

Programação Web

O que é um algoritmo ?

O que é um algoritmo ?

É uma sequência de regras ou uma receita. Instruções que devemos seguir para realizar determinado objetivo.

O que é um algoritmo ?

É uma sequência de regras ou uma receita. Instruções que devemos seguir para realizar determinado objetivo.

Escreva seu primeiro algoritmo. Nele você deve escrever a sequência de instruções para que você conclua seu dia.

O que é um algoritmo ?

Meu dia:

- Acordar;
 - Fazer um café;
 - Tomar meu café;
 - Tomar banho;
 - Ir para o trabalho;
 - Trabalhar;
 - Almoçar;
 - Voltar ao trabalho;
 - Tomar um café;
 - Ir para o Senac
- Dar aula;
 - Ir para casa;
 - Tomar banho;
 - Dormir;

O que é um algoritmo ?

Escreva agora um algoritmo para sua semana (Segunda à Sexta)

O que é um algoritmo ?

Escreva agora um algoritmo para sua semana (Segunda à Sexta)

Minha semana:

- Repita meus dias para toda semana:
 - Se o dia da semana for sexta-feira:
 - Tomar um cerveja com os amigos depois do trabalho;

The background features a solid blue gradient that transitions from a darker shade at the bottom to a lighter shade at the top. Overlaid on this are several abstract geometric shapes and patterns. In the upper left, there is a cluster of squares and rectangles in green, pink, white, orange, and blue. A diagonal line of orange and green shapes runs from the top right towards the center. A small green square is positioned in the middle left. The overall composition is modern and minimalist.

Variáveis

Variáveis ?

São propriedades onde os dados são guardados temporariamente ou em definitivo.

Variáveis ?

São propriedades onde os dados são guardados temporariamente ou em definitivo

Podemos falar que são caixas onde colocamos dados, onde podemos manipular ou deixar guardado e utilizar quando necessário.

Variáveis ?

São propriedades onde os dados são guardados temporariamente ou em definitivo

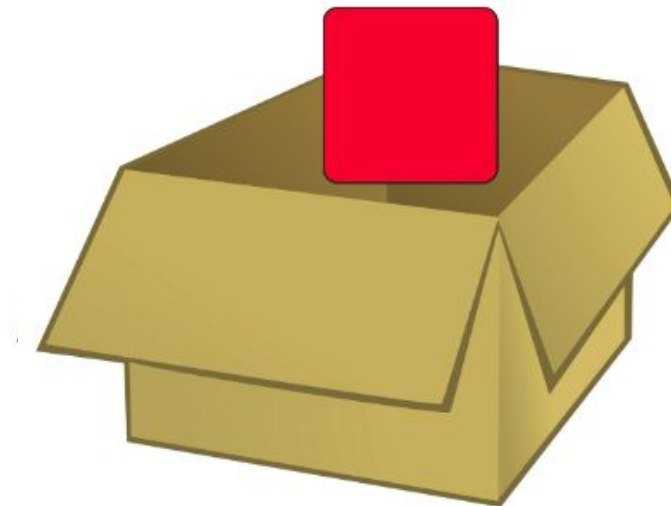
Podemos falar que são caixas onde colocamos dados, onde podemos manipular ou deixar guardado e utilizar quando necessário.



Variáveis ?

São propriedades onde os dados são guardados temporariamente ou em definitivo

Podemos falar que são caixas onde colocamos dados, onde podemos manipular ou deixar guardado e utilizar quando necessário.



Nome da caixa: **nota**

Variáveis ?

São propriedades onde os dados são guardados temporariamente ou em definitivo

Podemos falar que são caixas onde colocamos dados, onde podemos manipular ou deixar guardado e utilizar quando necessário.



Variáveis

Exemplos:

- nome = “Leonardo”
- nome_completo = “Leonardo Moreira Faêda”
- idade = 27
- altura = 1.72
- sexo = ‘M’
- possui_filhos = Falso

Como dar bons nomes de variáveis

Você deve conseguir identificar o que significa a variável pelo seu nome.

Como dar bons nomes de variáveis

Você deve conseguir identificar o que significa a variável pelo seu nome.

$x = 10$

Como dar bons nomes de variáveis

Você deve conseguir identificar o que significa a variável pelo seu nome.

`x = 14`

O que significa x ?

Se precisamos escrever em outro lugar, o que significa aquela variável é que o nome dela é ruim.

`distancia_entre_vicosa_texeiras_em_km = 14`



Tipos de variáveis

Tipos de variáveis

Os dados armazenados nas variáveis possuem tipos que devem ser ditos, os mais comuns são:

Tipos de variáveis

Os dados armazenados nas variáveis possuem tipos que devem ser ditos, os mais comuns são:

- **Inteiro:** variável do tipo inteiro, positivo ou negativo
 - inteiro idade = 27
 - inteiro temperatura = -10

Tipos de variáveis

Os dados armazenados nas variáveis possuem tipos que devem ser ditos, os mais comuns são:

- **Inteiro:** variável do tipo inteiro, positivo ou negativo
 - inteiro idade = 27
 - inteiro temperatura = -10
- **Real:** variável numérica do tipo decimal
 - real altura = 1.72
 - real valor = 99.90

Tipos de variáveis

Os dados armazenados nas variáveis possuem tipos que devem ser ditos, os mais comuns são:

- **Inteiro:** variável do tipo inteiro, positivo ou negativo
 - inteiro idade = 27
 - inteiro temperatura = -10
- **Real:** variável numérica do tipo decimal
 - real altura = 1.72
 - real valor = 99.90
- **Caracter:** variável que representa um caractere do tipo texto
 - caracter sexo = 'm'
 - caracter vogal = 'a'

Tipos de variáveis

Os dados armazenados nas variáveis possuem tipos que devem ser ditos, os mais comuns são:

- **Cadeia:** variável que representa um conjunto de caracteres do tipo texto
 - cadeia nome = “Leonardo”
 - cadeia escola = “Senac”

Tipos de variáveis

Os dados armazenados nas variáveis possuem tipos que devem ser ditos, os mais comuns são:

- **Cadeia:** variável que representa um conjunto de caracteres do tipo texto
 - cadeia nome = "Leonardo"
 - cadeia escola = "Senac"
- **Lógico:** variável que pode representar apenas os valores, verdadeiro ou falso
 - logico possui_filhos = falso
 - logico usa_oculos = falso

Declarar variáveis

```
tipo_da_variavel nome_da_variavel = valor_da_variavel
```

Declarar variáveis

```
tipo_da_variavel nome_da_variavel = valor_da_variavel
```

Podemos declarar a variável sem inicializar seu valor:


1. cadeia nome_completo
2. nome_completo = "Leonardo Moreira Faêda"

Linha 1:

- declaramos a variável

Linha 2:

- inicializar a variável

The background features a solid blue gradient. On the left side, there is a complex arrangement of overlapping geometric shapes in various colors: green, pink, white, orange, and dark blue. Some shapes are squares or rectangles, while others are triangles or polygons. A small green square is also visible in the middle-left area. The overall composition is modern and abstract.

