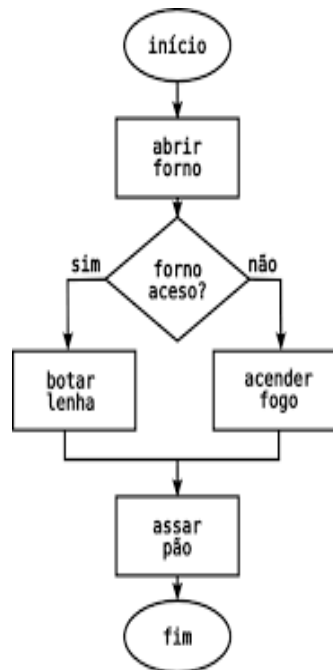


# Técnicas de Desenvolvimento de Algoritmos



- Raciocínio lógico;
- Noções de lógica de programação;
- Tipos de Algoritmos,
- Fluxogramas e Programas

**CLÓVIS JOSÉ RAMOS FERRARO**

cferraro@unicid.edu.br

# Questão 1

Descubra a lógica e complete o próximo elemento:

a) 1, 3, 5, 7, \_\_\_\_\_

b) 2, 4, 8, 16, 32, 64, \_\_\_\_\_

c) 0, 1, 4, 9, 16, 25, 36, \_\_\_\_\_

d) 4, 16, 36, 64, \_\_\_\_\_

e) 1, 1, 2, 3, 5, 8, \_\_\_\_\_

f) 2, 10, 12, 16, 17, 18, 19, \_\_\_\_\_

# Questão 2

(FGV/TCE-SE) Duas tartarugas estavam juntas e começaram a caminhar em linha reta em direção a um lago distante. A primeira tartaruga percorreu 30 metros por dia e demorou 16 dias para chegar ao lago. A segunda tartaruga só conseguiu percorrer 20 metros por dia e, portanto, chegou ao lago alguns dias depois da primeira. Quando a primeira tartaruga chegou ao lago, o número de dias que ela teve que esperar para a segunda tartaruga chegar foi:

- a) 8
- b) 9
- c) 10
- d) 12
- e) 15

# Questão 3

(UERJ) Em um sistema de codificação, AB representa os algarismos do dia do nascimento de uma pessoa e CD os algarismos de seu mês de nascimento. Nesse sistema, a data trinta de julho, por exemplo, corresponderia a:

$$A=3 \ B=0 \ C=0 \ D=7 \quad \text{Total}=3+0+0+7=10$$

Admita uma pessoa cuja data de nascimento obedeça à seguinte condição:

$$A+B+C+D=20$$

O mês de nascimento dessa pessoa é:

- a) agosto
- b) setembro
- c) outubro
- d) novembro

# Conceitos Básicos

## ✓ Raciocínio Lógico:

- **Pensar** antes de agir.
- Entender o problema proposto e suas possíveis **soluções**.
- Pensar com coerência sobre os vários caminhos possíveis e tomar a **melhor decisão** para alcançar o objetivo.

# Conceitos Básicos

## ✓ Raciocínio Lógico

Para usar o raciocínio lógico, é necessário ter:

- Domínio do pensar,
- Saber pensar.

Ou seja:

Possuir e usar a “Arte de pensar”

# Conceitos Básicos

## ✓ Lógica:



- A lógica teve sua formulação original feita por Aristóteles (384 BC – 322 BC), filósofo grego.
- Talvez por necessidade de encontrar um meio preciso para contrapor argumentações e responder às questões de seus adversários. A lógica tem como objeto de estudo o **pensamento** e os processos da razão.

# Conceitos Básicos

## ✓ Lógica:

- Usamos lógica para Pensar, Escrever e Falar.
- No exemplo “clássico”:

Todos os homens são mortais.

Sócrates é homem.

Logo, Sócrates é mortal.

A lógica é imprescindível para a elaboração de algoritmos



# Conceitos Básicos

✓ Lógica de programação:

- Transferência do pensar humano para a máquina.



# Conceitos Básicos

- ✓ Nós utilizamos a **lógica para resolver** um problema;
- ✓ A resolução do problema é o **objetivo**;
- ✓ Para chegar ao objetivo é necessário **seguir uma sequência**;
- ✓ Nesta sequência estão contido **instruções**.
- ✓ Sendo assim, tudo isso se resume em:

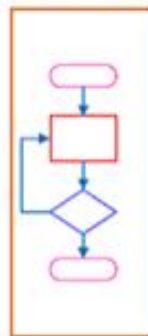


# Conceitos Básicos

O algoritmo não é a  
solução do problema,  
mas o caminho que leva a  
solução.



Algoritmo



Solução!



# Conceitos Básicos

- ✓ No desenvolvimento de ALGORITMOS deve-se seguir algumas premissas:
  - Ter um **COMEÇO** e um **FIM**, as ações devem ter uma sequência **finita** de passos;
  - As instruções devem ser organizadas em uma **sequência lógica**;
  - As instruções devem ser descritas de forma precisa, simples e sem **ambiguidades**.

# Conceitos Básicos

Possível resolução do algoritmo “Fritar ovos”:

- ✓ Separar ingredientes
  - Manteiga
  - Sal
  - ovos
- ✓ Separar utensílios
  - colher
  - Frigideira
- ✓ Preparar ambiente
  - Acender o fogo
  - Colocar a frigideira no fogão
- ✓ Modo de preparo
  - Colocar a manteiga na frigideira, depois junte os ovos, coloque sal e mexa.
  - Quando os ovos fritarem, desligue o fogo.



# Conceitos Básicos

## Alguns problemas com o algoritmo do “Fritar ovos”

✓ A linguagem natural é imprecisa:

- Quanto de manteiga?
- Quanto de Sal?
- Quantos ovos?
- Como mexer?
- O que define que está pronto?



- ✓ O uso de uma linguagem mais restritiva e estruturada seria mais adequado para certas situações;
- ✓ Sem ambiguidades (possíveis entendimentos diferentes);
- ✓ Utilizar uma linguagem simples e genérica.

# Conceitos Básicos

Veja o algoritmo do “Fritar ovos” de forma mais definida:

- separar 2 ovos;
- separar 100g de manteiga;
- Separar 5g de sal;
- separar frigideira;
- separar colher de pau;
- acender fogo do fogão;
- colocar frigideira no fogo;
- colocar manteiga na frigideira;
- deixar a manteiga derreter;
- colocar o sal;
- quebrar levemente a casca do primeiro ovo;
- colocar a clara e a gema na frigideira;
- repetir os dois passos anteriores para o segundo ovo;
- mexer os ovos com a colher;



# Conceitos Básicos

## ✓ Instrução de um Algoritmo:

- Cada linha do algoritmo pode ser chamada de **instrução**;
- Logo, pode-se dizer que um algoritmo é um **conjunto de instruções**;
- Uma instrução indica a um computador uma **ação elementar a ser executada**.



# Conceitos Básicos

- ✓ Qual a conclusão que podemos tirar sobre os algoritmos?
  - Algoritmos é uma espécie de **passo a passo**.
  - Algoritmos realizam uma saída (ex. Um ovo frito) a partir de uma entrada (um ovo) através de uma **sequencia**.
  - Os passos tem que ser executados **um após o outro**.
  - Um algoritmo está correto quando sua sequência de instrução resulta em uma **saída esperada**.
  - Podem existir um ou mais algoritmos para atingir um resultado, desde que o **resultado seja o mesmo**.

# Conceitos Básicos

- ✓ As principais tarefas realizadas por um Algoritmo são:
  - Ler e escrever dados;
  - Tomar decisões com base nos resultados das expressões avaliadas;
  - Repetir um conjunto de ações de acordo com uma condição.

