



# BACK-END

**Desenvolvimento de Software para Internet**

# ESTRUTURAS DE CONTROLE



# Variáveis em PHP: quais são os tipos mais comuns e como converter?





# Tipos de variáveis: básicas

- **Inteiros** (**int**): positivos ou negativos
- **Reais** (**float**): números reais com casas decimais (usa-se ponto “.”)
- **Cadeiras de caracteres** (**string**)
- **Booleanos** (**bool**): verdadeiros ou falsos





# Conversão de tipos

- PHP é uma linguagem **fracamente tipada**
- São identificados em tempo de execução
- Conversão explícita de tipos:
  - (int) - converte para int
  - (bool) - converte para bool
  - (float) - converte para float
  - (string) - converte para string

**gettype() : obtém o tipo da variável em PHP**

```
<?php
$idade = '27';
$idade_inteiro = (int) $idade;
echo gettype($idade);
echo gettype($idade_inteiro);
?>
```





# Operadores em PHP: para que servem e quais são as regras?





# Exemplos: operadores aritméticos

```
<?php
```

```
$a = 5;
```

```
$b = 10;
```

```
$c = 2;
```

```
$d = 3;
```

```
$s = $a + $b / $c;
```

```
$s = ($a + $b) / $c;
```

```
$s = $a % $c;
```

```
$s = $c * 3;
```

```
?>
```



\$s?

10

7.5

1

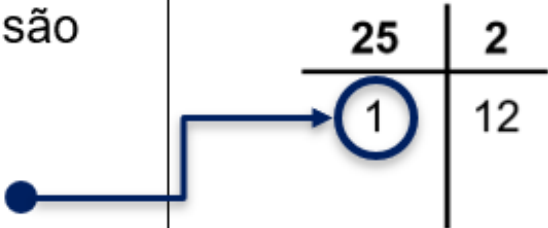
6

Crédito da ilustração: Freepik





# Operadores aritméticos

Operador	Descrição	Na matemática
+	soma	+
-	subtração	-
*	multiplicação	X ou .
/	divisão	÷ ou / ou $\frac{x}{y}$
%	resto da divisão inteira	
**	potenciação	$x^y$

Crédito: Slides do Prof. Leonardo Nardi







# Exemplos: operadores relacionais

```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
$s = ($a != $c);
$s = ($b <= $d);
$s = ($a > $d);
$s = ($c == "Maria");
?>
```



\$s?

True

False

True

False

Crédito da ilustração: Freepik





# Operadores relacionais

Operador	Descrição	Na matemática
<code>==</code>	igualdade(comparação)	$=$
<code>&gt;</code>	maior que	$>$
<code>&gt;=</code>	maior ou igual	$\geq$
<code>&lt;</code>	menor que	$<$
<code>&lt;=</code>	menor ou igual	$\leq$
<code>!=</code> ou <code>&lt;&gt;</code>	diferente	$\neq$

Crédito: Slides do Prof. Leonardo Nardi





# Exemplos: operadores de atribuição

```
<?php  
$a = 5;  
$b = 10;  
$c = 2;  
$d = 3;  
$a += 50;  
$b *= $c;  
$a %= $c;  
$d .= '-olá';  
?>
```



Crédito da ilustração: Freepik





# Operadores de atribuição

Operadores	Significado
=	Atribuição simples
+=	Atribuição com adição
-=	Atribuição com subtração
*=	Atribuição com multiplicação
/=	Atribuição com divisão
%=	Atribuição com resto da divisão inteira
.=	Atribuição com concatenação

Crédito: Slides do Prof. Leonardo Nardi



# Exemplos: operadores de incremento/decremento

```
<?php
```

```
$a = 5;
```

```
$b = 10;
```

```
$c = 2;
```

```
$d = 3;
```

```
echo -$a. "\n";
```

```
echo --$b + $c. "\n";
```

```
echo $d++ + $c. "\n";
```

```
echo ++$d + $c. "\n";
```

```
?>
```



\$a
5
\$c

\$b
\$d



Crédito da ilustração: Freepik

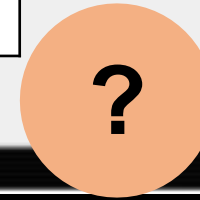


# Exemplos: operadores de incremento/decremento

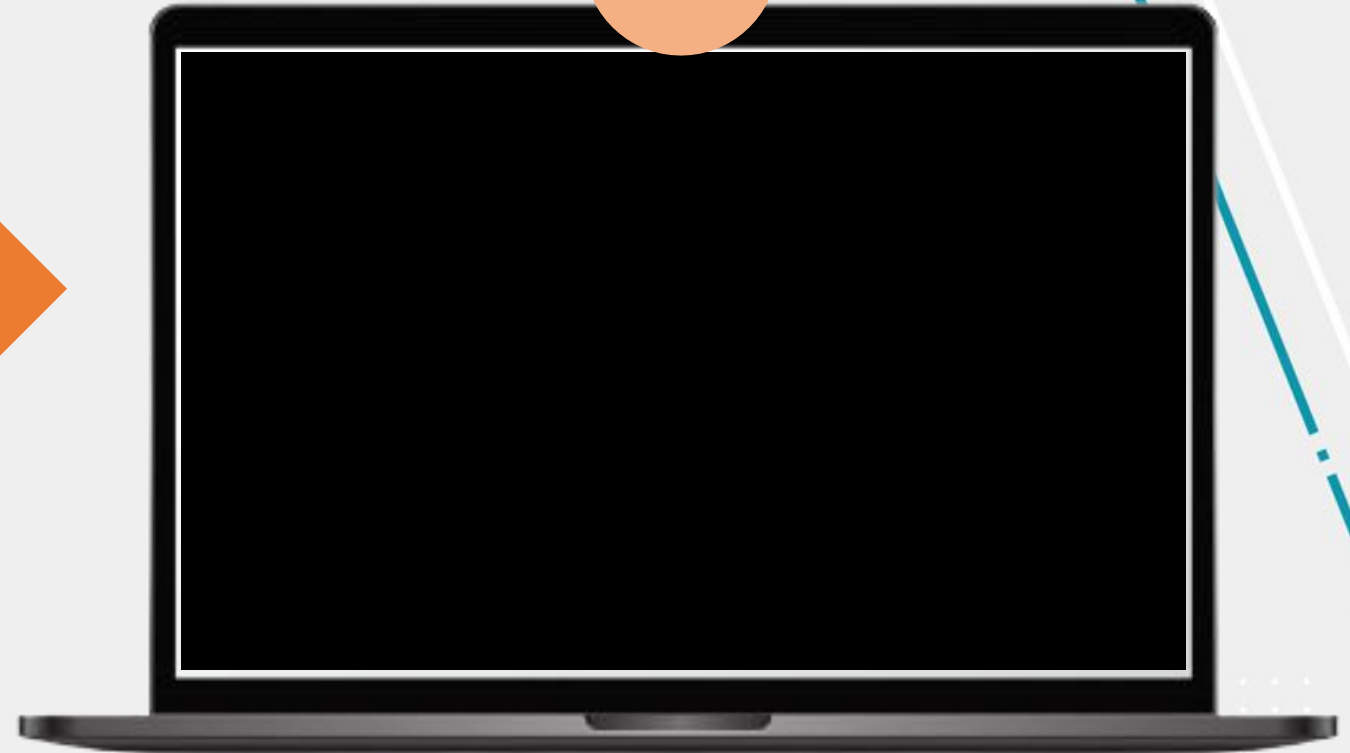


\$a
5
\$c

\$b
10
\$d



```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```



Crédito da ilustração: Freepik



# Exemplos: operadores de incremento/decremento



\$a
5

\$b
10

\$c
2

\$d



