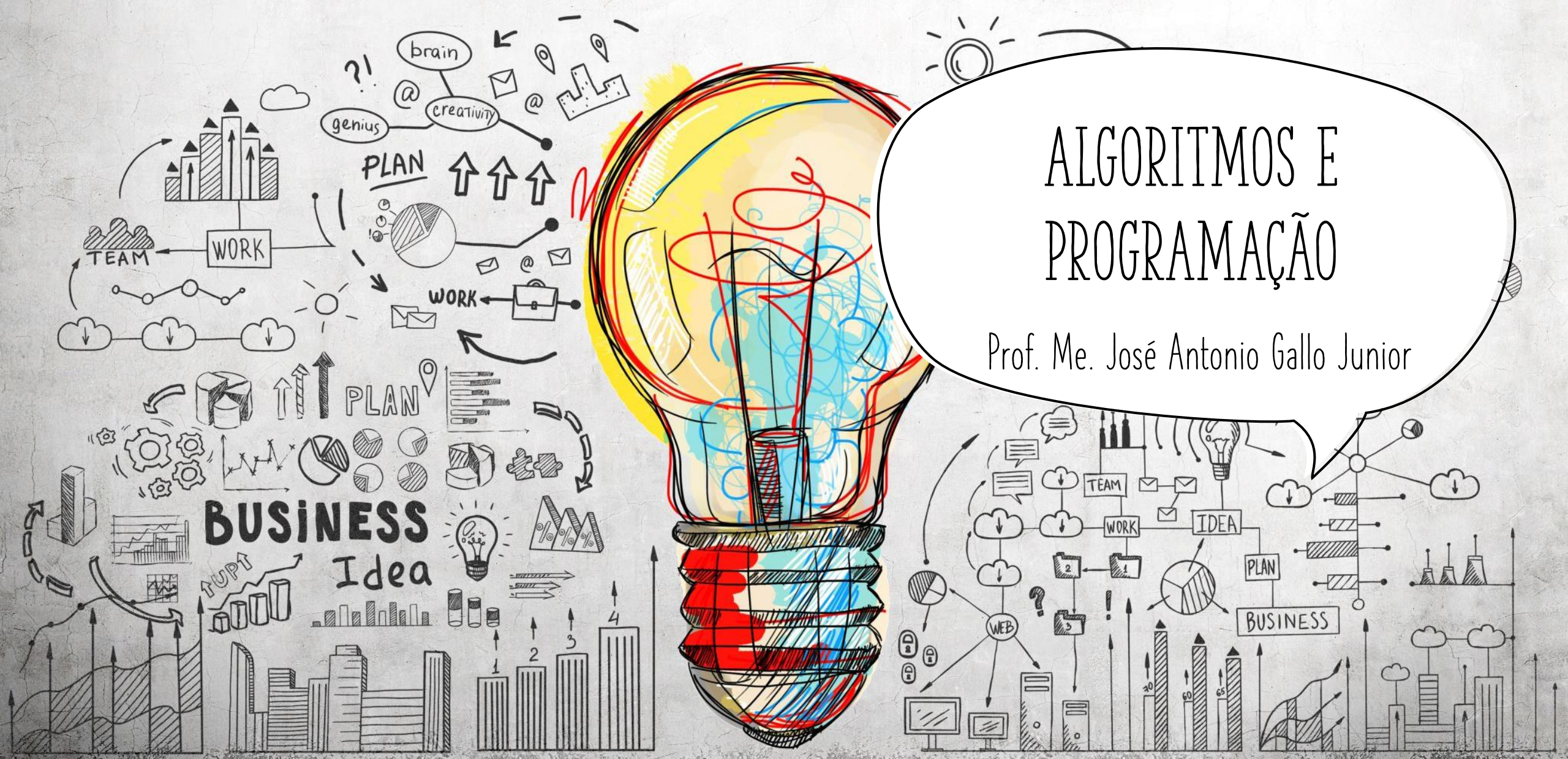


ALGORITMOS E PROGRAMAÇÃO

Prof. Me. José Antonio Gallo Junior



PROCEDIMENTOS E FUNÇÕES

Imagine que você precise escrever um algoritmo para preencher uma matriz.

Isso não é difícil. Mas e se fossem dez matrizes?

Escrever o mesmo código várias vezes deixaria o programa cansativo e repetitivo.

