

# INE5607 – Organização e Arquitetura de Computadores

Linguagem de Montagem e de Máquina

**Aula 11: Instruções de tomada de decisão: while, for, repeat**

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



# Sumário

- Desvios
- While
- For
- Repeat
- Exercícios
- Considerações finais

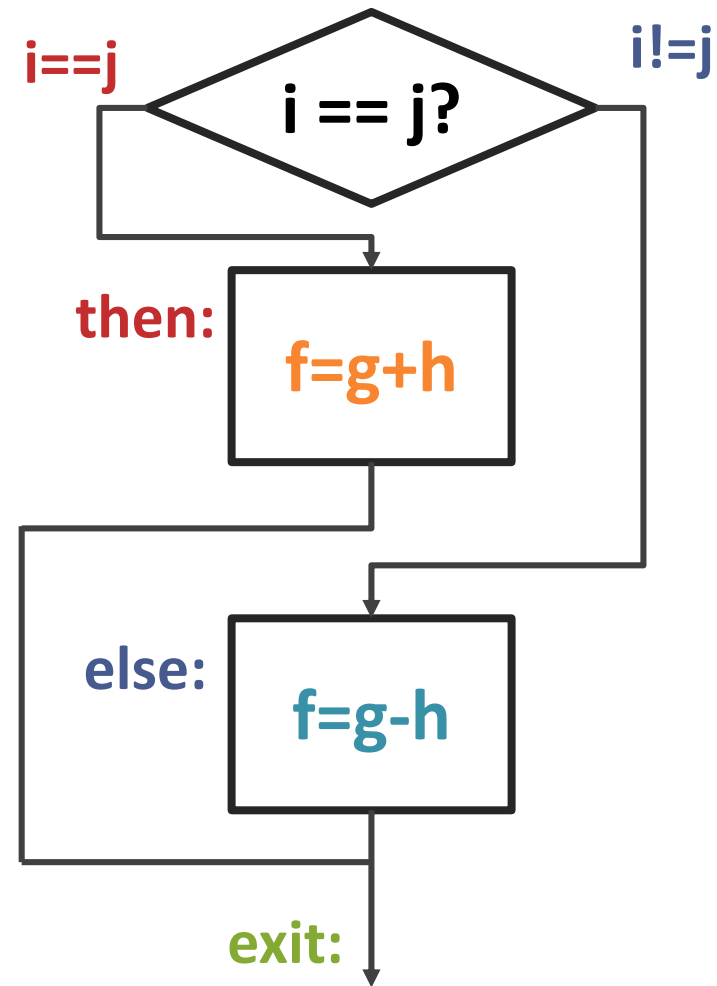
# DESVIOS

# Desvios

- Instruções de desvio incondicional
  - **j, jal, jr**
- Instruções de desvio condicional
  - **beq, bne**
  - Suporte: **slt, slti**
- **if-then-else**
  - Visão de programa monolítico
  - Switch como encadeamento de if-then-else

# Desvios

```
bne $s3, $s4, else
add $s0, $s1, $s2
j exit
else: sub $s0, $s1, $s2
exit: ...
```



# Desvios

- Computador != calculadora
  - Máquina: dá significado a operações e **testes**
  - Capacidade de tomar decisões
- Linguagens de programação
  - if-then-else, while, for, repeat until, switch
- Suporte na arquitetura do conjunto de instruções
  - **Desvio condicional (*branch*)**
  - **Desvio incondicional (*jump*)**