Entrada e saída
de dados

Entrada e saída de dados

Termo exclusivo da programação, utilizada exclusivamente para interação entre o usuário e o sistema.

Entrada e saída de dados

Termo exclusivo da programação, utilizada exclusivamente para interação entre o usuário e o sistema.



Entrada e saída de dados

Termo exclusivo da programação, utilizada exclusivamente para interação entre o usuário e o sistema.



Entrada:

- O sistema solicita dados para o usuário
- e o usuário informa o dado o sistema

Entrada e saída de dados

Termo exclusivo da programação, utilizada exclusivamente para interação entre o usuário e o sistema.

Saída:

- O sistema exibe dados ou mensagens para o usuário



Entrada:

- O sistema solicita dados para o usuário
- e o usuário informa o dado o sistema

Entrada e saída de dados

Exemplo

Saída

Sistema: Qual é seu nome ?

Usuário: Leonardo

Sistema: Qual é seu sobrenome ?

Usuário: Moreira Faêda

Sistema: Bem Vindo Leonardo Moreira Faêda!

Entrada

Entrada e saída de dados

Comandos

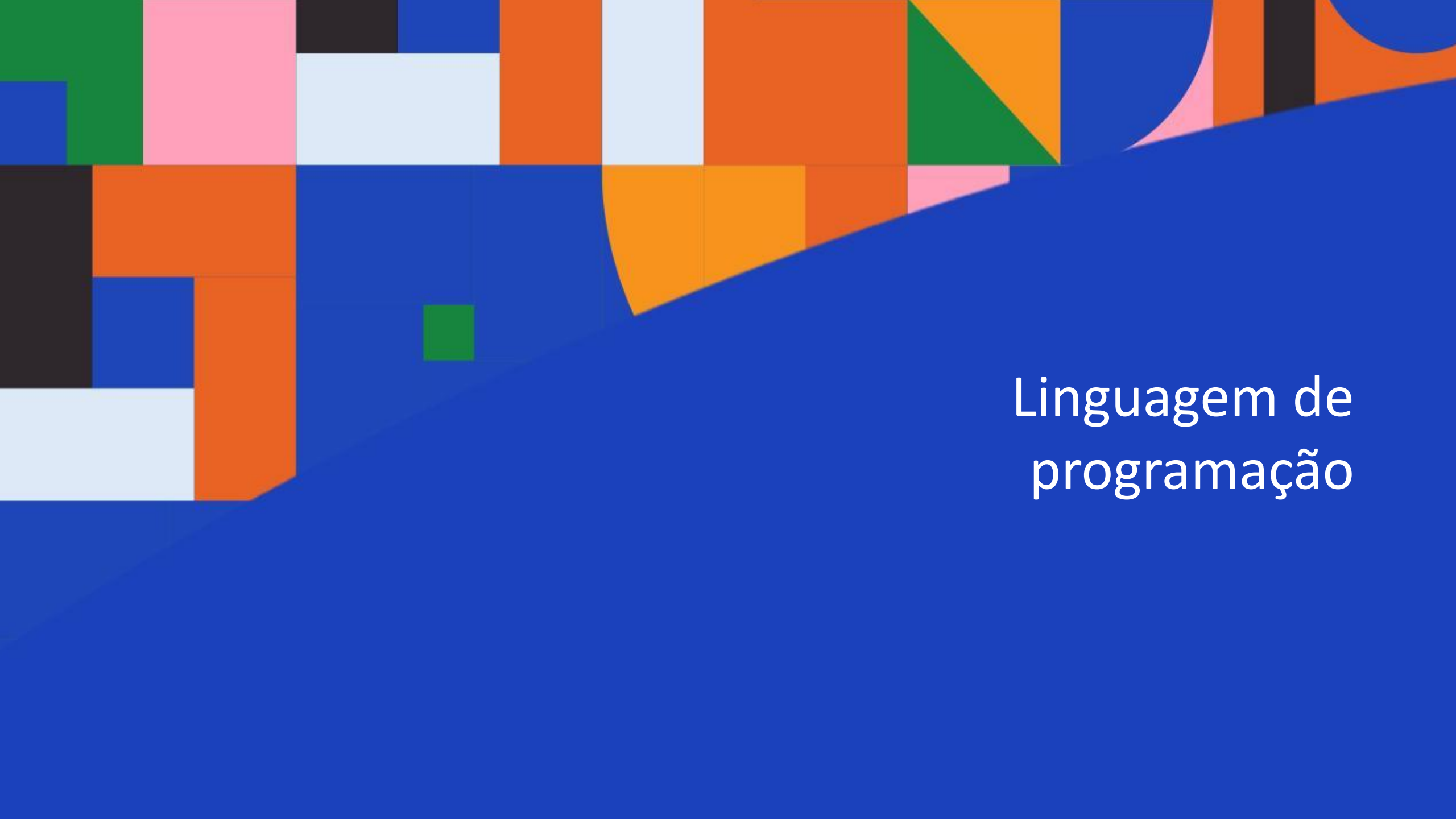
Sistema: Qual é seu nome ?

Usuário: Leonardo

Sistema: Qual é seu sobrenome ?

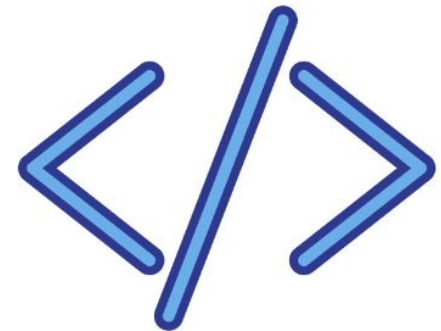
Usuário:Moreira Faêda

Sistema: Bem Vindo Leonardo Moreira Faêda!



