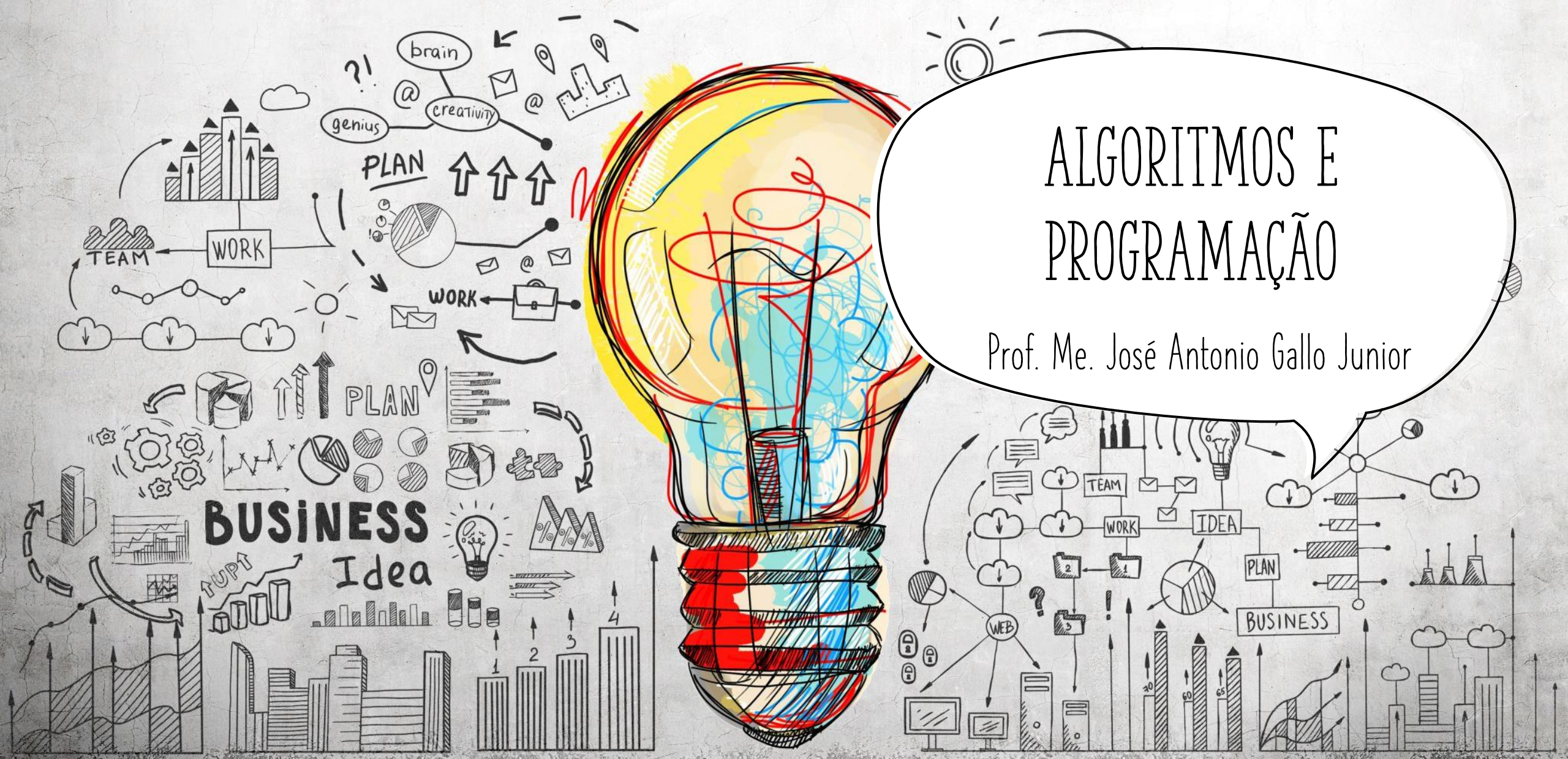


ALGORITMOS E PROGRAMAÇÃO

Prof. Me. José Antonio Gallo Junior



ALGORITMOS E PROGRAMAÇÃO - EMENTA

Vetores e matrizes.