Para evitar isso, podemos escrever o procedimento apenas uma vez e reutilizá-lo sempre que necessário.

É para isso que servem as funções.

Uma função é um bloco de código que resolve um problema específico, fazendo parte de um problema maior.

PROCEDIMENTOS E FUNÇÕES

Vantagens do uso de funções:

- Evitam código repetido.
- Permitem reutilizar o mesmo código em outros programas.
- Dividem problemas grandes em partes menores e mais fáceis de entender.
- Deixam o programa mais organizado e legível.
- Escondem detalhes internos do funcionamento, mostrando apenas o necessário para quem usa a função.

PROCEDIMENTOS E FUNÇÕES

Componentes de uma função:

- Protótipo: define o nome da função e os parâmetros que ela recebe.
- Corpo: contém o bloco de comandos que resolve o problema.
- Valor de retorno (opcional): resultado que a função pode devolver para ser usado no programa.

PROCEDIMENTOS E FUNÇÕES

Declaração de função no Portugol:

- Usa-se a palavra reservada **funcao**, seguida do tipo de retorno.
- O tipo de retorno da função.
- Depois vem o **nome** da função e, entre parênteses, os parâmetros (se existirem).
- Em seguida, o corpo da função é escrito entre **chaves** { }.
- É obrigatório usar o comando **retorne** dentro do corpo da função para o tipo de retorno.

PROCEDIMENTOS E FUNÇÕES

Além das funções com retorno, também podemos criar **procedimentos**.

Um **procedimento** é muito parecido com uma função, mas não devolve nenhum valor ao final da sua execução.

Ele serve para executar uma tarefa específica dentro do programa, sem a necessidade de calcular ou retornar um resultado.

PROCEDIMENTOS E FUNÇÕES

Exemplos de uso:

- Mostrar uma mensagem na tela.
- Preencher uma matriz com valores digitados pelo usuário.
- Exibir um relatório com os dados cadastrados.

PROCEDIMENTOS E FUNÇÕES

Declaração de procedimento no Portugol:

- Usa-se a palavra reservada `funcao`.
- Em seguida a palavra `vazio`.
- Em seguida, o `nome do procedimento` e entre parênteses os `parâmetros` se existirem.
- O corpo do procedimento fica entre `chaves { }`.
- Como é um procedimento não precisa do comando `retorne` dentro do corpo.

PROCEDIMENTOS E FUNÇÕES

Diferenças principais entre função e procedimento:

- Função: executa uma tarefa e devolve um resultado (usa retorne).
- Procedimento: executa uma tarefa, mas não retorna nada. (possui vazio como retorno).

PROCEDIMENTOS E FUNÇÕES

Característica	Função	Procedimento
Retorno	Sempre pode retornar um valor (ex.: texto, resultado de cálculo).	Não retorna valor.
Uso principal	Calcular ou processar algo e devolver um resultado.	Executar uma tarefa sem necessidade de retorno.
Retorne	Obrigatório se houver retorno (ex.: retorne soma).	Não utiliza retorne.
Exemplo típico	Calcular a média de notas e devolver o resultado.	Mostrar mensagens na tela ou preencher uma matriz.

EXEMPLO DE FUNÇÃO

Imagine que você precise criar um algoritmo que faça uma pergunta ao usuário e a resposta deve ser "Sim" ou "Não". Agora imagine que você precisa fazer várias perguntas como essa durante a execução do algoritmo?

Seria necessário, um escreva para a pergunta, uma variável para resposta e um condicional para verificar se a resposta é "S" ou "N", e isso sendo repetido várias vezes.

Nessa situação, podemos simplificar nosso código com uma função, que já retorna verdadeiro ou falso.

EXEMPLO DE FUNÇÃO

```
funcao logico pergunta(cadeia texto) {  
    cadeia resposta  
    escreva(texto, " (S/N): ")  
    leia(resposta)  
    se (resposta == "S") {  
        retorne verdadeiro  
    } senao {  
        retorne falso  
    }  
}
```

EXEMPLO DE FUNÇÃO

```
funcao inicio() {  
    se (pergunta("Acabou a aula?")) {  
        escreva("Vamos embora")  
    } senao {  
        escreva("Trabalhemmmm")  
    }  
}
```



VAMOS PRATICAR!?!