



Conceitos básicos de hardware e software

Disciplina: Algoritmos e Programação
Curso: Engenharia de Computação

Professora: Mariza Miola Dosciatti
mariza@utfpr.edu.br

Objetivos

- Identificar as características de um problema que possui solução computacional.
- Saber conceitos de *hardware* e *software*.
- Conhecer os itens fundamentais de composição de um computador.
- Entender a forma de armazenamento de dados na memória do computador e sua manipulação.
- Entender o conceito de *bit* e *byte*.
- Item da ementa (Plano de ensino):
 - Noções de hardware e software.

Sumário

- Computador
- Problemas resolvidos pelo computador
- Hardware
- Software

Computador

- Computador é uma máquina capaz de executar (realizar as instruções de) **algoritmos** denominados **programas, softwares, sistemas, aplicativos**.
- Um computador é uma máquina programável que processa dados com o objetivo de **resolver problemas**.
- Programar um computador consiste em elaborar a **sequência de instruções (algoritmo na forma de programa)** que define a **resolução de problemas**.

Problemas resolvidos pelo computador

- **Computador x Humano**

- **Computador**

- Tarefas bem definidas e com instruções necessárias e suficientes para resolver o problema.
 - Repetição.
 - Precisão de cálculo.

- **Humano**

- Raciocínio.
 - Análise subjetiva.

Problemas: exemplos

- **Problemas de solução fácil ou difícil para humanos e computadores?**

1. Previsão do tempo.
2. Identificar se uma pessoa está triste.
3. Dividir dois números reais muito grandes.
4. Verificar se um número é primo.
5. Reconhecer uma pessoa que faz tempo que não vê.
6. Responder perguntas que exigem interpretação de texto.
7. Controle de estoque de uma empresa.
8. Determinar a rota mais curta entre n cidades ([problema do caixeiro viajante](#)).
9. Organizar a carga de caixas diferentes em um caminhão ([problema do container](#)).
10. Encher uma mochila com a melhor relação peso x valor ([problema da mochila](#)).
11. [Problemas de Landau](#).

Problemas resolvidos pelo computador (cont.)

- **Problemas completamente especificados:**
 - Possuem todas as entradas necessárias para resolver o problema e as saídas adequadamente identificadas.
 - A solução para o problema é conhecida. Há um método, uma fórmula, procedimentos para resolver o problema. **São conhecidas as ações que devem ser realizadas para obter os valores de saída a partir dos dados de entrada.**
 - A solução é alcançada em tempo factível (finito) e com os recursos disponíveis.

Problemas resolvidos pelo computador (cont.)

- **Problemas P x NP**

- **P (polinomial)** - Problemas que podem ser computados em “tempo aceitável”.
 - **Exemplo:** A multiplicação de dois números.
- **NP (polinomial não determinista)** - Problemas que possuem respostas que podem ser rapidamente testadas, mas não são resolvidos de forma “tão rápida”.
 - **Exemplo:** A fatoração de um número que é produto de dois primos: 13.717.421.
 - É um processo demorado encontrar os números: 3.607 e 3.803.
 - É rápido testar a solução: $(3.607 * 3.803)$

Problemas resolvidos pelo computador (cont.)

- **Problemas P x NP**

- Os **problemas NP** ainda não resolvidos em tempo factível são categorizados de **NP-completo**.

- **Exemplos:**

- Determinar a melhor rota para um número considerável de cidades.
- Quebrar senhas com 10 caracteres (letras, dígitos e símbolos especiais) por meio de teste de todas as possibilidades.
- Encher uma mochila com objetos na melhor combinação entre peso e valor/benefício.

- $P == NP$ ou $P \neq NP$?

- Pode ser que:

- Não há uma forma polinomial de serem resolvidos ($P \neq NP$).
- Há solução, mas ainda não foi descoberta ($P == NP$).

Problemas resolvidos pelo computador (cont.)

- **A solução de um problema deve ser considerada em âmbito geral e não em casos particulares.**
 - As variáveis de entrada possuem valores fornecidos durante a execução do programa.
 - A solução deve ser adequada para um conjunto de valores de entrada especificados.
 - A solução pode abranger um determinado escopo ou parte de um problema.
 - A solução pode não ser a melhor.

Computador: hardware + software

- Computador é uma máquina (**hardware**) capaz de executar programas (**software**).
- O **Hardware** é formado pelos dispositivos **físicos** de um computador.
 - **São exemplos de hardware:** placa-mãe, memória, processador, placa de rede, placa de som, impressora, monitor, etc.
- O **Software** são as **instruções lógicas** do computador. Ou seja, são os todos **programas** que são executados no computador.
 - **São exemplos de software:** Sistemas Operacionais (Windows, Linux, etc.), editores de textos, jogos de computador, etc.

Computador: hardware + software (cont.)

- Um computador é uma máquina eletrônica que:
 - **Recebe dados**
 - De periféricos (teclado, mouse, sensores, ...), meios de armazenamento, aplicativos.
 - **Processa dados**
 - Realizando operações lógicas e aritméticas.
 - **Gera saídas**
 - Dados processados, dados enviados para periféricos e equipamentos (monitor de vídeo, impressora, ...), para aplicativos, para meios de armazenamento.