# Desvios

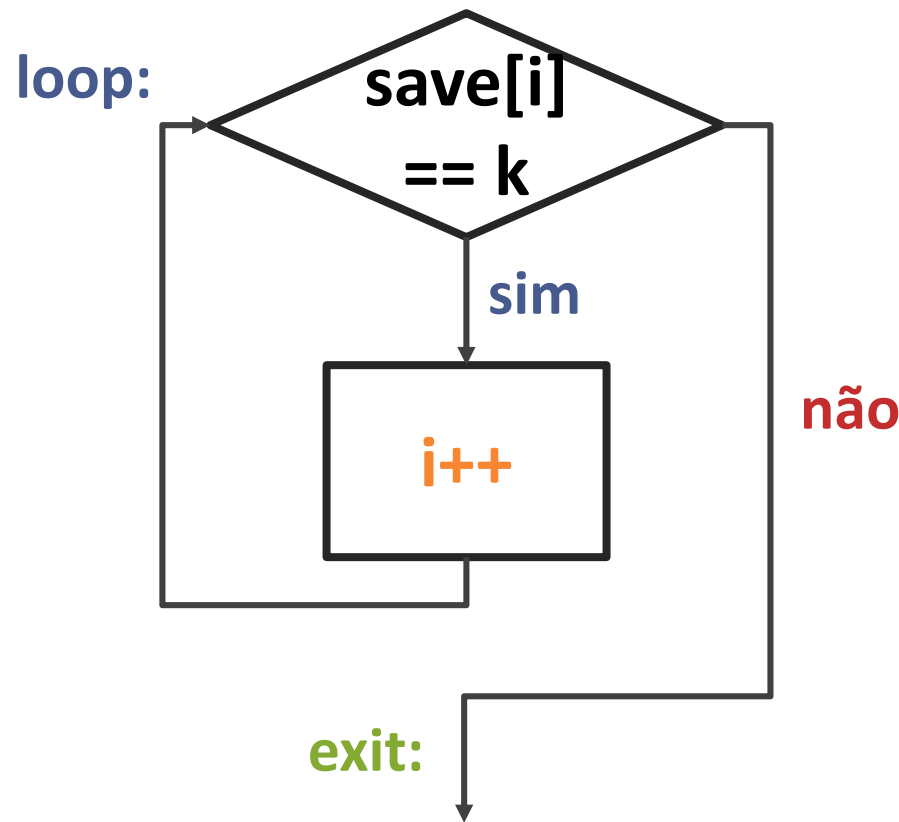
- Qual a diferença entre **while**, **for** e **repeat**?
  - **while** e **for**: teste no início
  - **repeat**: teste no fim
  - **for** define inicialização e iteração
  - **while** e **repeat** não definem

# WHILE



# While

- `while (save[i] == k) i++;`
- Programa monolítico em fluxograma



# While

- while (**save[i] == k**) **i++**;
  - Ler **save[i]** da memória
    - Necessário calcular o endereço primeiro
  - Comparar com **k**
    - Sair do laço caso seja diferente
  - Senão fazer **i++** e voltar no loop

# While

- while (save[i] == k) i++;  
– i = \$s3, k = \$s5, endereço base de save = \$s6

```
Loop:  sll $t1, $s3, 2           # $t1 = 4*i
        add $t1, $t1, $s6       # $t1 = *save[i]
        lw $t0, 0($t1)         # $t0 = save[i]
        bne $t0, $s5, Exit     # vá p/ Exit se
                                # save[i] ≠ k

        addi $s3, $s3, 1       # i = i+1
        j Loop
```

Exit: ...

