

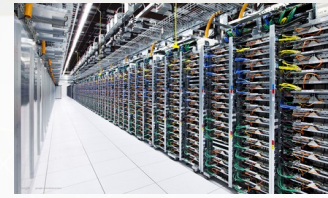
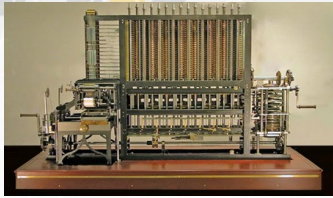
Engenharia de Computação

Fundamentos de Programação

Aula 01 - Introdução

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

Conceitos Básicos



- De forma geral, o que é um Computador?
 - É um equipamento eletrônico capaz de executar quatro tarefas básicas :
 - Receber Dados por uma Entrada de Dados
 - Armazenar Dados
 - Processar e Efetuar Cálculo sobre os Dados
 - Saída de Resultados (geralmente exibidos)

ENTRADA



ARMAZENAMENTO e
PROCESSAMENTO



SAÍDA

Hardware x Software

- Hardware

- Dispositivo e Componentes físicos

- Exemplos: Processadores, Placa Mãe, Periféricos em geral, Memória, Placas diversas (rede, som, etc...)....

- Software

- Conjunto de instruções representadas em vários Bytes que, ao serem executadas pelo Processador geram várias ordens de processamento/cálculo...

- Exemplos: Navegador, editor de texto, jogos, aplicativos, etc...

Tipos de Software

- Básico
 - Necessários para a utilização do sistema computacional
 - **Exemplos:** Sistema Operacional, Drivers
- Aplicativo
 - Que tem aplicação direta para o usuário
 - **Exemplos:** Navegador, editor de texto, jogos, aplicativos, etc...
 - **Programas que serão desenvolvidos nesta disciplina...**



Bits e Bytes

- As informações armazenadas no computador são medidas em BITS e BYTES.
- Medidas de Dados
 - BIT
 - Menor unidade de dado
 - Apresenta os valores 0 ou 1 – (Desligado ou Ligado)

BIT
0

BIT
1

Bits e Bytes

- BYTE

BIT BIT BIT BIT BIT BIT BIT BIT

8 BITS = 1 BYTE

- Conjunto de 8 bits, representa um caractere
- Menor unidade de dados que um computador trabalha
- 1 Byte pode representar $2^8 = 256$ combinações valores
 - Os Dados são representados por várias sequencias de Bytes....
 - Uma tecla digitada gera um sinal elétrico codificado (keycode) que é interpretado com 1 ou mais Bytes (ex: representar a letra "a"), geralmente na tabela ASCII
 - **Ex:** letra "a" em ASCII

0	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---

Bits e Bytes

- Medidas de Quantidade de Dados (base 2)

Medida	Representa o mesmo que:
Bit	0 ou 1 – menor unidade de dado
Byte	Conjunto de 8 bits ou 1 caracter
KiloByte (KB)	2^{10} bytes ou 1.024 bytes
MegaByte (MB)	2^{20} bytes ou 1.048.576 bytes
GigaByte (GB)	2^{30} bytes ou 1.073.741.824 bytes
TeraByte (TB)	2^{40} bytes ou 1.099.511.627.776 bytes

Bits e Bytes

- Padrões IEEE 1541-2002 e IEC 80000-13:2008
 - Evitar confusão base 10 e base 2 por conta de alguns fabricantes
 - Ex: Alguns fabricantes de disco vendem disco de 1 TeraByte como $10^{12} = 1.000.000.000.000$ bytes, sendo na base 2 é de $10^{12} / 240 = 909,49$ GigaBytes
 - Padrão atual que tende a ser adotado (já válido):