Hardware: conceito



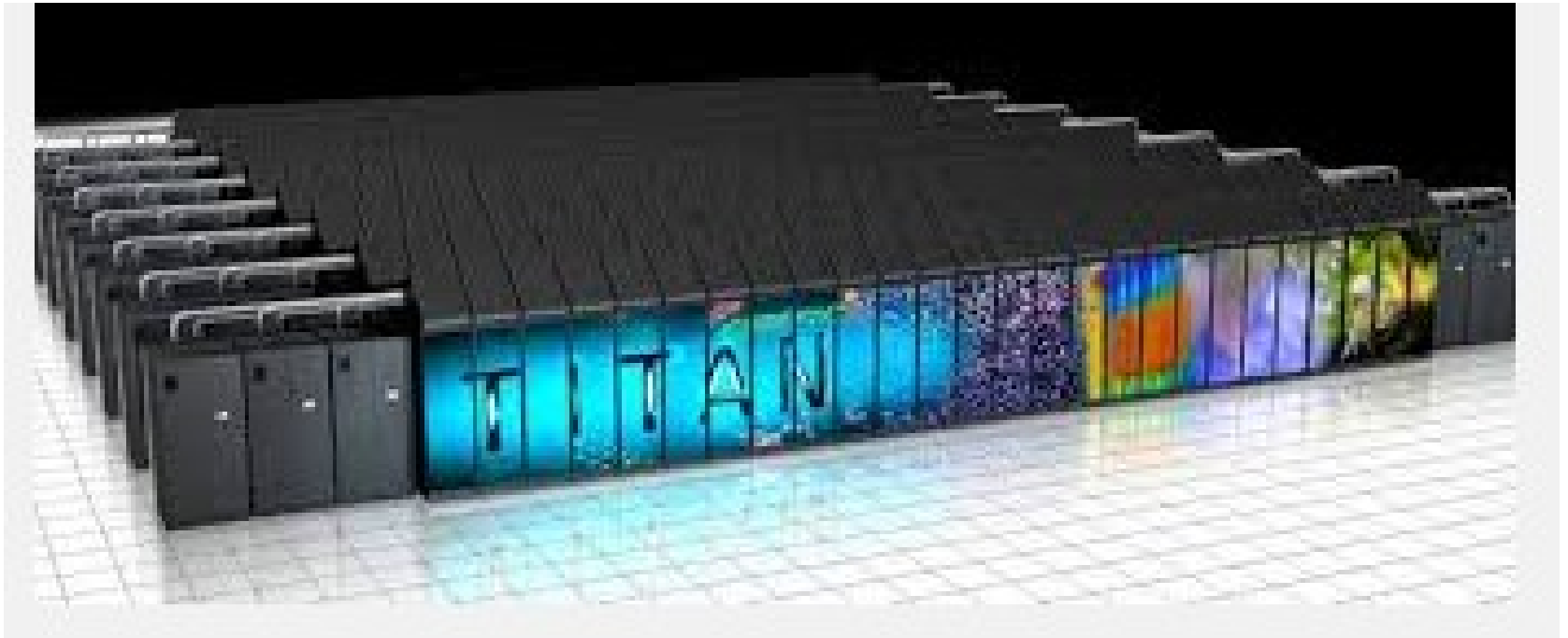
Tupã – Supercomputador brasileiro
(Instituto Nacional de Pesquisas Espaciais (INPE))

Hardware: conceito



[Sunway TaihuLight](#), o supercomputador mais poderoso do mundo

Hardware: conceito



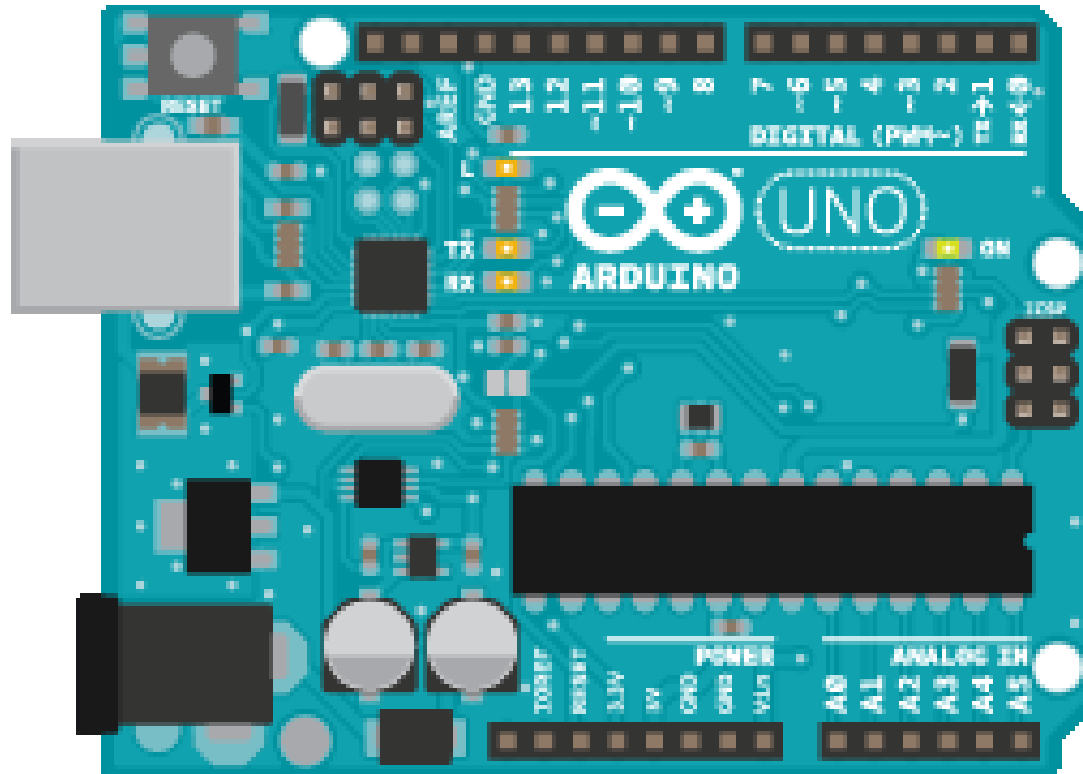
Titan – Supercomputador que faz 20 quatrilhões de operações aritméticas float por segundo

Hardware: conceito

- Arduino é uma plataforma de prototipagem de código aberto baseado no conceito ***easy-to-use*** para hardware e software.
- Placas Arduino são capazes de ler entradas (a luz em um sensor, o apertar de um botão, ou uma mensagem de *Twitter*) e transformá-las em uma saída (ativação de um motor, ligar um LED, publicar algo online).