```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```



Crédito da ilustração: Freepik



# Exemplos: operadores de incremento/decremento

\$a
5

\$c
2

\$b
10

\$d
3



Crédito da ilustração: Freepik

```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```





# Exemplos: operadores de incremento/decremento

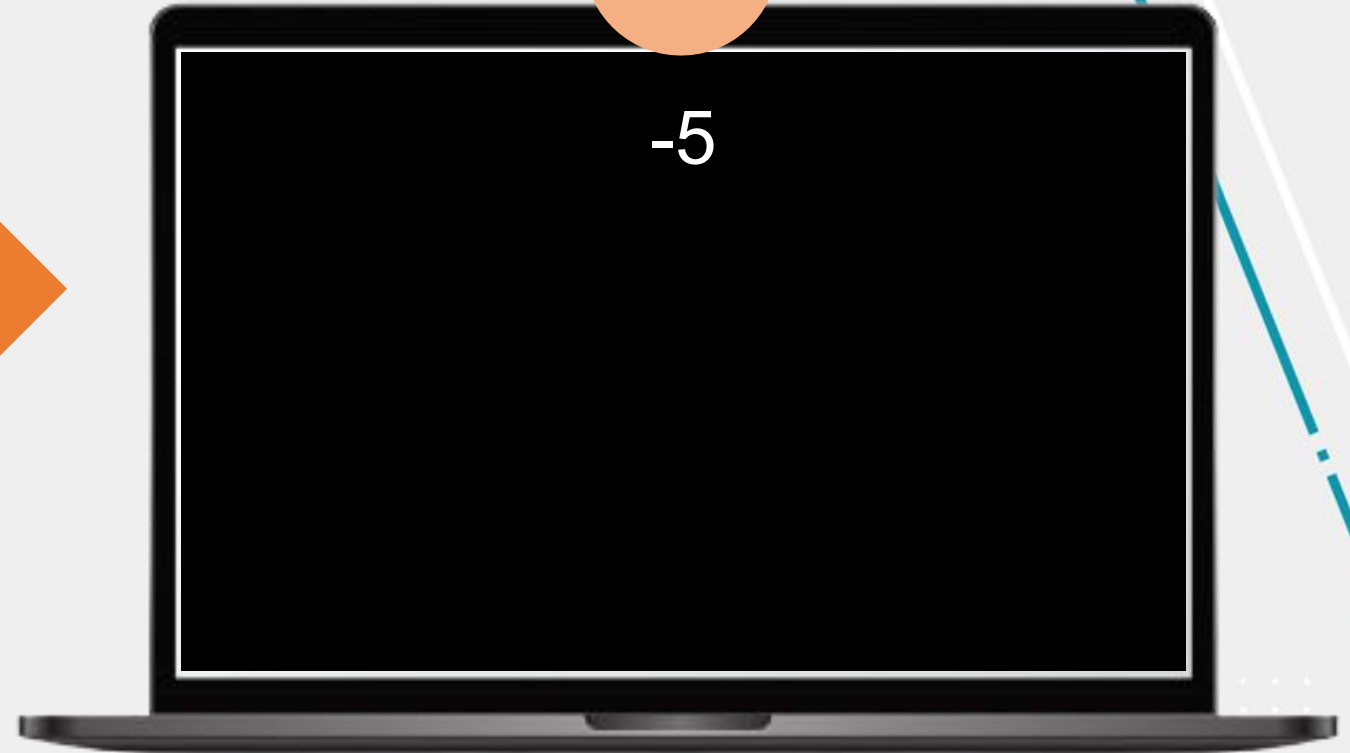


\$a
5

\$b
10

\$c
2

\$d
3



Crédito da ilustração: Freepik

```
<?php
```

```
$a = 5;
```

```
$b = 10;
```

```
$c = 2;
```

```
$d = 3;
```

```
echo -$a. "\n";
```

```
echo --$b + $c. "\n";
```

```
echo $d++ + $c. "\n";
```

```
echo ++$d + $c. "\n";
```

```
?>
```

# Exemplos: operadores de incremento/decremento

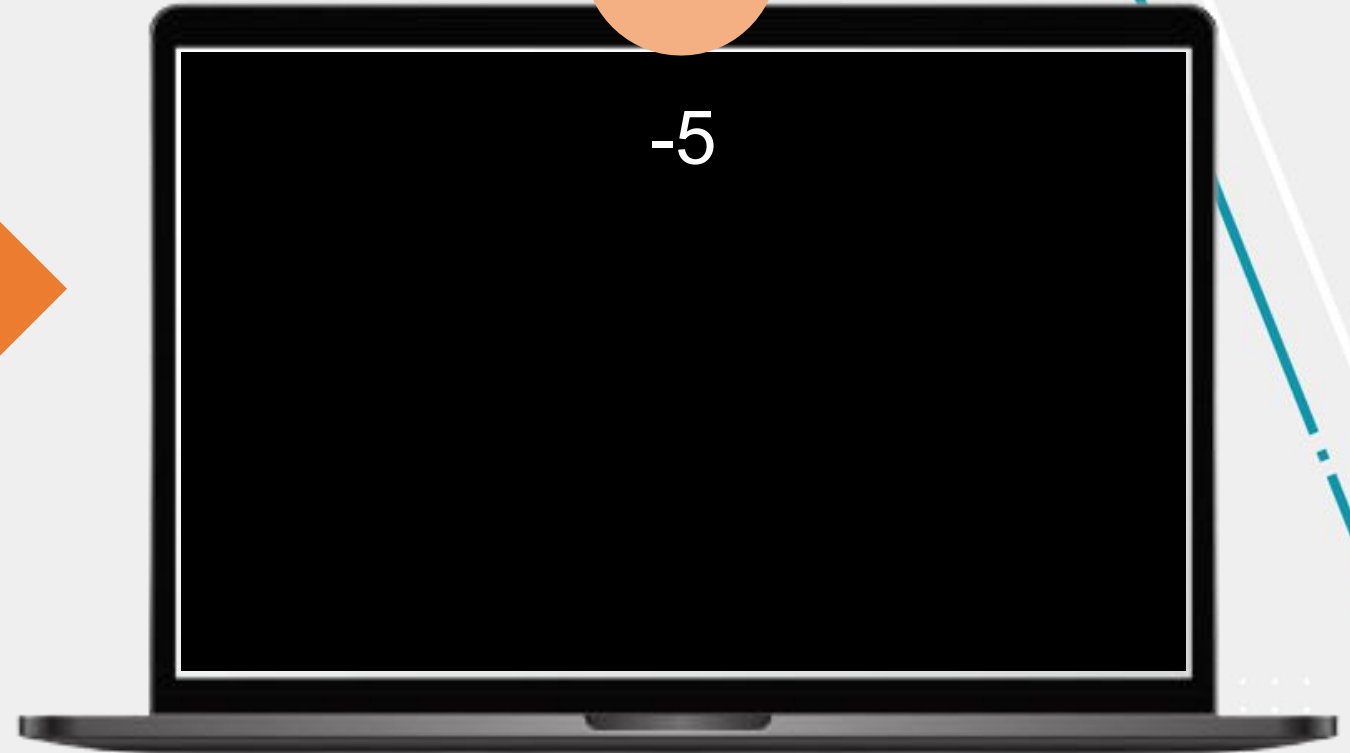


\$a
5

\$b
<del>10</del> 9

\$c
2

\$d
3



Crédito da ilustração: Freepik

```
<?php
```

```
$a = 5;
```

```
$b = 10;
```

```
$c = 2;
```

```
$d = 3;
```

```
echo -$a. "\n";
```

```
echo --$b + $c. "\n";
```

```
echo $d++ + $c. "\n";
```

```
echo ++$d + $c. "\n";
```

```
?>
```



# Exemplos: operadores de incremento/decremento

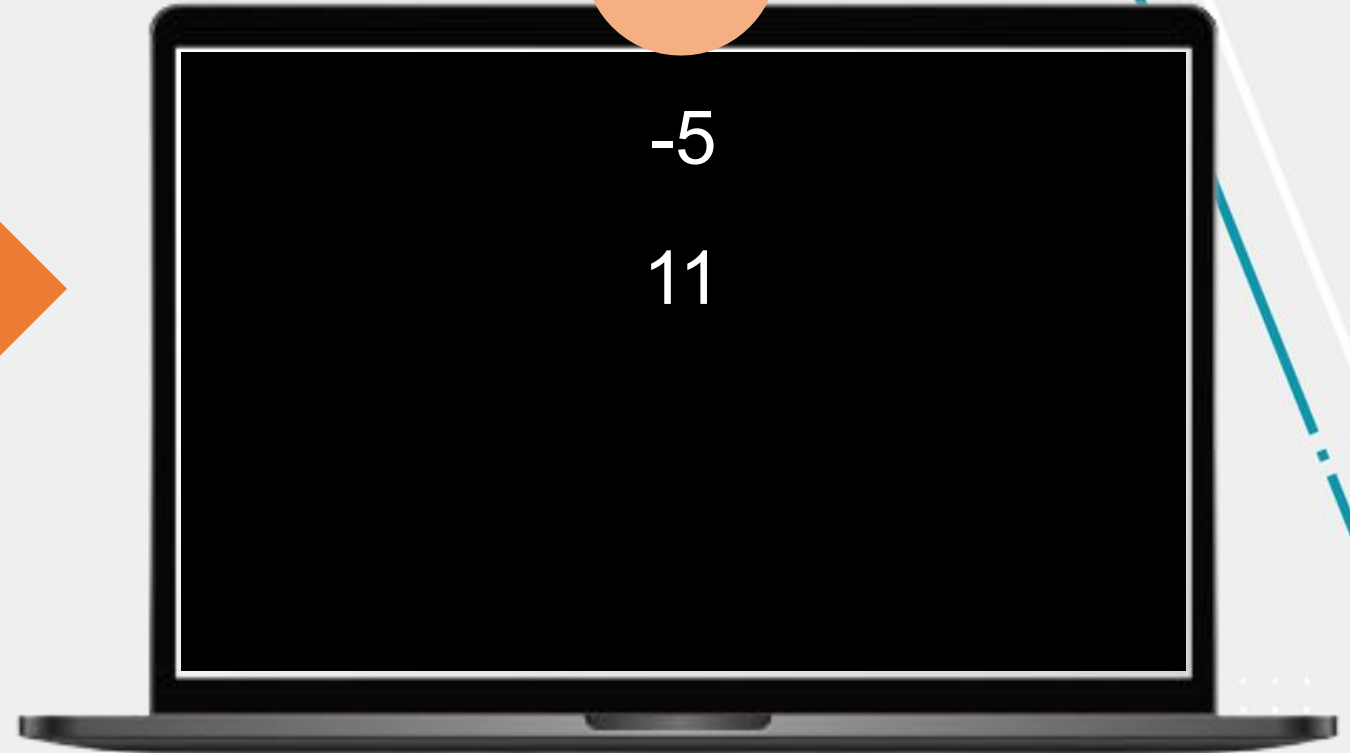
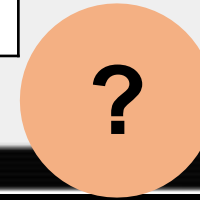


\$a
5

\$b
<del>10</del> 9

\$c
2

\$d
3



Crédito da ilustração: Freepik