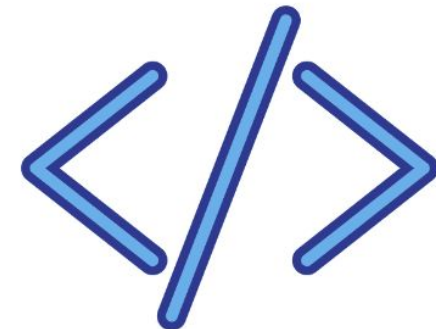
Linguagem de
programação

O que é uma linguagem de programação ?



O que é uma linguagem de programação ?

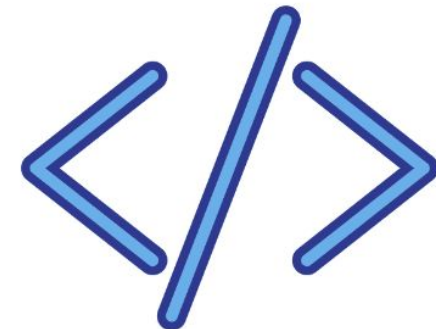
Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software).



O que é uma linguagem de programação ?

Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software).

Vocês já ouviram falar em alguma linguagem de programação ?

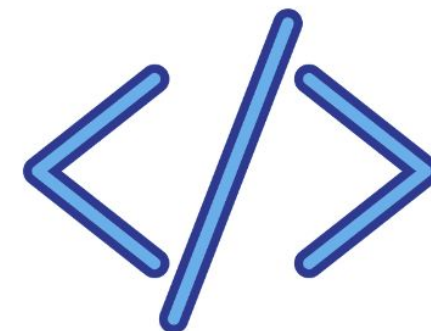


O que é uma linguagem de programação ?

Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software).

Vocês já ouviram falar em alguma linguagem de programação ?

Existem várias linguagens, com vários propósitos diferentes.



O que é uma linguagem de programação ?

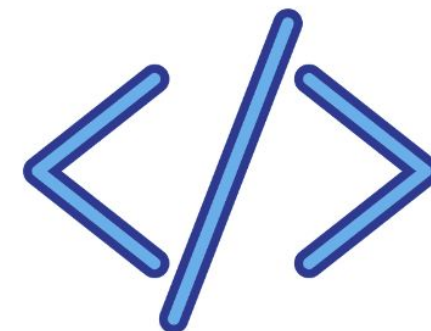
Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software).

Vocês já ouviram falar em alguma linguagem de programação ?

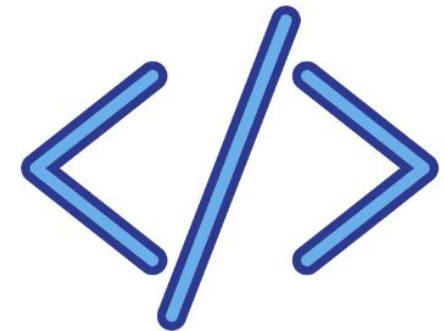
Existem várias linguagens, com vários propósitos diferentes.

Algumas delas são:

- Portugol, C, C++, C#, PHP, JavaScript, Python, Java, entre outros

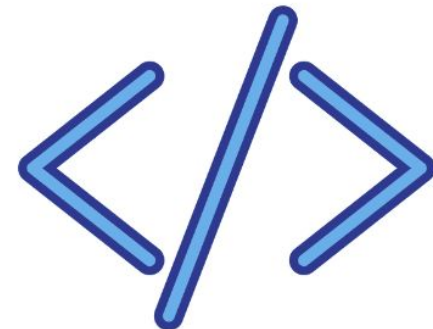


Quais linguagens vamos aprender nesse curso ?



Quais linguagens vamos aprender nesse curso ?

Vamos começar com uma linguagem para o desenvolvimento da lógica de programação, o Portugol.

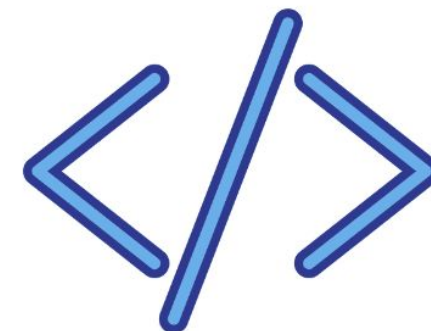


Quais linguagens vamos aprender nesse curso ?

Vamos começar com uma linguagem para o desenvolvimento da lógica de programação, o Portugol.

Mas lembre!!!!

- **Aprendendo a lógica de programação você consegue aprender qualquer linguagem de programação.**



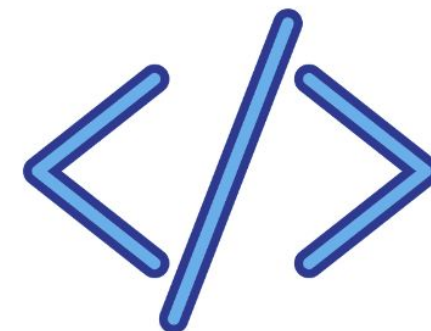
Quais linguagens vamos aprender nesse curso ?

Vamos começar com uma linguagem para o desenvolvimento da lógica de programação, o Portugol.

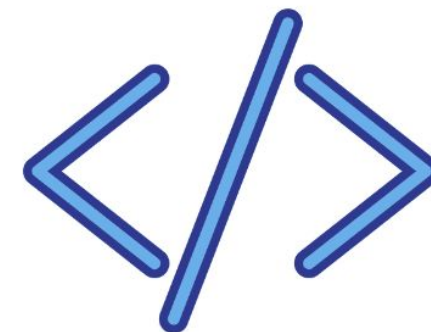
Mas lembre!!!!

- **Aprendendo a lógica de programação você consegue aprender qualquer linguagem de programação.**

Para o desenvolvimento web vamos aprender o **PHP!**

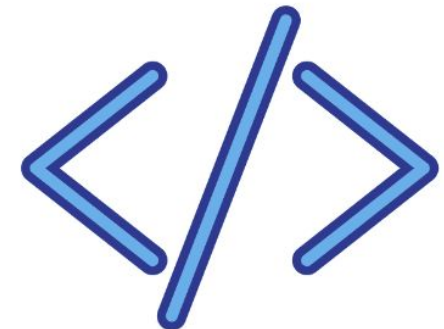


O que é uma IDE ?