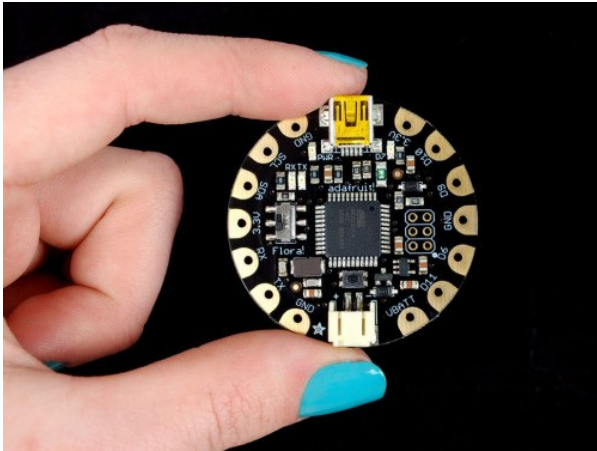
Fonte: <https://www.arduino.cc/en/Guide/Introduction>

Hardware: conceito

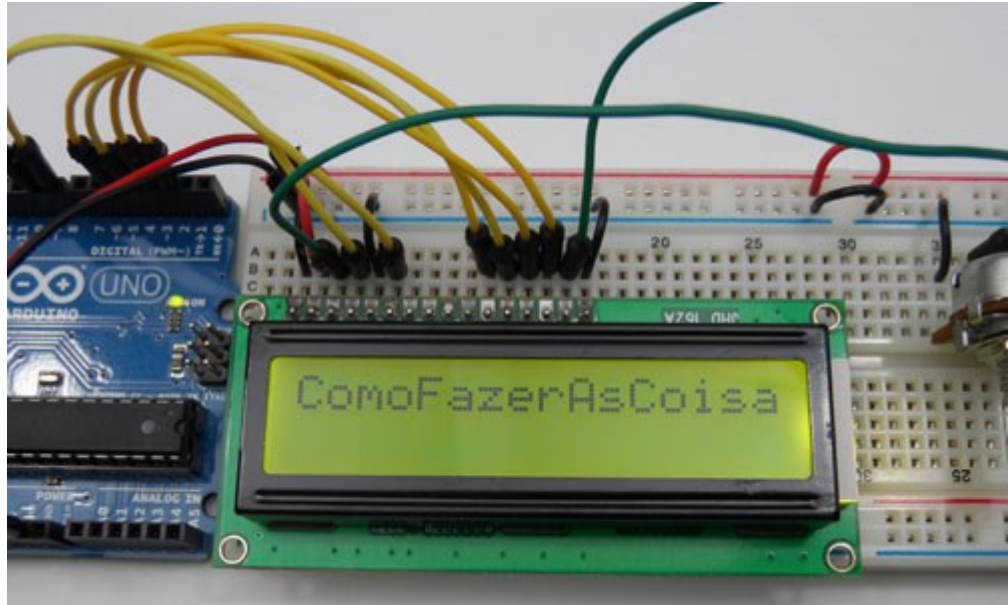


Arduino

Hardware: conceito

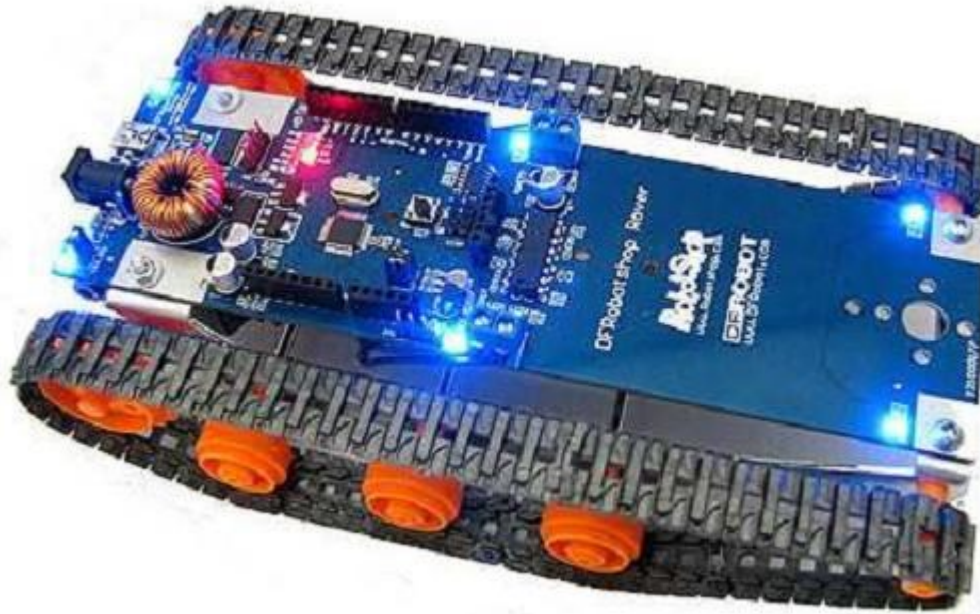


Arduino Flora



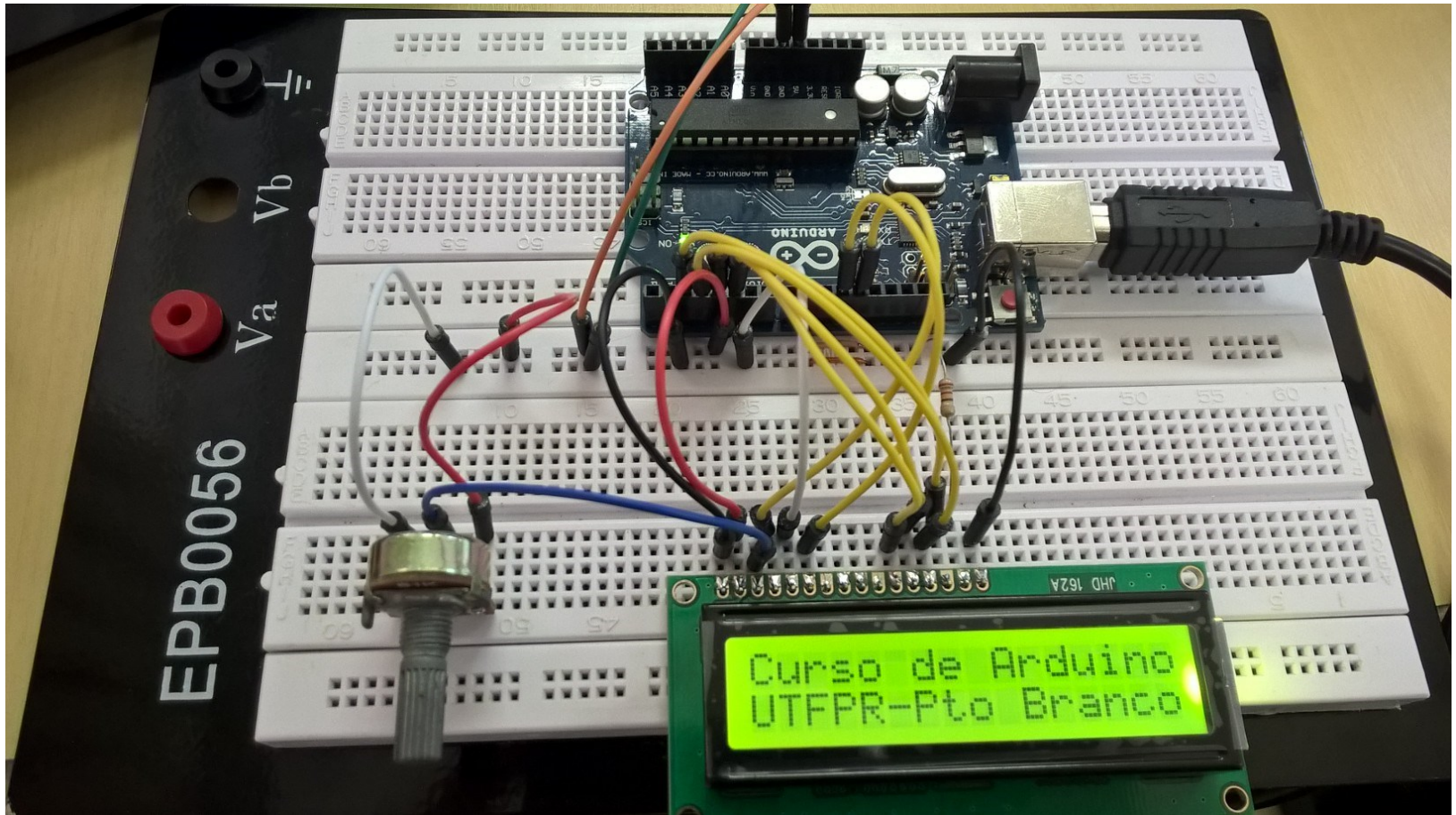
Arduino com display LCD

Hardware: conceito

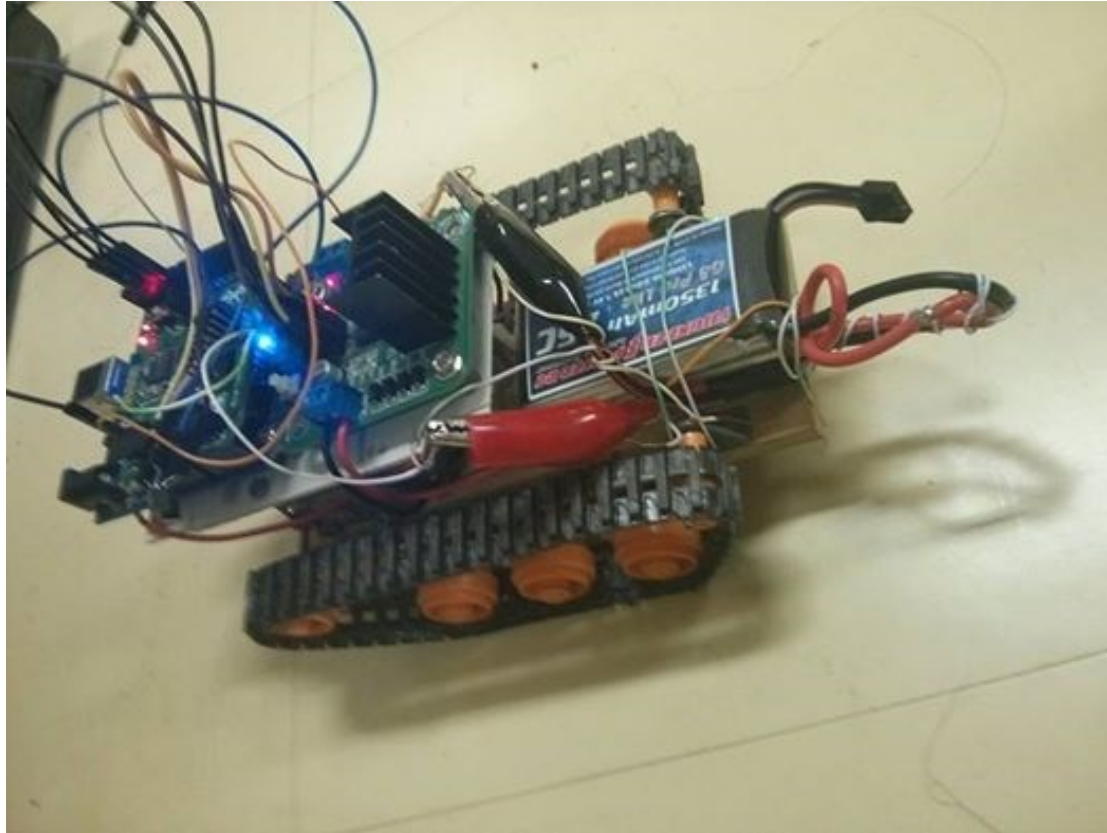