Funções.

Definição e declaração de novos tipos de variáveis.

Leitura e escrita de arquivos.

Análise da complexidade de algoritmos.

Algoritmos de busca e de ordenação.

Ponteiros.

Implementação de algoritmos utilizando linguagens de programação estruturadas.

ALGORITMOS E PROGRAMAÇÃO - CRITÉRIOS DE AVALIAÇÃO

Prova - 6

Trabalhos e listas de exercícios - 3

Assiduidade - 1

PROGRAMAS DE COMPUTADOR

Os programas de computadores nada mais são do que algoritmos escritos numa linguagem de computador (C, C#, Java, Python, Javascript, entre outras) e que são interpretados e executados por uma máquina, no caso um computador. Notem que dada esta interpretação rigorosa, um programa é por natureza muito específico e rígido em relação aos algoritmos da vida real, ou seja, deve ser programado de maneira com que não haja problemas quanto as instruções escritas, podendo ocasionar em erro grave, comprometendo assim a execução do programa.

LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

ALGORITMOS

Um **algoritmo** é formalmente uma sequência finita de passos que levam a execução de uma tarefa. Podemos pensar em algoritmo como uma receita, uma sequência de instruções para um fim específico. Para escrever um algoritmo precisamos descrever a sequência de instruções, de maneira simples e objetiva.

COMO VOCÊ PREPARA UM CAFÉ?



ALGORITIMOS

A importância do algoritmo está no fato de termos que especificar uma sequência de passos lógicos para que o computador possa executar uma tarefa qualquer, pois o mesmo por si só não tem vontade própria, faz apenas o que mandamos. Com uma ferramenta algorítmica, podemos conceber uma solução para um dado problema, independentemente de uma linguagem específica e até mesmo do próprio computador.

ALGORITMOS

- Partir de um ponto inicial e chegar a um ponto final;
- Não ser ambíguo (ter dupla interpretação);
- Poder receber dados externos e ser capaz de retornar resultados aos mesmos;
- Ter suas etapas alcançáveis em algum momento do programa.

ALGORITMOS

- ENTRADA: São os dados de entrada do algoritmo
- PROCESSAMENTO: São os procedimentos utilizados para chegar ao resultado final
- SAÍDA: São os dados já processados



LINGUAGEM PORTUGOL

O Portugol é uma representação que se assemelha bastante com a linguagem C, porém é escrito em português. A ideia é facilitar a construção e a leitura dos algoritmos usando uma linguagem mais fácil aos alunos.



Portugol
Webstudio

VARIÁVEIS

Variável é um recurso utilizado nos programas computacionais para armazenar e recuperar dados, ou seja, é simplesmente um espaço na memória que reservamos atribuindo um **nome**, para que possamos organizar os dados à serem manipulados pelo programa e um **tipo**.

É importante ressaltar que o nome de cada variável deve ser explicativo, facilitando assim a compreensão do conteúdo que está armazenado nela.

A **sintaxe** para se declarar uma variável é o **tipo da variável**, o **nome da variável** ou das **variáveis** (separadas por vírgula cada uma) e opcionalmente pode ser atribuído a ela um **valor de inicialização** (exceto se for declarado mais de uma na mesma linha)

VARIÁVEIS

Exemplo de Sintaxe:

caracter nome_variavel

inteiro variavel_inicializada = 42

real nome_variavel2

logico nome_variavel3

cadeia var1, var2, var3, var4

logico var4, var5, var6

CONSTANTES

Uma **constante** é responsável por armazenar um valor fixo em um espaço da memória, sendo que este valor não se altera durante a execução do programa.