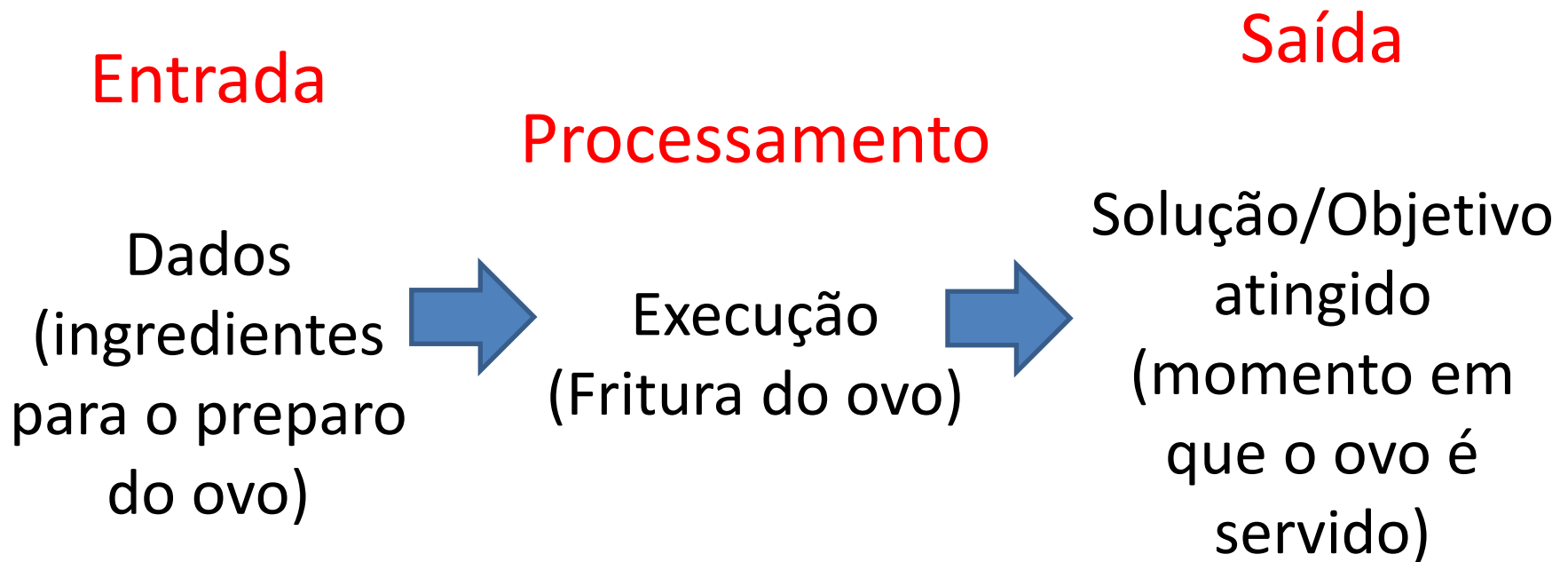
# Conceitos Básicos

## ✓ Fases de um Algoritmo:

Ao montar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais:

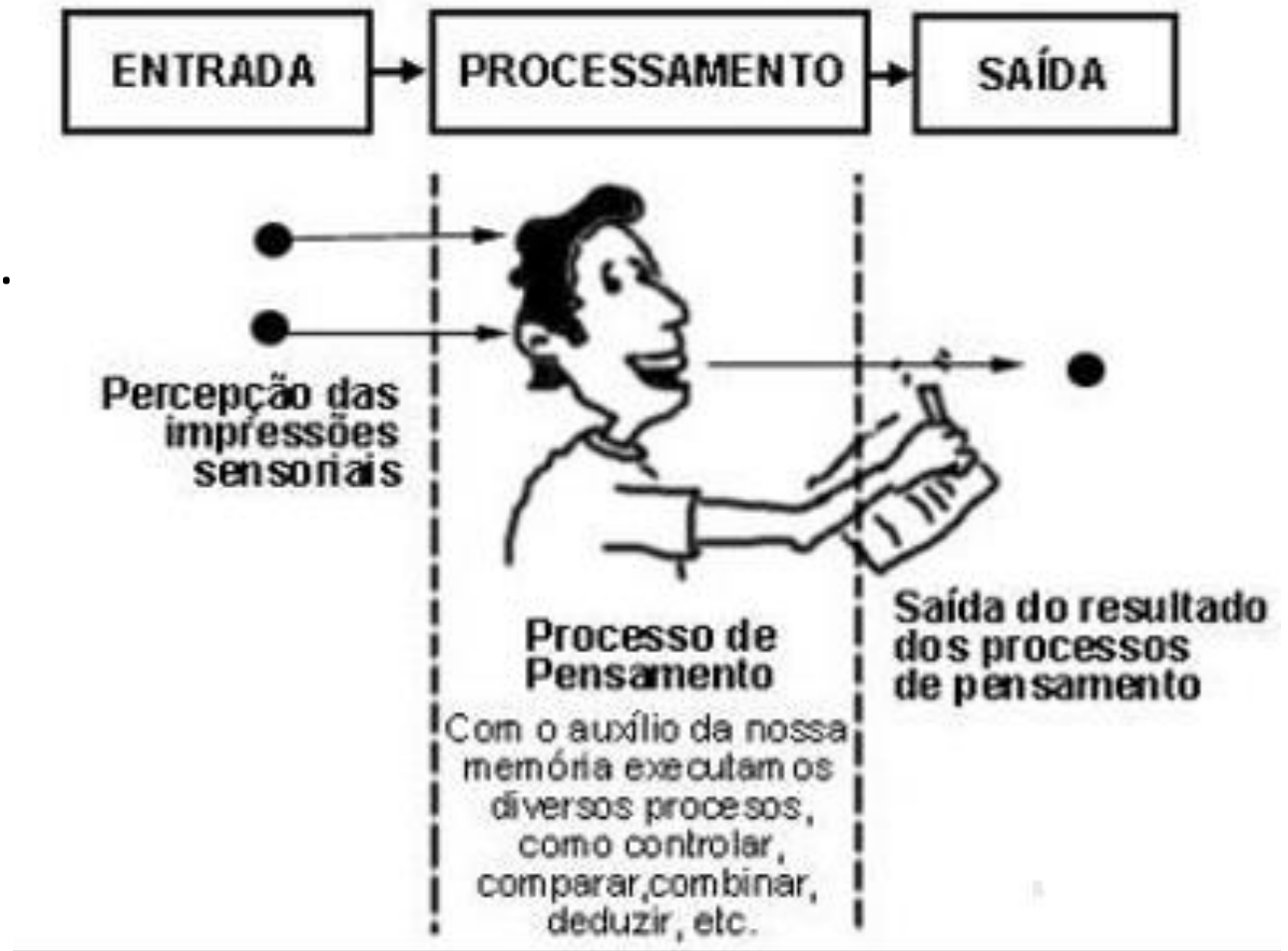
- **ENTRADA:** são os dados de entrada;
- **PROCESSAMENTO:** são os procedimentos utilizados para se chegar ao resultado final;
- **SAÍDA:** são os dados já processados, gerando uma informação.

# Processamento de dados



# Processamento de Dados

Forma humana ...