[Arduino em robótica](#)

Hardware: conceito



Hardware: conceito



Hardware: conceito



a.



b.



c.

Fonte: Bezerra (2013, p. 12).

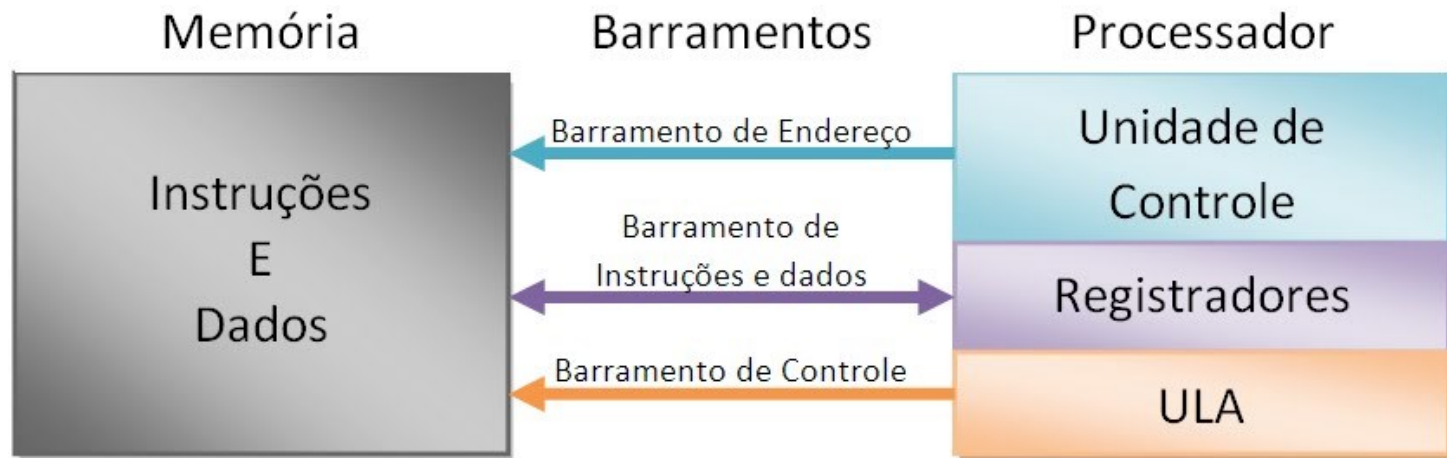
Arquitetura de um computador

- A arquitetura de computadores é o projeto conceitual e fundamental da estrutura operacional de um sistema computacional.
- A arquitetura **define os requisitos necessários** para que um computador funcione e como organizar os diversos componentes para obter o melhor desempenho (relação custo x benefício).
- Arquiteturas – modelos clássicos:
 - **CISC** (*Complex Instruction Set Computing*) - arquitetura von Neumann
 - **RISC** (*Reduced Instruction Set Computing*) - arquitetura Harvard
 - **CRISC** – híbrido