Para declarar uma **constante** basta adicionar a palavra reservada **const** seguida do tipo de dado, pelo nome da constante e atribuir um valor a ela.

Exemplo de Sintaxe:

```
const inteiro NOME_DA_CONSTANTE = 3
```

```
const real NOME_DA_CONSTANTE2 = 45
```


TIPOS DE DADOS

Ao armazenar variáveis na memória do computador, precisamos dizer que tipo elas são, para que seja reservado o espaço adequado, onde cada tipo define os valores que a variável pode armazenar e cada tipo ocupa uma certa quantidade de memória. Os tipos de dados influenciam na forma como o algoritmo irá trabalhar, o desempenho do algoritmo e o seu consumo de memória.

Existem alguns tipos básicos de dados nos quais valores podem ser armazenados no computador. O **Portugol** exige que o tipo de dado de um valor seja do mesmo tipo da variável ao qual este valor será atribuído.

TIPOS DE DADOS

cadeia: armazenar um texto ou uma quantidade grande de caracteres. Ex: cadeia nome = "Gallo"

caracter: armazena um caractere, que pode ser uma letra, número ou símbolo. Ex: caracter letra = 'a'

inteiro: armazena um valor inteiro. Ex: inteiro numero = 51

logico: armazena o valor verdadeiro ou falso. Ex: logico fimDaAula = falso

real: armazena um valor real. Ex: real preco = 1.99

vazio: usado em funções que não possui retorno. Ex: funcao vazio imprimeLinha() { }

ENTRADA E SAÍDA

Entrada/saída é um termo utilizado quase que exclusivamente no ramo da computação (ou informática), indicando **entrada** (inserção) de dados por meio de algum código ou programa, para algum outro programa ou hardware, bem como a sua **saída** (obtenção de dados) ou retorno de dados, como resultado de alguma operação de algum programa, conseqüentemente resultado de alguma entrada.

Entrada -> leia()

Saída -> escreva()

SAÍDA - ESCRIVA

Em determinadas situações precisamos mostrar ao usuário do programa alguma informação. Para isso, existe um comando na programação que exibe dados ao usuário. No **portugol** a instrução de **saída de dados** para a tela é chamada de "**escreva**", pois segue a ideia de que o algoritmo está escrevendo dados na tela do computador.

Para utilizar o comando **escreva**, você deverá escrever este comando e entre parênteses colocar a(s) **variável(eis)** ou **texto** que você quer mostrar no console.

SAÍDA - ESCRIVA

A sintaxe para utilização deste comando está demonstrada a seguir:

```
// escreve no console um texto qualquer
```

```
escreva("Oi mundo! ")
```

```
// escreve no console o valor de uma variável e "pula" a linha
```

```
cadeia mensagem = "Computação é coisa linda!!!!"
```

```
escreva(mensagem, "\n")
```


ENTRADA - LEIA

Em alguns problemas, precisamos que o usuário digite um valor a ser armazenado. Por exemplo, se quisermos elaborar um algoritmo para calcular a média de nota dos alunos, precisaremos que o usuário informe ao algoritmo quais as suas notas. No **portugol** a instrução de **entrada de dados** via teclado é chamada de "**leia**", pois segue a ideia de que o algoritmo está lendo dados do ambiente externo(usuário) para poder utilizá-los.

Para utilizar o comando **leia**, você deverá escrever este comando e entre parênteses colocar a(s) variável (eis) que você quer que recebam os valores a serem digitados.

ENTRADA - LEIA

A sintaxe deste comando está exemplificada a seguir:

inteiro x

cadeia y

real z

leia(x)

leia(y, z)

EXTRA - LIMPA O CONSOLE

À medida que um algoritmo está sendo executado ele exibe mensagens e executa ações no console. Assim, em alguns casos o console fica poluído com informações desnecessárias, que atrapalham a compreensão e visualização do programa. Para isso, podemos usar o comando limpa.