# Processamento de Dados

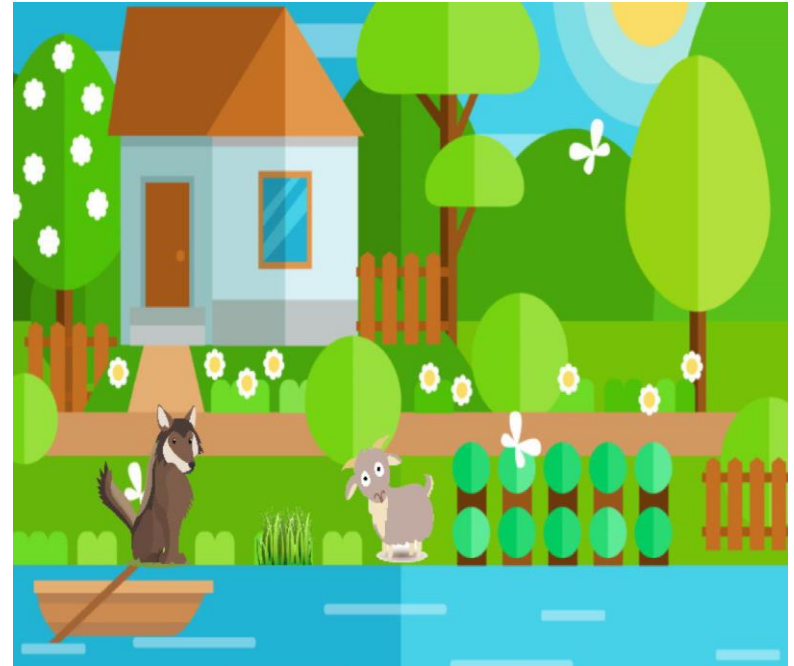
## Representação do funcionamento do computador



# Exercício de Lógica

## Exercício:

Um homem precisava atravessar um rio de uma margem para outra, com um barco que possui capacidade apenas para carregar ele mesmo e mais uma de suas três cargas: um lobo, um bode e um maço de capim. O que o homem deve fazer para conseguir atravessar o rio sem perder nenhuma de suas cargas? Escreva um algoritmo mostrando a resposta, ou seja, indicando todas as ações necessárias para efetuar uma travessia.

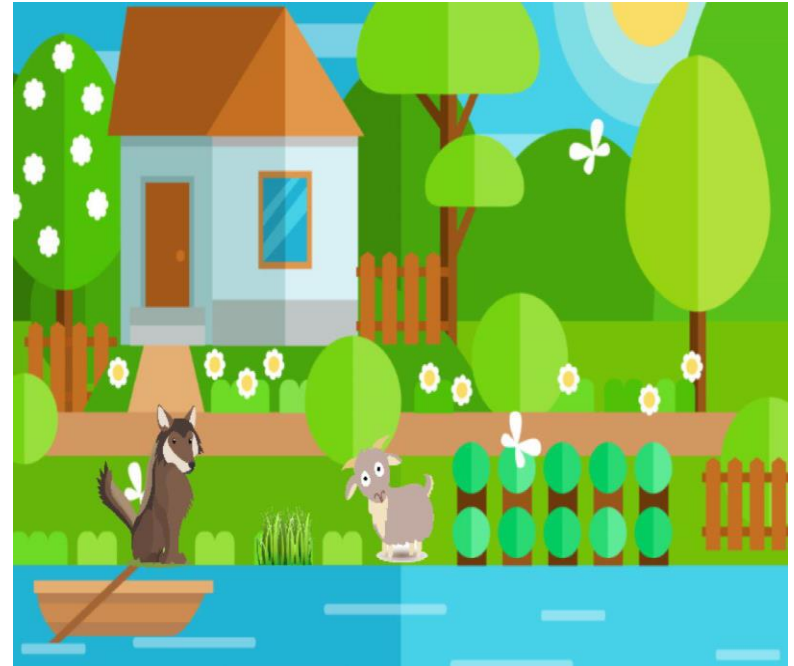


## Regras:

- O bode não pode ficar com o capim
- O lobo não pode ficar com o bode

# Resolução do exercício de Lógica

1. LEVAR O BODE PARA O OUTRO LADO DO RIO
2. VOLTAR SEM CARGA NENHUMA
3. LEVAR O LOBO PARA O OUTRO LADO DO RIO
4. VOLTAR COM O BODE
5. LEVAR O MAÇO DE CAPIM PARA O OUTRO LADO DO RIO
6. VOLTAR SEM CARGA ALGUMA
7. LEVAR O BODE PARA O OUTRO LADO DO RIO





# Exemplo das fases no Algoritmo

Calcule a média final dos alunos, sabendo que os alunos realizarão 4 provas: P1, P2, P3 e P4.

Para montar o algoritmo proposto, faremos 3 perguntas:

a) Quais são os dados de **ENTRADA**?

Resposta: P1, P2, P3 e P4.

b) Qual será o **PROCESSAMENTO** a ser utilizado?

Resposta: somar os dados de entrada e depois dividir o resultado da soma por 4.

c) Quais serão os dados de **SAÍDA**?

Resposta: será a média final.

# Exemplo das fases no Algoritmo

## Construindo o algoritmo...

- Ler a nota da prova 1.
- Ler a nota da prova 2.
- Ler a nota da prova 3.
- Ler a nota da prova 4.
- Somar todas as notas e dividir o resultado por 4.
- Exibir o resultado da operação anterior.

# Conceitos Básicos

## Introdução

### **O que é Programação Estruturada?**

A programação estruturada é uma técnica de programação, independente da linguagem de programação, que tem como objetivo construir programas claros, legíveis, eficientes e de fácil manutenção.

# Conceitos Básicos

## Programas de computador

Um programa é um conjunto de instruções escrito em uma determinada linguagem que diz ao computador o que deve ser feito. Existem muitas formas e diferentes tipos de linguagens de programação, cada qual com uma finalidade específica.

As linguagens podem ser classificadas em níveis, que vão desde o nível de dispositivo e lógico digital (conhecido como nível das portas lógicas, formadas pelos transistores e demais componentes de hardware), chamados de **baixo nível**, até o de linguagem orientada a problemas, chamado de **alto nível**.

*As linguagens de alto nível estão associadas ao fato de que nelas utilizam-se linguagens que procuram aproximar-se das linguagens naturais, usadas pelas pessoas.*

# Conceitos Básicos

## Programas de computador

*As linguagens de **alto nível** estão associadas ao fato de que nelas utilizam-se linguagens que procuram aproximar-se das linguagens naturais, usadas pelas pessoas.*

*Já as linguagens de **baixo nível** estão associadas ao fato de que nelas utilizam-se linguagens que procuram aproximar-se das linguagens máquina, ou seja, bit's e bytes.*

# Programas de computador

A maioria dos computadores modernos usam dois ou mais níveis de programação, sendo que, os programas escritos em linguagem de alto nível precisam ser traduzidos para o nível mais baixo, de forma que possam ser “entendidos” ou executados pela máquina.

# Conceitos Básicos

## Linguagem de programação

A linguagem de programação pode ser entendida como um conjunto de palavras (vocabulário) e um conjunto de regras gramaticais (para relacionar essas palavras) usados para instruir o sistema de computação a realizar tarefas específicas e com isso, criar os programas.

# Estrutura de Sequência

A estrutura de sequência define que as instruções do programa são executadas sequencialmente, de cima para baixo, linha a linha, do programa, de forma sequencial.

...

instrução 1

instrução 2

instrução 3

instrução N

...



## Estrutura de Seleção

A estrutura de seleção permite que o fluxo de execução das instruções seja executado de acordo com uma condição lógica que é avaliada e caso seja verdadeira permite que uma ou um grupo de instruções seja executado.

Se <condição for verdadeira> Então

instrução 1

instrução 2

instrução N

FimSe

## Estrutura de Repetição

Esta estrutura de controle de repetição permite que um grupo de instruções seja executado repetidamente de acordo com uma condição lógica.

