

Aula 1

Lógica de Programação e Algoritmos

Prof. Vinicius Pozzobon Borin

1

Conversa Inicial

2

- O objetivo de hoje é introduzir os principais conceitos inerentes a esta disciplina de programação
- Tais conceitos serão recorrentes

3

- A estrutura de conteúdos é a seguinte:
 - O que é lógica?
 - O que são algoritmos?
 - Como representamos algoritmos?
 - Que sistema computacional usamos para executar programas?
 - O que são linguagens de programação?
 - Qual linguagem vamos adotar?

4

Introdução à lógica e aos algoritmos

5

O que é lógica?

- Aristóteles (384 a 322 a.C.) – conceito de lógica (*logos*) – linguagem racional
- Parte da filosofia que se ocupa das formas do pensamento e das operações intelectuais (Michaelis, 2020)

6

- **Raciocínio lógico do nosso dia a dia para realizar atividades**
- **Na computação: maneira pela qual instruções, assertivas e pressupostos são organizados num algoritmo para viabilizar a implantação de um programa (Michaelis, 2020)**

7

O que são algoritmos?

- **Um algoritmo é dado como uma sequência de passos a serem realizados para que determinada tarefa seja concluída, ou um objetivo atingido**

8

- **Exemplo de algoritmo:**
 - Pegue uma fatia de pão de forma
 - Com a ponta da faca, raspe duas vezes na manteiga dentro do pote
 - Com a mesma faca que contém a manteiga, espalhe uniformemente a manteiga em um dos lados do pão de forma
 - No mesmo lado que você espalhou a manteiga, coloque uma fatia de queijo e uma de presunto, esta última em cima da de queijo
 - Em cima das fatias, coloque o outro pão de forma e pronto, seu sanduíche está finalizado

9

- **Passo a passo da equação:**

$$[(a + b) * c + d]$$
- **Realizar o cálculo dentro dos parênteses a+b**
- **Multiplicar o resultado de dentro dos parênteses por c**
- **Por fim, somar com d**

10

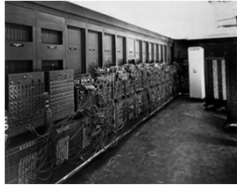
Sistemas de computação

11

- **Segunda Guerra Mundial:**
 - **Cálculo de mísseis**
 - **Mensagens codificadas**
 - **Computadores construídos com milhares de válvulas e relés, pesando toneladas e consumindo montantes gigantes de energia elétrica**

12

- **ENIAC (Electronic Numerical Integrator and Computer)**



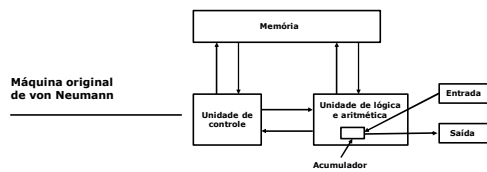
Everett Historical/SHUTTERSTOCK

13

- Percebeu-se a necessidade de mudar a maneira como computadores eram projetados, começando pela aritmética decimal para binária
- John von Neumann – matemático húngaro. Propôs o primeiro computador de programa armazenado e criado

14

A máquina de von Neumann



15

O bit

- Base decimal – dígitos de 0 a 9
- Base binária – dígitos 0 e 1
- Todo e qualquer computador é binário
- **Binary digit** – a menor unidade de armazenamento de dados

16

	Equivale a	Abreviação
8 bits	1 Byte	B
1024 Bytes	1 KiloByte	KB
1024 KB	1 MegaByte	MB
1024 MB	1 GigaByte	GB
1024 GB	1 TeraByte	TB
1024 TB	1 PetaByte	PB

17

A palavra (word)

- **Binary digit** – a menor unidade de armazenamento de dados
- **Palavra (word)** – a menor unidade útil de manipulação do dado

18

O sistema operacional

- Define quais *softwares* e quando serão executados
- Gerencia o uso de memória
- Abstrai o *hardware* para o usuário e para o desenvolvedor