Sintaxe:

```
limpa()
```

EXPRESSÕES

Uma expressão em uma linguagem de programação é uma combinação de valores explícitos, constantes, variáveis, operadores e funções que são interpretados de acordo com as regras específicas de precedência e de associação para uma linguagem de programação específica, que calcula e, em seguida, produz um outro valor. Este processo, tal como para as expressões matemáticas, chama-se avaliação. O valor pode ser de vários tipos, tais como numérico, cadeia, e lógico.

EXPRESSÕES - OPERADORES

As expressões possuem os seguintes operadores:

- Operador de Atribuição
- Operações Aritméticas
- Operações Relacionais
- Operações Lógicas

OPERADOR DE ATRIBUIÇÃO

Quando criamos uma variável, simplesmente separamos um espaço de memória para um conteúdo. Para especificar esse conteúdo, precisamos de alguma forma determinar um valor para essa variável. Para isso, usamos a operação de atribuição.

A instrução de **atribuição** serve para **alterar o valor de uma variável**. Ao fazer isso dizemos que estamos atribuindo um novo valor a esta variável. A atribuição de valores pode ser feita de variadas formas. Pode-se atribuir valores através de constantes, de dados digitados pelo usuário (Leia) ou mesmo através de comparações e operações com outras variáveis já existentes.

OPERADOR DE ATRIBUIÇÃO

O sinal de igual "=" é o símbolo da **atribuição** no Portugol. A variável a esquerda do sinal de igual recebe o valor das operações que estiverem à direita. Exemplos:

variavel = 6

variavel = variavel2

variavel = 6 + 4 / variavel2

leia (variavel)

OPERAÇÕES ARITMÉTICAS

As operações aritméticas são nossas velhas conhecidas da **Matemática**. Em algoritmos é muito comum usarmos operadores aritméticos para realizar cálculos.

Os símbolos que usamos para os operadores na **Matemática** mudam um pouquinho em algoritmos. A multiplicação, que na matemática é um xis 'x' ou um ponto ".", torna-se um '*', justamente para não confundir com o xis que pode ser uma variável e com o ponto que pode ser a parte decimal de um número real.

OPERAÇÕES ARITMÉTICAS

OPERAÇÃO	SÍMBOLO	PRIORIDADE
Adição	+	1
Subtração	-	1
Multiplicação	*	2
Divisão	/	2
Resto da divisão inteira	%	2

OPERAÇÕES ARITMÉTICAS

A prioridade indica qual operação deve ser realizada primeiro quando houverem várias juntas. Quanto maior a prioridade, antes a operação ocorre.

Por exemplo: $6 + 7 * 9$

A multiplicação $7 * 9$ é feita antes pois a operação de multiplicação tem prioridade maior que a soma. O resultado deste cálculo será 69.

OPERAÇÕES RELACIONAIS

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

Os símbolos que usamos para os operadores também mudam um pouco em relação ao que usamos no papel.

OPERAÇÕES RELACIONAIS

OPERAÇÃO	SÍMBOLO
Maior	>
Menor	<
Maior ou igual	>=
Menor ou igual	<=
Igual	==
Diferente	!=

OPERAÇÕES RELACIONAIS

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

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 a 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**.

Os operadores lógicos no Portugal são: 'e', 'ou' e 'nao'

OPERAÇÕES LÓGICAS

OPERAÇÃO	RESULTADO
$5 > 3 \text{ e } 2 < 1$	falso
$1 > 3 \text{ ou } 1 \leq 1$	verdadeiro
$\text{nao } (8 < 4)$	verdadeiro

OPERAÇÕES LÓGICAS

Assim como as operações aritméticas, as operações lógicas também possuem prioridades. Veja a tabela abaixo:

OPERAÇÃO	PRIORIDADE
ou	1
e	2
nao	3



VAMOS PRÁTICA!?!?