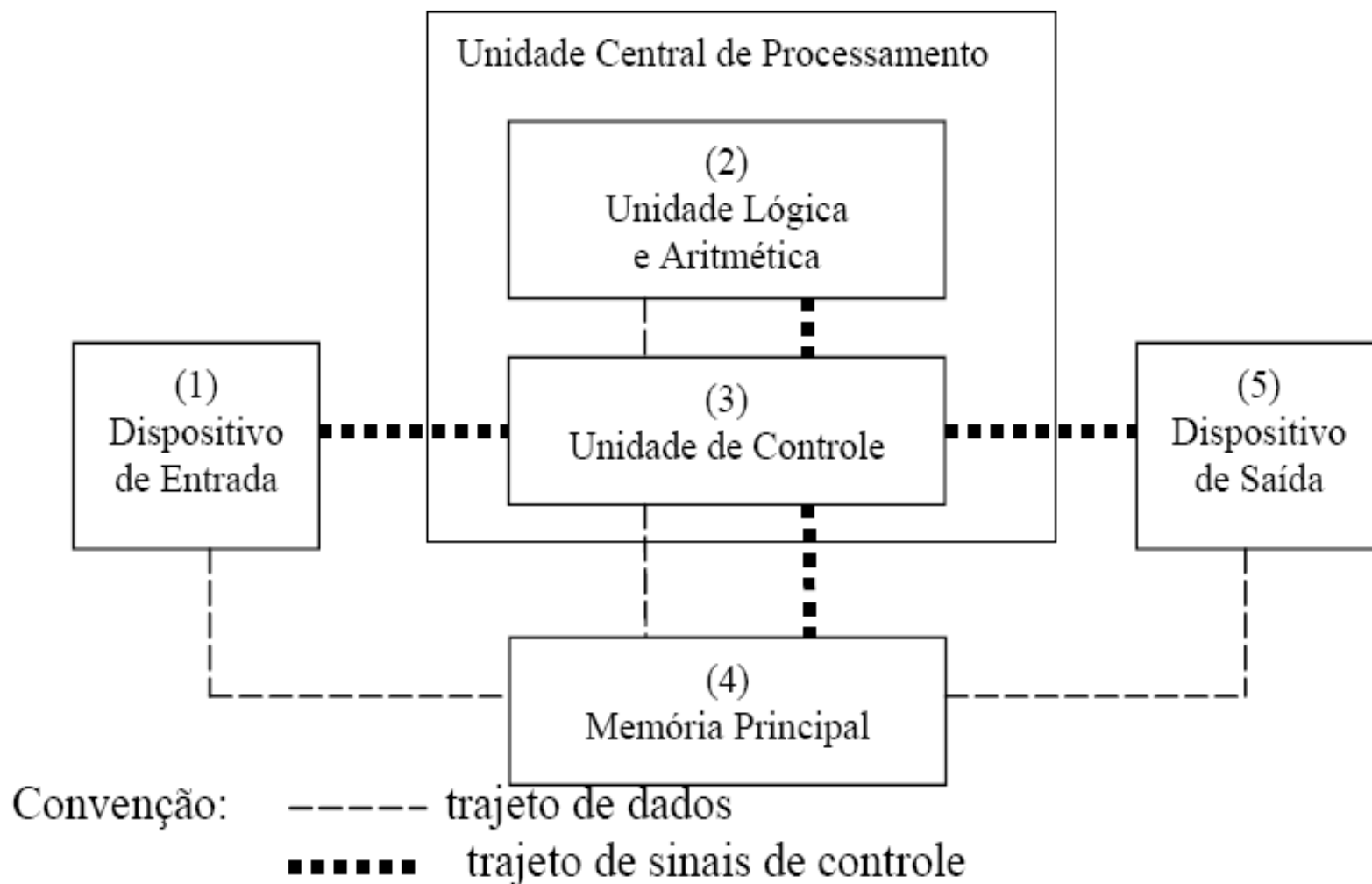
Arquitetura de von Neumann

- **Características:**

- Utiliza o mesmo espaço de memória para instruções e dados.
- Processa uma única instrução por vez.
- As instruções e os dados percorrem o mesmo barramento.