Binary Prefixes:				Binary Prefixes:			
Factor	Symbol	Name	Value	Factor	Symbol	Name	Derivation Value
10^3	k	Kilo	1,000	2^{10}	Ki	Kibi	Kilobinary 1,024
10^6	M	Mega	1,000,000	2^{20}	Mi	Mebi	Megabinary 1,048,576
10^9	G	Giga	1,000,000,000	2^{30}	Gi	Gibi	Gigabinary 1,073,741,824
10^{12}	T	Tera	1,000,000,000,000	2^{40}	Ti	Tebi	Terabinary 1,099,511,627,776
10^{15}	P	Peta	1,000,000,000,000,000	2^{50}	Pi	Pebi	Petabinary 1,125,899,906,842,624
10^{18}	E	Exa	1,000,000,000,000,000,000	2^{60}	Ei	Exbi	Exabinary 1,152,921,504,606,846,976
10^{21}	Z	Zetta	1,000,000,000,000,000,000,000	2^{70}	Zi	Zebi	Zettabinary 1,180,591,620,717,411,303,424

Bits e Bytes

- Medidas de Armazenamento de dados (Base 2)
 - Memória principal (RAM). Ex: 4GBytes = 4×2^{30} bytes
 - Memória secundária. Ex: 1TByte = 1×2^{40} bytes
- Medidas de Transferência de Dados (Base 10):
 - Velocidade de conexão/transferência de dados
 - Ex: Taxa de transferência de 20Mbps = 20×10^6 bits/s

• Linguagem de Programação

- Computadores entendem apenas sim/não (zero/um)
 - Números binários ou bits
- Computadores fazem o que queremos
 - Programas são constituído de instruções
 - Instruções: conjunto de bits que o processador entende...
 - Ex: 100011001010000
- Faz-se necessária uma linguagem mais semelhante ao nosso idioma e nossa forma de pensar
 - **Linguagem de Programação**

• Linguagem de Programação

- Conjunto de regras definidas e que devem ser organizadas para serem convertidas em linguagem de máquina (são executadas pelo computador)
 - Cada Linguagem de Programação adota um determinado conjunto de regras.
 - Exemplo: Linguagem C.
- Essas regras devem ser estruturadas segundo uma determinada lógica, que deve ser pensada com base em **algoritmos**

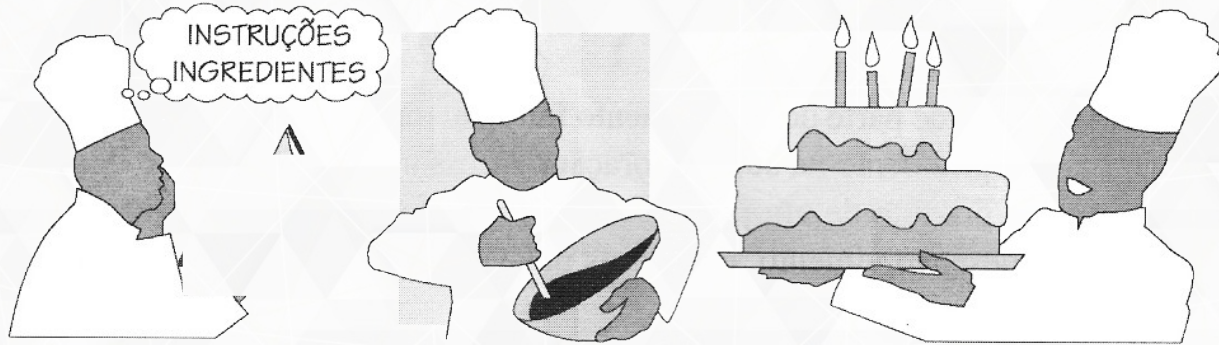
Algoritmo

- Definições:
 - Procedimento passo a passo para a solução de um problema
 - Sequência detalhada de ações a serem executadas para realizar alguma tarefa
 - Conjunto de regras para a solução de um problema
- O termo **algoritmo** não é restrito a computação. Pode ser aplicado a qualquer circunstância que exija a descrição de uma solução.