19



Stanislaw Mikulski / SHUTTERSTOCK

20

Representação de algoritmos

Descrição narrativa

- Linguagem natural
- Não utilizada em algoritmos computacionais

1. Ler dois valores (x e y)
2. Verificar se x e y são iguais
3. Se x for igual a y, mostrar a mensagem "Valores iguais!"
4. Se x for diferente de y, mostrar a mensagem "Valores diferentes!"
5. Fim

21

22

Pseudocódigo

- Português estruturado
- Representação mais próxima de um programa computacional, mas sem se preocupar com a linguagem de programação adotada
- Regras definidas
- Linguagem genérica

1. Algoritmo - exemplo
2. Var
3. x, y: inteiro
4. Início
5. Ler (x, y)
6. Se (x = y) então
7. Mostrar ("Valores iguais!")
8. Senão
9. Mostrar ("Valores diferentes!")
10. Fimse
11. Fim

23

24

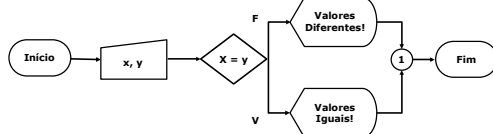
Comparando com a descrição narrativa...

1. Ler dois valores (x e y)
2. Verificar se x e y são iguais
3. Se x for igual a y, mostrar...
4. Se x for diferente de y, mostrar...
5. Fim

Fluxograma

- Representação gráfica de um algoritmo
- Usado para passar a ideia do seu código e organizar o raciocínio lógico
- Simbologia gráfica padrão ISO 5807:1985

Fluxograma do pseudocódigo apresentado anteriormente



Linguagens de programação e compiladores

- Um computador só compreende bits
- Você conseguiria escrever um programa somente usando bits?

```

001000000110100010111001000000110100010111001000001
00010111001001110010010000001010001100100100001011011
01101000110111011001000000101000101001101101101101
1110110110011011001001001011001001101000110001001010
0111001001011001100010110111011010100101110010011000
000000110100011101000010100001010100111110010111100
111000111100101101000001000001111011110010111001100
011010001100100110000101101100110011011000110000101
110011000110110111011011010010111001001100010001100110
00110110000110100001010110011111001011110000110010111
0010110100000100000111110111100101110010010000011100
11011011101000111011011100110000101001010010010001
  
```

lunewind/SHUTTERSTOCK

Altair 8800



Jefferson Schnaider

Linguagens de programação

- Uma linguagem de programação é, portanto, esse conjunto de regras, com palavras-chaves, verbos, símbolos e sequências específicas. Chamamos todo esse conjunto de *sintaxe da linguagem*
- Resultam em instruções compreendidas pelo computador e não geram ambiguidades

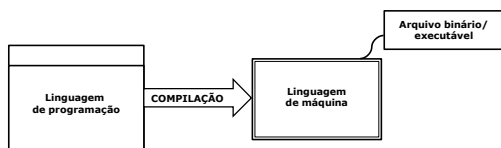
31

Software de compilação

- Se o computador compreende uma linguagem e você trabalha em outra, diferente e ilegível para ele, como o computador entende o código que você faz?
- Ele transforma o código que você escreveu (alto nível) em uma linguagem de máquina (baixo nível) compreendida pelo *hardware*

32

Processo de compilação

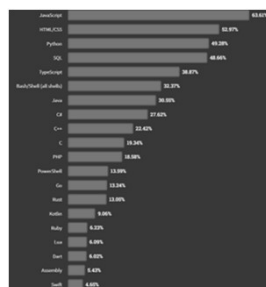


33

- Compilador – transforma um código-fonte em um arquivo binário
- Interpretador – o código não é convertido de uma só vez, mas, sim, executado instrução por instrução à medida que o programa vai requisitando
- Prós x contras

34

Language Ranking: Stack Overflow



35

Linguagem de programação Python

36

- Site oficial da linguagem:
<https://www.python.org/>
- Site oficial no Brasil:
<https://python.org.br/>



37

- O que torna o Python tão popular, amado e utilizado?
 - Linguagem de propósito geral
 - Vasta quantidade de bibliotecas existentes
 - Linguagem simples, intuitiva. Ótima para iniciantes
 - Linguagem multiplataforma
 - Comunidade ativa e atualizações constantes
 - Python Software Foundation (2001)

38

Histórico da linguagem

- Linguagem criada em 1982 por Guido van Rossum
- Centrum Wiskunde & Informatica (CWI), Amsterdam
- Inspiração: linguagem ABC

39

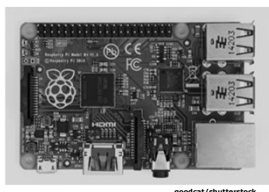
- Monty Python Flying Circus
- Programa de televisão britânico (1969-1974)



40

Exemplos de aplicações Python

- Sistemas operacionais. Linux, macOS
- Raspberry Pi



41

Análise de dados e inteligência artificial



42

Industrial Light and Magical (ILM)



pio3/SHUTTERSTOCK



Alex Cimbal/SHUTTERSTOCK

43



monticello/shutterstock



charnatt/shutterstock

44



sukku_1242/shutterstock

Praxsen_Namu/shutterstock

45

- **O Zen do Python, por Tim Peters**
 - Bonito é melhor que feio
 - Explícito é melhor que implícito
 - Simples é melhor que complexo
 - Complexo é melhor que complicado
 - Plano é melhor que aglomerado
 - Legibilidade faz diferença
 - Agora é melhor que nunca

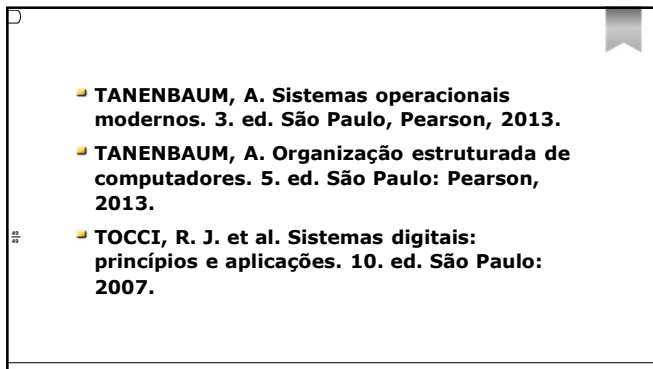
46

Referências

- **PERKOVIC, L. Introdução à computação usando Python – um foco no desenvolvimento de aplicações.** Rio de Janeiro: LTC, 2016.
- **PUGA, S.; RISSETI, G. Lógica de programação e estrutura de dados.** 3. ed. São Paulo: Pearson, 2016.
- **STALLINGS, W. Arquitetura e organização de computadores.** 10. ed. São Paulo: Pearson, 2017.

47

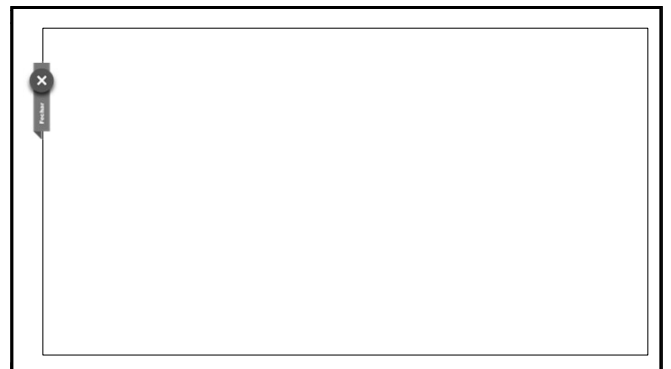
48



Slide 49 displays a list of three references. The first two are by Tanenbaum, A., and the third is by Tocci, R. J. et al. The slide has a small icon in the top left corner and a bookmark icon in the top right corner.

- **TANENBAUM, A. Sistemas operacionais modernos. 3. ed. São Paulo, Pearson, 2013.**
- **TANENBAUM, A. Organização estruturada de computadores. 5. ed. São Paulo: Pearson, 2013.**
- **TOCCI, R. J. et al. Sistemas digitais: princípios e aplicações. 10. ed. São Paulo: 2007.**

49



Slide 50 is an empty slide with a close button (X) in the top left corner.

50