# While

- while ( teste ) operação

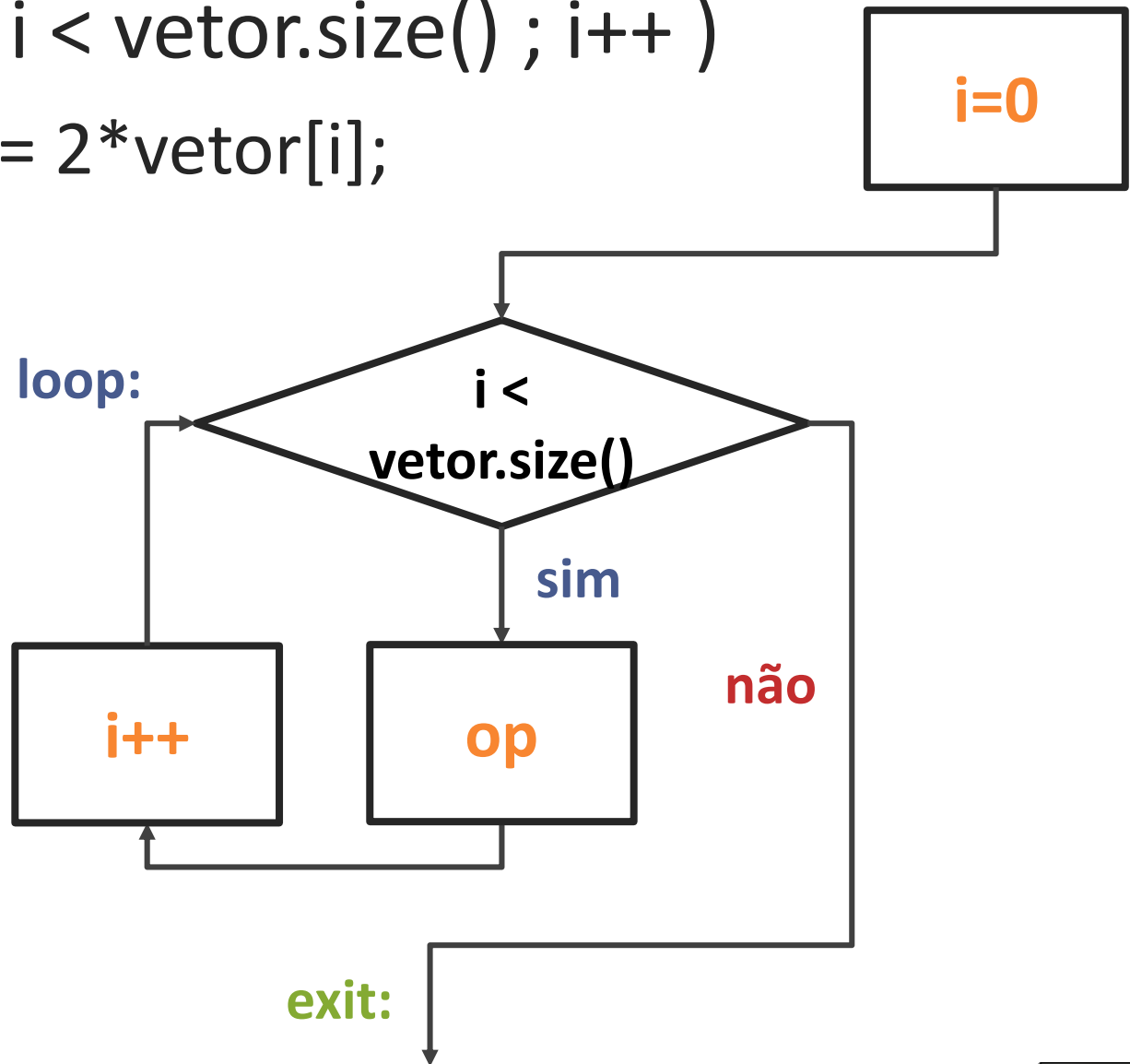
**loop:**     preparação teste  
             teste com salto condicional para **saída**  
             operação  
             salto incondicional para o **loop**

**saída:** ...

# FOR

# For

- `for ( i=0 ; i < vetor.size() ; i++ )`  
– `vetor[i] = 2*vetor[i];`



# For

- for ( início ; teste ; itera ) operação

início

loop:    preparação teste  
         teste com salto condicional para saída  
         operação  
         itera  
         salto incondicional para loop

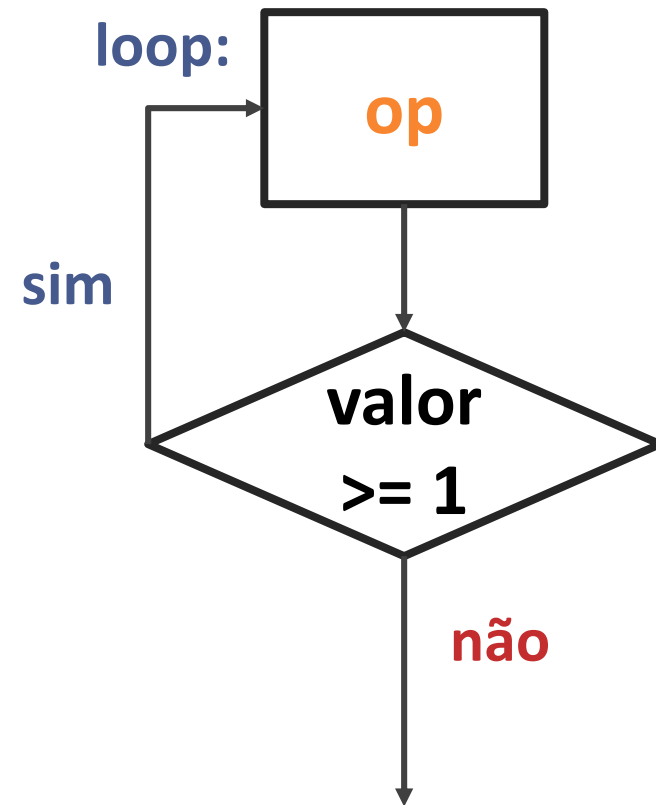
saída: ...

