

Aula 1

Algoritmo e Pseudo-código

Seiji Isotani, Rafaela V. Rocha

sisotani@icmc.usp.br

rafaela.vilela@gmail.com

PAE: Armando M. Toda, Geiser Chalco

armando.toda@gmail.com

geiser.gcc@gmail.com

Informações sobre o curso

- Site do curso

<https://edisciplinas.usp.br/course/view.php?id=39957>

- Atividades online:

<http://moodle.caed-lab.com>

- Cronograma das aulas

– Será disponibilizado no edisciplinas

Objetivos do Curso

Apresentar os conceitos básicos para o desenvolvimento de ***programas***, utilizando uma **linguagem de programação** como apoio

Desenvolver o **Pensamento Computacional**

O que é?

- **Programa:** *codificação de um algoritmo em uma linguagem de programação*
(Ascencio, 1999)

```
//oimundo.cpp -- imprime uma mensagem na tela
#include <iostream>
using namespace std;
int main( )
{
    cout << "Olá, mundo!";
    cin.get();
    return 0;
}
```

O que é?

- **Pensamento computacional:**
pensamento nos problemas de forma analítica e desenvolvimento de soluções em forma de algoritmos
(Easterbrook, 2014)

Pensamento Computacional

Técnicas específicas de pensamento computacional incluem:

- Decomposição do problema
- Reconhecimento de padrões
- Generalização de padrões para definir abstrações ou modelos
- Projeto de algoritmos
- Análise de dados e visualização

Problemas

- **Somar três números**

Passo 1: Receber três números

Passo 2: Somar os três números

Passo 3: Mostrar o resultado obtido

Problemas

- Trocar uma lâmpada



Problemas

- Sacar dinheiro no banco 24 horas



- Para resolver os problemas apresentados você precisa pelo menos:
 - Indicar quais são as entradas esperadas
 - Indicar quais são as saídas produzidas
 - Um conjunto de passos não ambíguos para transformar as entradas nas saídas.

Isso é um algoritmo

Algoritmo

Um **algoritmo** é um conjunto de **atividades** que podem ser executadas **passo a passo** para **resolver problemas**

Outras definições:

- <http://en.wikipedia.org/wiki/Algorithm>
- Robert Sedgewick – Algorithms in C, 3rd Ed, Addison Wesley

Algoritmo

- Características básicas
 - ter **fim**
 - não dar margem à dupla interpretação
 - capacidade de receber dado(s) de **entrada** do mundo exterior
 - poder gerar informações de **saída** para o mundo externo ao do ambiente do algoritmo
 - ser efetivo (todas as etapas especificadas no algoritmo devem ser alcançáveis em um tempo finito)

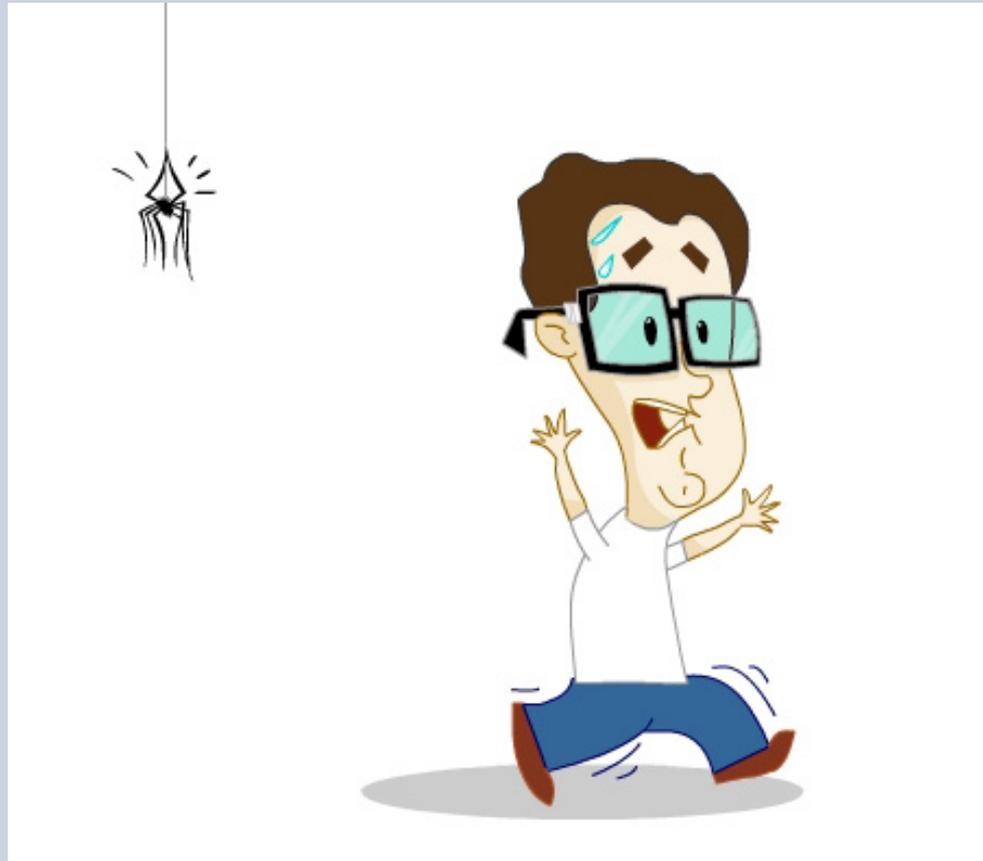
Mas peraí....