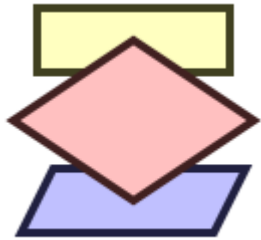
Enquanto <condição for verdadeira> Faça

instrução 1

instrução 2

instrução N

FimEnquanto



# Flowgorithm

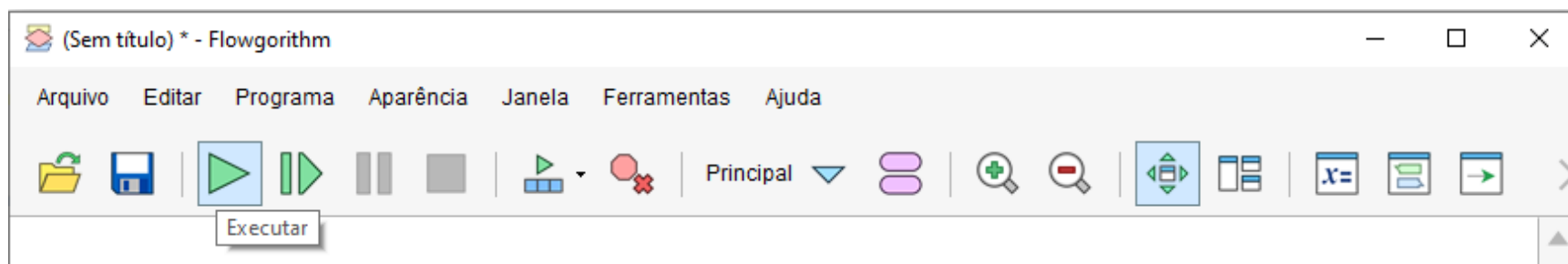
Ferramenta de desenvolvimento

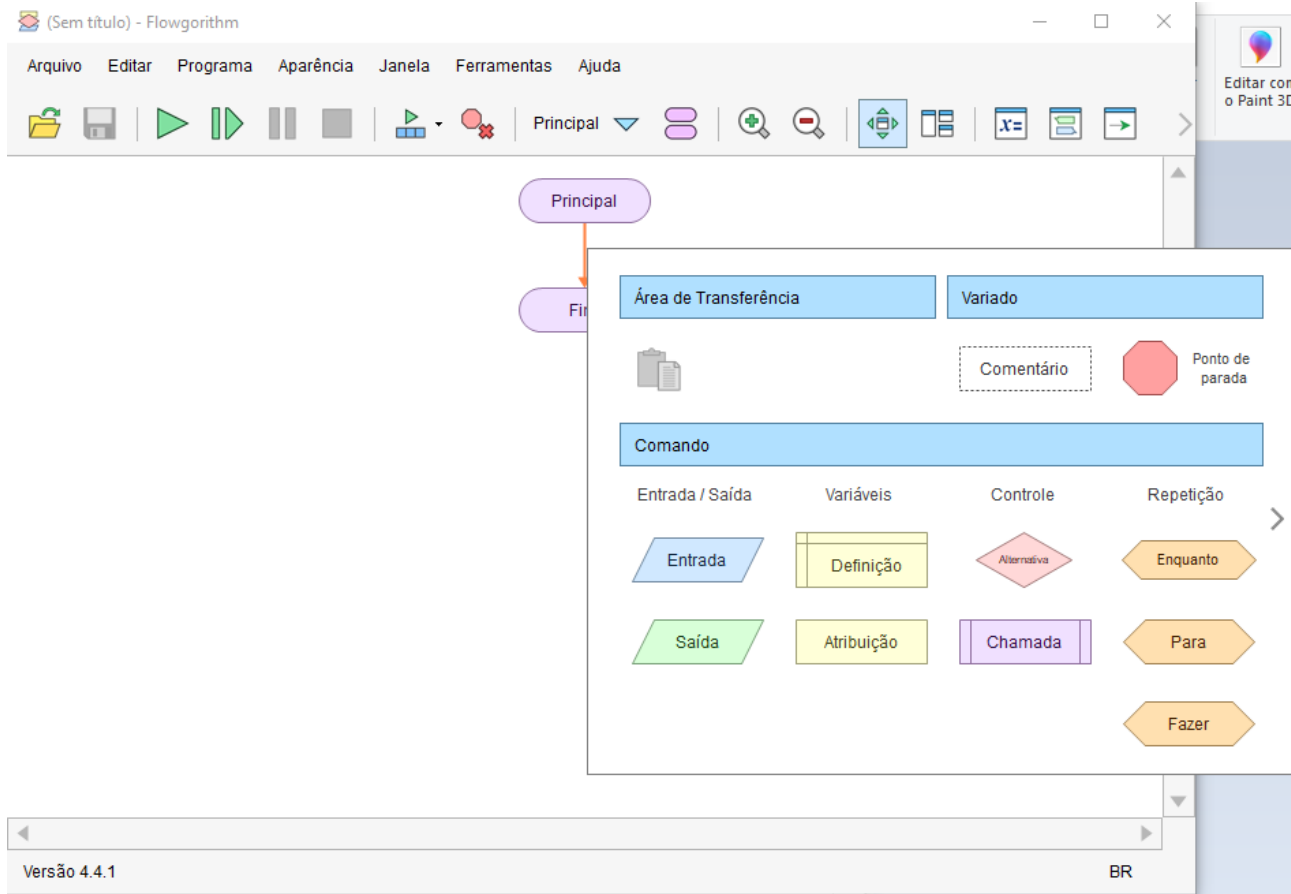
Fluxograma

# Diagrama de Blocos (Fluxograma)

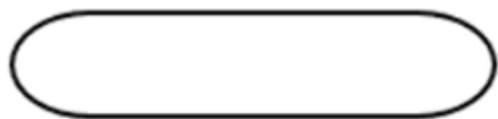
Diagrama de blocos é um conjunto de símbolos gráficos, onde cada um desses símbolos representa ações específicas a serem executadas pelo computador. Determina a linha de raciocínio utilizada pelo programador para resolver problemas.







# Símbolos utilizados em um diagrama de blocos/fluxograma utilizando Flowgorithm



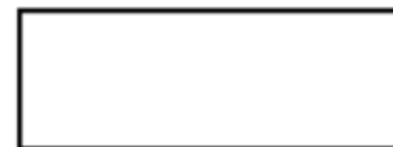
## Terminal

Representa o início ou o fim de um fluxo lógico. Em alguns casos define as sub-rotinas.



## Entrada manual

Determina a entrada manual dos dados, geralmente através de um teclado.



## Processamento

Representa a execução de ações de processamento.

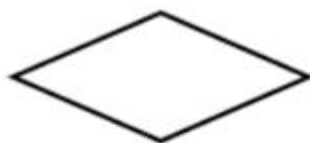


# Símbolos utilizados em um diagrama de blocos/fluxograma utilizando Flowgorithm



## Exibição

Mostra o resultado de uma ação, geralmente através da tela de um computador.



## Decisão

Representa os desvios condicionais nas operações de tomada de decisão e laços condicionais para repetição de alguns trechos do programa.



## Preparação

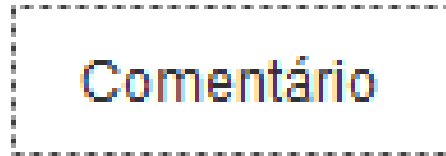
Representa a execução de um laço incondicional que permite a modificação de instruções do laço.

# Símbolos utilizados em um diagrama de blocos/fluxograma utilizando Flowgorithm

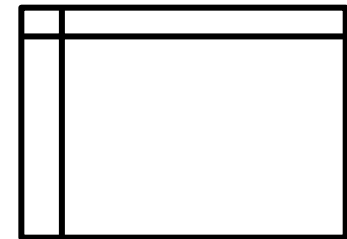


Símbolo de Subrotina

Normalmente usado para chamar uma subrotina.