Exemplo de Algoritmo

- **Receita de Bolo:**

- ◉ **Entrada:** ingredientes e quantidades
- ◉ **Processamento:** regras para o preparo, sequência de inclusão dos ingredientes, cozimento
- ◉ **Saída:** o bolo pronto



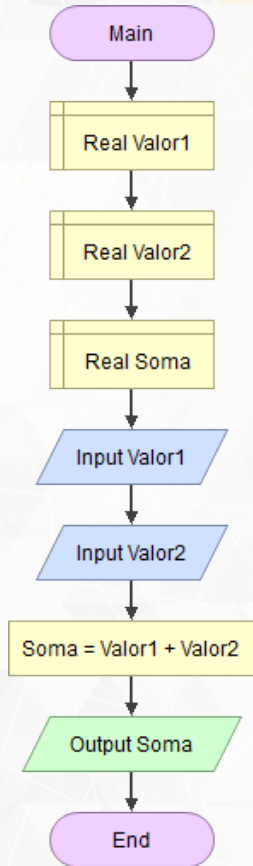
Uma receita de bolo é um algoritmo.

Exemplo de Algoritmo 2

- Troca de um pneu furado
 - Afrouxar ligeiramente as porcas
 - Suspender o carro
 - Retirar as porcas
 - Retirar o pneu furado
 - Pegar o pneu reserva
 - Colocar o pneu reserva
 - Apertar as porcas
 - Abaixar o carro
 - Dar o aperto final nas porcas

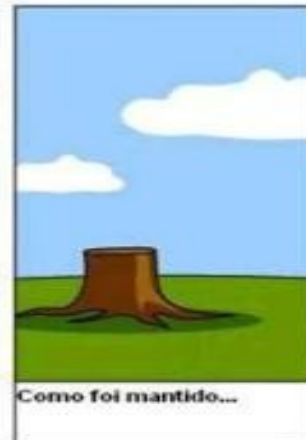
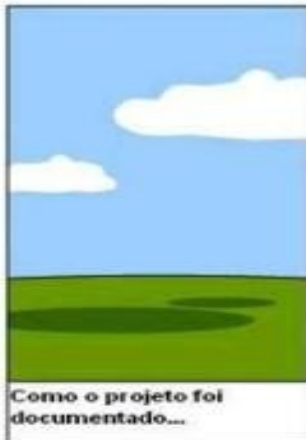
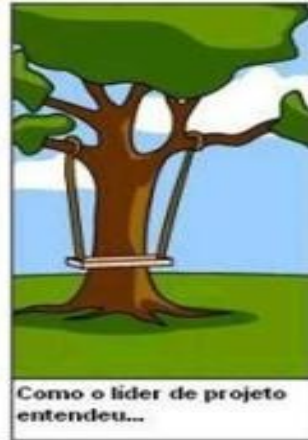


Algoritmo: Somar dois números



- Somar dois números e mostrar na tela
- Iniciar Algoritmo
- Definir local para armazenar 1º valor: **Valor1**
- Definir local para armazenar 2º valor: **Valor2**
- Definir local para armazenar resultado: **Soma**
- Ler 1º valor e armazene em **Valor1**
- Ler 2º valor e armazene em **Valor2**
- Somar **Valor1** com **Valor2** e armazenar em **Soma**
- Mostrar mensagem para mostrar **Soma**
- Fim Algoritmo

Dificuldade: Saber o problema



Passos para um algoritmo

Deve especificar com clareza e de forma correta as instruções que um software deverá conter para que, ao ser executado, forneça resultados esperados.

- **Etapas:**

- **Primeiro:** Conhecer qual é o problema a ser resolvido pelo software (objetivo do programa)
- **Segundo:** Extrair todas as informações a respeito desse problema (dados e operações)
- **Terceiro:** Descrever claramente os passos para se chegar à sua solução. Os passos devem ser colocados em uma sequência lógica.
- **Quarto:** Temos a especificação do algoritmo.