

Variáveis

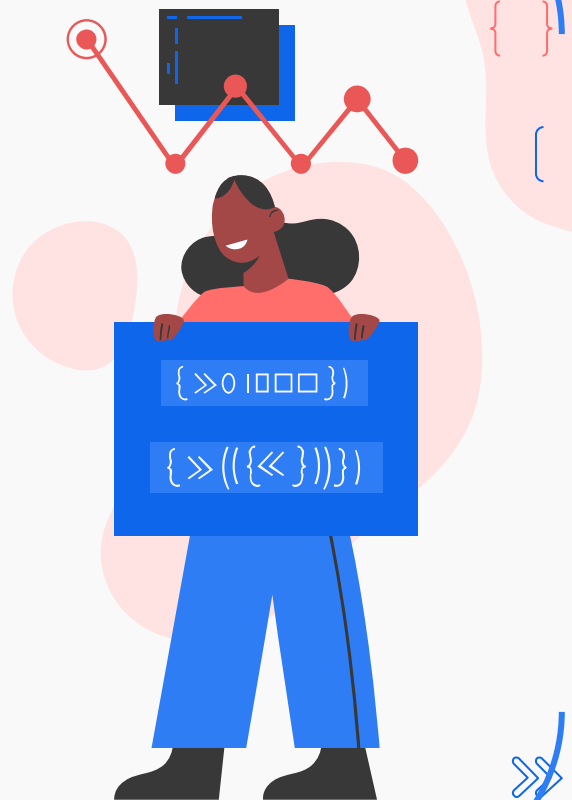
Desenvolver Algoritmos

Msc. Lucas G. F. Alves
e-mail: lgfalves@senacrs.com.br



Planejamento de Aula

Revisão Algoritmos
Atividade de Revisão





Algoritmos naturais x estruturados



Diferenças

A principal diferença reside na forma como as instruções são organizadas e executadas:

Algoritmos Naturais: São mais fáceis de compreender, como uma receita culinária tradicional. As instruções são escritas como fala, com passos sequenciais e pode ter alterações.



Algoritmos Estruturados: São um pouco mais difíceis de entender, como um código de programação. As instruções são organizadas em blocos lógicos, com sequência definida e condições específicas para cada passo.

{((({>>}))<<}





Algoritmos naturais x estruturados



[]

Algoritmos Naturais

Algoritmo Natural (receita de bolo):

2 ovos,
1 xícara de leite,
2 xícaras de farinha de trigo,
1 xícara de açúcar.

Misture os ingredientes secos.
Bata as claras em neve.
Adicione as claras à massa.
Asse em forno pré-aquecido.
Espere esfriar antes de servir.

{ }

{((({>> }))<<) }

- []



Algoritmos naturais x estruturados



[]

Algoritmo Estruturado

Algoritmo Estruturado (código para calcular a média):

1. Leia as notas dos alunos.
2. Calcule a soma das notas.
3. Divida a soma pelo número de alunos.
4. Exiba a média.

{ }

{((({>>}))<<}

-[]



Comandos de entrada

[]

O que são Inputs ou Entrada?

Os comandos de entrada em uma receita, são os ingredientes que você entrega ao cozinheiro.

No mundo da computação, isso significa fornecer informações ao computador, como:

- Digitar texto no teclado.
- Clicar com o mouse.
- Ler dados de um arquivo.
- Receber informações de outros dispositivos.

{ }

Exemplo de código: `prompt("Qual é o seu nome?")`

```
{((({>>}))<<}
```

-[]



Comandos de processamento

[]

O que são processamento?

Os comandos de processamento em uma receita, é o trabalho do cozinheiro transformando os ingredientes na comida.

No computador, é o processamento das informações que você forneceu, como:

- Realizar cálculos matemáticos.
- Comparar dados e tomar decisões.
- Ordenar informações.
- Criar novos dados a partir dos existentes.

{ }

Exemplo de código: `let somar = 1+1;`

```
{((({>>}))<<}
```

-[]



Comandos de saída

[]

O que são saídas ou Outputs?

Os comandos de saída em uma receita, é a comida pronta que o cozinheiro te entrega.
No computador, as saída são as informações resultantes do processamento, que podem ser:

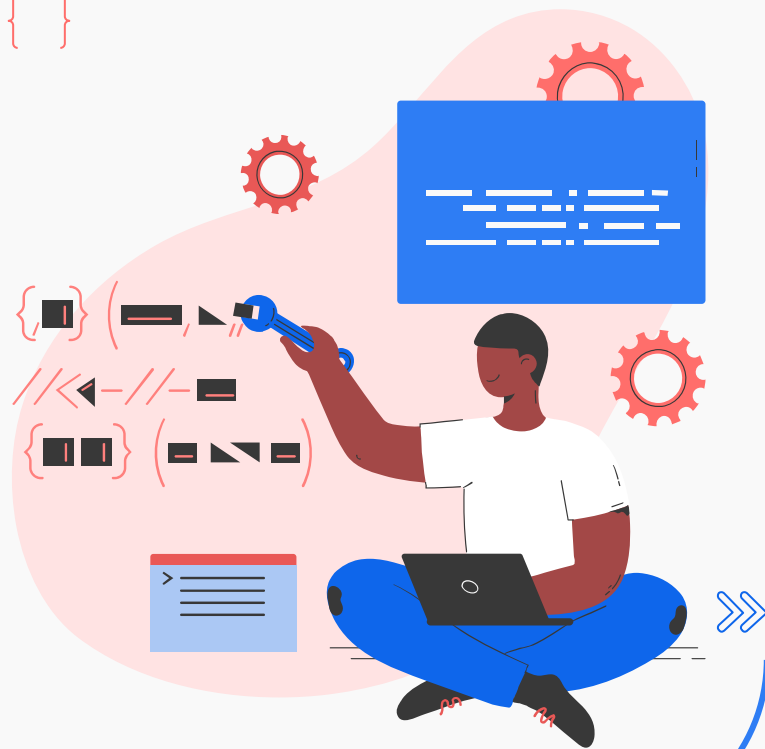
- Exibidas na tela.
- Salvas em um arquivo.
- Enviadas para outro dispositivo.
- Impressas em papel.

{ }

Exemplo de código: `console.log(somar);`

`({{{({>>}}))<<}`

- []





Representação visual

[]

Formas de representar algoritmos

Existem diversas formas de representar um algoritmo. As mais comuns são por textos livres, fluxogramas e pseudo-códigos.

Texto livre

Texto livre seria o algoritmo natural descrito em linguagem natural. Exemplo de uma receita de bolo.

{ }

{((({>> }))<< }

- []



Representação visual

[]

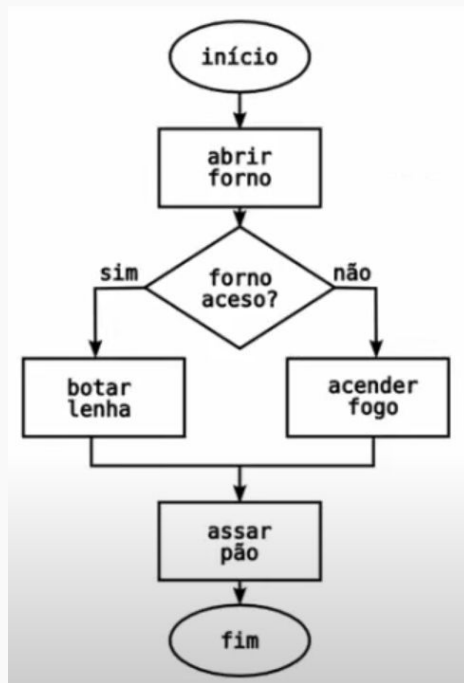
Fluxograma

É a representação gráfica de algoritmos onde formas geométricas diferentes Implicam em ações (instruções ou comandos) diferentes.

Um fluxograma é um diagrama que nos ajuda a entender a ordem em que cada coisa acontece na nossa solução de uma forma visual.

{ }

{((({>>}))<<}


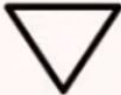






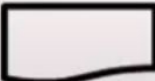



- []



Representação visual

Fluxograma

SIMBOLOGIA DE FLUXOGRAMAS PADRÃO ANSI - American National Standards Institute			
SÍMBOLO	SIGNIFICADO	SÍMBOLO	SIGNIFICADO
	Operação		Armazenagem
	Movimento / Transporte		Sentido de fluxo
	Ponto de decisão		Conexão¹
	Inspeção		Limites (início, pare, fim)
	Documento impresso	1 -	Utilizado quando o fluxograma não cabe em uma única página
	Espera		

{((({>>}))<<}





Atividade em grupo

[]

Desvendando o Mistério da Biblioteca - Algorithm Flowchart

Criar um fluxograma na ferramenta miro para um algoritmo que encontre um livro raro na biblioteca. Você sabe que o livro está na biblioteca, mas não sabe em qual estante ele está.

Pistas:

- O livro é sobre história.
- O livro tem mais de 100 anos.
- O livro tem uma capa vermelha.

Exemplo:

{ } Retângulo: "Ir para a seção de história da biblioteca."

Losango: "O livro está na estante?"

Seta: Conecta as etapas do fluxograma.

{((({ >> })) <<) }

- []

Atividade de Revisão





Atividade de Revisão

[]

Antes de começar os exercícios, baixe o exemplo de projeto (template) no drive ou crie os arquivos `index.html` e `index.js` no VS Code dentro da pasta `aula5-variaveis`.

Os exercícios de hoje são divididos em interpretação e escrita de código. Para os de interpretação, coloque as respostas em **comentários de código**. Já, nos de escrita, lembre-se de imprimir no console os resultados.

Tente responder os exercícios de interpretação sem executar o código. Se ficar muito difícil, pode rodar no seu computador para analisar e pensar sobre o resultado.