- Um algoritmo é um programa?
 - Não. Um programa é uma das possíveis representações de um algoritmo.
- Um algoritmo é um processo?
 - Não, processo é a atividade de executar um programa e, conseqüentemente, também é a atividade de executar um algoritmo.

Algoritmos estão em toda a parte



Não adianta fugir



- Imagine que você queira comprar num site online
 - Um algoritmo vai analisar suas atividades e recomendar coisas para comprar
 - Um algoritmo irá verificar seu cartão de crédito
 - Um algoritmo irá identificar a maneira mais eficiente de entregar o produto na sua casa
- Não vai comprar online?
 - Existe um algoritmo para maximizar as vendas colocando os produtos na prateleira de maneira diferenciada
- Em computação tudo vira algoritmos

Se não é possível fugir, então
junte-se a ele 😊.

**Seja um criador de algoritmo
de primeira linha!**

Como?

Como criar algoritmos?

- Para se tornar um bom **cozinheiro**, você não deve memorizar livros e livros de receitas.
- Você **estuda** receitas existentes.
- Você **pratica** fazê-los.
- Você **experimenta** com suas próprias variações.
- Até que possa **inventar** suas próprias receitas.

Como criar algoritmos?

- Para se tornar um bom **profissional da computação**, você não deve memorizar livros e livros de algoritmos.
- Você **estuda** algoritmos famosos.
- Você **pratica** implementá-los e executá-los.
- Depois passa a reconhecer padrões mais gerais, e a **experimental** implementações com suas próprias variações.
- Até que, finalmente, você irá **inventar** seus próprios algoritmos.

OK. Agora eu sei o que é um algoritmo!

Mas como “escrever” ou representar um algoritmo?

- **Descrição Narrativa**
- **Fluxograma**
- **Pseudocódigo**

Descrição Narrativa

Passo 1: Receber três números

Passo 2: Somar os três números

Passo 3: Mostrar o resultado obtido

E como representar um algoritmo?

- **Descrição narrativa:**
 - consiste em analisar o enunciado do problema e escrever, utilizando **linguagem natural**, os passos a serem seguidos para sua resolução (receita de bolo).
 - **Ponto positivo:** Não é necessário aprender novos conceitos, pois a língua natural já é bem conhecida.
 - **Ponto negativo:** A língua natural abre espaço para várias interpretações, dificultando a transcrição desse algoritmo para programa

Exemplo – Descrição Narrativa

Crie um algoritmo para exibir o resultado da multiplicação de dois números

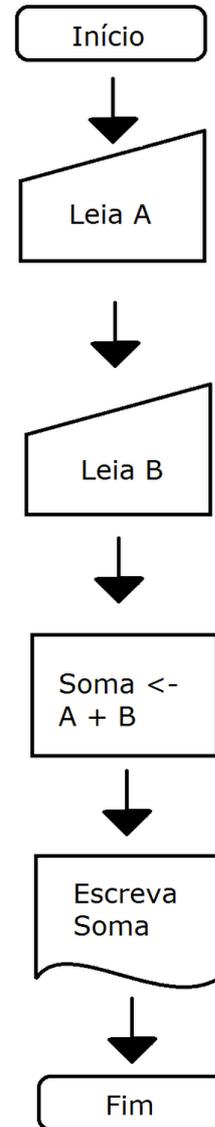
- Descrição narrativa:
 - PASSO 1 – Receber os dois números que serão multiplicados
 - PASSO 2 – Multiplicar os números
 - PASSO 3 – Mostrar o resultado obtido da multiplicação

Exemplo - Descrição Narrativa

Crie um algoritmo para exibir o dobro
($n*2$) de um número

- Descrição narrativa:
 - PASSO 1 – Receber um número
 - PASSO 2 – Multiplicar o número * 2
 - PASSO 3 – Mostrar o resultado obtido da multiplicação

Fluxograma



E como representar um algoritmo?

- **Fluxograma:**

- consiste em analisar o enunciado do problema e escrever, utilizando **símbolos gráficos**, os passos a serem seguidos para sua resolução
- **Ponto positivo:** O entendimento de elementos gráficos é mais simples que o entendimento de textos.
- **Ponto negativo:** Os fluxogramas devem ser entendidos e o algoritmo resultante não é detalhado. Isso dificulta sua transcrição para um programa

TABELA 1.1: Conjunto de símbolos utilizados no fluxograma.

	Símbolo utilizado para indicar o início e o fim do algoritmo.
	Permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.
	Símbolo utilizado para indicar cálculos e atribuições de valores.
	Símbolo utilizado para representar a entrada de dados.
	Símbolo utilizado para representar a saída de dados.
	Símbolo que indica que deve ser tomada uma decisão, indicando a possibilidade de desvios.

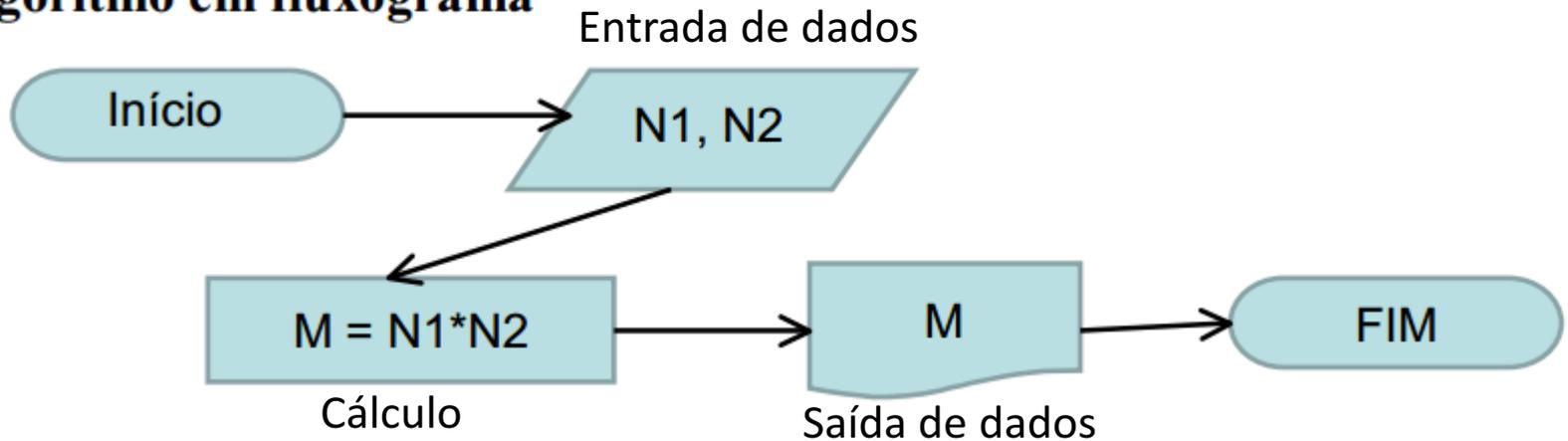
Fonte:

[http://wiki.icmc.usp.br/index.php/Scs-101\(2011101\)](http://wiki.icmc.usp.br/index.php/Scs-101(2011101))

