



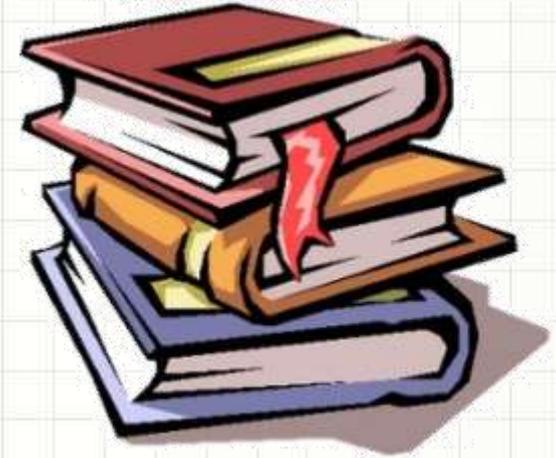
LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA

INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO – PARTE II

Prof. Dr. Daniel Caetano

2012 - 1

Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 4)

Apresentação

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 4)

Material Didático

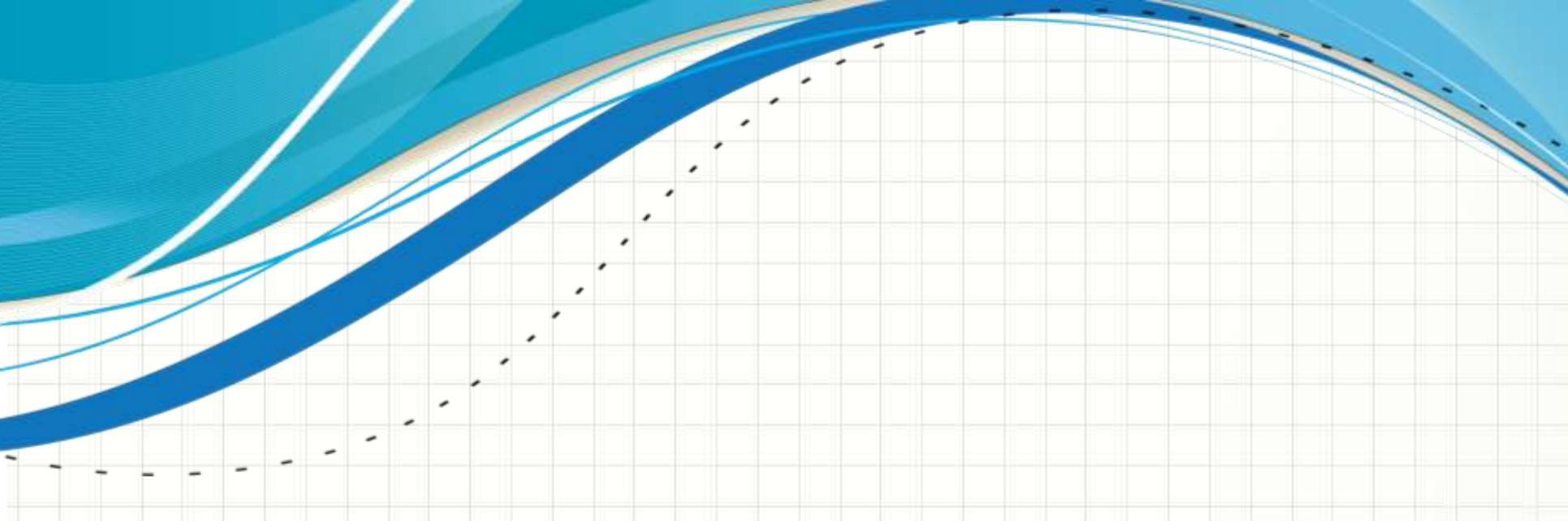
Lógica de Programação – Fundamentos da
Programação de Computadores, páginas 4 a 8, 12 e 13.

Objetivos

- Apresentar, na prática, os diferentes níveis de detalhamento de algoritmo
- Apresentar a estrutura do português estruturado
- Capacitar o aluno para escrever algoritmos sequenciais em português estruturado

- **Exercício**
- **LISTA 1**

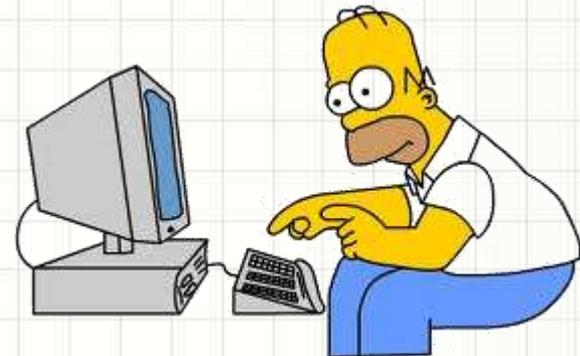




A ANALOGIA DO PROGRAMA DE COMPUTADOR

Analogia

- Relação Chefe-Funcionário-Cliente



Analogia

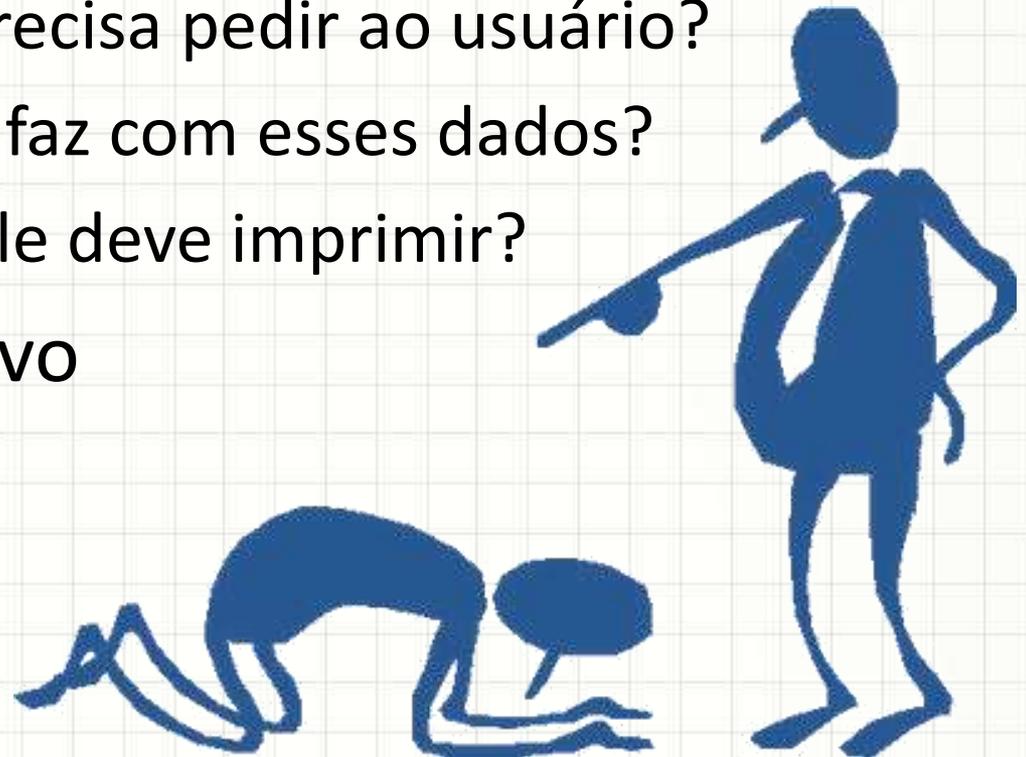
- Relação Chefe-Funcionário-Cliente



**Leve um café
para o cliente!**

Aja como Chefe!

