Introdução à Ciência da Computação

Priscila C. Calegari

Departamento de Informática e Estatística

Março de 2024

Tópicos

1. Apresentação da disciplina
2. Algoritmos.
3. Como funciona um computador.
4. Linguagens de programação, compiladores e interpretadores.

Algoritmos

Um algoritmo pode ser definido como uma sequência precisa de passos que visam atingir um objetivo bem definido.

Algoritmos fazem parte do nosso dia-a-dia:

- Tarefas cotidianas,
- Resolução de problemas,
- Simulações,
- Inteligência artificial,
- entre outras atividades...

Algoritmos

Como construir um algoritmo?

- Definir ações simples e sem ambiguidade.
- Organizar as ações de forma ordenada.
- Estabelecer as ações dentro de uma sequência finita de passos.

Dado um problema, vamos desenvolver algoritmos que possam servir como uma solução. Depois vamos usar o computador para automatizar a execução. Assim, a programação é uma habilidade que nos permite transformar um algoritmo em um programa que possa ser executado por um computador. Os programas são escritos por meio de linguagens de programação.

Algoritmos

Exemplo cotidiano

Troca de lâmpada (Versão 1):

1. Pegar uma escada.
2. Posicionar a escada embaixo da lâmpada.
3. Pegar uma lâmpada nova.
4. Subir na escada.
5. Retirar a lâmpada velha.
6. Colocar a lâmpada nova.
7. Guardar a escada.

Problemas? Podemos melhorar o algoritmo?

Algoritmos

Exemplo cotidiano

Troca de lâmpada (Versão 2):

1. Pegar uma escada.
2. Posicionar a escada embaixo da lâmpada.
3. Pegar uma lâmpada nova.
4. Acionar o interruptor.
5. **Se** a lâmpada não acender **então**:
   (5.1) Subir na escada.
   (5.2) Retirar a lâmpada velha.
   (5.3) Colocar a lâmpada nova.
6. Guardar a escada.

Ainda Podemos melhorar o algoritmo?

Algoritmos

Exemplo cotidiano

Troca de lâmpada (Versão 3):

1. Pegar uma escada.
2. Posicionar a escada embaixo da lâmpada.
3. Acionar o interruptor.
4. **Enquanto** a lâmpada não acender **faça**:
   (4.1) Pegar uma lâmpada nova.
   (4.2) Subir na escada.
   (4.3) Retirar a lâmpada velha.
   (4.4) Colocar a lâmpada nova.
   (4.5) Acionar o interruptor.
5. Guardar a escada.

Computação

Um brevíssimo histórico ...
Há muitos anos as máquinas vem sendo utilizadas para nos auxiliar em cálculos e tarefas.

- Os termos **algarismo** e **algoritmo** tem a origem no nome do matemático Persa, Abu Abdullah Muhammad ibn Musa **Al-Khwarizmi** (780-850).
- 1642 - Máquina de adição mecânica (Pascal).
- 1673 - Aperfeiçou a máquina de Pascal (4 operações) (Leibniz).
- 1835 - Máquina análitica, um computador mecânico programável (Babbage)
- Augusta Ada King (Ada Lovelace) (1815 – 1852) publicou o primeiro algoritmo para a máquina de Babbage. A primeira programadora da história.
- 1847 - Álgebra Booleana, base da computação lógica (Boole).
- 1890 - Herman Hollerith construiu uma máquina programável capaz de processar dados.

Computação

- 1936 - Alan Turing criou um modelo teórico de computador (Máquina Universal). Ideia de Computabilidade (Quais problemas podem ser resolvidos com computador?)
- 1938 – 1942 - Z1 computador eletrômecânico (Konrad Zuse) e ABC (Atanasoff-Berry *Computer*) computador eletrônico digital.
- Outros computadores, Colossus e Mark I 1944 e ENIAC (*Eletronic Numeric Integrator And Calculator*) com 30 metros de comprimento, 3 metros de largura e 30 toneladas.
- 1946 - Arquitetura de Von Neumann: programa armazenado no computador como os dados. Base dos computadores modernos, possui três características: (1) Codificação de instruções; (2) Armazenamento em memória das instruções e dos dados; (3) Busca de instruções a cada processamento na memória.
- Grace Hopper (1953) - desenvolve a primeira Linguagem de Programação (COBOL).
- 1970 - Linguagem C e UNIX.