```
<?php
```

```
$a = 5;
```

```
$b = 10;
```

```
$c = 2;
```

```
$d = 3;
```

```
echo -$a. "\n";
```

```
echo --$b + $c. "\n";
```

```
echo $d++ + $c. "\n";
```

```
echo ++$d + $c. "\n";
```

```
?>
```

# Exemplos: operadores de incremento/decremento

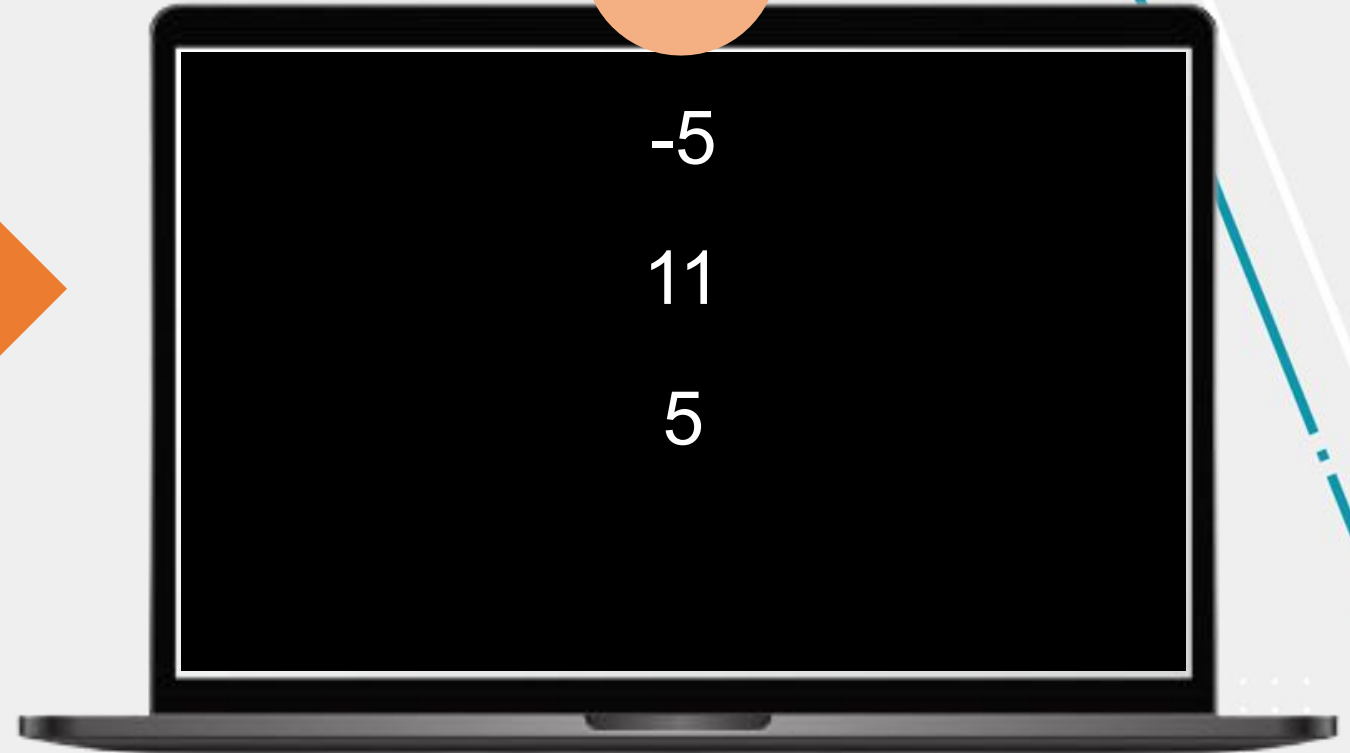
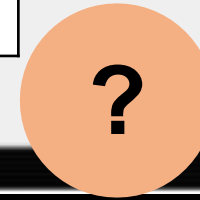


\$a
5

\$b
<del>10</del> 9

\$c
2

\$d
3



```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```

Crédito da ilustração: Freepik



# Exemplos: operadores de incremento/decremento

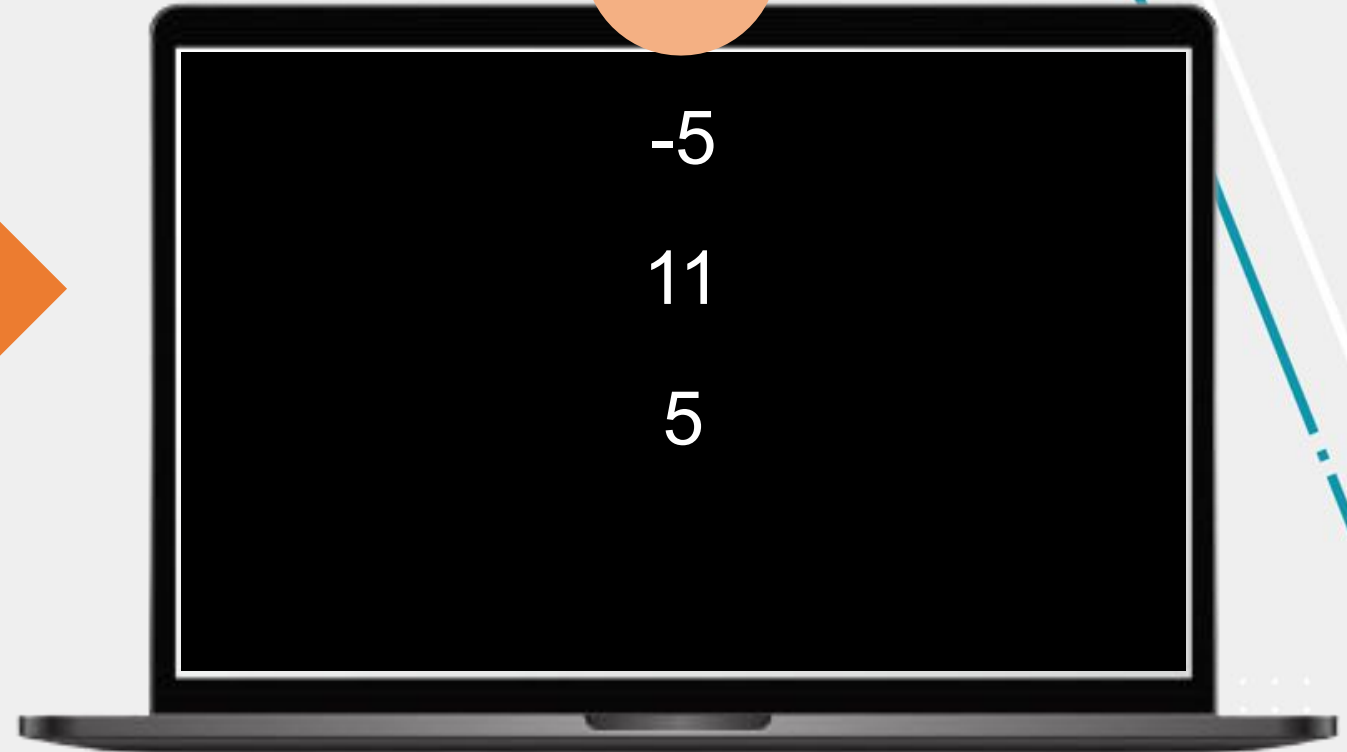


\$a
5

\$b
<del>10</del> 9

\$c
2

\$d
<del>3</del> 4



Crédito da ilustração: Freepik

```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```



# Exemplos: operadores de incremento/decremento

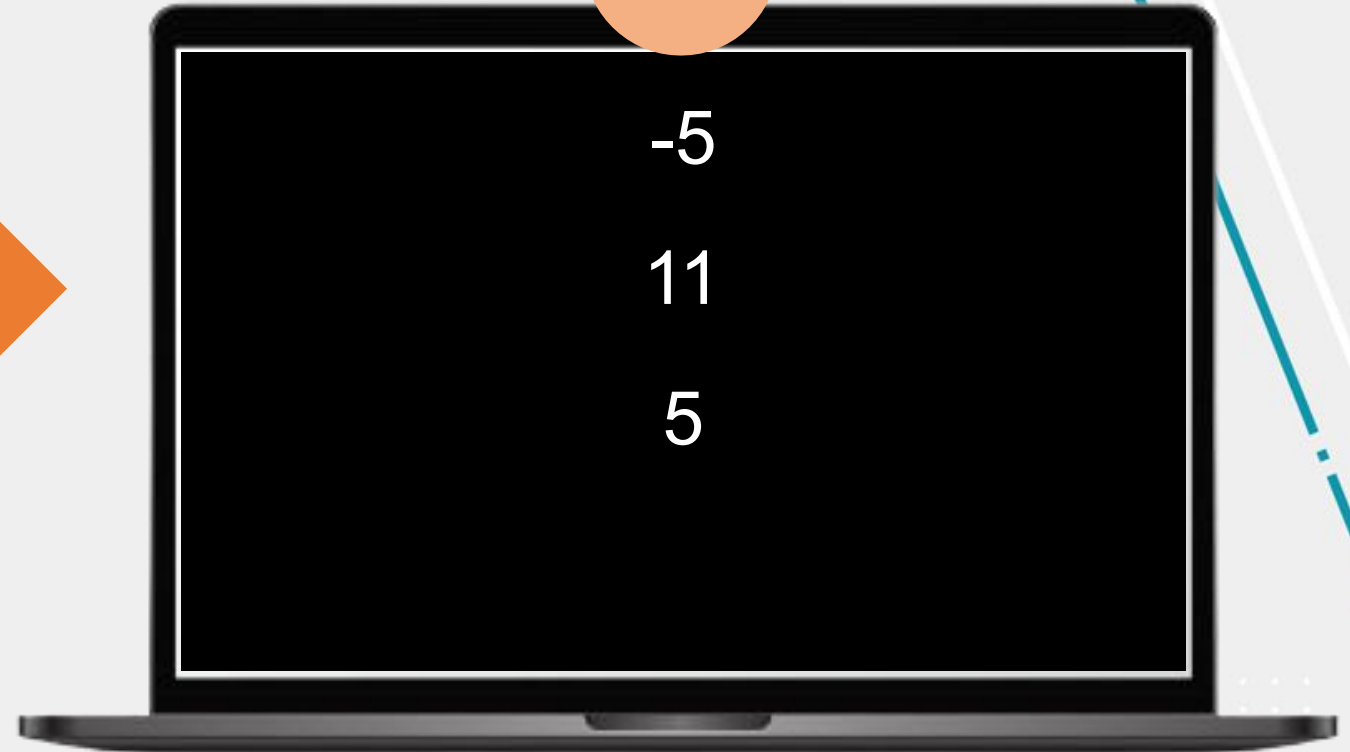


\$a
5

\$b
<del>10</del> 9

\$c
2

\$d
<del>3</del> <del>4</del> 5



Crédito da ilustração: Freepik