Exemplo - Fluxograma

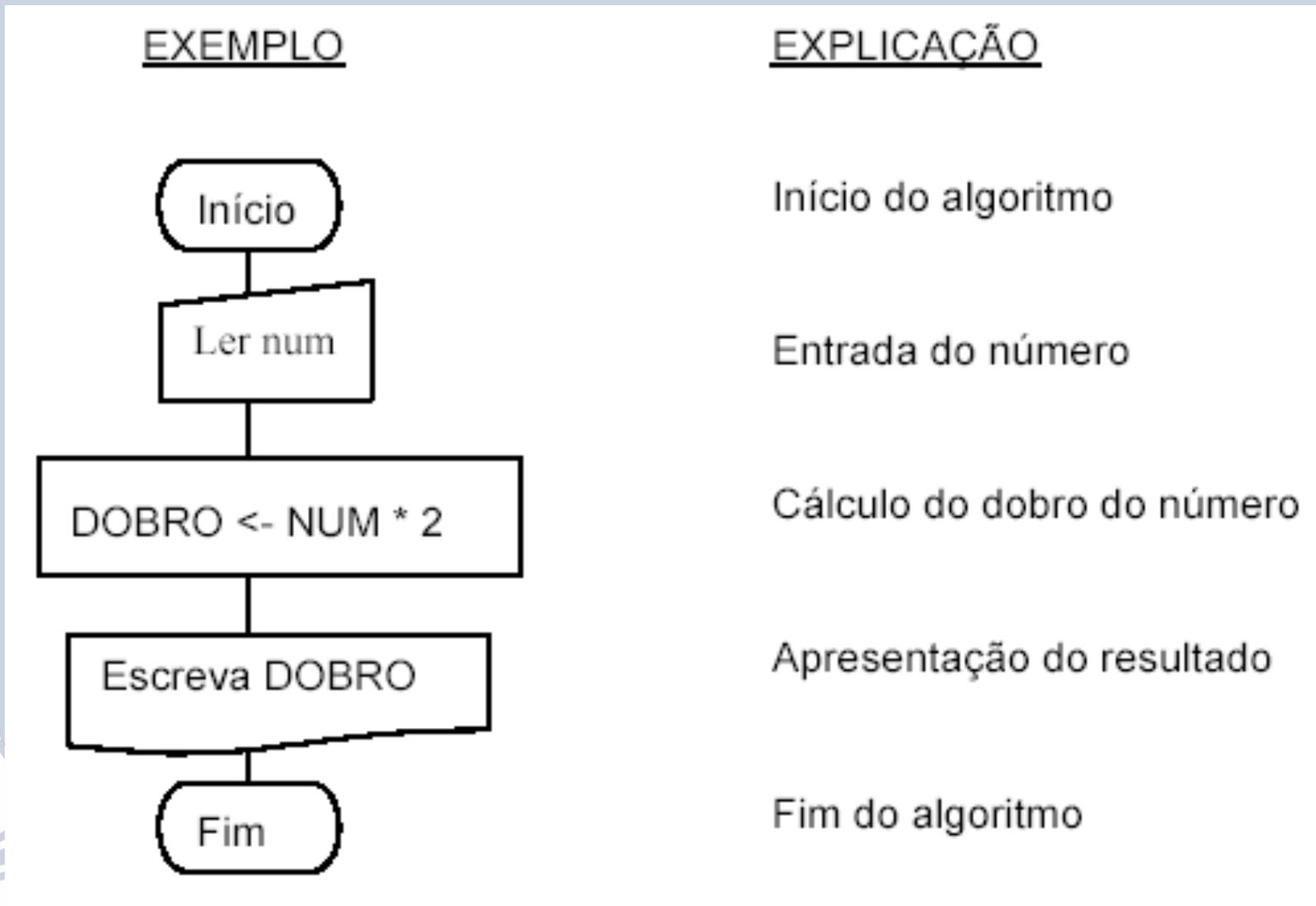
Crie um algoritmo para exibir o resultado da multiplicação de dois números

Algoritmo em fluxograma



Exemplo - Fluxograma

Crie um algoritmo para exibir o dobro de um número



Pseudocódigo

ALGORITMO

DECLARE n1, n2, m

LEIA n1, n2

m \leftarrow **n1*****n2**

ESCREVA m

FIM

E como representar um algoritmo?

- **Pseudocódigo:**

- consiste em analisar o enunciado do problema e escrever, por meio de **regras predefinidas**, os passos a serem seguidos para sua resolução
- **Ponto positivo:** Representação clara sem as especificações de linguagem de programação. A passagem do algoritmo para qualquer linguagem de programação é mais simples.
- **Ponto negativo:** As regras do pseudocódigo devem ser aprendidas

Exemplo - Pseudocódigo

Crie um algoritmo para exibir o resultado da multiplicação de dois números

ALGORITMO

DECLARE n1, n2, m

LEIA n1, n2

m ← n1*n2

ESCREVA m

FIM

Exemplo - Pseudocódigo

Existem várias formas de escrever o pseudocódigo.

Exemplos:

Multiplicacao (**n1, n2**)

m ← **n1*n2**

RETORNE m

FIM

ALGORITMO

DECLARE n1, n2, m

LEIA n1, n2

m ← **n1*n2**

ESCREVA m

FIM

Exemplo - Pseudocódigo

algoritmo “Multiplicação”

var

n1,n2,m: inteiro

inicio

escreva(“Digite dois números:”)

leia(n1)

leia(n2)

m ← n1 * n2

escreva(“Multiplicação = ”, m)

fimalgoritmo

ALGORITMO

DECLARE n1, n2, m

LEIA n1, n2

m ← n1*n2

ESCREVA m

FIM

Código C

```
#include<stdio.h>
void main() {
    int n1, n2, m;
    scanf("%d %d", &n1, &n2);
    m = n1*n2;
    printf("\n %d", m);
}
```

ALGORITMO

DECLARE n1, n2, m

LEIA n1, n2

m ← n1*n2

ESCREVA m

FIM

Exemplo - Fluxograma

Crie um algoritmo para exibir o dobro de um número

ALGORITMO

DECLARE n, m

LEIA n

m \leftarrow **n*2**

ESCREVA m

FIM

Exercício

- Faça um programa que receba o **raio** e calcule o **perímetro da circunferência** nas **três formas** de representação de algoritmos (descrição narrativa, fluxograma e pseudocódigo)

$$\text{perimetro} \leftarrow 2 * \text{pi} * \text{raio}$$