

# Lógica de Programação

- **Funções**

- **Função é um recurso da programação que possui a mesma utilidade de um Procedimento e ainda é capaz de devolver um retorno à rotina chamadora.**

## Criação da Função

Funcao <nome-de-função> () : <tipo-de-dado>

// Seção de Declarações Internas

inicio

// Seção de Comandos

Retorne <Valor>

FimFuncao

- **Após a criação da função, a mesma pode ser chamada pela rotina principal ou ainda por outro procedimento ou função. A rotina chamadora é a responsável por tratar o retorno da função.**

# Lógica de Programação

## • Função – Exemplo

### Algoritmo Soma (Tradicional)

```
algoritmo "Soma"  
  
var  
    Numero1, Numero2, Resultado: real  
  
inicio  
    Leia(Numero1)  
    Leia(Numero2)  
    Resultado <- Numero1 + Numero2  
    Escreva(Resultado)  
  
fimalgoritmo
```

### Algoritmo Soma (Modularizado com Função)

```
algoritmo "Função Soma"  
  
Funcao Soma():real  
inicio  
    Retorne Numero1 + Numero2  
FimFuncao  
  
var  
    Numero1, Numero2, Resultado:real  
  
inicio  
    Leia(Numero1)  
    Leia(Numero2)  
    Resultado <- Soma()  
    Escreva(Resultado)  
  
fimalgoritmo
```

Criação da função

Retorno da função

Chamada da função

Captura do retorno da função

# Lógica de Programação

## **Exemplo da declaração da função**

Funcao RetornaTexto () : Caractere

// Seção de Declarações Internas

var

inicio

// Seção de Comandos

    Retorne "Texto retornado pela função"

FimFuncao

## **Exemplo de chamada da função por uma rotina principal (Rotina chamadora)**

Algoritmo "Exibe texto retornado pela função"

Var

    X : Caractere

Inicio

    X <- RetornaTexto

    Escreva(x)

FimAlgoritmo

# Lógica de Programação

- **Funções - Exercícios**

**Exercício 2: Evoluir o algoritmo de calculadora modularizado com procedimentos para que utilize funções.**