```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```



# Exemplos: operadores de incremento/decremento

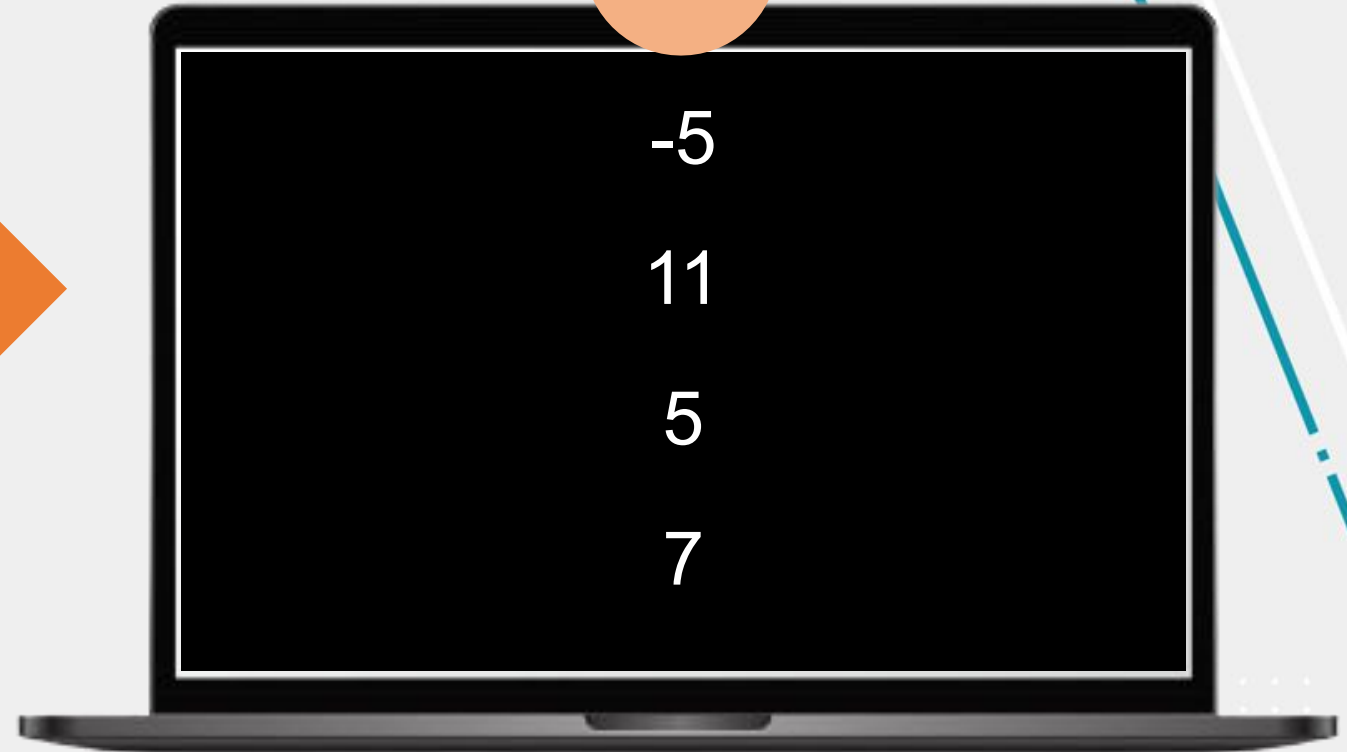


\$a
5

\$b
<del>10</del> 9

\$c
2

\$d
<del>3</del> <del>4</del> 5



Crédito da ilustração: Freepik

```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
echo -$a. "\n";
echo --$b + $c. "\n";
echo $d++ + $c. "\n";
echo ++$d + $c. "\n";
?>
```



# Operadores de decremento e incremento

Operadores	Significado
$-\$x$	Troca o sinal da variável $\$x$
$++\$x$	Pré-incremento: primeiro incrementa depois realiza a operação
$--\$x$	Pré-decremento: primeiro decrementa depois realiza a operação
$\$x++$	Pós-incremento: primeiro faz a operação depois incrementa
$\$x--$	Pós-decremento: primeiro faz a operação depois decrementa

Crédito: Slides do Prof. Leonardo Nardi







# Exemplos: operadores lógicos

```
<?php
$a = 5;
$b = 10;
$c = 2;
$d = 3;
$s = !($a == 1);
$s = (($a >= 1) and ($b < 10));
$s = (($c < $b) || ($b != 10));
$s = ((!($b > $a)) or (($c+$d) != $a));
?>
```



\$s?

True  
False  
True  
False

Crédito da ilustração: Freepik





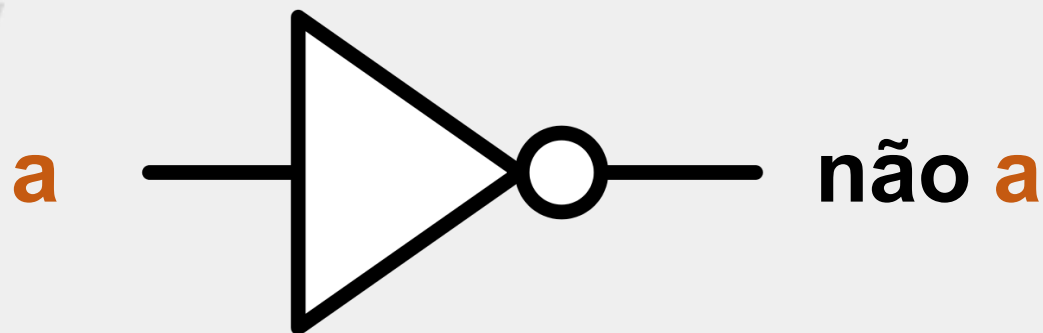
# Operadores lógicos

Operadores	Representação na lógica
!	$\neg$ (negação)
and ou &&	$\wedge$ (conjunção)
or ou	$\vee$ (disjunção)

Crédito: Slides do Prof. Leonardo Nardi



# Operador ! (NÃO)



**1** = ligado | **0** = desligado

**1** = **True** (verdadeiro) | **0** = **False** (falso)

a	not a
0	1
1	0

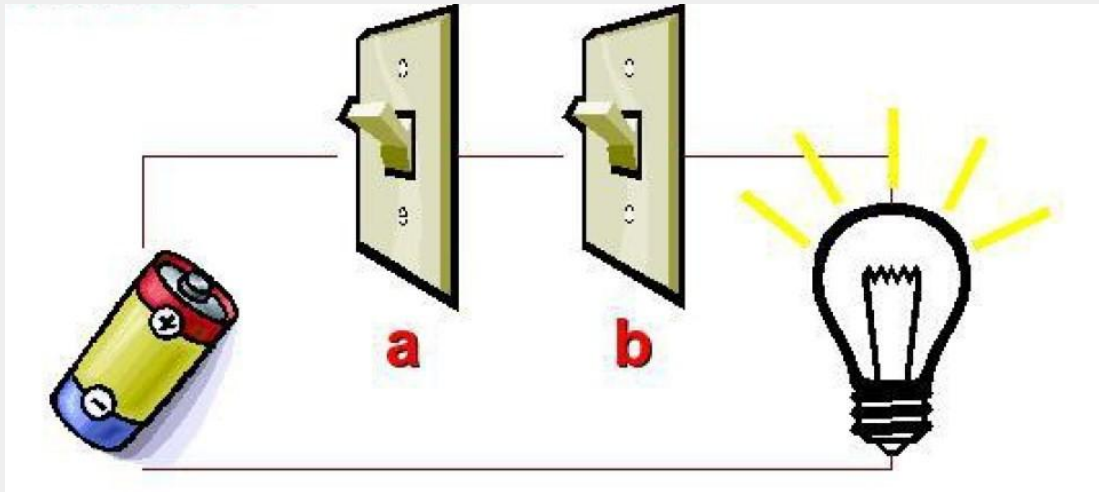
a	not a
<b>False</b>	<b>True</b>
<b>True</b>	<b>False</b>