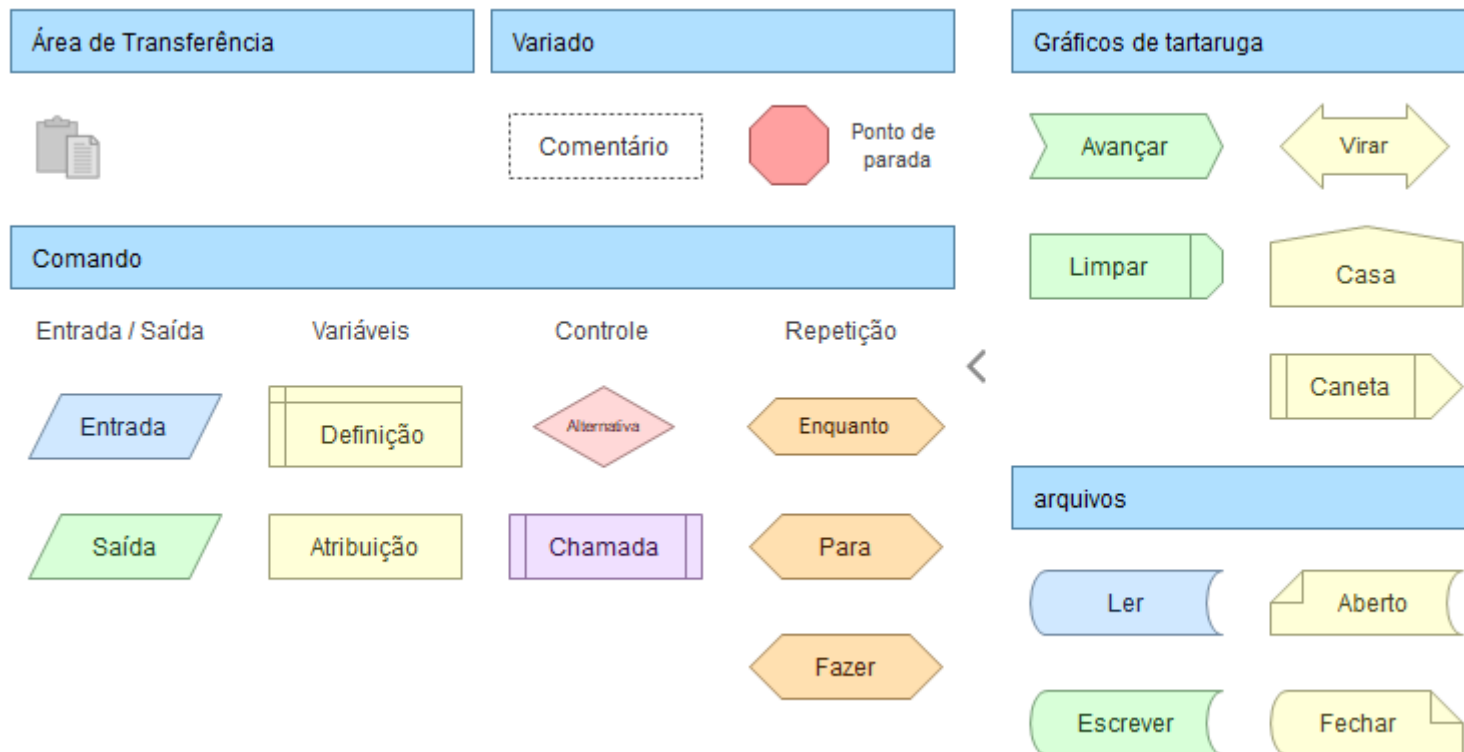


Comentário

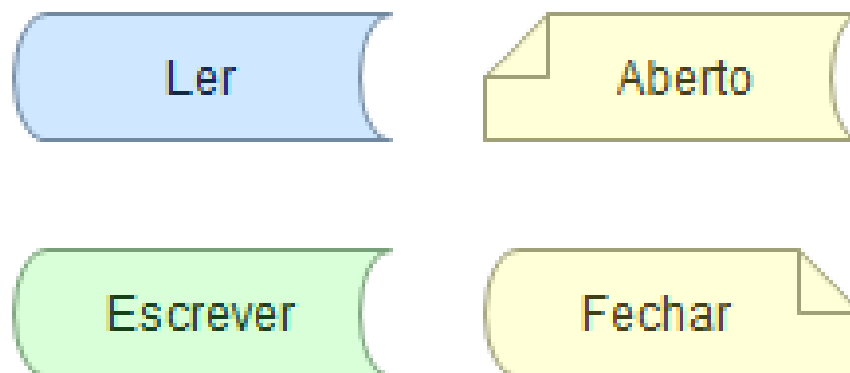


Símbolo de Armazenamento Interno

Normalmente usado nos fluxogramas de design de software e indica que os dados / informações são armazenados na memória durante um programa.



# Símbolos utilizados em um diagrama de blocos/fluxograma utilizando Flowgorithm



## Símbolos de Manipulação de um arquivo

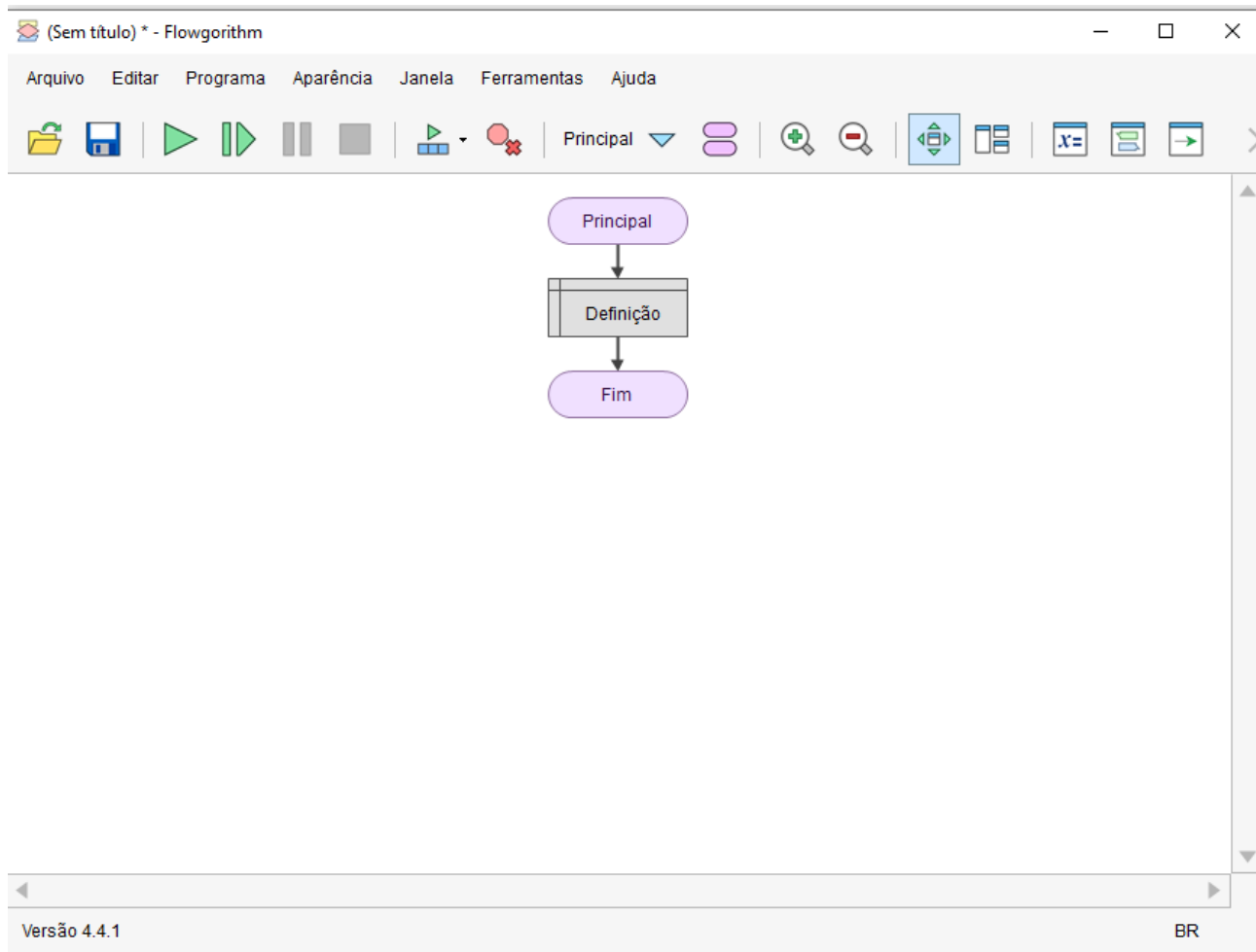
Aberto - abertura de arquivos;