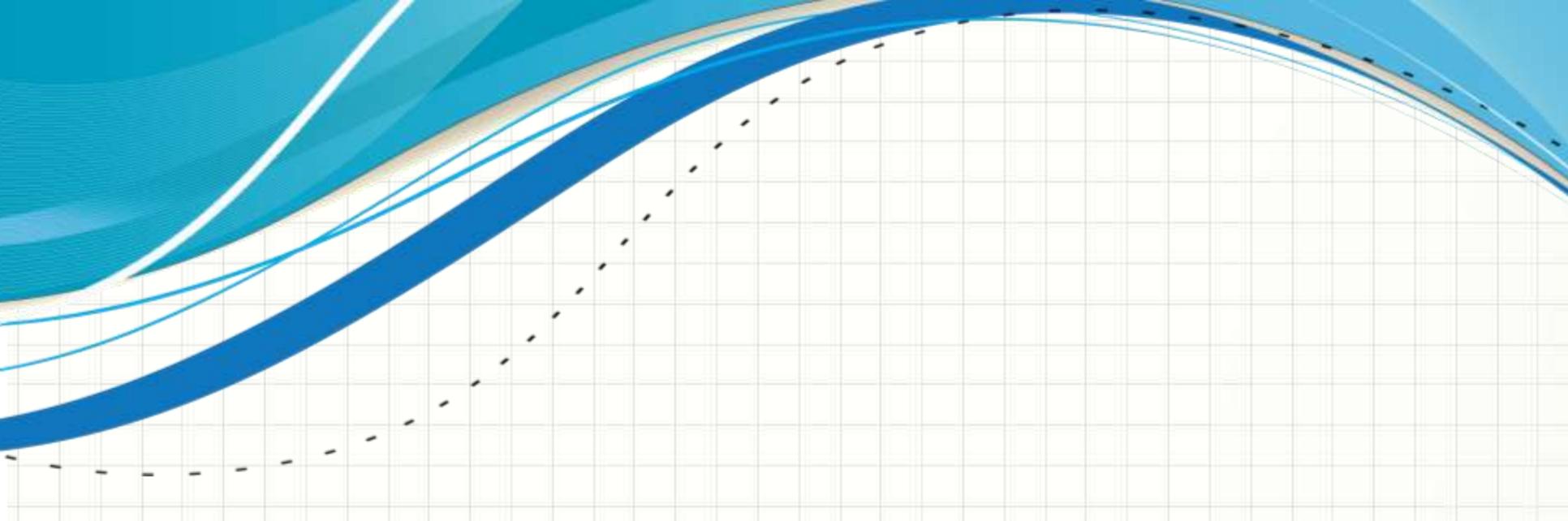
- Programar: dar ordens ao computador
- Como resolver um problema?
 - Que dados ele precisa pedir ao usuário?
 - Que cálculos ele faz com esses dados?
 - Qual resultado ele deve imprimir?
- Ordens: imperativo
 - Leia!
 - Calcule!
 - Imprima!



Dando Ordens

- Ensinar o computador a multiplicar por dois um número fornecido pelo usuário:
 - Leia um número digitado pelo usuário
 - Multiplique este número por dois
 - Imprima o resultado





NÍVEIS DE DETALHAMENTO

Nível de Detalhamento?

- Aula passada: vários níveis de detalhamento
- Qual usar?
- No início:
 - Linguagem natural
 - Fluxograma
 - Português estruturado



Algoritmo Mais Simples do Mundo

- Linguagem Natural
- Português Estruturado

1. Escreva “Olá!”

Algoritmo “Imprime Olá”

Var

Início

Escreva (“Olá! “)

FimAlgoritmo

- Fluxograma



Algoritmo Mais Simples do Mundo

- Linguagem Natural
- Português Estruturado

1. Escreva "Olá!"

Algoritmo "Imprime Olá"

Var

Início

Escreva ("Olá! ")

FimAlgoritmo

- Fluxograma



Aumentou o detalhamento: Agora são indicados o início e o fim do programa

Algoritmo Mais Simples do Mundo

Aumentou o detalhamento: Agora são indicados o nome do programa, área de variáveis e de código

- Fluxograma



- Português Estruturado

Algoritmo “Imprime Olá”

Var

Início

Escreva (“Olá! “)

FimAlgoritmo

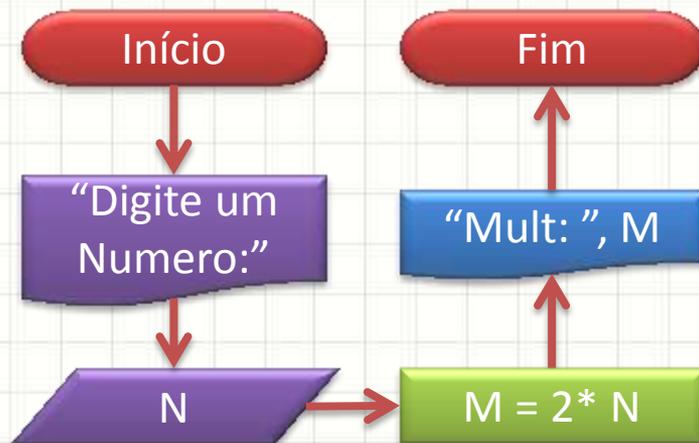


Multiplicar um Número por Dois

- Linguagem Natural

1. Leia um número
2. Multiplique o número por dois
3. Imprima o valor

- Fluxograma



- Português Estruturado

Algoritmo "Multiplicação"

Var

INTEIRO: N, M

Início

Escreva("Digite Um Número:")

Leia(N)

$M \leftarrow 2 * N$

Escreva("Mult:", M)