# Operador AND ou && (E)

Analogia a lâmpadas



Crédito: Slides Profs. Adriano Cruz e Jonas Knopman

1 = ligado | 0 = desligado

1 = **True** (verdadeiro) | 0 = **False** (falso)

a	b	a and b
0	0	0
0	1	0
1	0	0
1	1	1

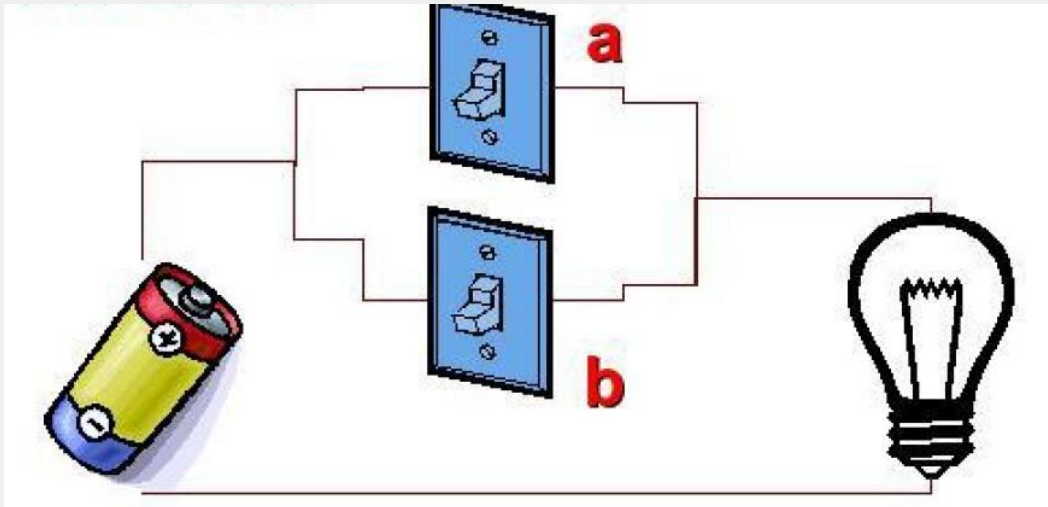
a	b	a and b
<b>False</b>	<b>False</b>	<b>False</b>
<b>False</b>	<b>True</b>	<b>False</b>
<b>True</b>	<b>False</b>	<b>False</b>
<b>True</b>	<b>True</b>	<b>True</b>





# Operador OR ou || (OU)

## Analogia a lâmpadas



Crédito: Slides Profs. Adriano Cruz e Jonas Knopman

1 = ligado | 0 = desligado

1 = **True** (verdadeiro) | 0 = **False** (falso)

a	b	a or b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	a or b
<b>False</b>	<b>False</b>	<b>False</b>
<b>False</b>	<b>True</b>	<b>True</b>
<b>True</b>	<b>False</b>	<b>True</b>
<b>True</b>	<b>True</b>	<b>True</b>





E por que os operadores relacionais  
e lógicos são tão importantes?

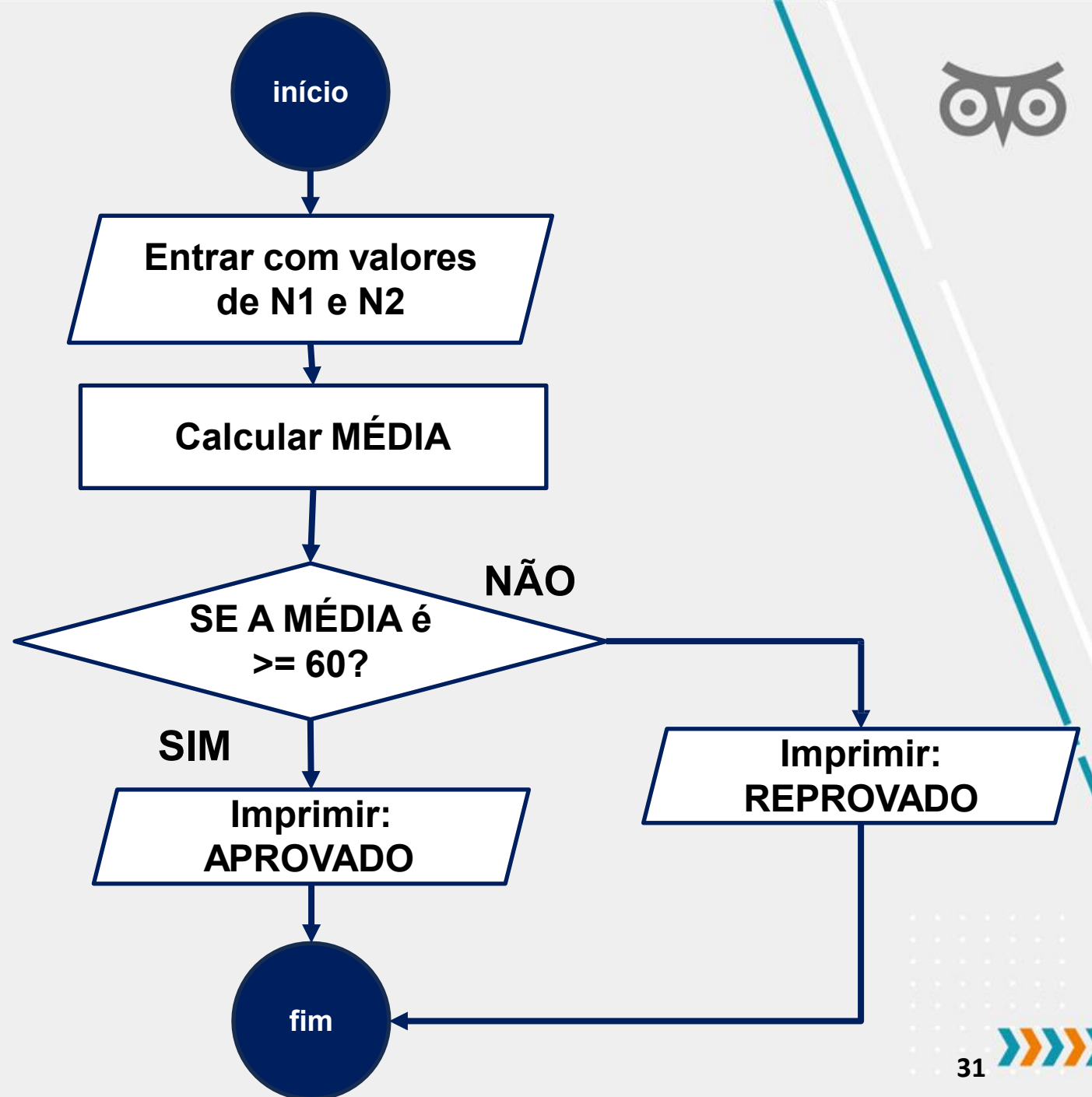




# Estruturas condicionais

Verificam condições por meio dos operadores relacionais e lógicos:

**se** atender a uma condição segue um caminho, caso contrário (**senão**) segue outro





## Exemplo: condicional



03

```
<?php
$n1 = 70.8;
$n2 = 45;
$media = ($n1 + $n2)/2;
if ($media >= 60)
    echo 'Aprovado!';
else
    echo 'Reprovado!';
?>
```