O que é uma IDE ?

É um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

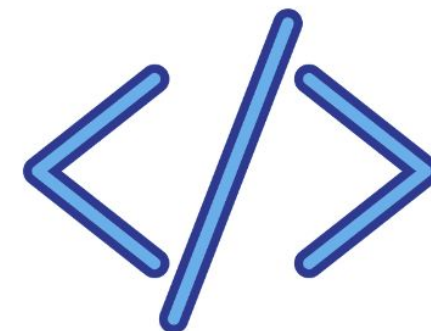


O que é uma IDE ?

É um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

Neste curso vamos utilizar duas IDE:

- Uma para desenvolver a lógica de programação (Portugol-WebStudio);
- outra para desenvolvimento web (Visual Studio Code);



O que é uma IDE ?

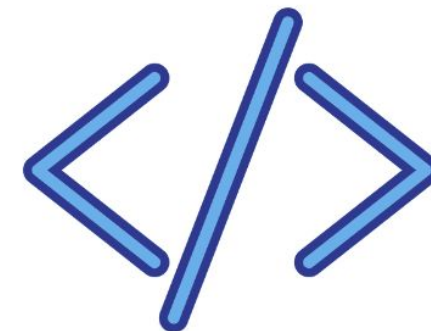
É um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

Neste curso vamos utilizar duas IDE:

- Uma para desenvolver a lógica de programação (Portugol-WebStudio);
- outra para desenvolvimento web (Visual Studio Code);

Bora iniciar no Protugol-WebStudio;

- Para acessar vamos em <https://portugol-webstudio.cubos.io/ide>



Exercícios: entrada e saída de dados

1: Escreva um algoritmo que simule uma pequena conversa entre o computador e o usuário:

- Inicialmente escreva: Olá, eu sou seu professor virtual, como é seu nome ?
- Em seguida, espere que o usuário responda.
- Por fim escreva: Bem vindo ao Senac (nome digitado).

2: Escreva um algoritmo que leia três números e os imprima na ordem inversa da que foram digitadas.

Exercícios: entrada e saída de dados

3:



The background features a solid blue gradient that transitions from a darker shade at the bottom to a lighter shade at the top. Overlaid on this are several abstract geometric shapes in various colors: green, pink, white, orange, and black. These shapes include rectangles, squares, and curved forms, some of which are partially cut off by the edges of the frame. A prominent white text label is positioned on the right side of the image.

Expressões

Expressões

Expressão é uma combinação de valores, variáveis e operadores

Expressões

Expressão é uma combinação de valores, variáveis e operadores

```
real valor_total = 19.90 + 10.50
```

Expressões

Expressão é uma combinação de valores, variáveis e operadores

```
real valor_total = 19.90 + 10.50
```

O valor_total é 30.40

Expressões

O dado da variável pode se composta por expressões de outras variáveis

```
real valor_blusa = 19.90  
real valor_calca = 10.50  
real valor_total = valor_blusa + valor_calca
```

O valor_total é 30.40



Operações Aritméticas

Operação Aritmética (Adição)

- Adição é uma das operações básicas da álgebra. Na sua forma mais simples, a adição combina dois números (termos, somados ou parcelas), em um único número, a soma ou total. Adicionar mais números corresponde a repetir a operação.
- A sintaxe é bem fácil, se coloca os operandos entre o sinal de mais.

```
01. escreva(1 + 5) //Operação Aritmética 1 + 5 sendo escrita na tela
02.
03. real numero = 50 + 30 //Operação Aritmética 50 + 30 sendo armazenada na variável numero
```


Tabela de compatibilidade de tipos da operação de adição

Operando Esquerdo	Operando Direito	Tipo Resultado	Exemplo	Resultado
cadeia	cadeia	cadeia	"Oi" + " mundo"	"Oi mundo"
cadeia	caracter	cadeia	"Banan" + 'a'	"Banana"
cadeia	inteiro	cadeia	"Faz um" + 21	"Faz um 21"
cadeia	real	cadeia	"Altura:" + 1.78	"Altura: 1.78"
cadeia	logico	cadeia	"Help bom =" + verdadeiro	"Help bom = verdadeiro"
caracter	cadeia	cadeia	'P' + "anqueca"	"Panqueca"
caracter	caracter	cadeia	'C' + 'a' + 'd' + 'e' + 'i' + 'a'	"Cadeia"
inteiro	cadeia	cadeia	22 + " de agosto"	"22 de agosto"
inteiro	inteiro	inteiro	12 + 34	46
inteiro	real	real	76 + 3.25	79.25
real	cadeia	cadeia	3.24 + " Kg"	"3.24 Kg"
real	inteiro	real	9.87 + 1	10.87
real	real	real	9.87 + 0.13	10.0
logico	cadeia	cadeia	verdadeiro + " amigo"	"verdadeiro amigo"

Exercício de Adição

- Escreva um programa que solicite dois números inteiros do usuário e exiba o somatório dos números.

Operação Aritmética (Subtração)

01. `escreva(1 - 5) //Operação Aritmética 1 - 5 sendo escrita na tela`

02.

03. `real numero = 50 - 30 //Operação Aritmética 50 - 30 sendo armazenada na variável numero`

Tabela de compatibilidade de tipos da operação de subtração

Operando Esquerdo	Operando Direito	Tipo Resultado	Exemplo	Resultado
inteiro	inteiro	inteiro	$20 - 10$	10
inteiro	real	real	$90 - 0.5$	89.5
real	inteiro	real	$11.421 - 3$	8.421
real	real	real	$12.59 - 24.59$	-12.0

Exercício de Subtração

- Escreva um programa que solicite o ano de nascimento e retorne as possibilidades de idade.

Operação Aritmética (Multiplicação)

01. escreva(1 * 5) //Operação Aritmética 1 * 5 sendo escrita na tela

02.

03. **real** numero = 50 * 30 //Operação Aritmética 50 * 30 sendo armazenada na variável numero

Tabela de compatibilidade de tipos da operação de multiplicação

Operando Esquerdo	Operando Direito	Tipo Resultado	Exemplo	Resultado
inteiro	inteiro	inteiro	$6 * 8$	48
inteiro	real	real	$4 * 1.11$	4.44
real	inteiro	real	$6.712 * 174$	1167.888
real	real	real	$207.65 * 1.23$	255.4095

Exercício de Multiplicação

- Escreva um programa que solicite do usuário o preço de um produto, e a quantidade de produtos que deseja comprar. O sistema deve retornar o preço total da compra.