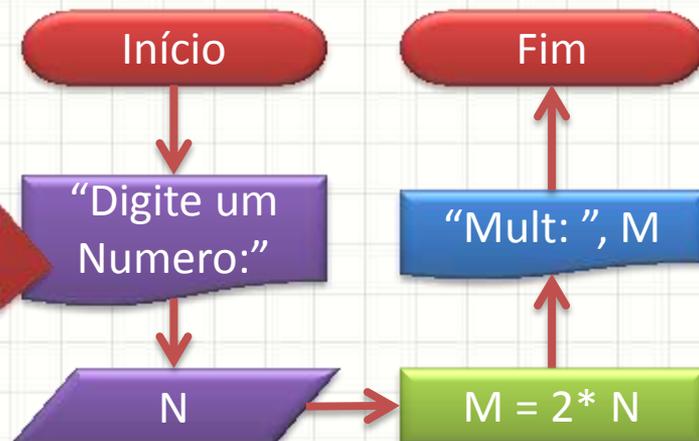
FimAlgoritmo

Multiplicar um Número por Dois

- Linguagem Natural

1. Leia um número
2. Multiplique o número por dois
3. Imprima o valor

- Fluxograma



- Português Estruturado

Algoritmo "Multiplicação"

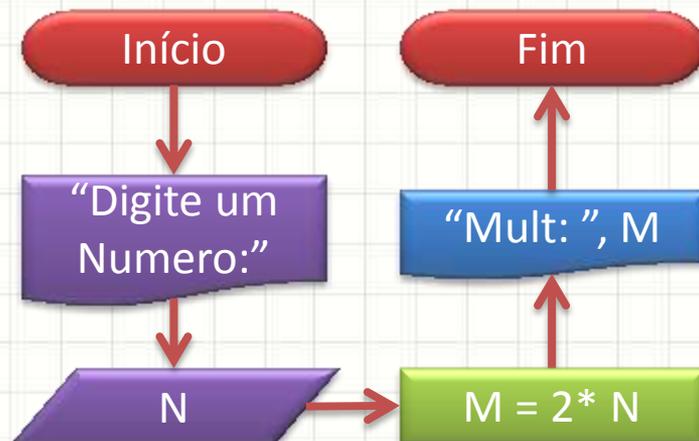
Var

Aumentou o detalhamento: indicação das variáveis, explicitação das operações e definição da forma da impressão

Multiplicar um Número por Dois

**Aumentou o detalhamento:
declaração das variáveis
(com tipo)**

- Fluxograma



- Português Estruturado

Algoritmo "Multiplicação"

Var

INTEIRO: N, M

Início

Escreva("Digite Um Número:")

Leia(N)

$M \leftarrow 2 * N$

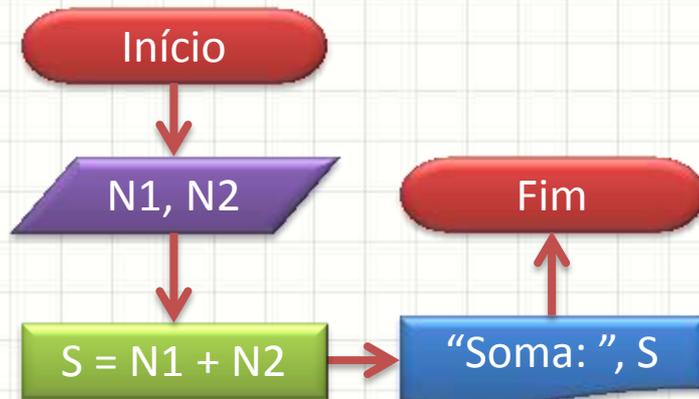
Escreva("Mult:", M)

FimAlgoritmo

Somando Dois Números

- Linguagem Natural
 1. Leia dois números
 2. Some os dois números
 3. Imprima a soma

- Fluxograma



- Português Estruturado

Algoritmo “Soma 2 Números”

Var

INTEIRO: N1, N2, S

Início

Escreva(“Digite Um Número:”)

Leia(N1)

Escreva(“Digite Outro Número:”)

Leia(N2)

$S \leftarrow N1 + N2$

Escreva(“Soma:”, S)

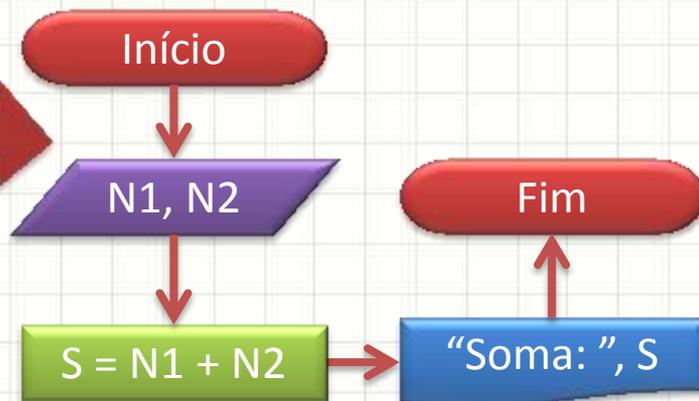
FimAlgoritmo

Somando Dois Números

- Linguagem Natural

1. Leia dois números
2. Some os dois números
3. Imprima a soma

- Fluxograma



- Português Estruturado

Algoritmo “Soma 2 Números”

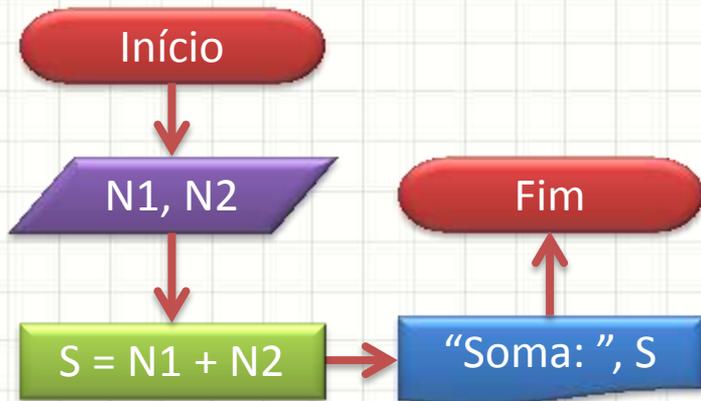
Var

Aumentou o detalhamento:
indicação das variáveis, explicitação das operações e definição da forma da impressão

Somando Dois Números

Aumentou o detalhamento:
declaração das variáveis
(com tipo), mensagem de solicitação dos valores a digitar

Fluxograma



- Português Estruturado

Algoritmo “Soma 2 Números”

Var

INTEIRO: N1, N2, S

Início

Escreva(“Digite Um Número:”)

Leia(N1)

Escreva(“Digite Outro Número:”)

Leia(N2)

$S \leftarrow N1 + N2$

Escreva(“Soma:”, S)