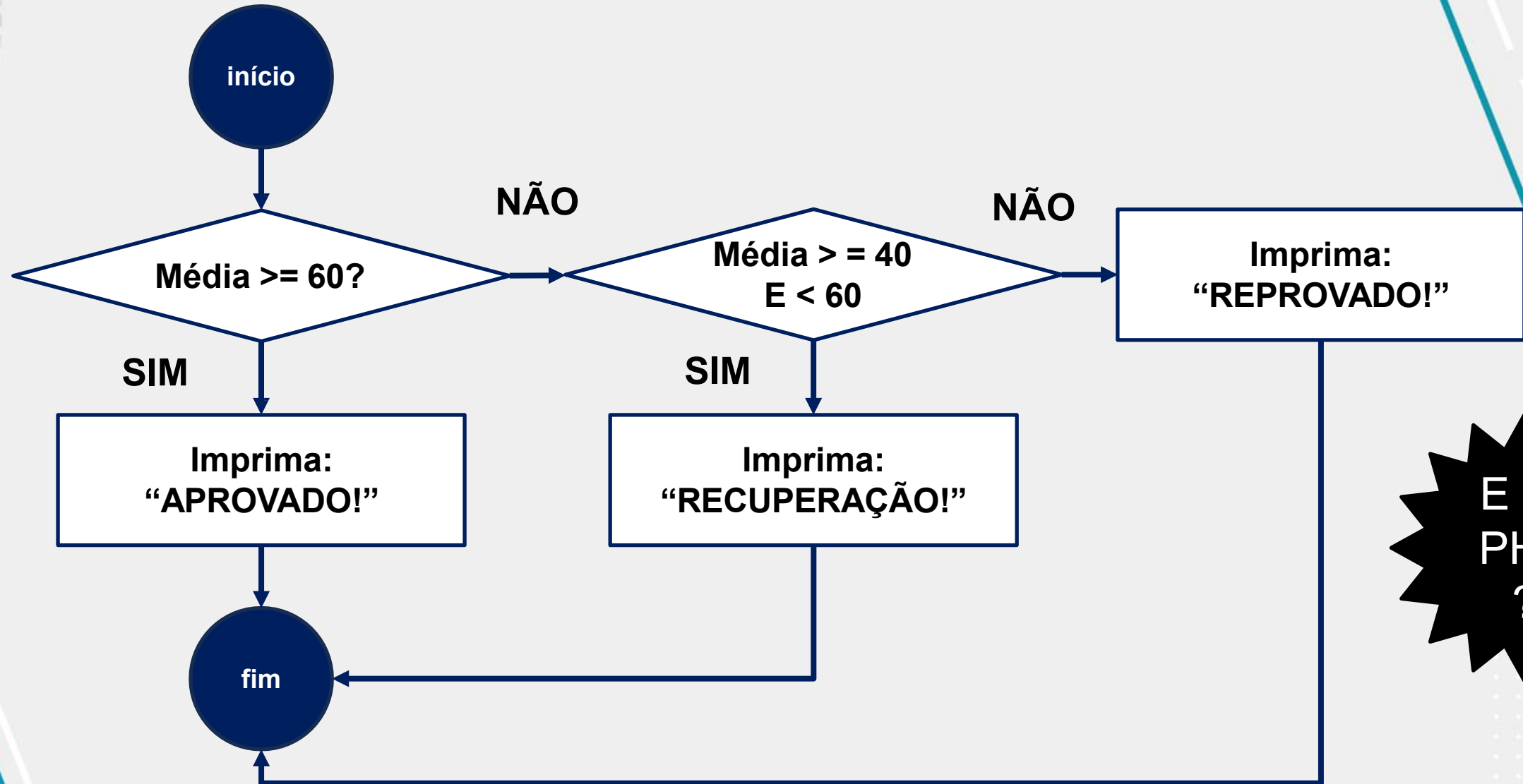




Há situações onde é necessário fazer  
mais testes...



# Estruturas condicionais: mais condições



if ...  
else



04

```
<?php
$n1 = readline("Digite a nota 1: ");
$n2 = readline("Digite a nota 2: ");
$media = ($n1 + $n2)/2;
echo "\nA média é: ". $media. ". Você está: ";
if ($media >= 60)
    echo "Aprovado!";
elseif ($media >= 40 and $media < 60)
    echo "Em recuperação";
else
    echo "Reprovado!";
?>
```



elseif



05

```
1  <?php
2  $n1 = readline("Digite a nota 1: ");
3  $n2 = readline("Digite a nota 2: ");
4  $media = ($n1 + $n2)/2;
5  echo "\nA média é: ". $media. ". Você está: ";
6  if ($media >= 60)
7  |   echo "Aprovado!";
8  elseif ($media >= 40 and $media < 60)
9  |   echo "Em recuperação";
10 else
11 |   echo "Reprovado!";
12 ?>
```





# Mas e se forem muitas as condições?

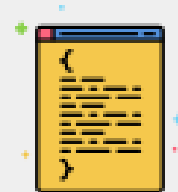
Crie um algoritmo que, dado duas notas N1 e N2, calcule a média e informe ao usuário:

- **“Aprovado com louvor”**, se a média for maior ou igual a 90
- **“Aprovado com boa nota”**, se a média for maior ou igual a 80 e menor que 90
- **“Aprovado na média”**, se a média for maior ou igual a 60 e menor que 80
- **“Em recuperação”**, se a média for maior ou igual a 40 e menor que 60
- **“Reprovado”**, se a média for menor que 40 e maior que 0
- **“Reprovado com Zero. Faltou?”**, se menor ou igual a zero



if ...  
else

```
1  <?php
2  $n1 = readline("Digite a nota 1: ");
3  $n2 = readline("Digite a nota 2: ");
4  $media = ($n1 + $n2)/2;
5  echo "\nA média é: ". $media. ". Você está: ";
6  if ($media >= 90)
7      echo "\nAprovado com louvor";
8  else {
9      if ($media >= 70 && $media < 90)
10         echo "\nAprovado com boa nota";
11     else {
12         if ($media >= 60 && $media < 80)
13             echo "\nAprovado na média";
14         else {
15             # code...
16             if ($media >= 40 && $media < 60)
17                 echo "\nEm recuperação";
18             else {
19                 if ($media > 0)
20                     echo "\nReprovado";
21                 else
22                     echo "\nReprovado com Zero. Faltou?";
23             }
24         }
25     }
26 }
27 ?>
```



06







## elseif



06

```
1  <?php
2  $n1 = readline("Digite a nota 1: ");
3  $n2 = readline("Digite a nota 2: ");
4  $media = ($n1 + $n2)/2;
5  echo "\nA média é: ". $media. ". Você está: ";
6  if ($media >= 90)
7      echo "\nAprovado com louvor";
8  elseif ($media >= 70 && $media < 90)
9      echo "\nAprovado com boa nota";
10 elseif ($media >= 60 && $media < 80)
11     echo "\nAprovado na média";
12 elseif ($media >= 40 && $media < 60)
13     echo "\nEm recuperação";
14 elseif ($media > 0)
15     echo "\nReprovado";
16 else
17     echo "\nReprovado com Zero. Faltou?";
18 ?>
```



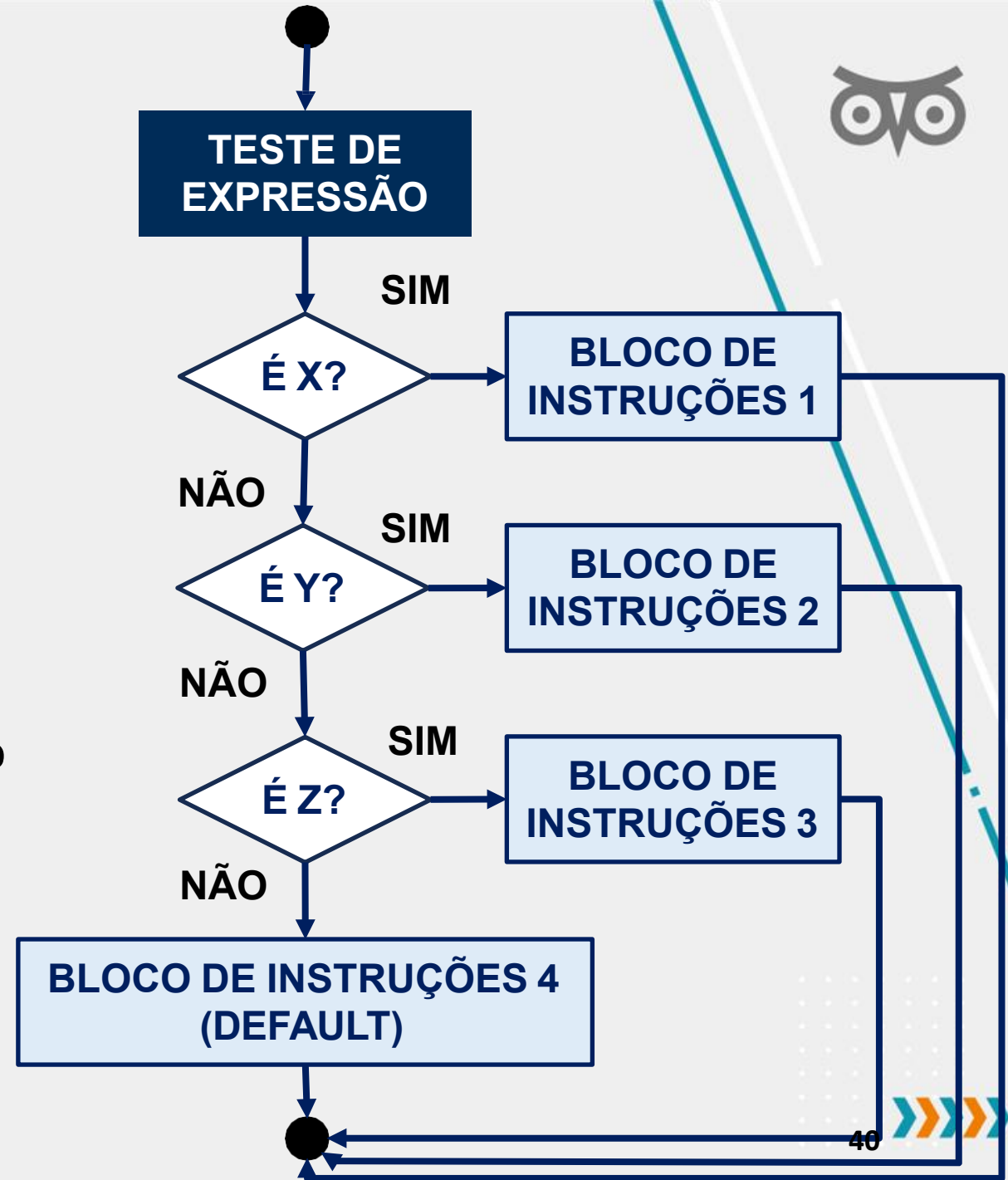


# switch ... case

Mais uma estrutura condicional para uso

**quando há muitos casos a testar**

- A expressão é testada no **switch**
- O **case** permite executar um conjunto de instruções conforme o valor da expressão
- Há ainda um caso **default**, para quando nenhum dos outros casos é acionado.