# REPEAT



# Repeat

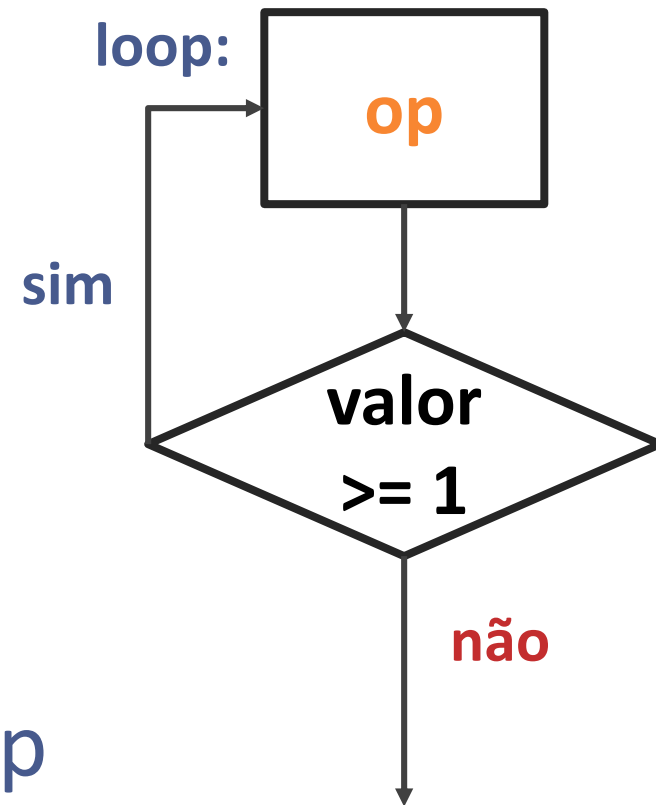
- repeat...until == do...while
- do {valor = valor/2; i++}
- while (valor >= 1)



# Repeat

- do {valor = valor/2; i++}
- while (valor >= 1)
- \$s0 = valor, \$s5 = i

```
loop:      srl $s0, $s0, 1  
           addi $s5, $s5, 1  
           slti $t0, $s0, 1  
           beq $t0, $zero, loop
```