Computação

- A partir da década de 70:
- 1971 – 1973 Ray Tomlinson cria um sistema de correio eletrônico e Robert Metcalfe o sistema de conectividade *Ethernet*.
- 1975 - Bill Gates e Paull Allen fundaram a *Microsoft Corporation*.
- 1976 - Steve Jobs, Steve Wozniak e Ronald Wayne fundaram a *Apple Computer*.
- Na década de 80 foram lançados os primeiros computadores pessoais IBM PC, Mac, ...
- Na década de 90, novos processadores Intel e os precursores dos *smartphones*, ...
- em 1998, Larry Page e Segey Brin criaram a Google.
- Nos anos 2000 – 2010 acompanhamos a evolução *smart* e as redes sociais e o desenvolvimento dos primeiros computadores quânticos.
- 2020 – OpenAi se populariza com as ferramentas de Inteligêncial Artificial e computação quântica.

Programação e Computação

- Além de realizar operações aritméticas, os computadores podem ser programados.
- Os programas de computador permitem a repetição de instruções.
- O **hardware** é a parte física do computador. A parte responsável pelo funcionamento do *hardware* é chamada **software**.
- O objetivo da programação é criar *softwares* para resolver algum problema computacional.

Programação e Computação

- Além de realizar operações aritméticas, os computadores podem ser programados.
- Os programas de computador permitem a repetição de instruções.
- O **hardware** é a parte física do computador. A parte responsável pelo funcionamento do *hardware* é chamada **software**.
- O objetivo da programação é criar *softwares* para resolver algum problema computacional.

Programação e Computação

- A (ciência da) computação vai além da programação. Como ciência, ela estuda os fundamentos teóricos da informação (dados) e da computação (instrução ou transformação de dados) e as considerações práticas para aplicar essa teoria em sistemas computacionais.

- Para a solução de problemas computacionais vamos desenvolver algoritmos e implementá-los usando a linguagem de programação *Python*.

- *Python* possui uma sintaxe clara e concisa que favorece a legibilidade do código fonte (programa), tornando-a mais produtiva (rápido desenvolvimento).

Programação e Computação

- A (ciência da) computação vai além da programação. Como ciência, ela estuda os fundamentos teóricos da informação (dados) e da computação (instrução ou transformação de dados) e as considerações práticas para aplicar essa teoria em sistemas computacionais.

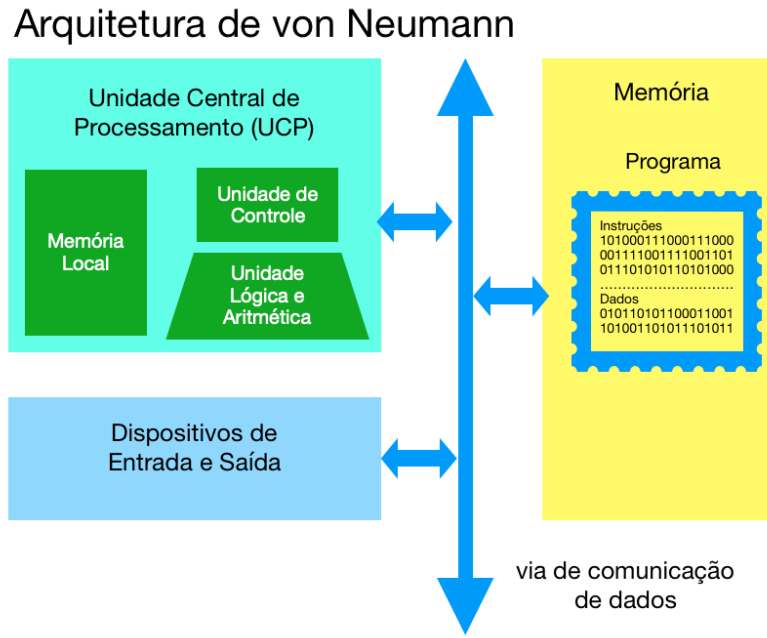
- Para a solução de problemas computacionais vamos desenvolver algoritmos e implementá-los usando a linguagem de programação *Python*.

- *Python* possui uma sintaxe clara e concisa que favorece a legibilidade do código fonte (programa), tornando-a mais produtiva (rápido desenvolvimento).

Como funciona um computador

A maioria dos computadores segue a estrutura de Von Neumann, composta por:

- UCP (Unidade Central de Processamento): composta pela Unidade Lógico-Aritmética (ULA) e Unidade de Controle (UC);
- Memória;
- Dispositivos de entrada e saída (E/S).



Fonte: `https://en.wikipedia.org/wiki/Von_Neumann_architecture`

Como funciona um computador

- Os **dispositivos de entrada e saída** são necessários para que o computador se comunique com outros computadores ou com as pessoas (teclado, microfone, disco, internet, ...) e transmita resultados (impressora, monitor, ...).

- A **memória** pode ser vista como uma lista de células. Cada célula possui um endereço numerado sequencialmente que armazena uma quantidade fixa de *informação*. A informação pode ser uma **instrução**, que diz ao computador o que deve ser feito, ou **dados**, informação que deve ser processada pelo computador.
- Os dados ficam armazenado na forma binária. Um dígito binário pode assumir 2 valores (0 ou 1) e é chamado *bit*. Um *byte* é composto por 8 *bits*.
- A capacidade da memória é representada em *kilobytes* (KB) $2^{10} = 1024$ *bytes*, *megabytes* (MB) 2^{10} *kilobytes*, *gigabytes* (GB) 2^{10} *megabytes*, ...

Como funciona um computador

- Os **dispositivos de entrada e saída** são necessários para que o computador se comunique com outros computadores ou com as pessoas (teclado, microfone, disco, internet, ...) e transmita resultados (impressora, monitor, ...).

- A **memória** pode ser vista como uma lista de células. Cada célula possui um endereço numerado sequencialmente que armazena uma quantidade fixa de *informação*. A informação pode ser uma **instrução**, que diz ao computador o que deve ser feito, ou **dados**, informação que deve ser processada pelo computador.
- Os dados ficam armazenado na forma binária. Um dígito binário pode assumir 2 valores (0 ou 1) e é chamado *bit*. Um *byte* é composto por 8 *bits*.
- A capacidade da memória é representada em *kilobytes* (KB) $2^{10} = 1024$ *bytes*, *megabytes* (MB) 2^{10} *kilobytes*, *gigabytes* (GB) 2^{10} *megabytes*, ...

Como funciona um computador

- Os **dispositivos de entrada e saída** são necessários para que o computador se comunique com outros computadores ou com as pessoas (teclado, microfone, disco, internet, ...) e transmita resultados (impressora, monitor, ...).

- A **memória** pode ser vista como uma lista de células. Cada célula possui um endereço numerado sequencialmente que armazena uma quantidade fixa de *informação*. A informação pode ser uma **instrução**, que diz ao computador o que deve ser feito, ou **dados**, informação que deve ser processada pelo computador.
- Os dados ficam armazenado na forma binária. Um dígito binário pode assumir 2 valores (0 ou 1) e é chamado *bit*. Um *byte* é composto por 8 *bits*.
- A capacidade da memória é representada em *kilobytes* (KB) $2^{10} = 1024$ *bytes*, *megabytes* (MB) 2^{10} *kilobytes*, *gigabytes* (GB) 2^{10} *megabytes*, ...

Como funciona um computador

- A **ULA** desempenha duas operaçõe: aritméticas (soma) e comparações.
- A **UC** é responsável por todo o resto: leitura de instruções e dados da memória ou dos dispositivos E/S., decodificação de instruções, alimentação da ULA e envio de resultados para a memória ou dispositivos E/S.
- Um programa de computador é uma sequência de instruções que é executada pela **UCP**.
- A **UCP** contém um pouco de memória local (grupo de registradores) para trabalhar os dados internamente. A ULA e a UC estão inseridas no microprocessador.
- O **Apontador de Instruções** possibilita a execução de um programa. Ele aponta para o endereço de memória que contém a próxima instrução a ser executada.

Como funciona um computador

- A **ULA** desempenha duas operaçõe: aritméticas (soma) e comparações.
- A **UC** é responsável por todo o resto: leitura de instruções e dados da memória ou dos dispositivos E/S., decodificação de instruções, alimentação da ULA e envio de resultados para a memória ou dispositivos E/S.
- Um programa de computador é uma sequência de instruções que é executada pela **UCP**.
- A **UCP** contém um pouco de memória local (grupo de registradores) para trabalhar os dados internamente. A ULA e a UC estão inseridas no microprocessador.
- O **Apontador de Instruções** possibilita a execução de um programa. Ele aponta para o endereço de memória que contém a próxima instrução a ser executada.

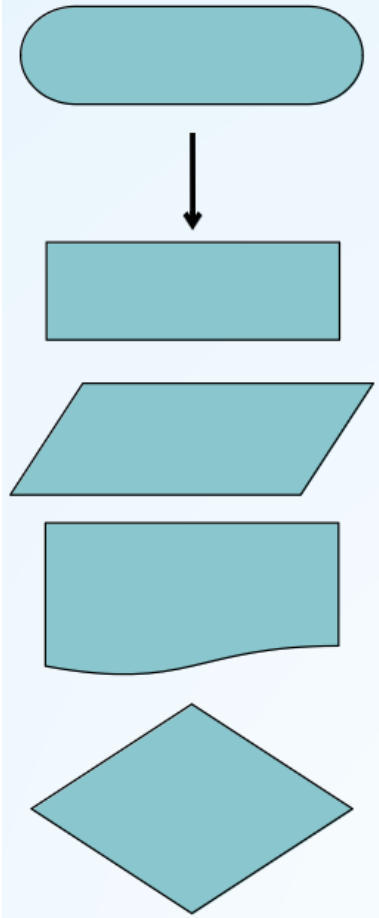
Como representar algoritmos

Um programa consiste em uma sequência de instruções definidas por um algoritmo.

- **Narrativa:** maneira informal.
- **Pseudocódigo:** independente da linguagem de programação.
- **Fluxograma:** representação gráfica.

Como representar algoritmos

Símbolos do 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 utilizado para indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.

Exemplo de Algoritmo

- Narrativa:

Algoritmo para a divisão inteira de a por b, usando somas e subtrações.

1. Receba os valores de a e b;
2. Inicialize c com o valor zero;
3. Enquanto $a > b$ faça:
   3.1. Incremente o valor c com 1;
   3.2. Subtraia o valor b de a;
4. Mostre o valor de c;

Exemplo de Algoritmo

- Pseudocódigo:

Algoritmo para a divisão inteira de a por b, usando somas e subtrações.

1. Receba a e b (números inteiros)
2. $c \leftarrow 0$
3. Enquanto $a > b$ faça:
   3.1. $c \leftarrow c + 1$
   3.2. $a \leftarrow a - b$
4. Escreva c

Obs. O símbolo $\leftarrow$ representa a atribuição de valor. Utilizamos o símbolo $=$ para comparar valores.

Programas

- Um **programa** é uma sequência de instruções que especificam como executar uma computação.

- A estrutura básica de um programa em diferentes linguagens:

  (1) **Entrada:** recebe dados dos dispositivos de entrada ou de um arquivo.

  (2) **Saída:** transmite os resultados para os dispositivos de saída ou para um arquivo.

  (3) **Lógica e matemática:** realiza operações aritméticas e lógicas.

  (4) **Execução condicional:** verifica se alguma condição é satisfeita antes de executar uma sequência de comandos.

  (5) **Repetição:** Realiza algumas instruções repetidamente.

Linguagem de Programação

- Um programa executado pela UCP é escrito em **linguagem de máquina** ou **linguagem de baixo nível**.
- A dificuldade de se programar na **linguagem de máquina** fez com que surgissem as diversas linguagens de programação existentes, chamadas **linguagens de alto nível**.
- O programa escrito na linguagem de alto nível (programa/código fonte) precisa ser convertido para um programa em linguagem de máquina.
- Dois tipos de programa fazem esta conversão: **compiladores** e **interpretadores**.

Linguagem de Programação

• O compilador lê um programa e o traduz completamente antes que o programa comece a ser executado.

• O interpretador lê um programa escrito em uma linguagem de alto nível e o executa, ou seja, faz o que o programa diz.

Ambientes de desenvolvimento

- Instalação do Python - veja apresentação da disciplina.
- Versão online: Colab, moodle, ...
- O *Python Shell* faz parte do ambiente Python que permite você digitar comandos de forma iterativa.
- O *Spyder* e o *Visual Studio* são ambiente integrados de desenvolvimento, pois reune várias ferramentas em um só lugar, dentro de uma mesma interface gráfica. Por exemplo, editor, *Python shell*, o depurador de programas, ...

Resumo

- Um **algoritmo** é uma sequência de passos bem definida, que visa atingir um objetivo.
- Um **programa** é uma sequência de instruções que especificam como executar uma computação.

- A estrutura básica de um programa em diferentes linguagens:

  (1) **Entrada:** recebe dados dos dispositivos de entrada ou de um arquivo.

  (2) **Saída:** transmite os resultados para os dispositivos de saída ou para um arquivo.

  (3) **Lógica e matemática:** realiza operações aritméticas e lógicas.

  (4) **Execução condicional:** verifica se alguma condição é satisfeita antes de executar uma sequência de comandos.

  (5) **Repetição:** Realiza algumas instruções repetidamente.

Exercícios:

1. Escreva um algoritmo que receba dois numéros inteiros e devolva a soma destes números.
2. A prefeitura de uma cidade gostaria de atualizar as informações dos terrenos de seus moradores. Supondo que todos os terrenos da cidade são retangulares, escreva um algoritmo que receba as medidas de dois lados do terreno e calcule a área e o perímetro do terreno.
3. Um professor quer calcular as notas finais dos estudantes da sua turma. A nota final é composta por 75% da média aritmética de duas provas e 25% da nota de um trabalho. Escreva um algoritmo que receba as três notas e calcule a nota final de **um** estudante.
4. A tarifa de energia elétrica convencional de uma residência normal é de 0.53224 R$KWh. Escreva um algoritmo que receba o consumo mensal de uma residência e calcule o valor a ser pago pelo consumo mensal.

Referências

[1] J.C.de Pina Jr. e C.H. Morimoto.
Introdução à Computação com Python: um curso iterativo
https://panda.ime.usp.br/cc110/static/cc110/index.html.

[2] C.H. Marimoto e R.F. Hashimoto.
Introdução à Ciência da Computação em C,
https://www.ime.usp.br/~hitoshi/introducao/.

[3] B. Miller and D. Ranum (Traduzido por C.H. Morimoto, J.C. de Pina Jr. e J.A. Soares.
Como pensar como um Cientista da Computação
https://panda.ime.usp.br/pensepy/static/pensepy/

[4] R. C. S. Schouery.
Algoritmos e Programação de Computadores
https://ic.unicamp.br/~rafael/slides/mc102/unidade02-historia.pdf

[5] Aulas de Introdução a Computação com Python: O Caminho do Programa — Como pensar como um Cientista da Computação: Edição Interativa em Python.

https://panda.ime.usp.br/pensepy/static/pensepy/01-Introducao/introducao.html.