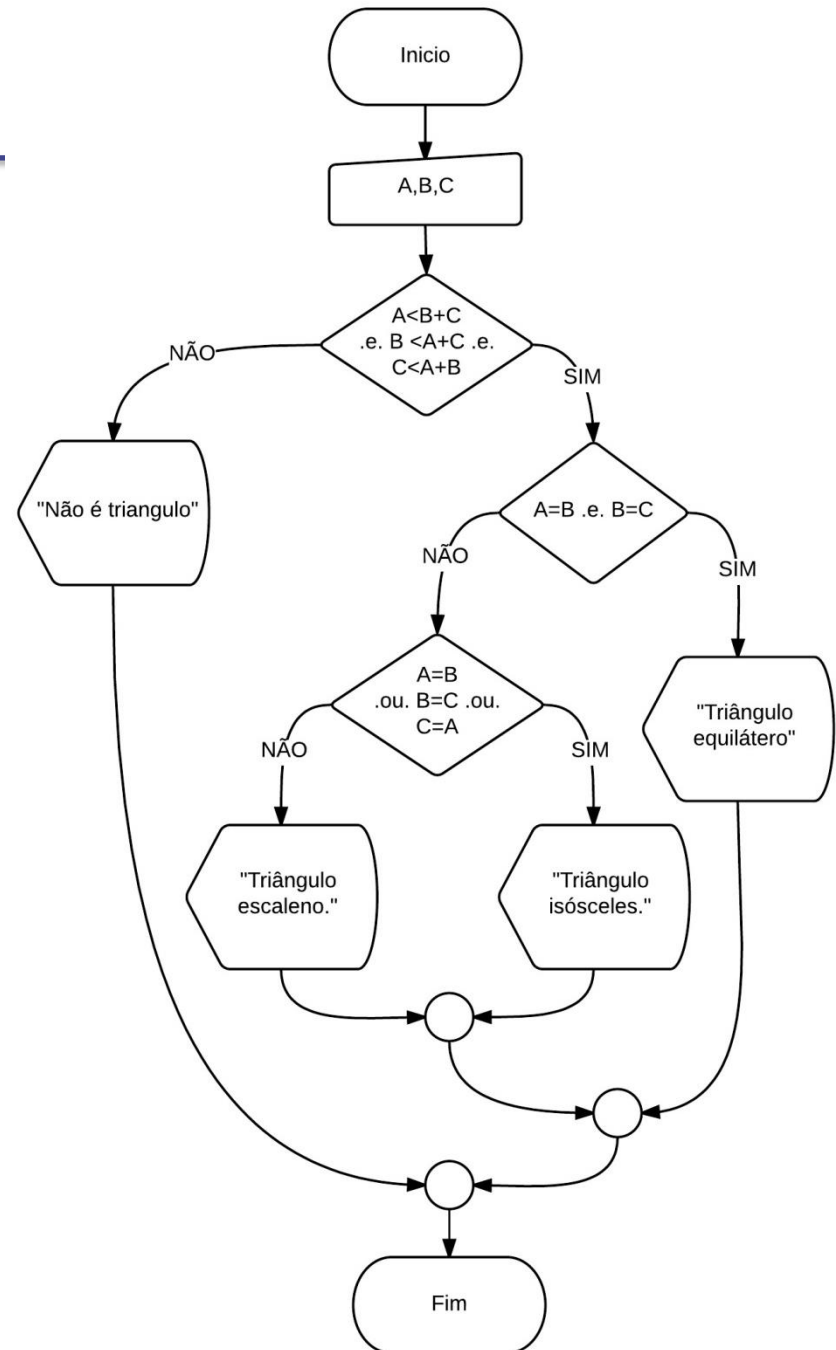
Algoritmo da solução:

- 1 - Ler três valores para os lados de um triângulo: A, B e C;
- 2 - Verificar se cada lado é menor que a soma dos outros dois lados. Se sim, saber se  $A=B$  e se  $B=C$ , sendo verdade o triângulo é equilátero.  
Se não, verificar  $A=B$  ou se  $A=C$  ou se  $B=C$ , sendo verdade o triângulo é isósceles; caso contrário, o triângulo é escaleno;
- 3 - Caso os lados fornecidos não caracterizem um triângulo, avisar a ocorrência.

# Desafio 1 (Dicas)

- Vamos exercitar?

Diagrama de blocos:



# Desafio 1 (Dicas)

---

- Vamos exercitar?

## Pseudocódigo

```
programa TRIÂNGULO
var
    A,B,C: real
início
    leia A,B,C
    se (A<B+C) .e. (B<A+C) .e. (C<B+A) então
        se (A=B) .e. (B=C) então
            escreva "Triângulo equilátero"
        senão
            se (A=B) .ou. (B=C) .ou. (A=C) então
                escreva "Triângulo isósceles"
            senão
                escreva "Triângulo escaleno"
            fim_se
        fim_se
    senão
        escreva "Não é um triângulo"
    fim_se
fim
```

# **ESTRUTURAS DE CONTROLE**

LAÇOS OU MALHAS DE REPETIÇÃO

# Repetição: Teste lógico no início do looping

---

- Vamos exercitar?