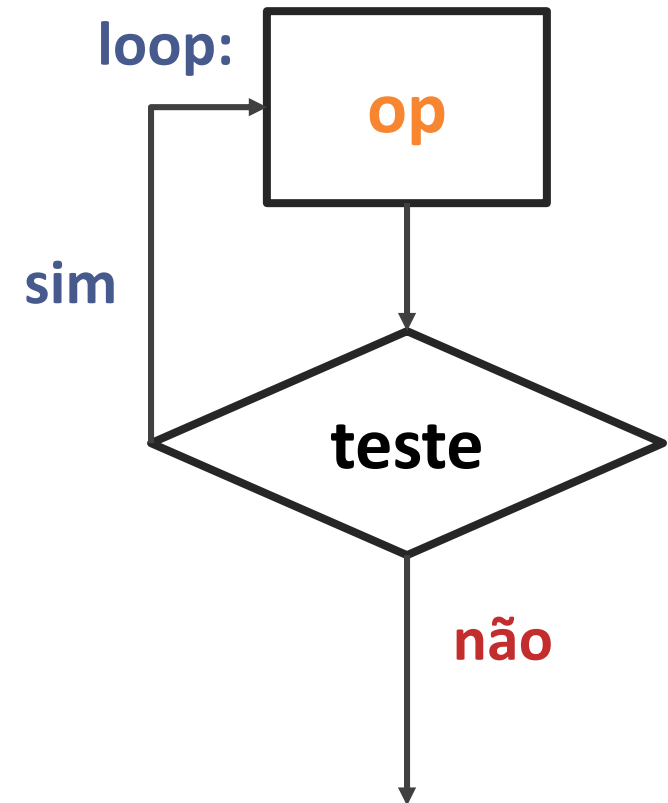
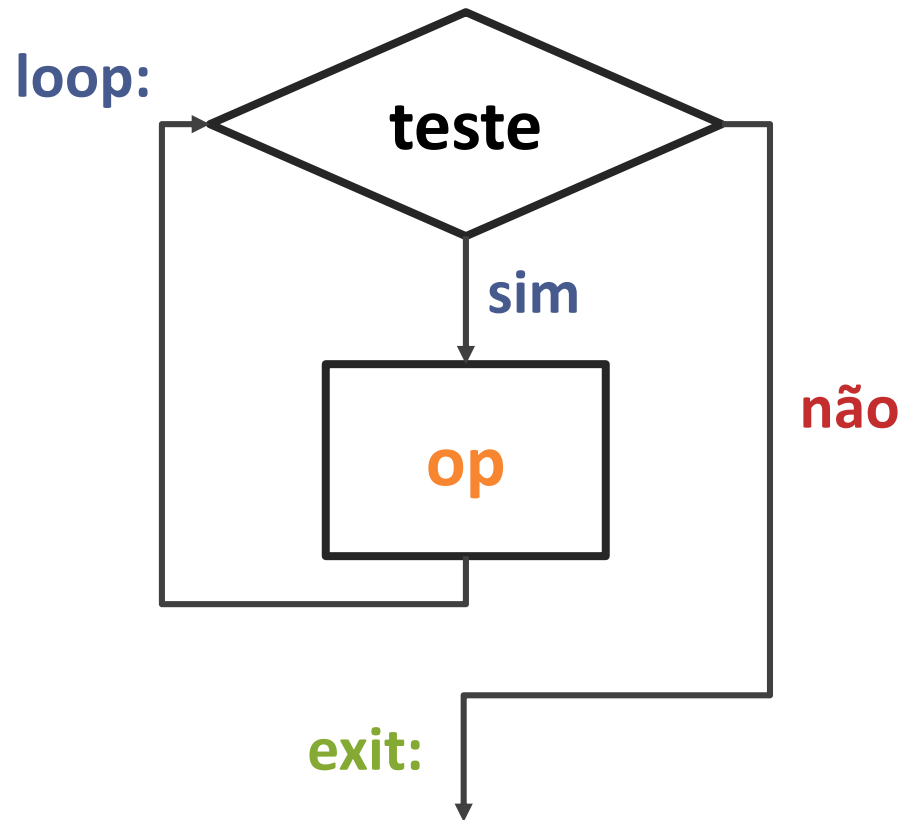
# Repeat

- do operação while ( teste )

loop: operação  
preparação teste  
teste com salto condicional para loop

# Repeat

- while vs repeat



# EXERCÍCIOS

# Exercícios

- Faça o papel do compilador para o seguinte código
  - `for ( i = 0 ; i < y ; i ++ ) x *= z;`
    - Considere que existe uma pseudoinstrução `mult` que é composta de várias instruções mais simples
      - `mult $t0, $t1, $t2 -> $t0 = $t1 * $t2`
  - Otimize o código anterior sabendo que  $y = 5$  e  $z = 2$

# Exercícios

- Que construção de iteração é usada abaixo e o que faz o código?

```
move $t0, $zero
```

```
loop: sll $t1, $t0, 2
```

```
add $t2, $a0, $t1
```

```
sw $zero, 0($t2)
```

```
addi $t0, $t0, 1
```

```
slt $t3, $t0, $a1
```

```
bne $t3, $zero, loop
```

\$a0 = endereço vetor

\$a1 = tamanho de vetor

# CONSIDERAÇÕES FINAIS



# Considerações finais

- Operações
- Instruções de desvio incondicional
  - j, jal, jr
- Instruções de desvio condicional
  - beq, bne
  - Suporte: slt, slti
- if-then-else, switch
- while, for, repeat

# Considerações finais

- Próxima aula
  - Suporte à chamada de procedimentos

# INE5607 – Organização e Arquitetura de Computadores

Linguagem de Montagem e de Máquina

**Aula 11: Instruções de tomada de decisão: while, for, repeat**

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

