

Algoritmos

André Tavares da Silva

andre.silva@udesc.br

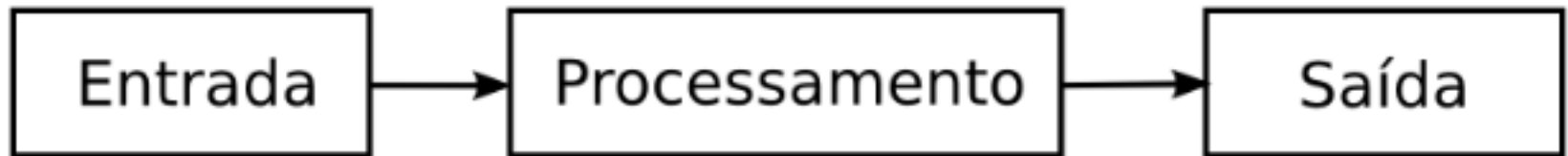
Roteiro

- Programação – conceito iniciais
- Algoritmos
- Linguagens de programação
- Representação de dados (introdução)

Partes de um Algoritmo

- Um algoritmo quando programado num computador é constituído pelo menos das 3 partes, sendo elas:
 1. Entrada de dados;
 2. Processamento de dados;
 3. Saída de dados;
- Na parte de entrada, são fornecidas as informações necessárias para que o algoritmo possa ser executado. Estas informações podem ser fornecidas no momento em que o programa está sendo executado ou podem estar embutidas dentro do mesmo.

Partes básicas de um Algoritmo



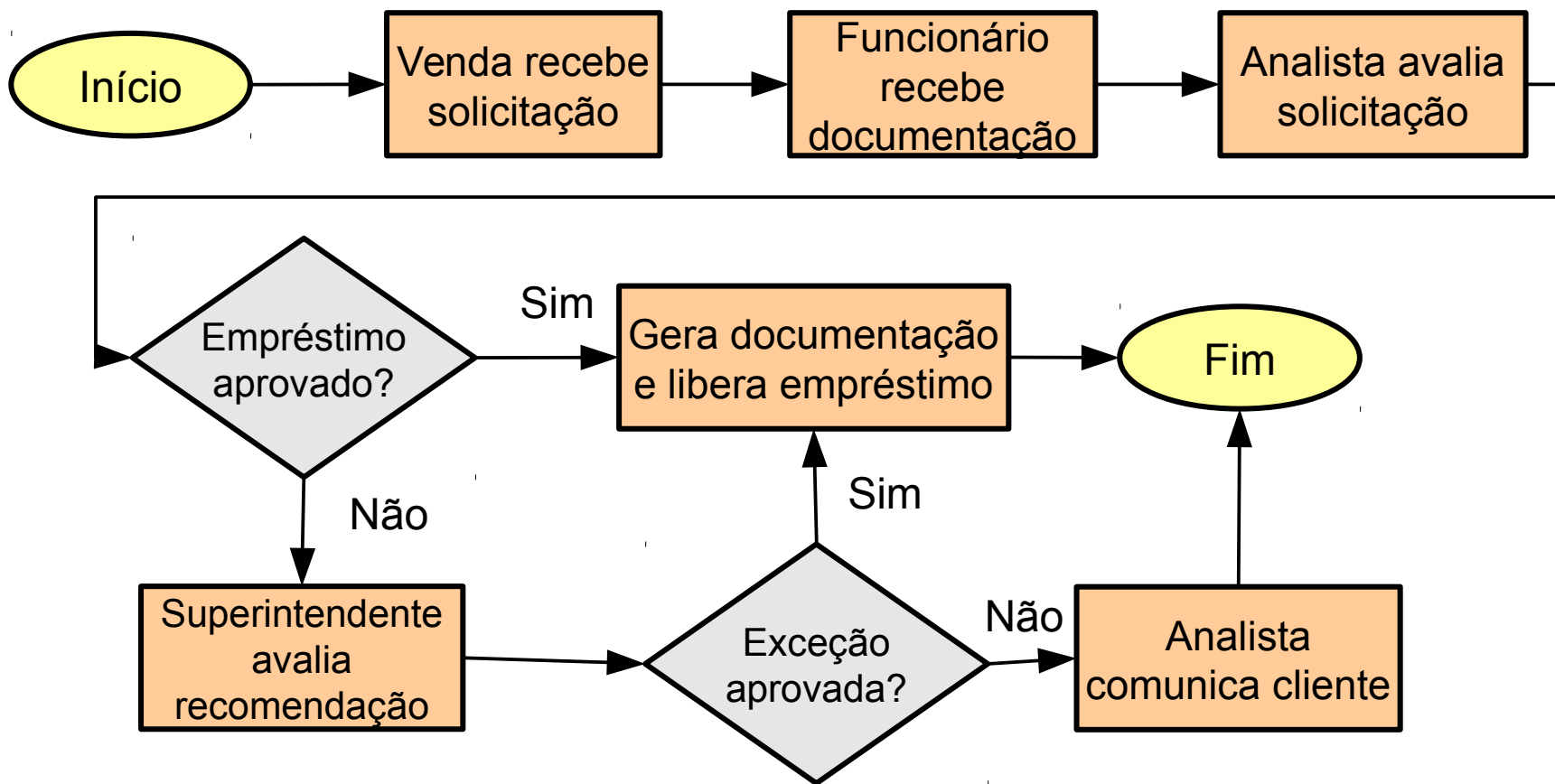
Representações

- Fluxograma
- Pseudocódigo
- Outros diagramas (ER, UML,...)