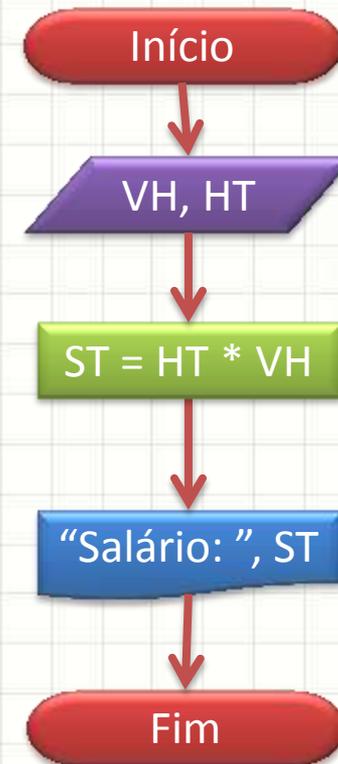
FimAlgoritmo

Calculando o Salário de Horista

- Linguagem Natural

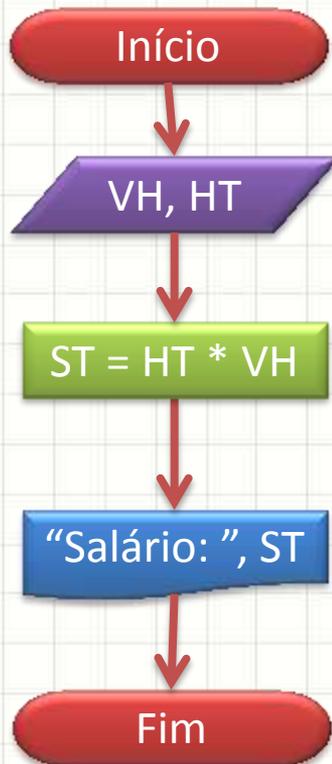
1. Leia o valor da hora de trabalho, em R\$ por hora
2. Leia o número de horas trabalhadas
3. Calcule o salário total, multiplicando o número de horas pelo valor da hora
4. Imprima o salário total

- Fluxograma



Calculando o Salário de Horista

- Fluxograma



- Português Estruturado

Algoritmo “Calcula salário de horista”

Var

REAL: VH, HT, S

Início

Escreva(“Digite o valor da hora, em R\$/h:”)

Leia(VH)

Escreva(“Digite as horas trabalhadas:”)

Leia(HT)

$ST \leftarrow HT * VH$

Escreva(“Salário: ”, ST)

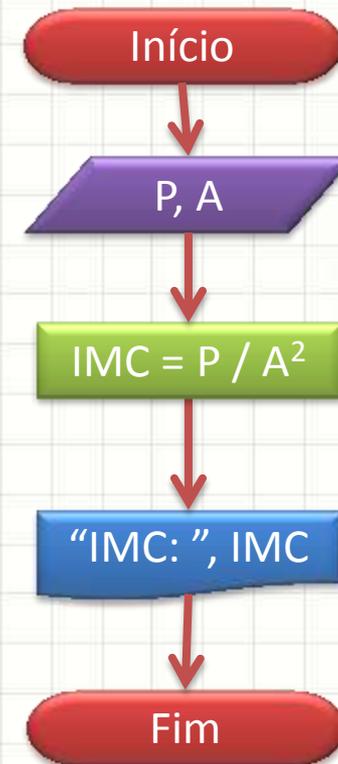
FimAlgoritmo

Índice de Massa Corporal

- Linguagem Natural

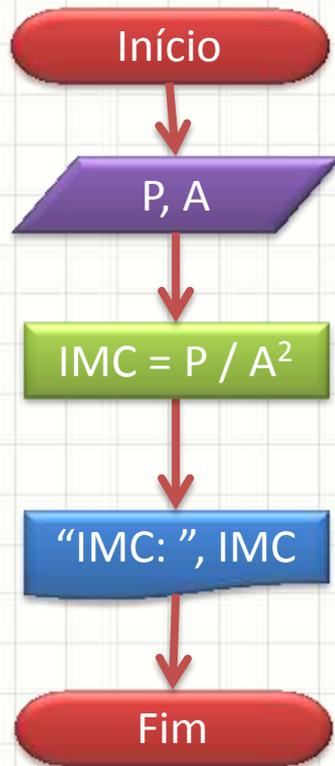
1. Leia o peso do sujeito, em kg
2. Leia a altura do sujeito, em metros
3. Calcule o IMC, dividindo o peso pelo quadrado da altura
4. Imprima o IMC

- Fluxograma



Índice de Massa Corporal

- Fluxograma



- Português Estruturado

Algoritmo “Calcula IMC”

Var

REAL: P, A, IMC

Início

Escreva(“Digite o seu peso, em Kg: ”)

Leia(P)

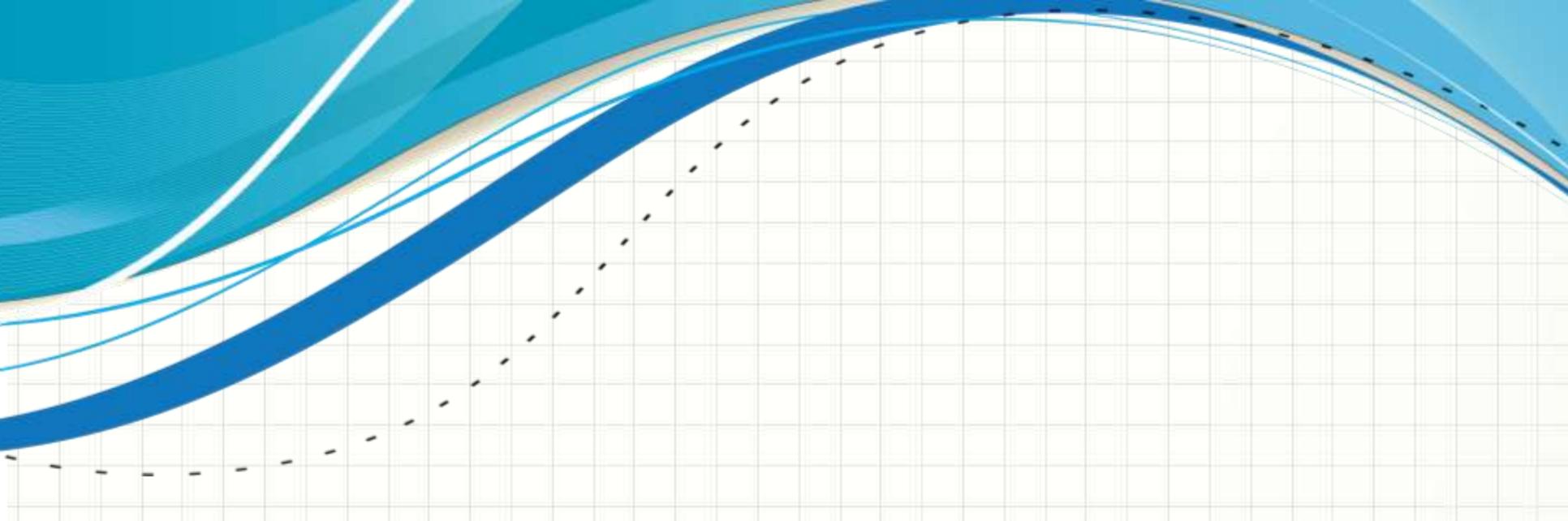
Escreva(“Digite sua altura, em metros: ”)