{ }

{ ((({ >> })) <<) }

- []



Atividade de Revisão

[]

Exercícios de interpretação de código.

1)

```
let a = 10
let b = 10

console.log(b)

b = 5
console.log(a, b)
```

2)

```
let a = 10
let b = 20
c = a
b = c
a = b

console.log(a, b, c)
```

{ }

{((({>>}))<<)}

-[]



Atividade de Revisão

[]

Exercícios de interpretação de código.

3) Analise o programa abaixo, entenda o que ele faz e sugira nomes melhores para as variáveis. Lembre-se que devemos escolher nomes significativos, usar o padrão **camelCase**. Além disso, os nomes não podem começar com números ou caracteres especiais.

```
let p = prompt("Quantas horas você trabalha por dia?")
let t = prompt("Quanto você recebe por dia?")
console.log("Voce recebe",t,"/",p,"por hora")
```

{ }

{((({>>}))<<}

-[]



Atividade de Revisão

[]

Exercícios de escrita de código

1) Construa um programa, seguindo os seguintes passos:

1. Declare uma variável para armazenar um nome, sem atribuir um valor.
2. Declare uma variável para armazenar uma idade, sem atribuir um valor.
3. Imprima na tela o tipo dessas variáveis que acabou de criar, usando o comando ``typeof``.
4. Reflita: por que esse tipo foi impresso? Escreva a resposta em um comentário de código.
5. Pergunte ao usuário seu nome e sua idade, atribuindo esses dois valores às variáveis que acabou de criar.
6. Imprima na tela o tipo dessas variáveis. Escreva em um comentário de código.
7. Para finalizar, imprima na tela a mensagem: "Olá `nome`, você tem `idade` anos". Onde `{((({>>}))<<}` `nome` e `idade` são os valores que o usuário inseriu.

{ }

{ }



Atividade de Revisão

[]

Exercícios de escrita de código

2) Escreva um programa que faça 3 perguntas de Sim ou Não, armazenado em uma variável. Por exemplo: "Você está usando uma roupa azul hoje?". Depois, siga os passos:

a) Crie três novas variáveis, contendo as respostas

b) Imprima na tela todas as perguntas seguidas por suas respectivas respostas. Por exemplo:

{ }

Você está usando uma roupa azul hoje? - SIM

{((({>>}))<<}

- []



Atividade de Revisão

[]

Exercícios de escrita de código

3) Suponha que temos duas variáveis a e b, cada uma com um valor inicial.

let a = 10

let b = 25

Agora, queremos trocar os valores delas, de forma que `a` passe a ter o valor de `b` e `b` passe a ter o valor de `a`. Ou seja, no caso desse exemplo acima, `a` passaria a ser 25 e `b` passaria a ser 10.

{ }

{((({>>}))<<)}

```
let a = 10
let b = 25

// Aqui faremos uma lógica para trocar os valores

// Depois de trocados, teremos o seguinte resultado:
console.log("O novo valor de a é", a) // O novo valor de a é 25
console.log("O novo valor de b é", b) // O novo valor de b é 10
```

- []



Atividade de Revisão



Desafio Opcional!

Faça um programa que receba dois números do usuário e faça as seguintes operações, imprimindo os resultados no console da seguinte forma:

1. O primeiro número somado ao segundo número resulta em: x.
2. O primeiro número multiplicado pelo segundo número resulta em: y.

Dica: Se os resultados das operações não estiverem corretos, reflita sobre o tipo dos valores recebidos pelo prompt.



```
{((({>>}))<<}
```





Atividade de Revisão

[]

Entrega

- Crie uma pasta chamado `aula5-variaveis`
- Envie o link do repositório para o professor por email

Terminou os exercícios? Primeiramente, parabéns 🙌

Agora, aproveite para oferecer ajuda à turma. Começar a aprender programação pode ser muito difícil, então se essa lista pareceu muito fácil para você, imagine-se no lugar de quem nunca tinha visto nada parecido na vida. Compartilhar conhecimento é sempre muito

{ } bem-vindo por aqui!

{((({>>}))<<}

- []

Obrigado!



E-mail: lgfalves@senacrs.com.br



{({({ >> })) << }

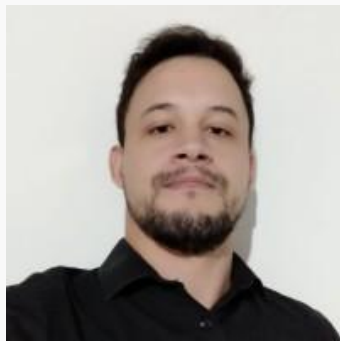


(({ >> 0 i □ □ □ }))

```
((: 00 - =>> } )  
{ (<1 00 1 000 >> )}  
((: 0)>"< )  
<01 001} +100 0}>  
((: 0)>"< )  
{ (<1 00 1 000 >> )}
```



Professor



Lucas G. F. Alves