“Construir um programa que calcule o fatorial de um número digitado pelo usuário”.

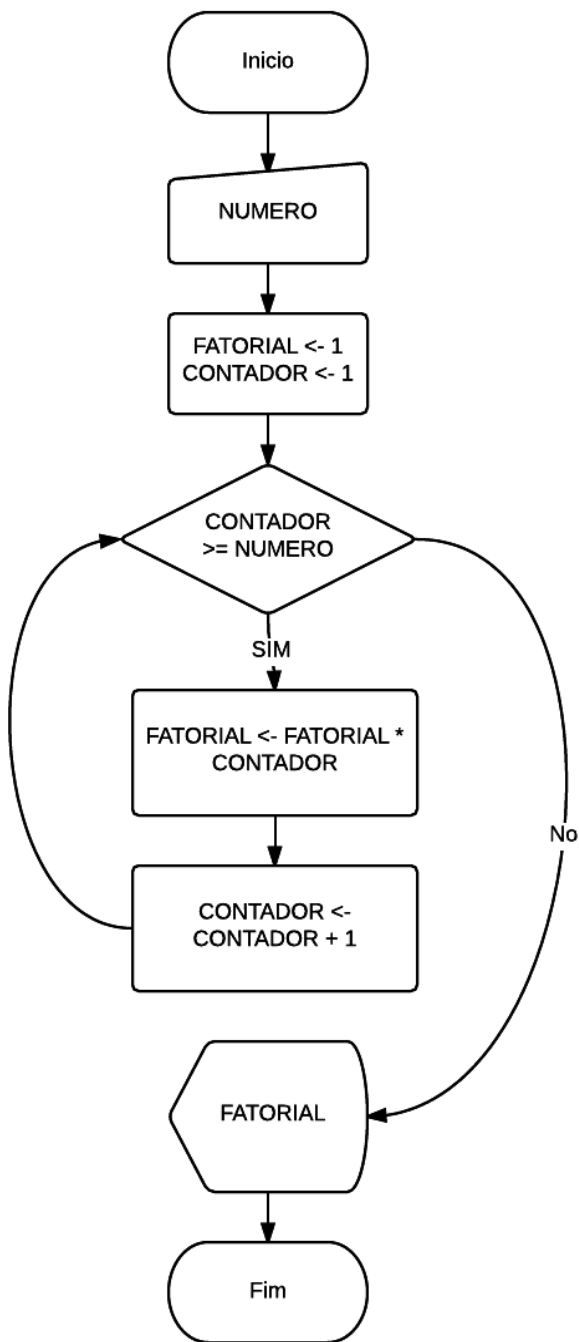
Lembrando que Fatorial de um número é igual a  $N! = N * N-1 * N-2 * ... * 1$

Exemplo: Fatorial de 5

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

Ou podemos escrever também

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$



programa FATORIAL\_DE\_N

var

CONTADOR, FATORIAL, NUMERO: inteiro

inicio

FATORIAL <- 1

CONTADOR <- 1

escreva "Digite o número para calcular o seu fatorial"

leia NUMERO

enquanto (CONTADOR <= NUMERO) faça

FATORIAL <- FATORIAL \* CONTADOR

CONTADOR <- CONTADOR + 1

fim\_enquanto

escreva "O fatorial de", NUMERO, "é: ", FATORIAL

fim

# Repetição: Teste lógico no fim do looping

---

- Vamos exercitar?
- 1) Elabore um programa que apresente os quadrados dos números inteiros de 15 a 200
  - 2) Apresentar os resultados de uma tabuada de um número qualquer, a qual deve ser impressa no seguinte formato: Considerando como exemplo o fornecimento do número 2 para o número qualquer, ter-se-ia a seguinte situação:  
$$2 \times 1 = 2$$
$$2 \times 2 = 4$$
$$2 \times 3 = 6$$
$$2 \times 4 = 8$$
$$2 \times 5 = 10$$
$$(\dots)$$
$$2 \times 10 = 20$$

# Repetição: Variável de controle (FOR)

---

- Vamos exercitar?
- 1) Elaborar um programa que apresente no final o somatório dos valores pares existentes na faixa de 1 até 500.
  - 2) Apresentar todos os números divisíveis por 4 que sejam menores que 200. Para verificar se o número é divisível por 4, efetuar dentro da malha a verificação lógica desta condição com a instrução “se”, perguntando se o número é divisível; sendo, mostre-o, não sendo, passe para o próximo passo. A variável que controlará o contador deverá ser iniciada com valor 1 .