Leia(A)

$IMC \leftarrow P / (A * A)$

Escreva(“IMC: ”, IMC)

FimAlgoritmo



O PORTUGUÊS ESTRUTURADO

Estrutura do Programa

- Todo programa em Português Estruturado tem **duas** partes importantes:
 1. A Declaração
 2. O Código (Algoritmo)

Algoritmo “Nome do Algoritmo”

Var

Início

FimAlgoritmo

Estrutura do Programa

- Área de Declaração indica
 1. Nome do Algoritmo
 2. As variáveis usadas no programa

Algoritmo "Nome do Algoritmo"

Var

Inicio

FimAlgoritmo

Estrutura do Programa

- Declarando a variável A para números REAIS

Algoritmo “Nome do Algoritmo”

Var

REAL : A

Inicio

FimAlgoritmo

A declaração de variáveis tem sempre o seguinte formato:

TipoDaVariável : NomeDaVariável



REAL: A

Inicio

FimAlgoritmo

Inicio "Nome do Algoritmo"

A declaração de variáveis tem sempre o seguinte formato:

TipoDaVariável : NomeDaVariável

Algoritmo “Nome do Algoritmo”

Var

REAL : A

Inicio

FimAlgoritmo

Estrutura do Programa

- Declarando a variável A para números REAIS
- Declarando também a variável B para REAIS

Algoritmo “Nome do Algoritmo”

Var

REAL : A, B

Inicio

FimAlgoritmo

Podemos declarar mais de uma variável do mesmo tipo em uma única linha, separando-as por vírgulas

TipoDaVariável : Variável1, Variável2

Algoritmo “ Algoritmo”

Var

REAL: A, B

Inicio

FimAlgoritmo

Estrutura do Programa

- Declarando a variável A para números REAIS
- Declarando também a variável B para REAIS
- Agora, a variável C para números INTEIROS

Algoritmo “Nome do Algoritmo”

Var

REAL : A, B

INTEIRO : C

Inicio

FimAlgoritmo

Para declarar **variáveis de tipos diferentes**, é necessário **declarar em linhas diferentes**

Algoritmo "Nome do Algoritmo"

Var

REAL : A, B

INTEIRO : C

Inicio

FimAlgoritmo

Estrutura do Programa

- Declarando a variável A para números REAIS
- Declarando também a variável B para REAIS
- Agora, a variável C para números INTEIROS

Algoritmo “N

Var

REAL : A, B

INTEIRO : C

Início

FimAlgoritmo

Simplificadamente, os tipos de variáveis são:

INTEIRO

REAL

LOGICO

CARACTERE/LITERAL

Há regras para os NOMES das variáveis:

1. Não podem ter espaços
2. Não podem iniciar com números
3. Não podem usar caracteres especiais
(acentos etc.)
4. Não podem ser uma palavra da linguagem
(REAL ou Inicio, por exemplo)

Var

REAL : A, B

INTEIRO : C

Inicio

FimAlgoritmo

Estrutura do Programa

- A área de código é onde será colocado nosso programa
- Nessa área, daremos ordens ao computador

Algoritmo “Nome do Algoritmo”

Var

Inicio

FimAlgoritmo

Área de Código

- Tarefa mais comum: escrever texto na tela
- Escrever “Uma frase qualquer”

Algoritmo “Nome do Algoritmo”

Var

Início

Escreva (“Uma frase qualquer”)

FimAlgoritmo

Área

Uma frase qualquer

- Ta
- Es

la

Início

Escreva (“Uma frase qualquer”)

FimAlgoritmo

Área de Código

- Tarefa mais comum: escrever texto na tela
- Escrever “Uma frase qualquer”
- Escrever o número **1934**

Algoritmo “Nome do Algoritmo”

Var

Início

Escreva (“Uma frase qualquer”)

Escreva (1934)

FimAlgoritmo

Área

Uma frase qualquer1934

- Ta
- Es
- Es

la

Início

Escreva (“Uma frase qualquer”)

Escreva (1934)

FimAlgoritmo

Área de Código

- Tarefa mais comum: escrever texto na tela
- Escrever “Uma frase qualquer” e pular linha
- Escrever o número **1934**

Algoritmo “Nome do Algoritmo”

Var

Início

Escreval (“Uma frase qualquer”)

Escreva (1934)

FimAlgoritmo

Área

Uma frase qualquer
1934

- Ta
- Es
- Es

la
nha



Início

Escreva ("Uma frase qualquer")

Escreva (1934)

FimAlgoritmo

Área de Código

- Vamos fazer algo mais complexo
- Escrever “2 + 3 = ”
- Escrever o resultado de **2+3**

Algoritmo “Nome do Algoritmo”

Var

Início

Escreva (“2 + 3 = ”)

Escreva (2+3)

FimAlgoritmo

Área

$$2 + 3 = 5$$

- Va
- Es
- Es

Início

Escreva ("2 + 3 = ")

Escreva (2+3)

FimAlgoritmo

Área

2 + 3 = 5

- Va
- Es
- Es

Quando mando
escrever algo **entre**
aspas, como "2+3=", o
computador **reproduz**
exatamente na tela

Início

Escreva ("2 + 3 = ")

Escreva (2+3)

FimAlgoritmo

Área

2 + 3 = 5

- Va
- Es
- Es

Esses conteúdos o computador considera como **texto**. O nome técnico para texto é string

Início

Escreva ("2 + 3 = ")

Escreva (2+3)

FimAlgoritmo

Área

$2 + 3 = 5$

- Va
- Es
- Es

Quando mando escrever algo **sem aspas**, como $2+3$, o computador **resolve** e imprime o **resultado**.

Início

Escreva $2 + 3 = "$)

Escreva $(2+3)$

FimAlgoritmo

Área

$$2 + 3 = 5$$

- Va
- Es
- Es

O nome técnico dos elementos que o computador resolve antes de continuar é expressão

Início

Escreva "2 + 3 = "

Escreva (2+3)

FimAlgoritmo

Área de Código

- Se quero que as impressões apareçam uma do lado da outra... Ainda assim preciso de duas linhas?

Algoritmo “Nome do Algoritmo”

Var

Início

Escreva (“2 + 3 = ”)

Escreva (2+3)

FimAlgoritmo

Área de Código

- **NÃO!** Podemos imprimir com uma linha só, separando as informações com vírgula

Algoritmo “Nome do Algoritmo”

Var

Início

Escreva (“2 + 3 = ”)

Escreva (2+3)

FimAlgoritmo

Área de Código

- **NÃO!** Podemos imprimir com uma linha só, separando as informações com vírgula

Algoritmo “Nome do Algoritmo”

Var

Início

Escreva (“2 + 3 = ”, 2+3)

FimAlgoritmo

Área

$$2 + 3 = 5$$

- Na
se

só,

Início

Escreva ("2 + 3 =", 2+3)

FimAlgoritmo

Nas expressões, podemos usar as operações matemáticas usuais:

Soma:	+	Divisão Inteira:	\
Subtração:	-	Resto da Divisão:	%
Multiplicação:	*	Potência:	^
Divisão:	/	Parênteses:	()

Var

Início

Escreva ("2 + 3 = ", 2+3)

FimAlgoritmo

As regras da matemática se aplicam:

$$2 + 4 * 2 =$$

$$2 + (4 * 2) =$$

$$2 + 8 =$$

10



$$(2 + 4) * 2 =$$

$$6 * 2 =$$

12

Var

Início

Escreva ("2 + 3 = ", 2+3)

FimAlgoritmo

Área de Código

- Para fazer coisas mais elaboradas, podemos usar variáveis

Algoritmo “Nome do Algoritmo”

Var

INTEIRO : SOMA

Início

SOMA \leftarrow 2+3

Escreva (“2 + 3 = ”, SOMA)

FimAlgoritmo

Área de Código

- Para fazer coisas mais fáceis usar variáveis

O símbolo \leftarrow é chamado atribuição e ele serve para guardar um valor em uma variável

Algoritmo "N"

Var

INTEIRO SOMA

Início

SOMA \leftarrow 2+3

Escreva ("2 + 3 = ", SOMA)

FimAlgoritmo

Uma atribuição sempre tem o nome de uma variável do lado esquerdo...

Algoritmo “Nome do Algoritmo”

Var

Inicio IRO : SOMA

Início

SOMA ← 2+3

Escreva (“2 + 3 = ”, SOMA)

FimAlgoritmo

...e um valor ou expressão do lado direito

Algoritmo “Nome do Algoritmo”

Var

INTEIRO : MA

Início

SOMA ← 2+3

Escreva (“2 + 3 = ”, SOMA)

FimAlgoritmo

Como a variável representa uma posição de memória, e a memória só armazena números, toda expressão é calculada antes que seu resultado seja armazenado

Algoritmo “Nome do Algoritmo”

Var

INTEIRO : MA

Início

SOMA ← **2+3**

Escreva (“2 + 3 = ”, SOMA)

FimAlgoritmo

Isso significa que o computador resolve a linha destacada assim:

SOMA \leftarrow 2+3

SOMA \leftarrow 5

A leitura dessa linha é “SOMA recebe 2+3”

Algoritmo “Nome do Algoritmo”

Var

INTE : SOMA

Inicio

SOMA \leftarrow 2+3

Escreva (“2 + 3 =”, SOMA)

FimAlgoritmo

Área de Código

- Finalmente, falta vermos como ler um valor
- Para ler um valor, precisamos de uma variável

Algoritmo “Nome do Algoritmo”

Var

INTEIRO : A

Início

Leia(A)

FimAlgoritmo

Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

O comando de entrada é simples: ele espera o usuário digitar algo e guarda na variável indicada (no caso, a variável A)

Algoritmo "N

Var

INTEIRO : A

Inicio

Leia(A)



FimAlgoritmo

Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

Observe, entretanto, que nada aparecerá na tela, além de um cursor, também chamado de prompt

Algoritmo "N

Var

INTEIRO : A

Início

Leia(A)



FimAlgoritmo

Ár -

- F
- P

anto,
erá na
m
m
mpt

Início

Leia(A)



FimAlgoritmo

Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

Devemos, então, imprimir uma mensagem para que o usuário saiba o que ele precisa digitar

Algoritmo "N

Var

INTEIRO : A

Início

Leia(A)

FimAlgoritmo

Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

Devemos, então, imprimir uma mensagem para que o usuário saiba o que ele precisa digitar

Algoritmo "N

Var

INTEIRO : A

Inicio

Leia(A)

FimAlgoritmo

Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

Devemos, então, imprimir uma mensagem para que o usuário saiba o que ele precisa digitar

Algoritmo "N

Var

INTEIRO : A

Início

Escreva("Digite um número: ")

Leia(A)

FimAlgoritmo

Área

Digite um número: _

- Fin
- Pa

o,

ue o
e ele

Inicio

Escreva("Digite um número: ")

Leia(A)

FimAlgoritmo

Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

O computador irá esperar até que o usuário digite algo e aperte a tecla enter

Algoritmo "N

Var

INTEIRO : A

Início

Escreva("Digite um

Leia(A)

FimAlgoritmo



Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

O tipo de informação que o computador irá aceitar, depende do tipo da variável!

Algoritmo "N

Var

INTEIRO : A

Inicio

Escreva("Digite um número: ")

Lea(A)

FimAlgoritmo



Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

Se a variável é inteira e o usuário digitar **20,5** ... o valor armazenado será 20

Algoritmo "N

Var

INTEIRO : A

Inicio

Escreva("Digite um número: ")

Leia(A)

FimAlgoritmo



Área de Código

- Finalmente, falta ver
- Para ler um valor, pre

Se a variável é inteira e o usuário digitar um texto ... o programa irá gerar um erro

Algoritmo "N

Var

INTEIRO : A

Inicio

Escreva("Digite um número: ")

Lea(A)

FimAlgoritmo



Área de Código

- Agora que já temos um valor em A, podemos imprimi-lo

Algoritmo “Nome do Algoritmo”

Var

INTEIRO : A

Início

Escreva(“Digite um número: ”)

Leia(A)

Escreva(“O número digitado foi: ”, A)

FimAlgoritmo

Áre

Digite um número: 1357
O número digitado foi: 1357

- Ag
im

mos

Início

Escreva("Digite um número: ")

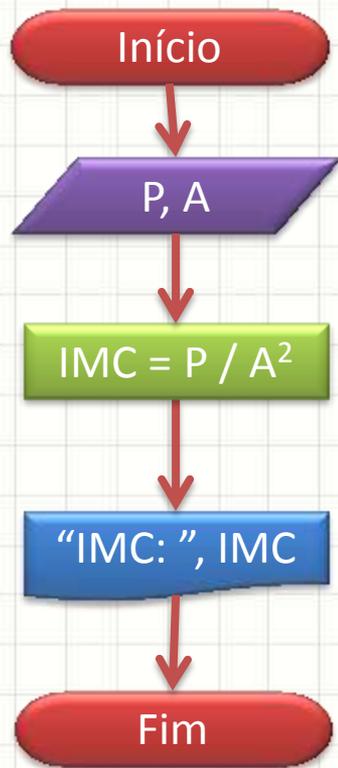
Leia(A)

Escreva("O número digitado foi: ", A)

FimAlgoritmo

Reverendo o Algoritmo de IMC

- Fluxograma



- Português Estruturado

Algoritmo “Calcula IMC”

Var

REAL : P, A, IMC

Inicio

Escreva(“Digite o seu peso, em Kg: ”)

Leia(P)

Escreva(“Digite sua altura, em metros: ”)

Leia(A)

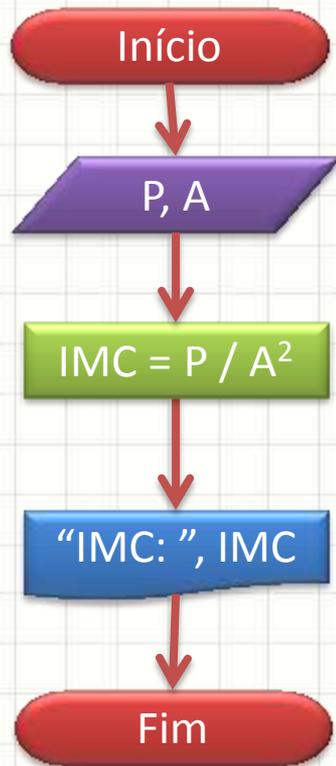
$IMC \leftarrow P / (A * A)$

Escreva(“IMC: ”, IMC)

FimAlgoritmo

Reverendo o Algoritmo de IMC

- Fluxograma



- Português Estruturado

Algoritmo “Calcula IMC”

Var

REAL : P, A, IMC

Inicio

Escreva(“Digite o seu peso, em Kg: ”)

Leia(P)

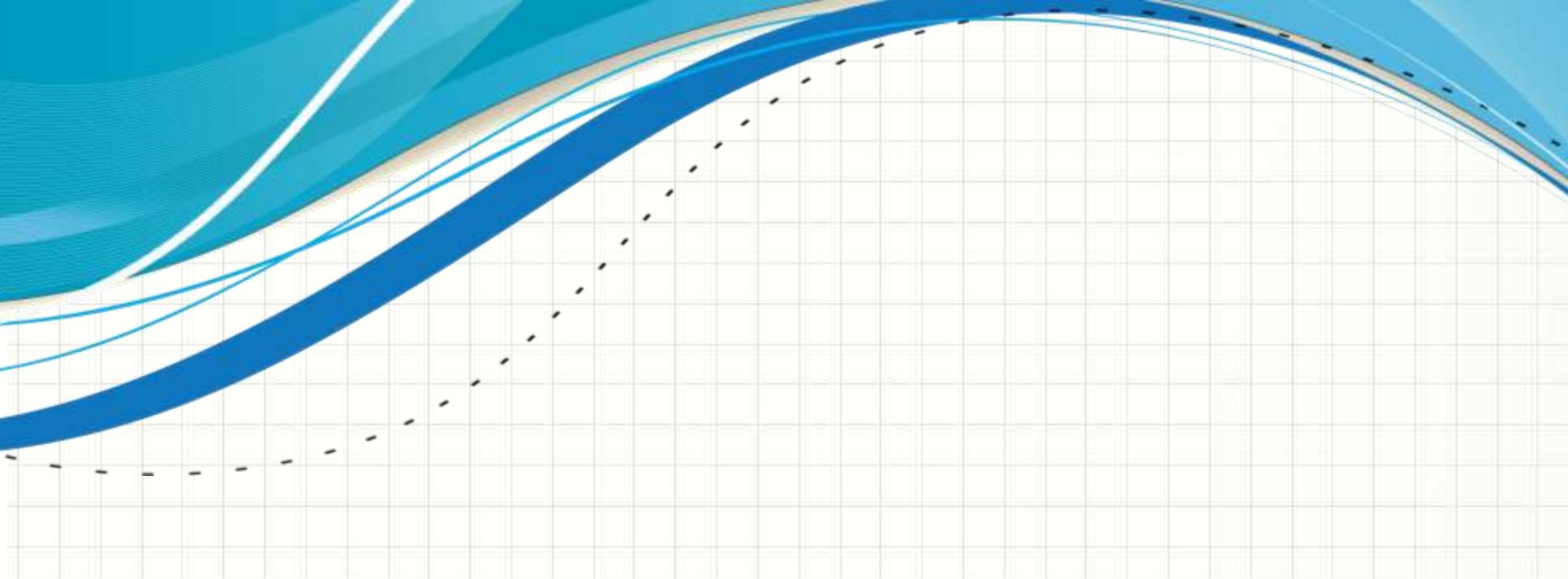
Escreva(“Digite sua altura, em metros: ”)

Leia(A)

$IMC \leftarrow P / (A^2)$

Escreva(“IMC: ”, IMC)

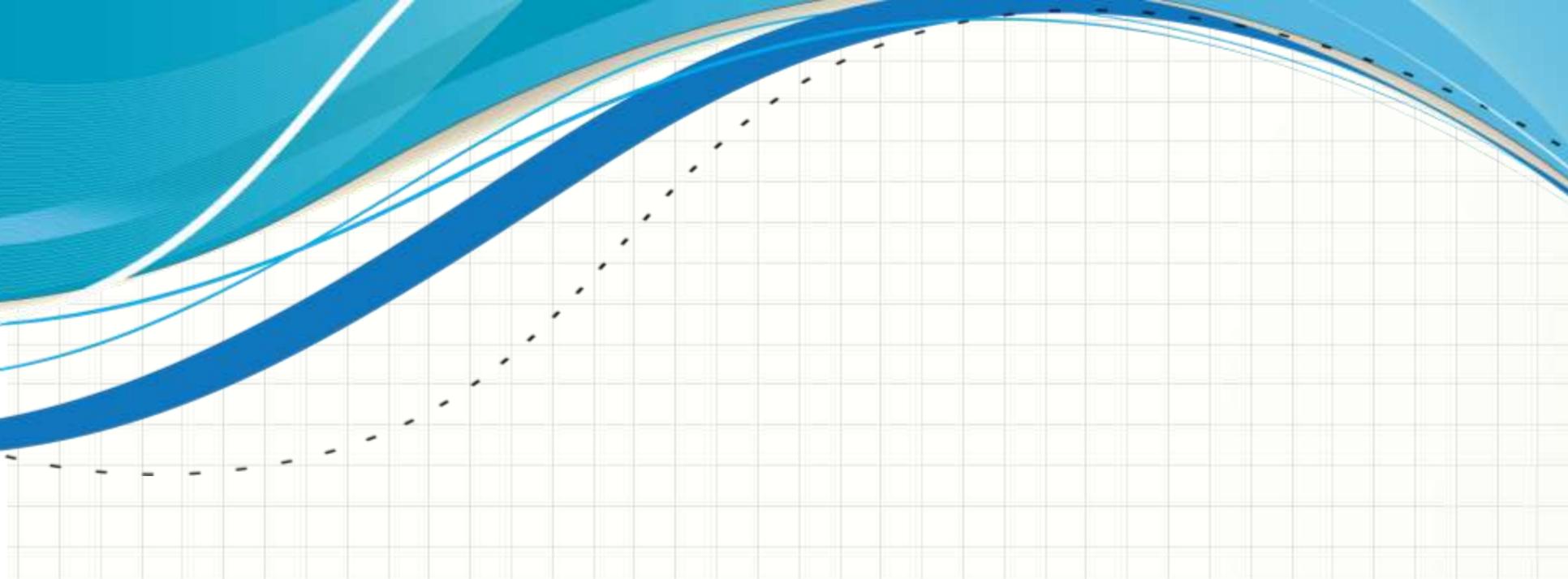
FimAlgoritmo



EXERCÍCIO

Exercício

- Escreva, em **linguagem natural, fluxograma e português estruturado** um algoritmo que receba do usuário um número de anos e calcule, aproximadamente, o número de dias total deste número anos
 - **Atenção:** use 365,25 como sendo o número de dias por ano
 - **Em duplas**



CONCLUSÕES

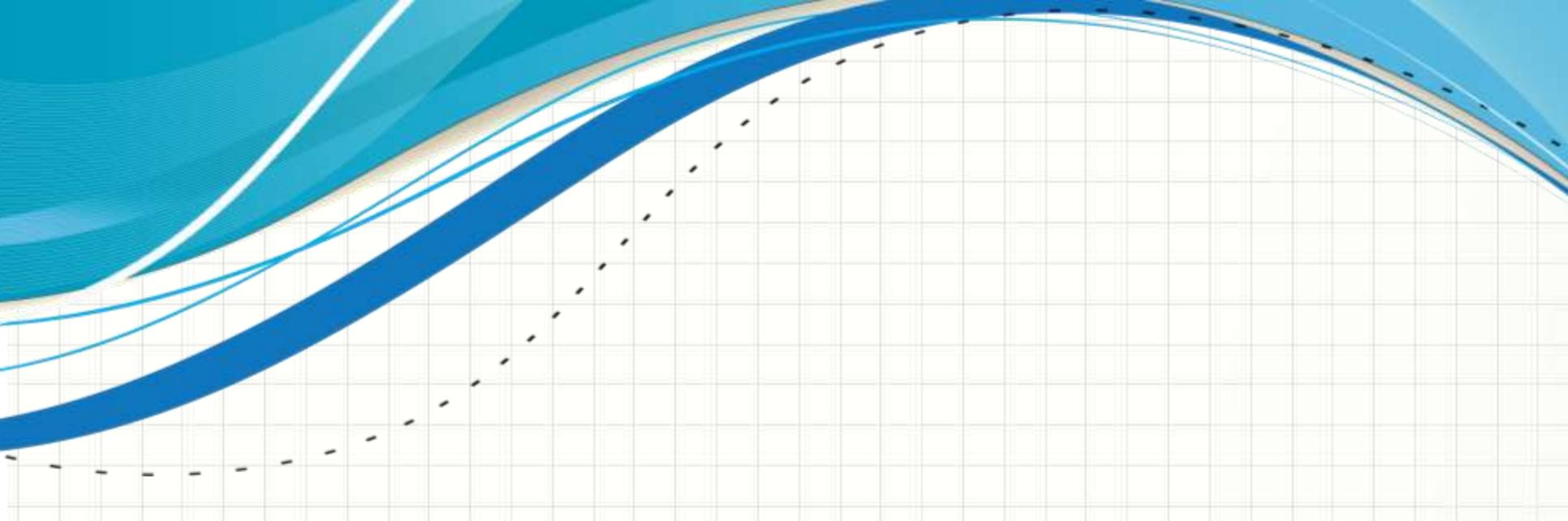
Resumo

- Escrever um algoritmo de computador requer formalização
- As diferentes representações nos obrigam a diferentes níveis de formalização e detalhe
- O Português Estruturado tem um conjunto de regras que precisa ser seguido
- **TAREFA!**
 - Lista de Exercícios 1

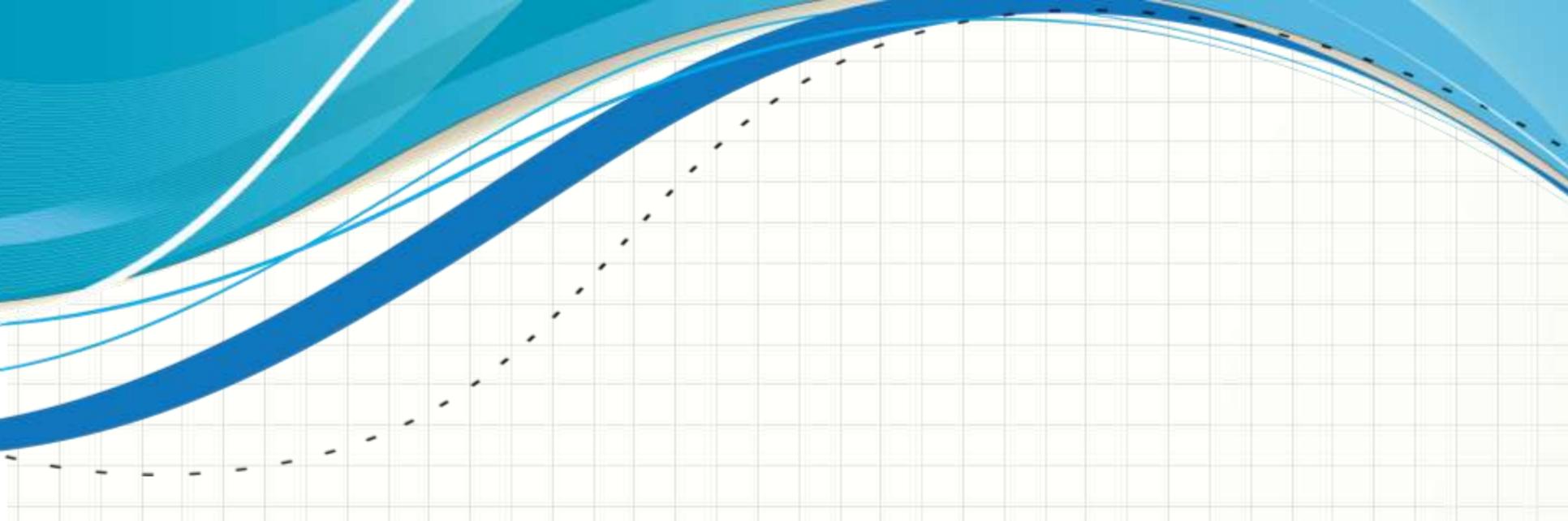
Próxima Aula



- Como testar isso?
 - Usaremos o Português Estruturado como base
 - Para construir um programa C/C++ que pode ser executado!



PERGUNTAS?



**BOM DESCANSO
A TODOS!**