Operação Aritmética (Divisão)

01. `escreva(15 / 5) //Operação Aritmética 1 * 5 sendo escrita na tela`

02.

03. `real numero = 50 / 25.6 //Operação Aritmética 50 * 30 sendo armazenada na variável numero`

Tabela de compatibilidade de tipos da operação de divisão

Operando Esquerdo	Operando Direito	Tipo Resultado	Exemplo	Resultado
inteiro	inteiro	inteiro	5 / 2	2
inteiro	real	real	125 / 4.5	27.777777
real	inteiro	real	785.4 / 3	261.8
real	real	real	40.351 / 3.12	12.9333333

Exercício de Divisão

- Escreva um programa que solicite do usuário o seu peso e altura e retorne para o usuário seu IMC (índice de massa corporal).
- Fórmula do IMC:
 - $\text{peso} / (\text{altura} * \text{altura})$

Operação Aritmética (Módulo)

- Em algumas situações faz-se necessário manipular o resto de algumas divisões. Por exemplo, se você quiser saber se um determinado valor é par ou ímpar, como faria? Para isso podemos utilizar o módulo.
- A operação módulo encontra o resto da divisão de um número por outro.

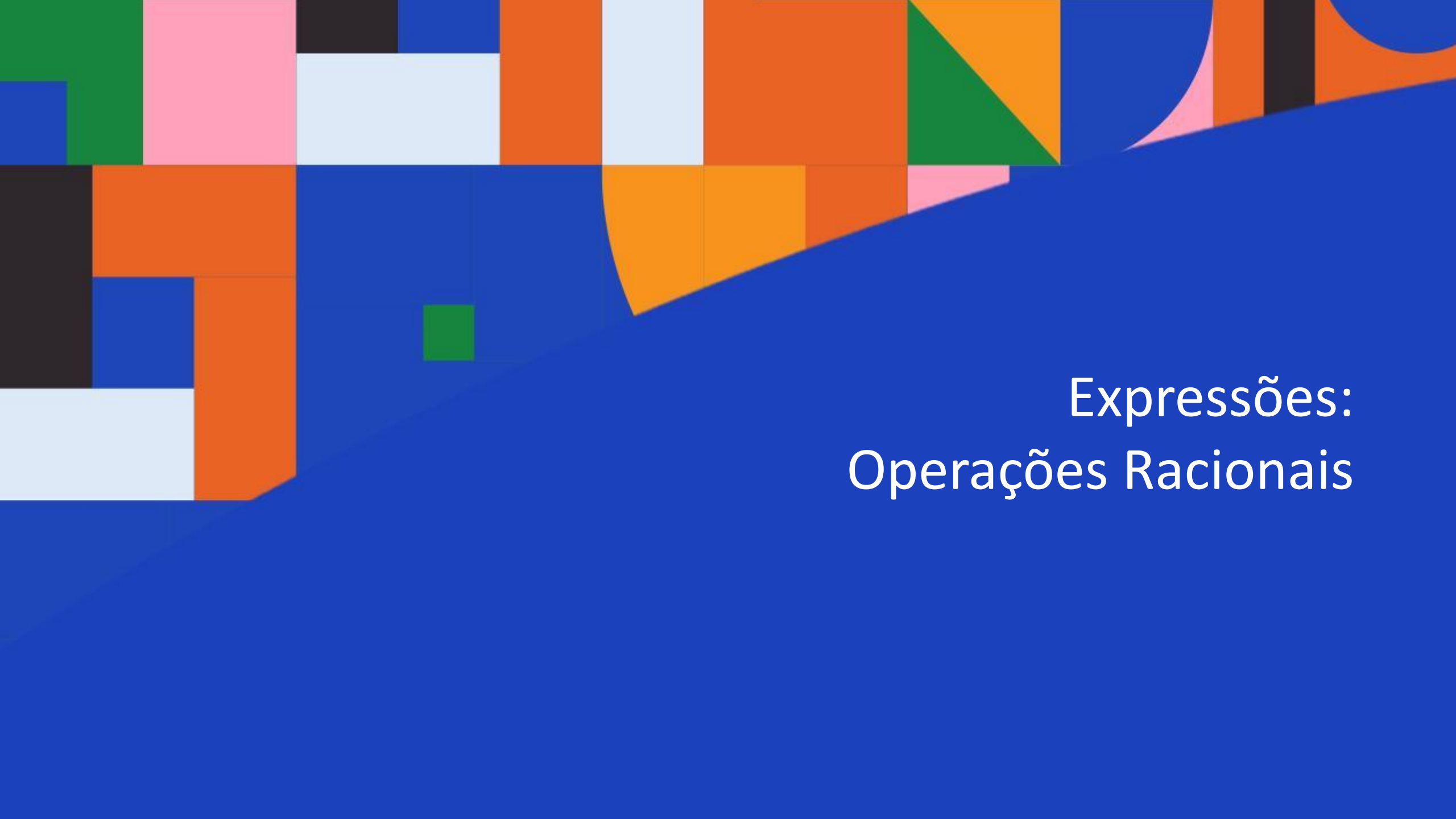
```
01. escreva(13 % 5) //Operação Aritmética 13 % 5 sendo escrita na tela
```

```
02.
```

```
03. real numero = 50 % 4 //Operação Aritmética 50 % 4 sendo armazenada na variável numero
```

Tabela de compatibilidade de tipos da operação de módulo

Operando Esquerdo	Operando Direito	Tipo Resultado	Exemplo	Resultado
inteiro	inteiro	inteiro	45 % 7	3



Expressões: Operações Racionais

Operações Relacionais

- Vamos imaginar que você precise verificar se um número digitado pelo usuário é positivo ou negativo. Como poderíamos verificar isso?

Operações Relacionais

- Vamos imaginar que você precise verificar se um número digitado pelo usuário é positivo ou negativo. Como poderíamos verificar isso?
- Através de uma operação relacional.

Operações Relacionais

- Vamos imaginar que você precise verificar se um número digitado pelo usuário é positivo ou negativo. Como poderíamos verificar isso?
- Através de uma operação relacional.
- As operações relacionais também são nossas conhecidas da Matemática. Em algoritmos, os operadores relacionais são importantes, pois permitem realizar comparações que terão como resultado um valor lógico (verdadeiro ou falso).