# Exemplo: **switch ... case**

Considerando algoritmo anterior:

- **“Aprovado com louvor”**, se a média for maior ou igual a 90
- **“Aprovado com boa nota”**, se a média for maior ou igual a 80 e menor que 90
- **“Aprovado na média”**, se a média for maior ou igual a 60 e menor que 80
- **“Em recuperação”**, se a média for maior ou igual a 40 e menor que 60
- **“Reprovado”**, se a média for menor que 40
- **“Reprovado com Zero. Faltou?”**, se menor ou igual a zero



# switch ... case

```
1  <?php
2  $n1 = readline("Digite a nota 1: ");
3  $n2 = readline("Digite a nota 2: ");
4  $media = ($n1 + $n2)/2;
5  echo "\nA média é: ". $media. ". Você está: ";
6  switch($media)
7  {
8      case $media >= 90:
9          echo "\nAprovado com louvor";
10         break;
11     case $media >= 70 and $media < 90:
12         echo "\nAprovado com boa nota";
13         break;
14     case $media >= 60 and $media < 70:
15         echo "\nAprovado na média";
16         break;
17     case $media >= 40 and $media < 60:
18         echo "\nEm recuperação";
19         break;
20     case $media > 0:
21         echo "\nReprovado";
22         break;
23     default:
24         echo "\nReprovado com Zero. Faltou?";
25 }
26 ?>
```



07

Interrompe a  
execução!





# Vamos exercitar?

## PARTE 1

**Acessibilidade em um Evento Cultural**



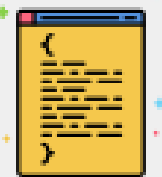


# Entrada de dados: função **readline(\$string)**

Permite a entrada de dados digitados **apenas via console**

Cálculo da área do círculo, com constante

```
<?php
define( 'PI', 3.14159265358979323846 );
$raio = readline("Digite o raio: ");
$area = PI * ($raio**2) ;
echo 'A área do triângulo é: ' . $area;
?>
```

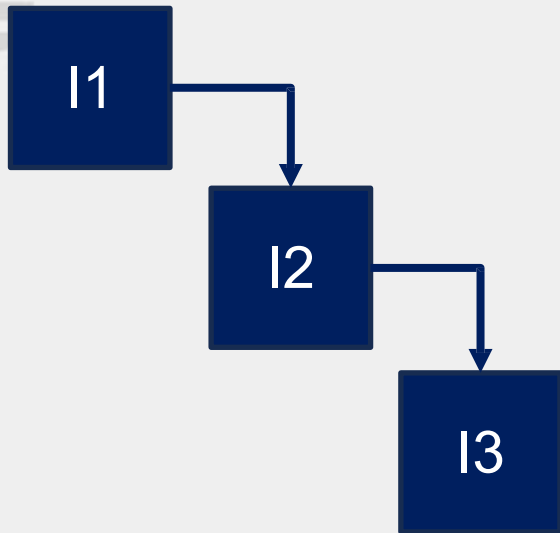


02

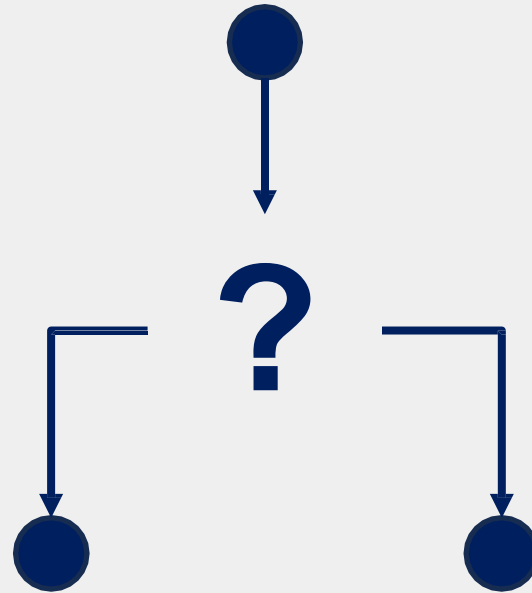




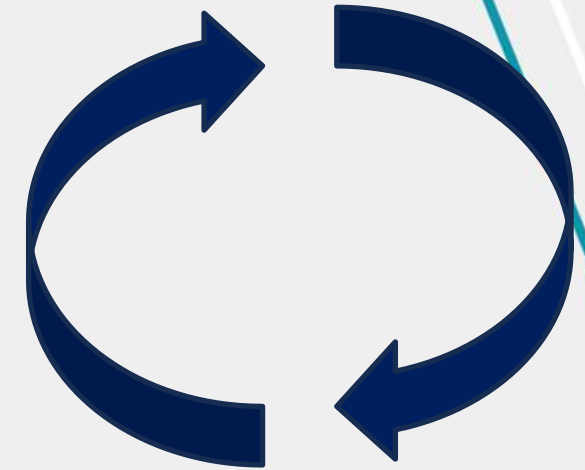
# Bases dos algoritmos



Execução de instruções  
de forma sequencial



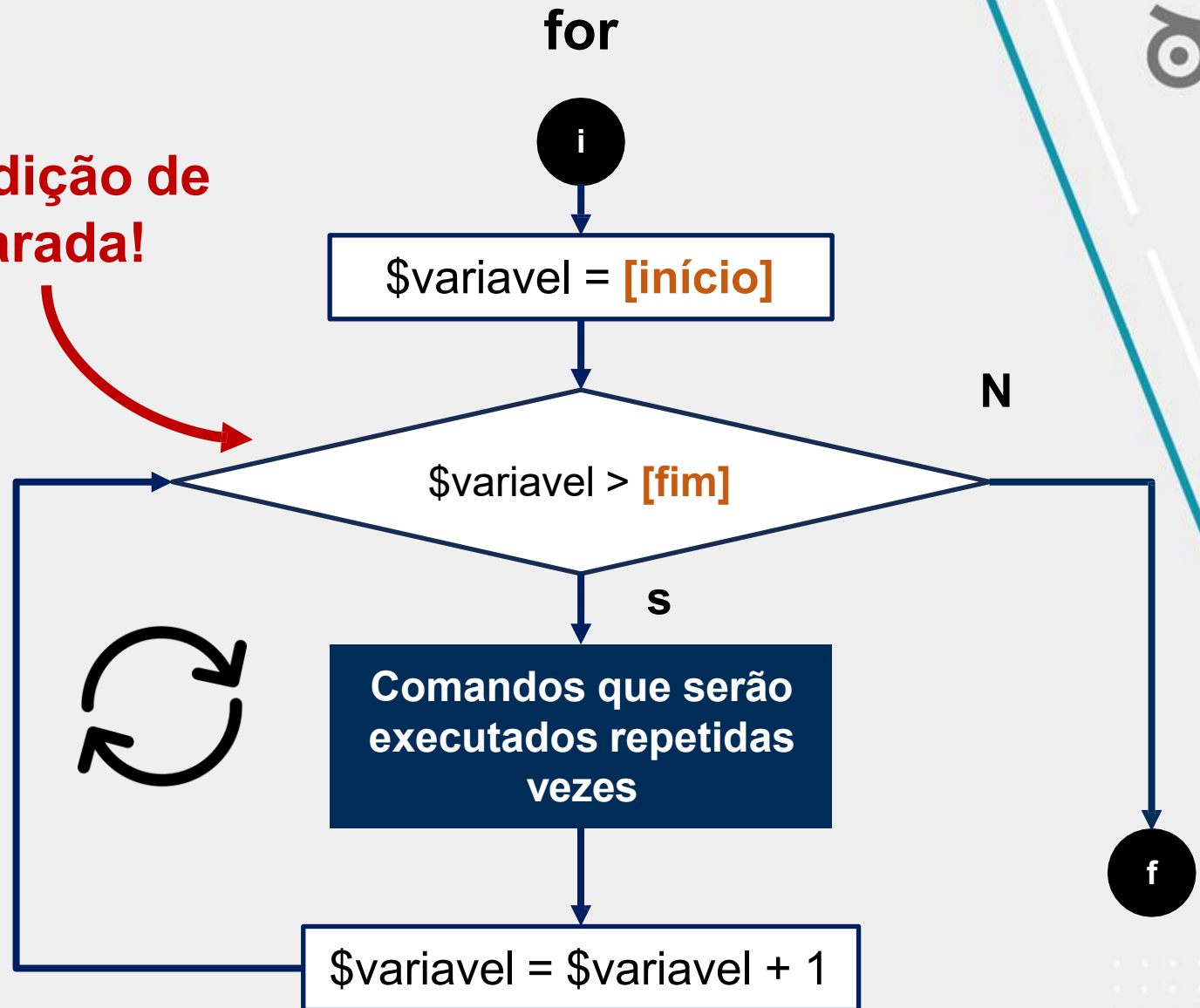
Teste que define um  
caminho a seguir  
(condicional)



Repetição de testes até  
uma condição de  
parada (loop)