Fechar – fechar um arquivo;

Ler – Leitura de um registro de um arquivo

Escrever – gravação de um registro em um arquivo

# Definindo uma variável



# Declaração de Variáveis

Todas as variáveis **devem ser declaradas** *antes* de serem usadas;

Não há uma inicialização implícita na declaração;

# Variáveis, Constantes e Atribuições

As **variáveis** como o próprio nome sugere, é algo que pode sofrer variações, ou seja, estão relacionadas a identificação de uma informação.

Exemplos: valor1, nome.

Constantes são identificadores que não podem ter seus valores alterados durante a execução de um programa.

Exemplo: o número Pi

A **atribuição** (= ) tem a função de indicar valores para as variáveis, ou seja, atribuir informação para variável. Exemplos:

Valor1 = 8

nome = "marcio"

Significa que o número "8" está sendo atribuído para variável "valor1" e que o texto "marcio" está sendo atribuído para variável "nome".

# Nomes de Variáveis

As variáveis no C podem ter qualquer nome se duas condições forem satisfeitas: o nome deve começar com uma letra ou sublinhado (\_) e os caracteres subsequentes devem ser letras, números ou sublinhado (\_).

Restrições:

- o nome de uma variável não pode ser igual a uma palavra **reservada**, nem igual ao nome de uma **função**.
- Variáveis de até 32 caracteres são aceitas.
- O C é "**case sensitive**" e portanto deve-se prestar atenção às maiúsculas e minúsculas.



# Tipos de variáveis

O uso de variáveis é um dos fundamentos básicos da programação. Temos vários tipos destas, é necessário compreender como utilizar cada tipo.

# ***INTEIRO (int)***

*O tipo inteiro compreende números sem casas decimais (“não quebrados”, ex: 1.01) positivos e negativos. EX: 1,2,3*

# ***REAL (float, double)***

*Compreende números com casas decimais(ex: 1.01).*

# Uma observação sobre a escolha do tipo de variáveis

Exemplos:

A variável idade será do tipo INTEIRO pois não existe idade 10.5;

A variável altura será REAL pois alturas pode ser 1,60 ou 1.

# ***CARACTERE (char)***

*Compreende qualquer tipo de dado porém entende como texto, ou seja caso seja números não terá nenhum tipo de valor aritmético.*

# ***LÓGICO (bool)***

*Compreende valores lógicos(verdadeiro e falso)*

FUNDATEC - 2023 Na lógica de programação, um tipo de dado \_\_\_\_\_ é caracterizado como um número que possui parte decimal, podendo ser positivo, negativo ou zero.

Assinale a alternativa que preenche corretamente a lacuna do trecho acima.

<b>A</b> Real	<b>B</b> Inteiro
<b>C</b> Cartesiano	<b>D</b> Booleano
<b>E</b> flutuante	

- [CESPE / CEBRASPE - 2024](#) Assinale a opção que apresenta a representação correta de instruções, em pseudocódigo tipado, para definir uma variável e modificar e imprimir seu conteúdo em uma saída.

<b>A</b> variavel = 0; variavel = 1 + 2; imprime(variavel);	<b>B</b> inteiro variavel = 0; variavel = 1 + 2; imprime(inteiro);
<b>C</b> inteiro variavel = 0; variavel = "1" + "2"; imprime(variavel);	<b>D</b> inteiro variavel = 0; variavel = 1 + 2; imprime(variavel);
<b>E</b> inteiro variavel = "0"; variavel = 1 + 2; imprime(variavel);	



COMVEST UFAM – 2023. Considere o algoritmo a seguir, escrito em uma variação simplificada de PORTUGOL:

algoritmo Valores

inteiros A,B,C

leia(A)

leia(B)

C = A

A = B

B = C

imprima(A)

imprima(B)

fim\_algoritmo

Testando para os valores de entrada 4, para a variável A, e 8 para a variável B, os valores de saída são:

Definição - Propriedades

Definição

Um Comando de Definição é usado para criar variáveis e arranjos. Esses são usados para armazenar dados enquanto o program estiver em execução.

Nomes de Variáveis:

Tipo:

☒ Inteiro

☐ Real

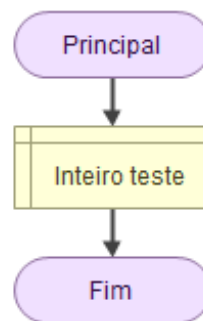
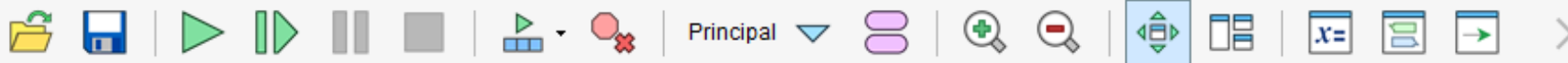
☐ Caracteres

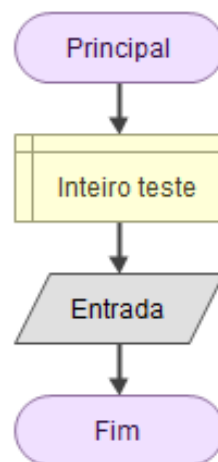
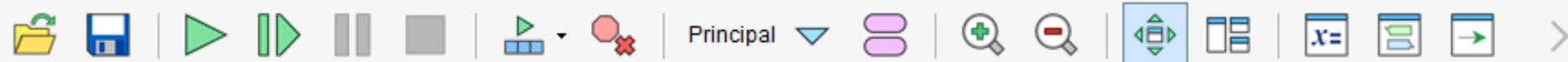
☐ Lógico

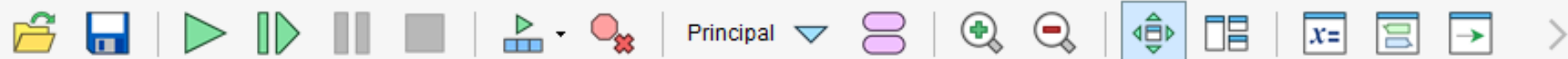
☐ Arranjo?

OK

Cancelar







Principal

### Entrada - Propriedades

Entrada

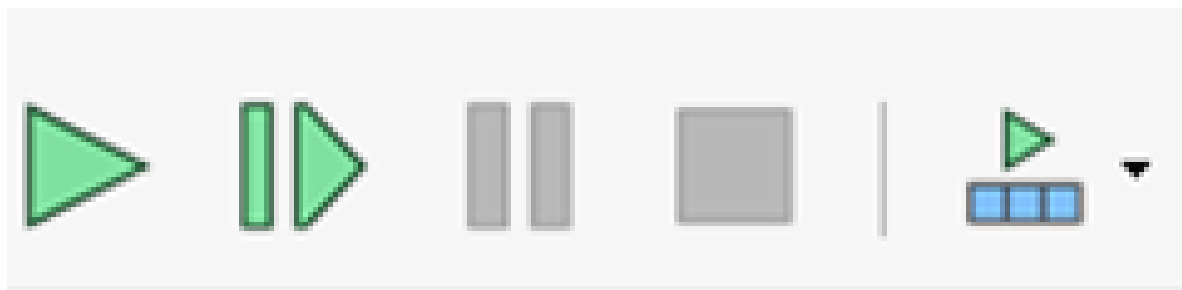
Um Comando de Entrada lê um valor do teclado e armazena o resultado em uma variável.