Operações Relacionais

- Os símbolos que usamos para os operadores também mudam um pouco em relação ao que usamos no papel. Os símbolos para diferente, maior ou igual e menor ou igual mudam pois não existem nos teclados convencionais. A tabela a seguir mostra todas as operações relacionais e os símbolos que o Portugol utiliza.

Operações Relacionais

Operação	Símbolo
Maior	>
Menor	<
Maior ou igual	>=
Menor ou igual	<=
Igual	==
Diferente	!=

Exercício

Operação
$3 > 4$
$7 \neq 7$
$9 == 10 - 1$
$33 \leq 100$
$6 \geq 5 + 1$
$5 + 4 \leq 11 - 2$



Expressões: Operações Lógicas

Operações lógicas

- As operações lógicas são uma novidade para muitos, pois raramente são vistas na escola.

Operações lógicas

- As operações lógicas são uma novidade para muitos, pois raramente são vistas na escola.
- Um operador lógico opera somente valores lógicos, ou seja, é necessário que o valor à esquerda e à direita do operador sejam valores lógicos (verdadeiro ou falso).

Operações lógicas

- As operações lógicas são uma novidade para muitos, pois raramente são vistas na escola.
- Um operador lógico opera somente valores lógicos, ou seja, é necessário que o valor à esquerda e à direita do operador sejam valores lógicos (verdadeiro ou falso).
- É muito comum usar expressões relacionais (que dão resultado lógico) e combiná-las usando operadores lógicos. Por exemplo:

Operações lógicas

Operações	Resultado
$5 > 3$ e $2 < 1$	falso
nao ($8 < 4$)	verdadeiro
$1 > 3$ ou $1 \leq 1$	verdadeiro

- Assim como as operações aritméticas, as operações lógicas também possuem prioridades.

Operador	Prioridade
ou	1
e	2
nao	3

Operações lógicas

- Ou seja, o **nao** tem maior prioridade que todos, e o **ou** tem a menor. Veja os exemplos a seguir:

Passo	Exemplo 1	Exemplo 2
Passo 1	nao verdadeiro ou falso	verdadeiro e falso ou verdadeiro
Passo 2	falso ou falso	falso ou verdadeiro
Passo 3	falso	verdadeiro

Operador lógico e

- Em algumas situações, necessitamos que alguma instrução só seja executada se outras condições forem verdadeiras.

Operador lógico e

- Em algumas situações, necessitamos que alguma instrução só seja executada se outras condições forem verdadeiras.
- Por exemplo, se você quisesse testar se duas variáveis distintas têm valor igual a 2, como faria? Para isso podemos utilizar o operador lógico **e**.

Operador lógico e

- Em algumas situações, necessitamos que alguma instrução só seja executada se outras condições forem verdadeiras.
- Por exemplo, se você quisesse testar se duas variáveis distintas têm valor igual a 2, como faria? Para isso podemos utilizar o operador lógico **e**.
- Quando usamos o operador **e** o resultado de uma operação lógica será verdadeiro somente quando AMBOS os operandos forem verdadeiros. Ou seja, basta que um deles seja falso e a resposta será falsa.

Operador lógico e

Operação 1	Operação 2	Operação 1 e Operação 2
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

Operador lógico **ou**

- Em algumas situações, necessitamos que alguma instrução seja executada se uma entre várias condições forem verdadeiras.

Operador lógico **ou**

- Em algumas situações, necessitamos que alguma instrução seja executada se uma entre várias condições forem verdadeiras.
- Por exemplo, se você quisesse testar se pelo menos uma entre duas variáveis distintas têm valor igual a 2, como faria? Para isso podemos utilizar o operador lógico **ou**.

Operador lógico **ou**

- Em algumas situações, necessitamos que alguma instrução seja executada se uma entre várias condições forem verdadeiras.
- Por exemplo, se você quisesse testar se pelo menos uma entre duas variáveis distintas têm valor igual a 2, como faria? Para isso podemos utilizar o operador lógico **ou**.
- Quando usamos o operador **ou** o resultado de uma operação lógica será verdadeiro sempre que UM dos operandos for verdadeiro. A tabela verdade a seguir ilustra o comportamento do operador **ou**.

Operador lógico **ou**

Operação 1	Operação 2	Operação 1 ou Operação 2
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

Operador lógico **nao**

- Em algumas situações precisamos verificar se o contrário de uma sentença é verdadeiro ou não.

Operador lógico **nao**

- Em algumas situações precisamos verificar se o contrário de uma sentença é verdadeiro ou não.
- Por exemplo, se você tem uma variável com um valor falso, e quer fazer um teste que será positivo sempre que essa variável for falsa, como faria? Para isso podemos utilizar o operador lógico **nao**.

Operador lógico **nao**

- Em algumas situações precisamos verificar se o contrário de uma sentença é verdadeiro ou não.
- Por exemplo, se você tem uma variável com um valor falso, e quer fazer um teste que será positivo sempre que essa variável for falsa, como faria? Para isso podemos utilizar o operador lógico **nao**.
- O operador **nao** funciona de forma diferente pois necessita apenas de um operando. Quando usamos o operador **nao**, o valor lógico do operando é invertido, ou seja, o valor falso torna-se verdadeiro e o verdadeiro torna-se falso.



Estrutura de Controle: Desvios Condicionais

Condicionais

- Não é só na vida que fazemos escolhas.

Condicionais

- Não é só na vida que fazemos escolhas.
- Nos algoritmos encontramos situações onde um conjunto de instruções deve ser executado caso uma condição seja verdadeira

Condicionais

- Não é só na vida que fazemos escolhas.
- Nos algoritmos encontramos situações onde um conjunto de instruções deve ser executado caso uma condição seja verdadeira
- Por exemplo: para que você possa tirar carteira de motorista, você deve ter 18 anos ou mais, caso contrário, você não vai ter permissão.

Condicionais se

- Aqui veremos como dizer a um algoritmo quando um conjunto de instruções deve ser executado.
- Esta determinação é estabelecida se uma condição for verdadeira
- Mas o que seria esta condição?
- Ao executar um teste lógico teremos como resultado um valor verdadeiro ou falso. A condição descrita anteriormente nada mais é que um teste lógico.

Condicionais **se**

- Se este teste lógico resultar verdadeiro, as instruções definidas dentro do desvio condicional serão executadas. Se o teste for falso, o algoritmo pulará o trecho e continuará sua execução a partir do ponto onde o desvio condicional foi finalizado.