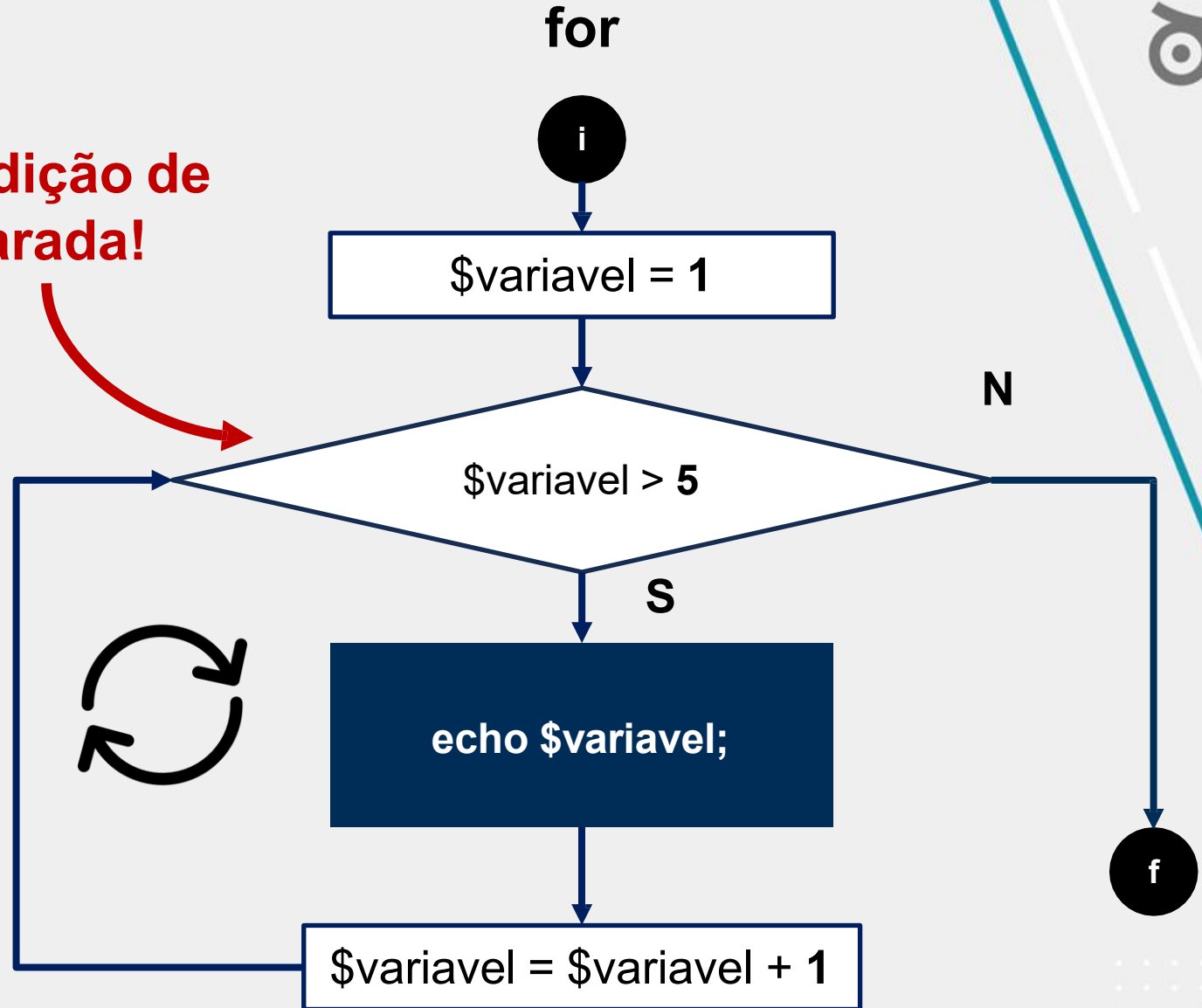
# Estruturas de repetição (laços com número determinado de repetições)

Condição de  
parada!



Estruturas de  
repetição  
(laços com  
número  
**determinado**  
de repetições)

Condição de  
parada!





# Repetindo sem laço **for**

Calcular e imprimir a  
média de 3 alunos

```
<?php
#aluno 1
$n1 = readline("Digite a nota 1: ");
$n2 = readline("Digite a nota 2: ");
$media = ($n1 + $n2)/2;
echo "\nA média é: ". $media;
#aluno 2
$n1 = readline("Digite a nota 1: ");
$n2 = readline("Digite a nota 2: ");
$media = ($n1 + $n2)/2;
echo "\nA média é: ". $media;
#aluno 3
$n1 = readline("Digite a nota 1: ");
$n2 = readline("Digite a nota 2: ");
$media = ($n1 + $n2)/2;
echo "\nA média é: ". $media;
?>
```







# Repetindo com laço **for**

Calcular e imprimir a média de 3 alunos



08

atribuição inicial

condição de parada

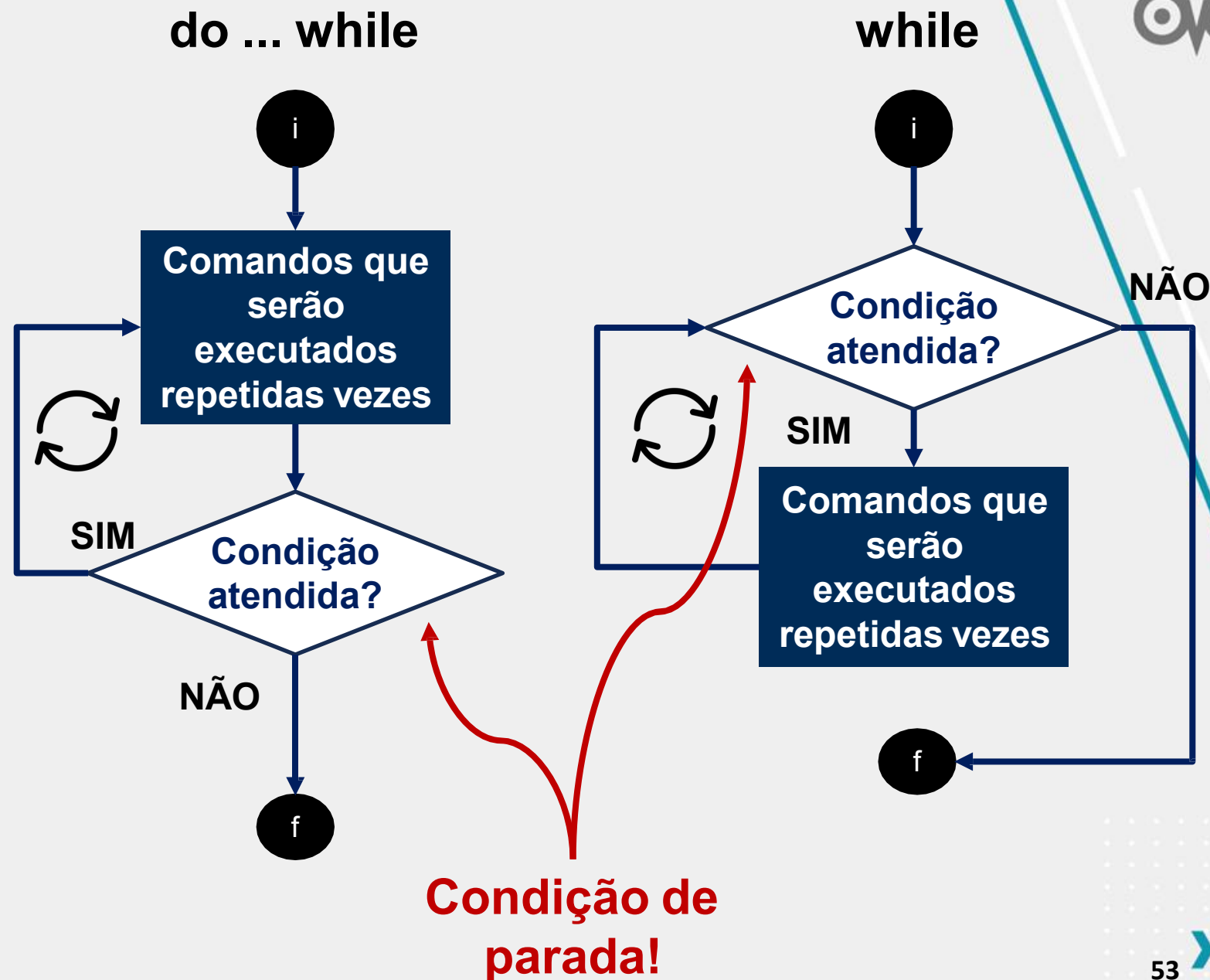
incremento da variável índice

```
1 <?php
2 for ($i = 1; $i <= 3; $i++)
3 {
4     $n1 = readline("Digite a nota 1: ");
5     $n2 = readline("Digite a nota 2: ");
6     $media = ($n1 + $n2)/2;
7     echo "\nA média é: ". $media;
8 }
9 ?>
```





Estruturas de repetição  
(laços com número **indeterminado** de repetições)





# Repetindo com laço **while**

Calcular e imprimir a  
média de 3 alunos



09

```
<?php
$i = 1;
while ($i <= 3)
{
    $n1 = readline("Digite a nota 1: ");
    $n2 = readline("Digite a nota 2: ");
    $media = ($n1 + $n2)/2;
    echo "\nA média é: ". $media;
    $i++;
}
?>
```

# Repetindo com laço **do ...while**

Calcular e imprimir a  
média de 3 alunos



10

```
<?php  
$i = 1;  
do  
{  
    $n1 = readline("Digite a nota 1: ");  
    $n2 = readline("Digite a nota 2: ");  
    $media = ($n1 + $n2)/2;  
    echo "\nA média é: ". $media;  
    $i++;  
} while ($i <= 3)  
?>
```



# Diferença dos laços **while** e **do...while**



Número  
indeterminado  
de repetições



11

```
<?php
$acabou = false;
while (!$acabou)
{
    $n1 = readline("Digite a nota 1: ");
    $n2 = readline("Digite a nota 2: ");
    $media = ($n1 + $n2)/2;
    echo "\nA média é: ". $media. "\n";
    $entrada = readline("Deseja calcular novamente? ");
    if (($entrada == 'n') or ($entrada == 'N'))
        $acabou = true;
}
?>
```







# **CENAS DOS PRÓXIMOS CAPÍTULOS...**

**foreach + arrays**



# Vamos exercitar?

## PARTE 2

**Aprimore o algoritmo para que ele permita o registro de 5 usuários**





# DÚVIDAS?



Créditos:

Profª: Carolina Sacramento  
[Carolina.sacramento@fiocruz.br](mailto:Carolina.sacramento@fiocruz.br)