Fluxograma

- Os fluxogramas são uma apresentação do algoritmo em formato gráfico.
- Cada ação ou situação é representada por uma caixa.
- Tomadas de decisões são indicadas por caixas especiais, possibilitando ao fluxo de ações tomar caminhos distintos.

Exemplo



Fluxograma

- O início e o fim do algoritmo são marcados com uma figura elíptica; as ações a serem executadas estão em retângulos; sendo que as estruturas de controle condicionais estão em losangos e indicam duas possibilidades de prosseguimento do algoritmo, uma para o caso da expressão avaliada (condição) ser verdadeira e outra para o caso de ser falsa.

Pseudocódigo

- O pseudocódigo é uma maneira intermediária entre a *linguagem natural* e uma *linguagem de programação* de representar um algoritmo. Ela utiliza um conjunto restrito de palavras-chave, em geral na língua nativa do programador, que tem equivalentes nas linguagens de programação.

Pseudocódigo

- Além disso, o pseudocódigo não requer toda a rigidez sintática necessária numa linguagem de programação, permitindo que o aprendiz se detenha na lógica do algoritmos e não no formalismo da sua representação. Na medida em que se obtém mais familiaridade com os algoritmos, então o pseudocódigo pode ser traduzido para uma linguagem de programação.

Linguagem Natural

- A linguagem natural é a maneira como expressamos nosso raciocínio e trocamos informação.
- Como é a expressão da cultura de uma sociedade, desenvolvida através das gerações e em diferentes situações, raramente constitui um sistema de regras rígidas que possa ser implementada numa máquina ou que possa ser transcrita logicamente.

Linguagem de Máquina

- Ao contrário dos seres humanos, as máquinas (dentre elas os computadores) são projetados para executar tarefas bem determinadas a partir de determinadas instruções.
- O computador é somente capaz de realizar estritamente as tarefas que lhe forem delegadas e que façam parte do conjunto daquelas ações que ele pode executar.

Linguagem de Máquina

- Além do fato de o computador necessitar que lhe instrua com ações bem específicas, estas ações devem ser passadas para o computador numa linguagem que ele possa entendê-las, chamada linguagem de máquina.
- Esta linguagem é composta somente por números, representados de forma binária, que, sob o ponto de vista do computador, representam as operações e os operandos que serão usados no processamento do programa. Para um ser humano, a linguagem de máquina é difícil de se compreender.

Linguagem de Máquina

```
000000 00001 00010
00110 00000 100000
100011 00011 01000
00000 00001 000100
000010 00000 00000
00000 10000 000000
```



Linguagem de Programação

- Para facilitar a tarefa de programar um computador, foram criadas várias linguagens de programação.
- Estas linguagens são um maneira de tentar escrever as tarefas que o computador vai realizar de maneira mais parecida com a linguagem natural.
- Um programa escrito em uma linguagem de programação é mais fácil de ser implementado, compreendido e modificado, se comparado à linguagem de máquina.

Linguagens de Programação

- As linguagens de programação são um meio termo entre a linguagem de máquina e a linguagem natural.
- Por isso são classificadas de acordo com o nível entre a linguagem natural ou de máquina. As linguagens muito parecidas com linguagem de máquina são chamadas de **linguagens de baixo nível** e suas instruções parecem-se muito com aquelas que serão executadas pelo processador (Ex.: *Assembly*). As **linguagens de alto nível** são as que guardam mais semelhanças com a linguagem natural (Ex.: *Pascal, Java, Perl, Python*).

Linguagem de Programação

- Como o processador não pode executar o código numa linguagem de programação, esta deve ser traduzida em código de máquina antes de ser executada.
- Para isso precisamos de ferramentas como **montadores, interpretadores e compiladores.**

Linguagem de Programação

- **Montadores** (*assemblers*): é um tradutor para a linguagem de montagem (*Assembly*) de um computador em particular. Geralmente uma instrução de linguagem simbólica (de montagem) para uma instrução de máquina.
- **Compiladores**: são tradutores que mapeiam programas escritos em linguagem de alto nível para programas equivalentes em linguagem simbólica ou de máquina.

Linguagem de Programação

- Por vezes um compilador irá gerar uma linguagem de montagem como sua linguagem alvo e, em seguida, contar com um montador para concluir a tradução para o código executável (instrução de máquina).
- **Interpretadores:** é um tradutor de linguagens, assim como um compilador. A diferença é que o interpretador executa o programa fonte de imediato, em vez de gerar um código executável que seja executado após o término da tradução (Ex. VisuAlg).