Condicionais se

```
03.      funcao inicio()
04.      {
05.
06.          inteiro num
07.
08.          escreva ("Digite um número: ")
09.          leia (num)
10.
11.          se (num==0)
12.          {
13.              escreva ("O número digitado é 0")
14.          }
15.
16.      }
```

Exercício

- 1- Escreva um algoritmo que solicite a idade do usuário e exiba se ele possui idade para tirar carteira de motorista.
- 2- Faça um algoritmo que leia os valores A, B, C e imprima na tela se a soma de $A + B$ é menor que C
- 3-) Faça um algoritmo que leia o nome, o sexo e o estado civil de uma pessoa. Caso sexo seja “F” e estado civil seja “CASADA”, solicitar o tempo de casada (anos).

Condicionais (**senao**)

- Agora vamos imaginar que se a condição for falsa um outro conjunto de comandos deve ser executado. Quando iremos encontrar esta situação?

Condicionais (**senao**)

- Agora vamos imaginar que se a condição for falsa um outro conjunto de comandos deve ser executado. Quando iremos encontrar esta situação?
- Imagine um programa onde um aluno com média final igual ou maior a 6 é aprovado. Se quisermos construir um algoritmo onde após calculada a média, seja mostrada na tela uma mensagem indicando se o aluno foi aprovado ou reprovado. Como fazer isto? Utilizando o comando **se** junto com o **senao**.

Condicionais (senao)

```
01. logico condicao = falso
02. se (condicao)
03. {
04.     //Instruções a serem executadas se o desvio for verdadeiro
05. }
06. senao
07. {
08.     //Instruções a serem executadas se o desvio for falso
09. }
```


Exercícios



Condicionais (**senao se**)

- Agora imagine que você precise verificar a nota da prova de um aluno e falar se ele foi muito bem, bem, razoável ou mau em uma prova como fazer isto ?

Condicionais (**senao se**)

- Agora imagine que você precise verificar a nota da prova de um aluno e falar se ele foi muito bem, bem, razoável ou mau em uma prova como fazer isto ?
- Quando você precisa verificar se uma condição é verdadeira, e se não for, precise verificar se outra condição é verdadeira uma das formas de se fazer esta verificação é utilizando do **se ... senao se**;
-

Condicionais (**senao se**)

- Agora imagine que você precise verificar a nota da prova de um aluno e falar se ele foi muito bem, bem, razoável ou mau em uma prova como fazer isto ?
- Quando você precisa verificar se uma condição é verdadeira, e se não for, precise verificar se outra condição é verdadeira uma das formas de se fazer esta verificação é utilizando do **se ... senao se**;
- A sua sintaxe é parecida com a do **senao**, mas usando o comando **se** imediatamente após escrever o comando **senao**.

Condicionais (senao se)

```
01. logico condicao = falso
02. logico condicao2 = verdadeiro
03. se (condicao)
04. {
05.     //Instruções a serem executadas se o desvio for verdadeiro
06. }
07. senao se (condicao2)
08. {
09.     //Instruções a serem executadas se o desvio anterior for falso e este desvio for verdadeiro
10. }
```

Condicionais (**senao se**)

- Também pode-se colocar o comando **senao** no final do ultimo **senao se**, assim quando todos os testes falharem, ele irá executar as instruções dentro do **senao**

```
01.  se (12 < 5)
02.  {
03.      //Instruções a serem executadas se o desvio for verdadeiro
04.  }
05.  senao se ("palavra" == "texto")
06.  {
07.      //Instruções a serem executadas se o desvio anterior for falso e este desvio for verdadeiro
08.  }
09.  senao
10.  {
11.      //Instruções a serem executadas se o desvio anterior for falso
12.  }
```

- Escreva um programa que leia a nota de uma aluno e se a nota for maior ou igual que 9, retornar que o aluno foi muito bem na prova. Caso sua nota for maior ou igual a 7 e menor que 9, retornar que o aluno foi bem na prova. Caso sua nota for maior igual a 6 e menor 7, retornar o aluno foi razoável. Caso tire outra nota, retornar o aluno teve um desempenho ruim.

```
01.  se (12 < 5)
02.  {
03.      //Instruções a serem executadas se o desvio for verdadeiro
04.  }
05.  senao se ("palavra" == "texto")
06.  {
07.      //Instruções a serem executadas se o desvio anterior for falso e este desvio for verdadeiro
08.  }
09.  senao
10.  {
11.      //Instruções a serem executadas se o desvio anterior for falso
12.  }
```

Escolha caso

- Qual a melhor forma para programar um menu de, por exemplo, uma calculadora? Esta tarefa poderia ser executada através de desvios condicionais **se** e **senão**, porém esta solução seria complexa e demorada. Pode-se executar esta tarefa de uma maneira melhor, através de outro tipo de desvio condicional: o **escolha** junto com o **caso**. Este comando é similar aos comandos **se** e **senão**, e reduz a complexidade do problema.

Escolha caso

- Qual a melhor forma para programar um menu de, por exemplo, uma calculadora? Esta tarefa poderia ser executada através de desvios condicionais **se** e **senão**, porém esta solução seria complexa e demorada. Pode-se executar esta tarefa de uma maneira melhor, através de outro tipo de desvio condicional: o **escolha** junto com o **caso**. Este comando é similar aos comandos **se** e **senão**, e reduz a complexidade do problema.
- Apesar de suas similaridades com o **se**, ele possui algumas diferenças. Neste comando não é possível o uso de operadores lógicos, ele apenas trabalha com valores definidos, ou o valor é igual ou diferente. Além disto, o **escolha** e o **caso** tem alguns casos testes, e se a instrução **pare** não for colocada ao fim de cada um destes testes, o comando executará todos casos existentes.