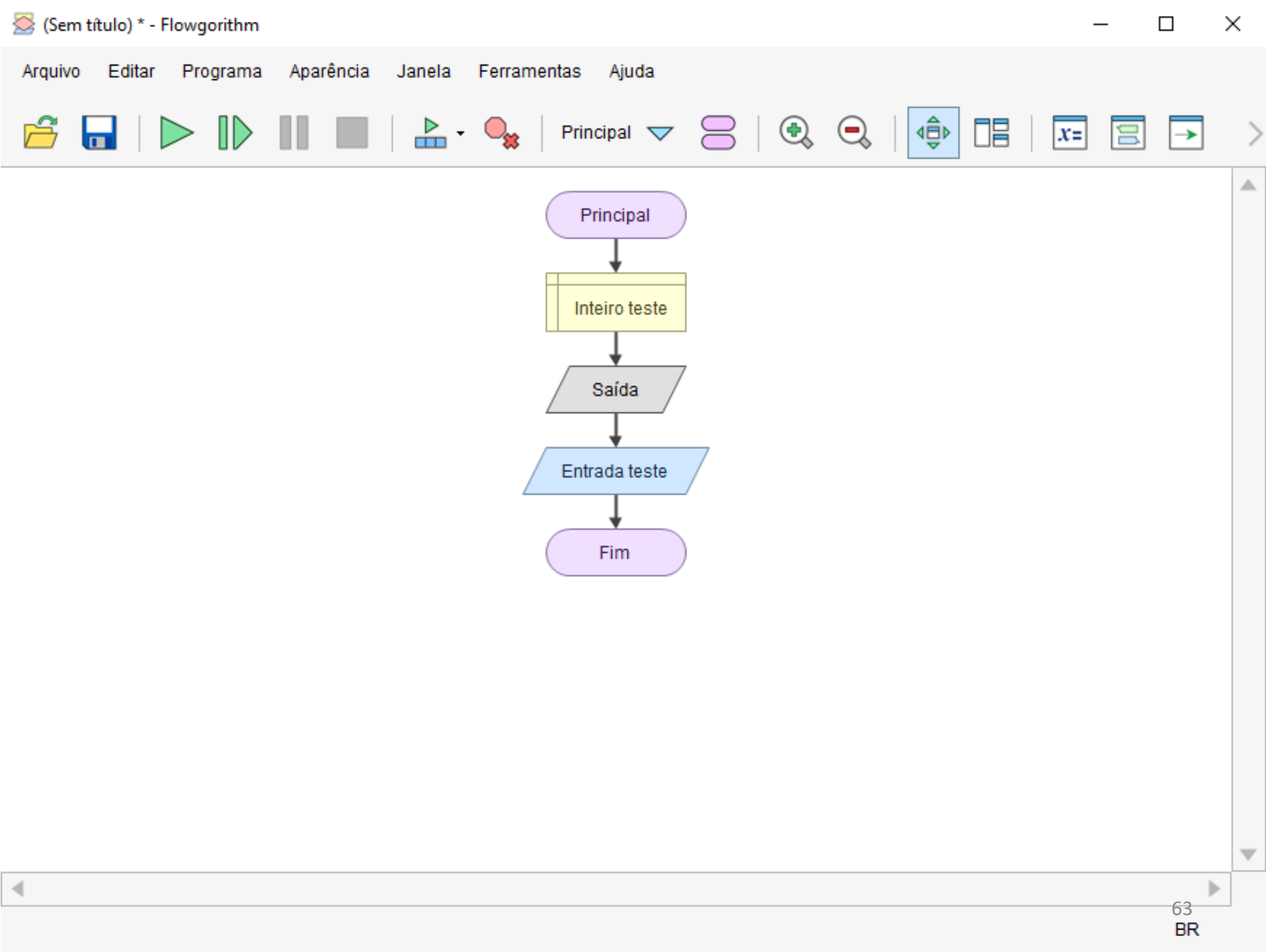
Fornecer um nome de variável abaixo:

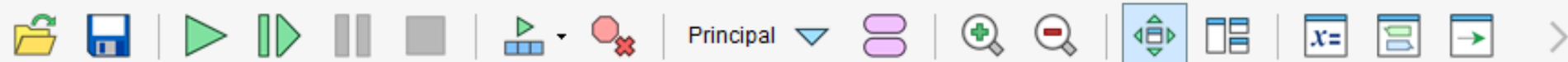
teste

OK

Cancelar







Saída - Propriedades

**Saída** Um Comando de Saída avalia uma expressão e mostra o resultado na tela.

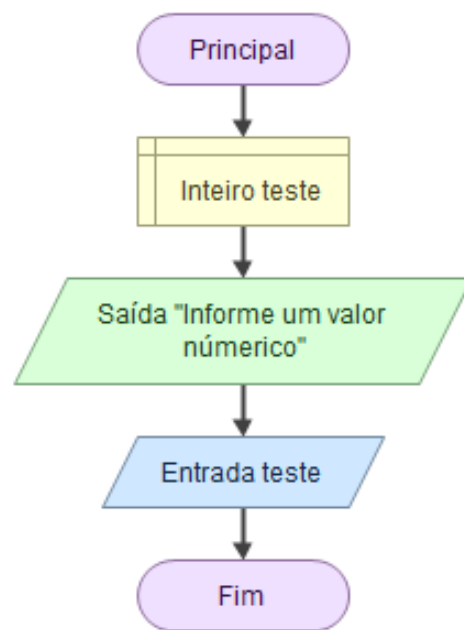
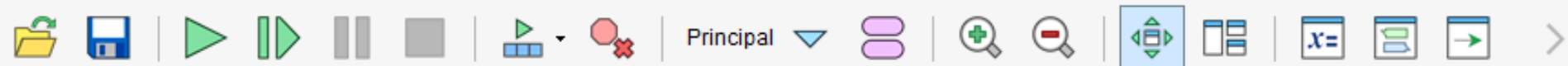
Fornecer uma expressão abaixo:

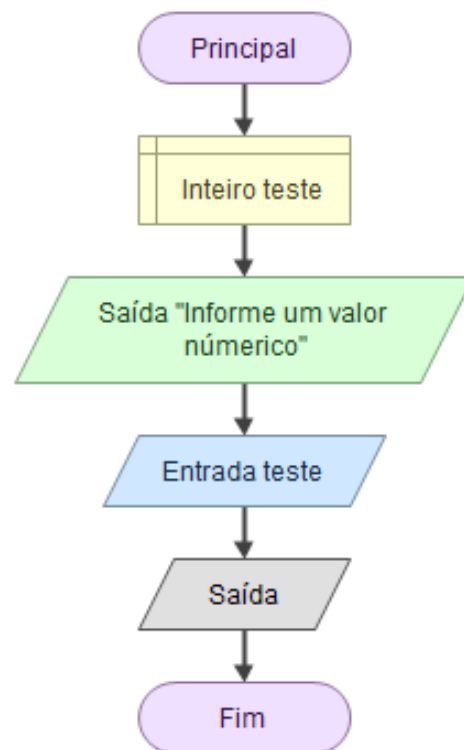
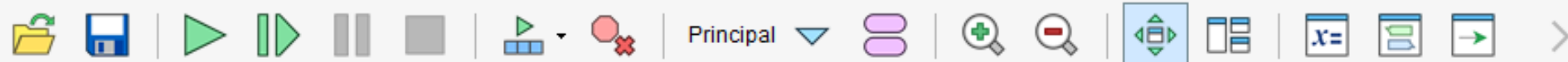
"Informe um valor numérico"

☒ Nova linha

OK Cancelar







## Saída

Um Comando de Saída avalia uma expressão e mostra o resultado na tela.

Fornecer uma expressão abaixo:

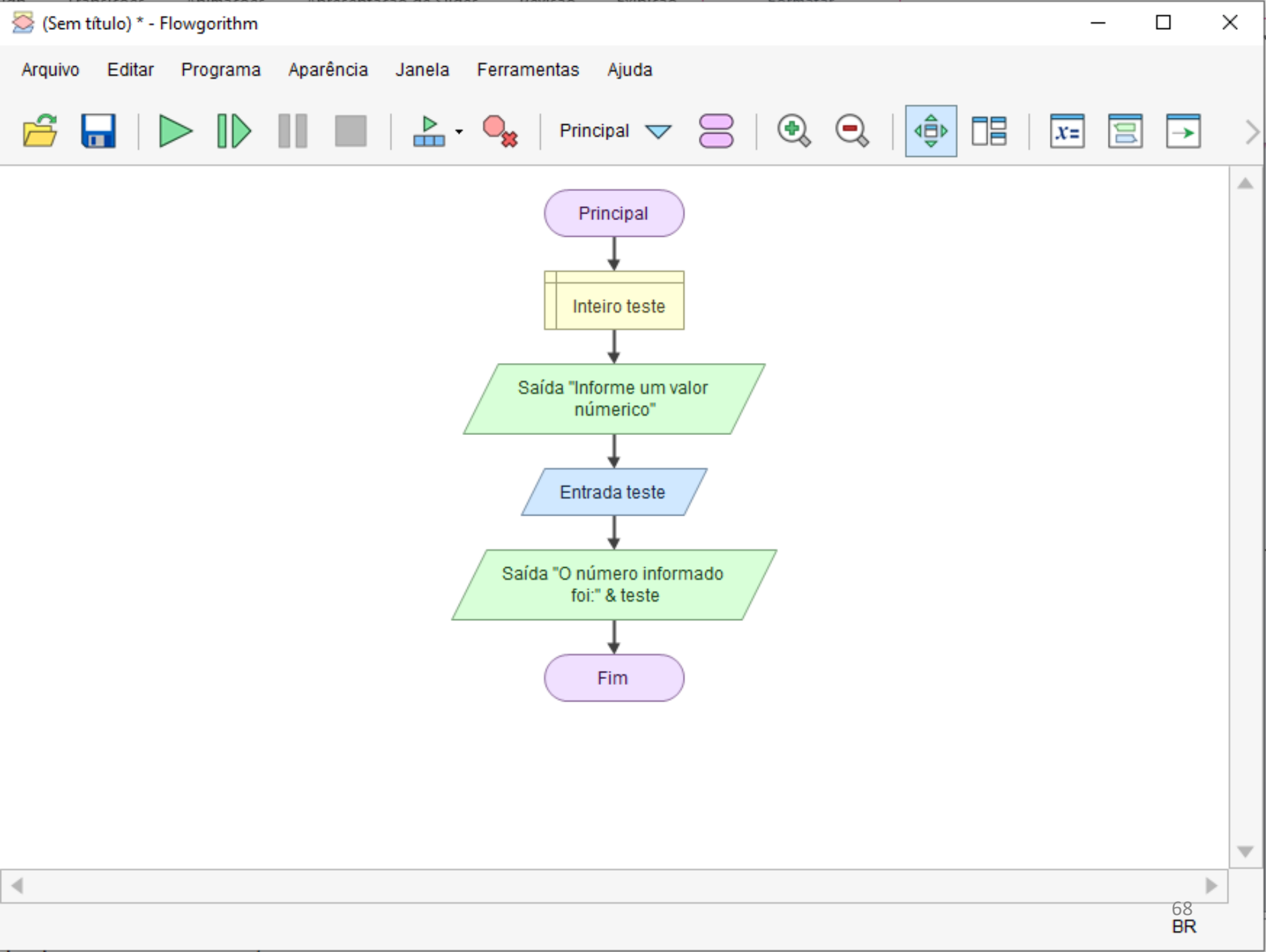
"O número informado foi:" & teste|



Nova linha

OK

Cancelar



# Atividade 1

Elabore um fluxograma que leia o nome de uma pessoa, na sequência imprima uma saudação seguida do nome informado

# Operadores matemáticos

Os principais operadores aritméticos são descritos abaixo.  
Considere que  $A=10$  e  $B=20$ .

Operador	Descrição	Exemplo	Resultado
+	soma	$A + B$	30
-	subtração	$A - B$	-10
*	multiplicação	$A * B$	200
/	divisão inteira	$B / A$	2
%	Módulo	$B \% A$	0

# Atividade 2

Elabore um fluxograma que leia o valor o valor de um lado de um quadrado (em cm) e exiba sua área (em  $\text{cm}^2$ )

# Atividade 3

Elabore um fluxograma que leia dois números, na sequência imprima o resultado da soma, da subtração, multiplicação e divisão.

$N1+N2$

$N1-N2$

$N1*N2$

$N1/N2$



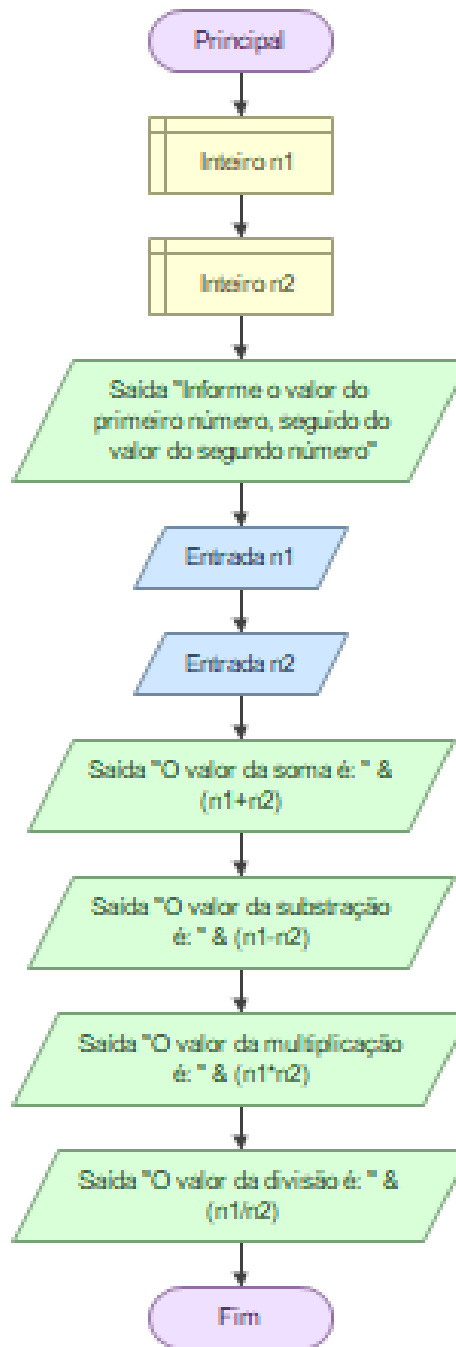
# Atividade 4

Elabore um fluxograma que dado um número inteiro em segundos, transforme-o em Horas, minutos e segundos.

Lembrando:

60 segundos -> 1 minuto

60 minutos -> 1 hora





**Obrigado!!!**