Representação de Dados

- Para que seja possível armazenar e manipular dados no computador é necessário representá-los internamente de alguma forma.
- Nós seres humanos, representamos nossos números usando um sistema que chamamos de **sistema decimal** (ou sistema na base 10).

Representação de Dados

- Nos caso dos computadores digitais, a notação que é utilizada possui apenas 2 algarismos ou dígitos para representar uma quantidade desejada, 0 e 1.
- Esse sistema de representação é chamado de **sistema binário** (ou sistema na base 2) e utiliza a noção de ligado/desligado, ou verdadeiro/falso, ou 0/1.

Representação de Dados

- Pelo fato de um número precisar de muitos algarismos para ser expresso no sistema binário, outras formas de representação auxiliares também são utilizadas nos computadores, como por exemplo a representação pelo sistema **hexadecimal** (ou sistema na base 16) que utiliza 16 dígitos (0 1 2 3 4 5 6 7 8 9 A B C D E F), e a representação no sistema **octal** (ou sistema na base 8) que utiliza 8 dígitos (0 1 2 3 4 6 7).

Tipos Primitivos

- Os dados em um computador devem ser armazenados de acordo com o tipo de informação que se deseja representar e com o tipo de operação que será realizada com eles.
- A representação correta e adequada de uma informação permite otimizar os recursos computacionais disponíveis, além de acelerar o processamento.

Tipos Primitivos

- Os tipos de dados mais comuns encontrados na maioria das linguagens de programação são:
 - **Inteiro:** são os números pertencentes ao conjunto dos inteiros, isto é, que não possuem parte fracionária. Ex.: 2, 27, -32, 0.
 - **Real:** são os números pertencentes ao conjunto dos Reais, isto é, que podem possuir parte fracionária. Também são chamados de ponto flutuante devido à maneira como o computador os armazena. Ex.: 5.0, 2.12, -3.5, 3.1515692.

Tipos Primitivos

- **Caractere:** são os valores pertencentes ao conjunto de todos os caracteres numéricos (0...9), alfabéticos (a...z,A...Z) e especiais (! @ # \$ % * ~ = ç...). Esse conjunto também é conhecido como conjunto de caracteres **alfanuméricos**. Os caracteres alfanuméricos são armazenados internamente no computador na forma numérica (binária) utilizando o padrão ASCII.

Tipos Primitivos

- Cadeia de caractere (string): um conjunto de caracteres para formar palavras e outros tipos de informações textuais. Ex.: "João da Silva", "Rua Porto Alegre", "Cidade de Joinville". Nestes exemplos, as aspas duplas (") são usadas para indicar o início e o fim das cadeias de caracteres, porém não fazem parte da informação contida nas mesmas. É importante ressaltar que o espaço em branco entre as palavras também é um caractere.

Tipos Primitivos

- Lógico: é utilizado para representar informações que só podem assumir dois valores, o valor verdadeiro (V) ou o valor falso (F). Estes valores também podem ser entendidos como: ligado/desligado, 1/0, alto/baixo, fechado/aberto, etc.

Constantes e Variáveis

- Dentro de um algoritmo podemos encontrar basicamente duas classes diferentes de dados, os dados **constantes** e os **variáveis**.
- Um dado é uma constante quando seu valor não se altera ao longo do tempo em que o algoritmo é executado, ou seja, permanece o mesmo desde o início até o final da execução. Já um dado que pode ter seu valor alterado durante a execução do programa é tido como uma variável.

Manipulação de Dados

- Para que os dados sejam manipulados no computador, é necessário que estes estejam associados a um nome - um **identificador**.
- Sendo assim, toda vez que se deseja acessar uma determinada informação utilizamos o seu identificador para recuperamos o conteúdo que está localizado dentro da memória.

Identificador

- A nomeação dos identificadores deve obedecer a algumas regras, sendo elas:
 1. Sempre começar com um caractere alfabético ou sublinhado (_);
 2. Pode ser seguido por um ou mais caracteres alfanuméricos;
 3. Não conter caracteres especiais nem espaços com exceção do sublinhado.
 4. Não é permitido utilizar palavras reservadas (palavras próprias da linguagem de programação, como os comandos, tipos de variáveis, etc).

Identificador - exemplos

- Exemplos de identificadores válidos:
telefone, _nome, num_clientes, ftQQfIS.
- Exemplos de identificadores inválidos:
(ee), nome@udesc.br, 123abc, :p

Identificadores

- Ao nomearmos os identificadores dos nossos dados é conveniente usarmos palavras mnemônicas, ou seja, palavras que nos façam lembrar o caráter do conteúdo armazenado. Isso facilita a leitura do código programado e possibilita uma melhor documentação do mesmo.
- Por exemplo, ao armazenarmos o nome completo, a idade e a quantidade de filhos de uma pessoa, é mais prático e coerente usarmos os identificadores `nome_sobrenome`, `idade` e `NumFilhos` do que usarmos nomes aleatórios como `X`, `Y`, `Z`.