Escolha caso

- A sintaxe do **escolha** é respectivamente o comando **escolha** a condição a ser testada e entre chaves se coloca os casos

Escolha caso

- A sintaxe do **escolha** é respectivamente o comando **escolha** a condição a ser testada e entre chaves se coloca os casos
- A sintaxe para se criar um caso é a palavra reservada **caso**, o valor que a condição testada deve possuir, dois pontos e suas instruções. Lembre-se de terminá-las com o comando **pare**

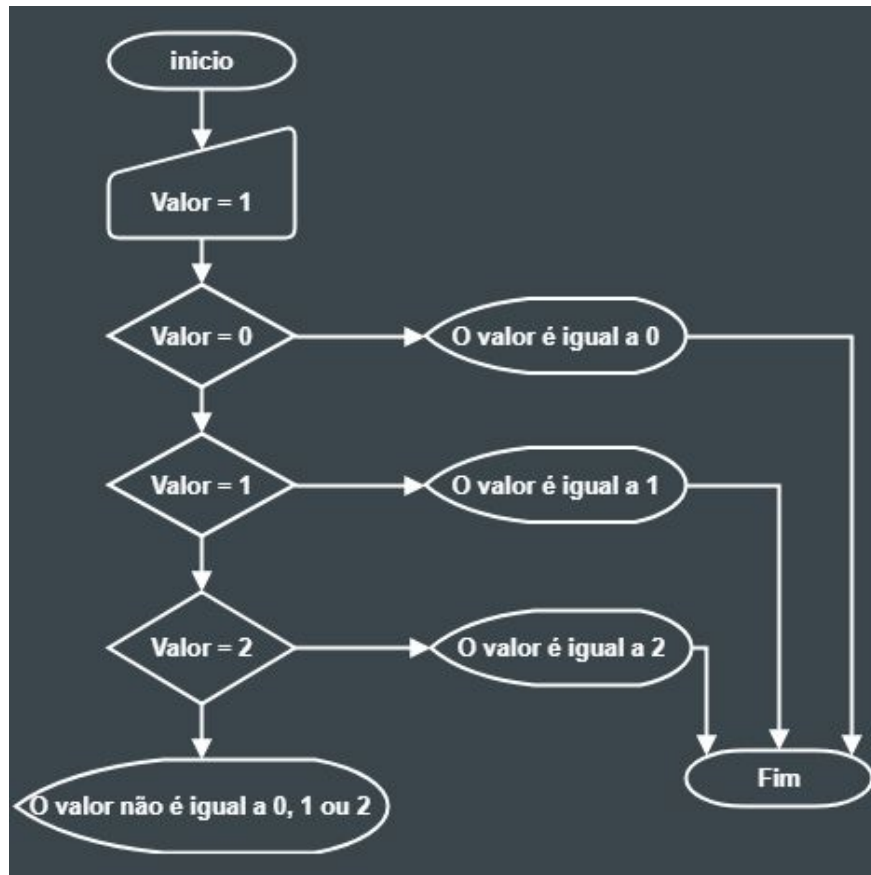
Escolha caso

```
01. inteiro numero
02. leia(numero)
03. escolha(numero)
04. {
05.     caso 1:
06.         //Instruções caso o numero for igual a 1
07.     pare
08.
09.     caso 2:
10.         //Instruções caso o numero for igual a 2
11.     pare
12.
13.     caso 50:
14.         //Instruções caso o numero for igual a 50
15.     pare
```

Escolha caso

```
01. inteiro numero
02. leia(numero)
03. escolha(numero)
04. {
05.     caso 1:
06.         //Instruções caso o numero for igual a 1
07.     pare
08.
09.     caso 2:
10.         //Instruções caso o numero for igual a 2
11.     pare
12.
13.     caso 50:
14.         //Instruções caso o numero for igual a 50
15.     pare
```

Exercício (escolha caso)



```
01. inteiro numero
02. leia(numero)
03. escolha(numero)
04. {
05.     caso 1:
06.         //Instruções caso o numero for igual a 1
07.     pare
08.
09.     caso 2:
10.         //Instruções caso o numero for igual a 2
11.     pare
12.
13.     caso 50:
14.         //Instruções caso o numero for igual a 50
15.     pare
```

Exercícios de Avaliação

- 4) **Faça um algoritmo que leia dois valores inteiros A e B se os valores forem iguais deverá se somar os dois, caso contrário multiplique A por B. Ao final de qualquer um dos cálculos deve-se atribuir o resultado para uma variável C e mostrar seu conteúdo na tela.**
- 5) **Encontrar o dobro de um número caso ele seja positivo e o seu triplo caso seja negativo, imprimindo o resultado.**
- 6) **Escreva um algoritmo que lê dois valores booleanos (lógicos) e então determina se ambos são VERDADEIROS ou FALSOS.**
- 7) **Faça um algoritmo que leia uma variável e some 5 caso seja par ou some 8 caso seja ímpar, imprimir o resultado desta operação.**
- 8) **Escreva um algoritmo que leia três valores inteiros e diferentes e mostre-os em ordem decrescente.**

Exercícios de Avaliação

9) Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:

- para homens: $(72.7 * h) - 58$;
- para mulheres: $(62.1 * h) - 44.7$.

10) O IMC – Índice de Massa Corporal é um critério da Organização Mundial de Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula é $IMC = \text{peso} / (\text{altura})^2$. Elabore um algoritmo que leia o peso e a altura de um adulto e mostre sua condição de acordo com a tabela abaixo.

IMC em adultos Condição
Abaixo de 18,5 Abaixo do peso
Entre 18,5 e 25 Peso normal
Entre 25 e 30 Acima do peso
Acima de 30 obeso

Exercícios de Avaliação

11) Elabore um algoritmo que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual condição de pagamento escolhida e efetuar o cálculo adequado.

Código Condição de pagamento

1 À vista em dinheiro ou cheque, recebe 10% de desconto

2 À vista no cartão de crédito, recebe 15% de desconto

3 Em duas vezes, preço normal de etiqueta sem juros

4 Em duas vezes, preço normal de etiqueta mais juros de 10%

12) Escreva um algoritmo que leia o número de identificação, as 3 notas obtidas por um aluno nas 3 verificações e a média dos exercícios que fazem parte da avaliação, e calcule a média de aproveitamento, usando a fórmula:

$MA := (nota1 + nota2 * 2 + nota3 * 3 + ME) / 7$

A atribuição dos conceitos obedece a tabela abaixo. O algoritmo deve escrever o número do aluno, suas notas, a média dos exercícios, a média de aproveitamento, o conceito correspondente e a mensagem 'Aprovado' se o conceito for A, B ou C, e 'Reprovado' se o conceito for D ou E.

Média de aproveitamento Conceito

≥ 90 A

≥ 75 e < 90 B

≥ 60 e < 75 C

≥ 40 e < 60 D

< 40 E



Siga o Senac em Minas nas Redes Sociais:

