

Lógica de Programação e Algoritmos

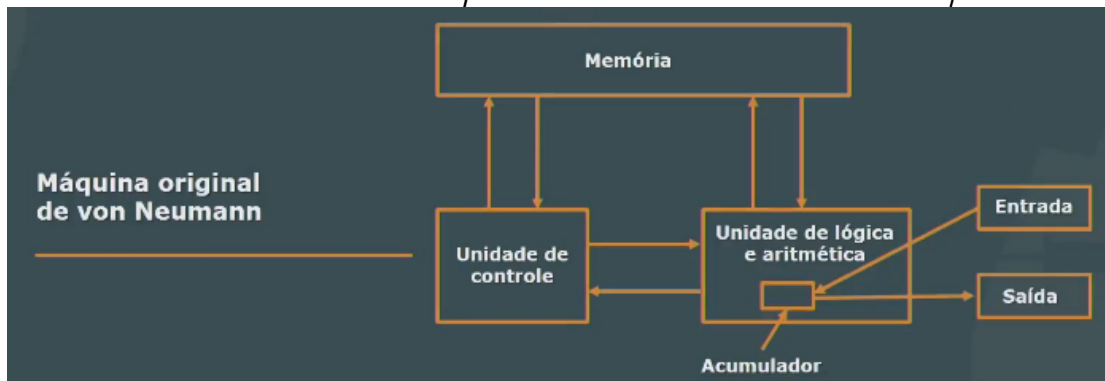
Introdução:

Lógica = Aristóteles conceituou, palavra grega logos, linguagem racional.

Na computação = Maneira pela qual instruções, assertivas e pressupostos são organizados num algoritmo para viabilizar a implantação de programas.

Algoritmos = Sequência de passos para atingir um objetivo.

John Von Neumann = transformou de aritmética decimal para aritmética binária, e desenvolveu a Arquitetura de Von Neumann:



Unidade de Lógica e Aritmética = realiza todos os cálculos do computador.

Unidade de Controle = diz qual vai ser o próximo cálculo, cuida da agenda.

Memória = armazena todos os programas em execução.

Bit = 0 ou 1. É a menor unidade de armazenamento. Também é chamado de binary digit.

Unidades	Equivalente	Abreviação
8 Bits	Byte	B
1.024 Byte	KiloByte	KB
1.024 KiloByte	MegaByte	MB
1.024 MegaByte	GigaByte	GB
1.024 GigaByte	TeraByte	TB
1.024 TeraByte	PetaByte	PB

Sistemas Operacionais = facilitam a vida do programador. Tornam desnecessário escrever programas específicos para cada tipo de máquina, apenas para cada tipo de sistema operacional.

Algoritmos:

Descrição Narrativa = linguagem natural, não se usa em algoritmos computacionais (gera muitas ambiguidades).

Pseudocódigo = português estruturado, como uma linguagem de programação geral, para todas. Tem regras definidas e linguagem geral.

Fluxograma = representação gráfica, passa a ideia geral do algoritmo de forma visual. Seguir o padrão ISO 5807:1985.

Linguagem de Programação:

Conjunto de regras com palavras-chave, símbolos e sequências específicas para escrever algoritmos.

O computador só entende 0 e 1, bits, a gente não. Até tiveram uns malucos que fizeram isso, no Altair 8.800, mas enfim.

Software de Compilação ou Interpretação = Tem a linguagem de máquina, que o computador entende, com 0 e 1; e tem a linguagem de programação, sem ambiguidades. O software de compilação/interpretação faz uma coisa conversar com a outra.

Compilador = transforma o código em arquivo binário. O código-fonte é perdido.

Interpretador = vai interpretando o código aos poucos, conforme necessário. O código-fonte não é perdido.

Python:

Resolver problemas de propósito geral, não específicos; muitas bibliotecas existentes; é simples, intuitiva, boa para iniciantes; é multiplataforma; tem uma comunidade ativa, com atualizações constantes.

História = surgiu em 1982, por Guido Van Rossum. O nome é derivado do programa Monty Python Flying Circus.

Site oficial = python.org.br

Lista de Erros Contornáveis = <https://docs.python.org/pt-br/3/library/exceptions.html#builtin-exceptions>

Algoritmos:

Receita de bolo para construir algo, só que isso no computador.

Entrada → Processamento → Saída

Dados = informações manipuláveis que ficam na memória. Há os numéricos (inteiros e pontos flutuantes), caracteres e booleanos.

Variável = dado que varia, o nome. Não aceita números, espaços ou caracteres especiais no início do nome (exceto _).

Algoritmos Sequenciais = seguem uma sequência, as instruções são executadas uma após a outra.

Algoritmos Condicionais = condicional simples, composta, aninhada e de múltipla escolha.

Tentar não repetir passos.

Indentação = o espaço de recuo do bloco condicional. Tudo o que estiver em recuo vai ser executado para aquela condição.

Condicionais Simples = tem uma pergunta de verdadeiro ou falso, e dependendo da resposta, pula uma parte, só não executa algo.

Condicionais Compostas = tem uma pergunta de verdadeiro ou falso, e dependendo da resposta, executa algo a parte.

Condicionais Aninhadas = uma condicional dentro da outra.

Estruturas de Condição = as instruções contidas nelas se repetem até que uma condição seja satisfeita.

Truthy e False = true e false. O false também pode ser 0 ou um espaço vazio.

Lambda = função criada para coisas simples, de uma linha só.

Desempacotamento de Parâmetros = não precisa passar o número de termos que serão adicionados.

Variáveis:

Local = só existe dentro da função específica, láááá na raíz. Em outro lugar, em outra função, no programa principal, não.

Global = existe no programa principal, nos códigos que criamos.

Variável Local > Variável Global

Variável Simples = armazena um dado.

Variável Composta = armazena vários dados, um conjunto deles..

Codificação de String:

Os caracteres são armazenados na memória através de números, e existem tabelas de conversões para acessar esses números. A tabela mais antiga é a ASCII (8 bits), mas hoje há outras, como a UNICODE (32 bits).

String = cadeia de caracteres, em sequência.



Relação de métodos para uso com strings

Função/método	Objetivo
startswith	Verifica se caracteres existem no início da string
endswith	Verifica se caracteres existem no final da string
lower	Converte string para minúscula
upper	Converte string para maiúscula
find	Busca a primeira ocorrência de um padrão de caracteres em uma string
rfind	Idêntico ao find, mas inicia a busca da direita para a esquerda
center	Centraliza uma string
ljust, rjust	Ajustam uma string com alinhamentos à esquerda ou à direita, respectivamente
split	Divide uma string
replace	Substitui caracteres em uma string
lstrip, rstrip	Removem espaços em branco à esquerda ou à direita, respectivamente
strip	Remove espaços em branco das extremidades

Relação de métodos para validação de dados em strings

Função/método	Retorna True para uma string com...
isalnum	Somente letras e números; acentos são aceitos
isalpha	Somente letras; acentos são aceitos
isdigit	Somente números
isnumeric	Somente números; aceita também caracteres matemáticos, como frações
isupper	Somente caracteres maiúsculos
islower	Somente caracteres minúsculos
isspace	Somente espaços. Inclui TAB, quebra de linha, retorno etc.
isprintable	Somente caracteres possíveis de serem impressos na tela

Estrutura de Dados:

Conjunto de dados organizados de uma maneira específica na memória.

Tupla = (). Estática e imutável.

Lista = []. Dinâmica, dá pra alterar os dados e tamanho.

Dicionário = {}. Dinâmica, dá pra alterar os dados e tamanho.

PDO = Programação Orientada a Objetos.