Arquitetura de von Neumann



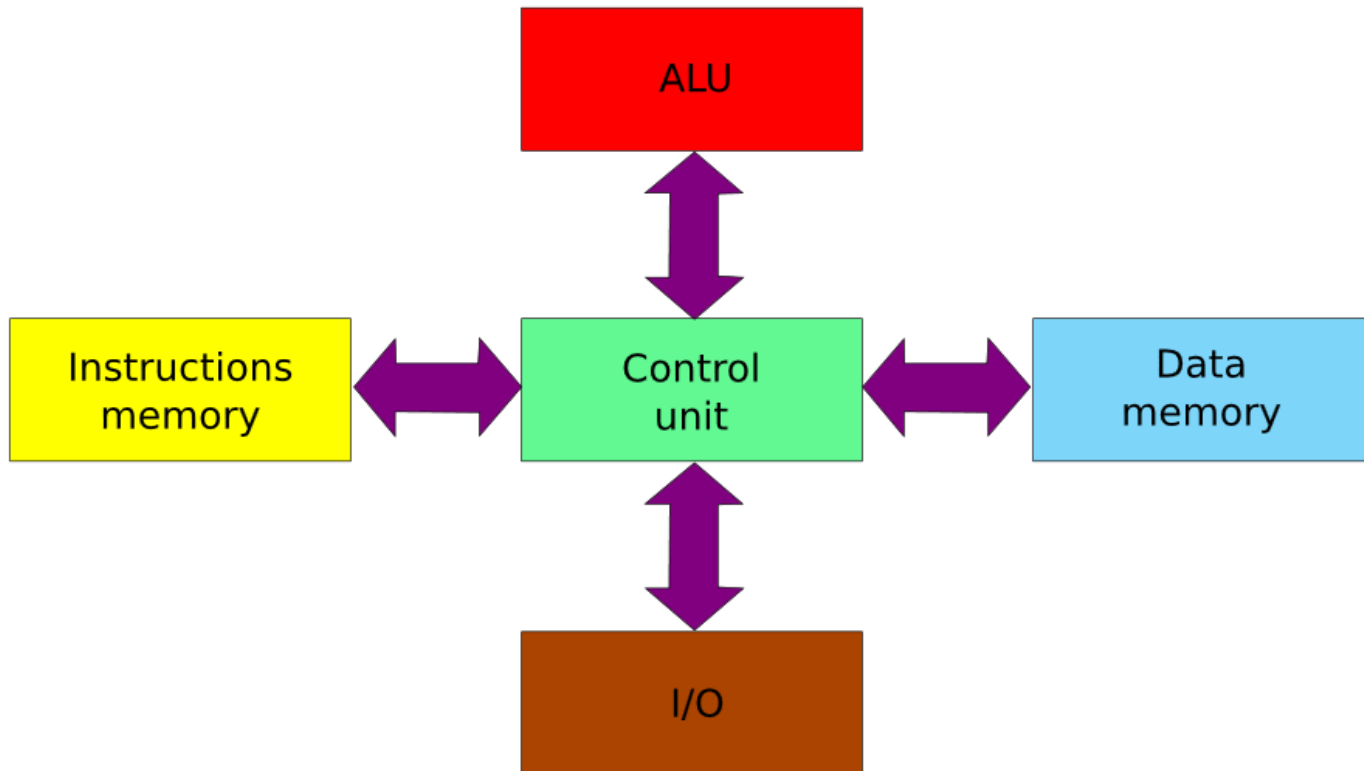
Arquitetura de von Neumann

- **1) Dispositivos de entrada:**
 - interação com o programa, entrada de dados.
- **2) Unidade central de processamento:**
 - o processador que executa as instruções do programa:
 - 2.1) [Unidade lógica e aritmética](#) - realiza cálculos e operações lógicas.
 - 2.2) [Unidade de controle](#) – gerencia o fluxo interno de dados.
 - 2.3) [Registradores](#) – armazenam dados necessários ao processamento.
- **3) Memória principal – [RAM](#) (*Random Access Memory*):**
 - armazena os dados e o programa em execução.
- **4) Dispositivos de saída:**
 - apresentação dos resultados (saídas) do programa.

Arquitetura Harvard

- A arquitetura Harvard **surgiu com a necessidade de melhorar o desempenho do microprocessador.**
- Sua principal característica está **na separação das memórias de dados e de instruções**, fazendo com que o processador possa acessar as duas simultaneamente.
- Isso pode representar um **desempenho melhor** do que o obtido com a arquitetura de Von Neumann, já que o processador busca uma nova instrução ao mesmo tempo que executa outra.

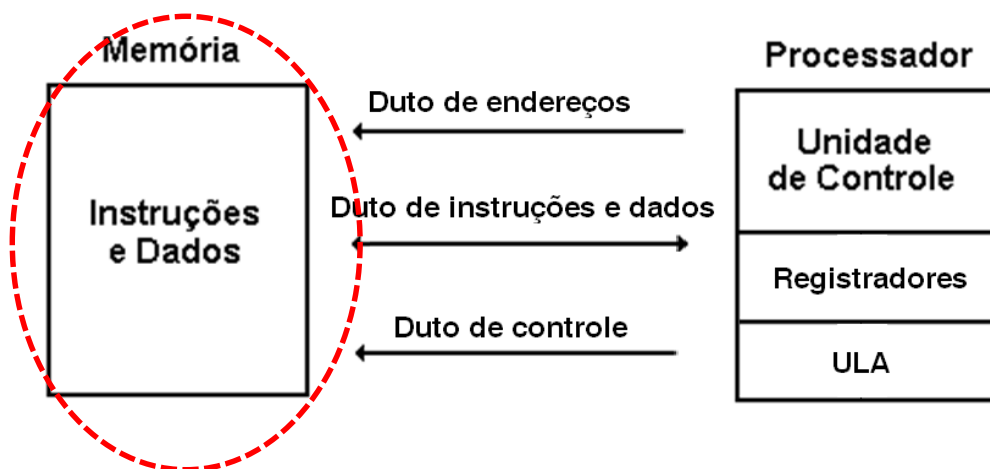
Arquitetura Harvard (cont.)



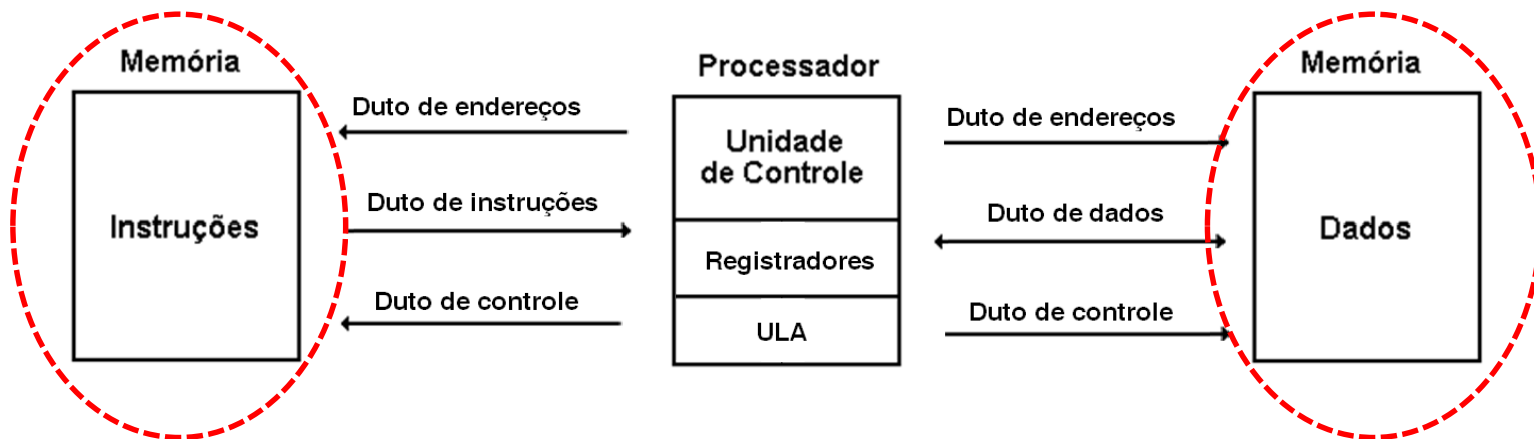
Fonte: Wikipédia (2016)

Arquiteturas von Neumann e Harvard

von Neumann

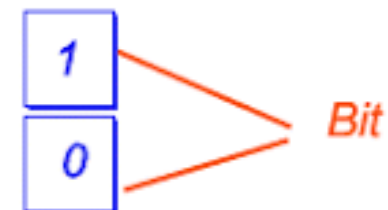


Harvard



Hardware: bit

- O computador trabalha com o **sistema binário** que utiliza 2 dígitos (0 e 1) e assim é de **base 2**.
- No sistema binário, os dígitos 0 e 1 são conhecidos como bits (***Binary Digit***).
- **Bit** é a menor unidade de representação de dado em um computador e pode assumir dois valores: 0 ou 1 (representativamente).
- **Bit** é a menor unidade binária de informação que pode ser armazenada ou transmitida.

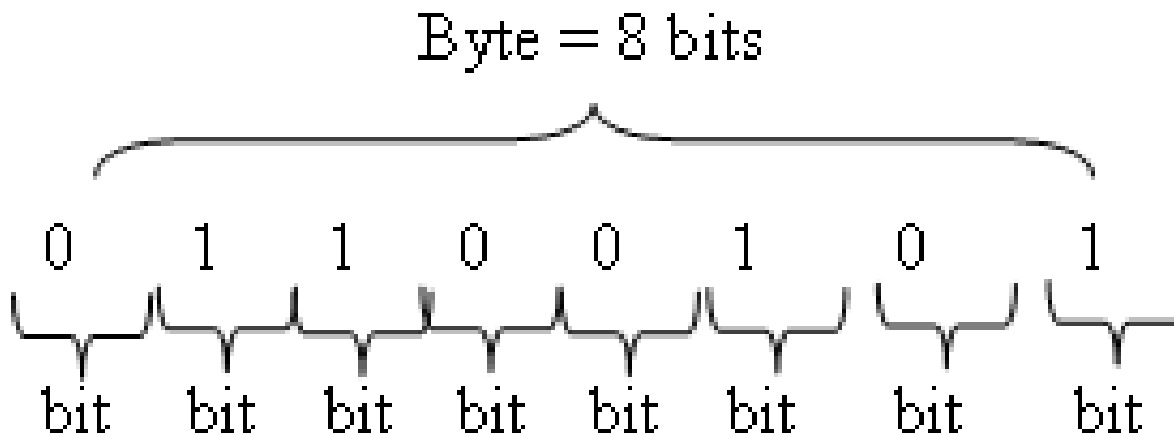


Hardware: byte

- Bit - dois valores (0 e 1).

Problema – representar os caracteres das linguagens escritas.

Solução – agrupamento de bits. Utilizar **8 bits** para representar **um caractere (byte)**.



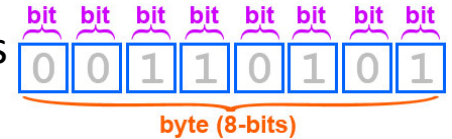
$$2^8 = 256$$

↑ ↑
8 bits
2 valores (0 ou 1)

Hardware: byte (cont.)

- **Byte** - é uma unidade de informação digital composta por 8 bits.

- **Byte - *Binary term*** é um conjunto de 8 bits. As combinações



de 8 bits representam 256 caracteres:

- Dígitos numéricos (0 a 9).
- Letras maiúsculas e minúsculas do alfabeto (A ... Z, a ... z).
- Sinais de pontuação, símbolos aritméticos e outros (. , % \$)
- Caracteres de controle.
- As 256 combinações compõem a [tabela ASCII](#).

- **Exemplos:**

- O número **0** em binário é : 00110000
- O número **1** em binário é : 00110001
- O caractere **B** em binário é : 01000010

- **Byte** especifica o tamanho ou a quantidade de memória ou de capacidade de armazenamento.

Hardware: byte

- **Byte** - É a unidade de medida básica e universal para a capacidade de armazenamento de informação (incluindo memória) para computadores e os seus dispositivos.

PREFIXO	SÍMBOLO	TAMANHO EM BYTES
byte	byte	1
quilo	KB	1.024
mega	MB	1.048.576
giga	GB	1.073.741.824
tera	TB	1.099.511.627.776
peta	PB	1.125.899.906.843.624
exa	EB	1.152.921.504.607.870.976
zeta	ZB	1.180.591.620.718.458.879.424
yotta	YB	1.208.925.819.615.701.892.530.176

Hardware: byte (cont.)

- **Tabela ASCII**
 - 256 combinações de 8 bits (caracteres)
- **Tabela UNICODE**
 - Possibilita que todos os caracteres de todas as linguagens escritas possam ser representados ([ANDRADE, 2016](#)).
 - Fornece uma identificação única para cada caractere.
 - UNICODE (<http://www.unicode.org>).
 - Permite a representação de 1.114.112 símbolos diferentes ([code points](#)).
 - Suporta praticamente todos idiomas do mundo.



Hardware: memória

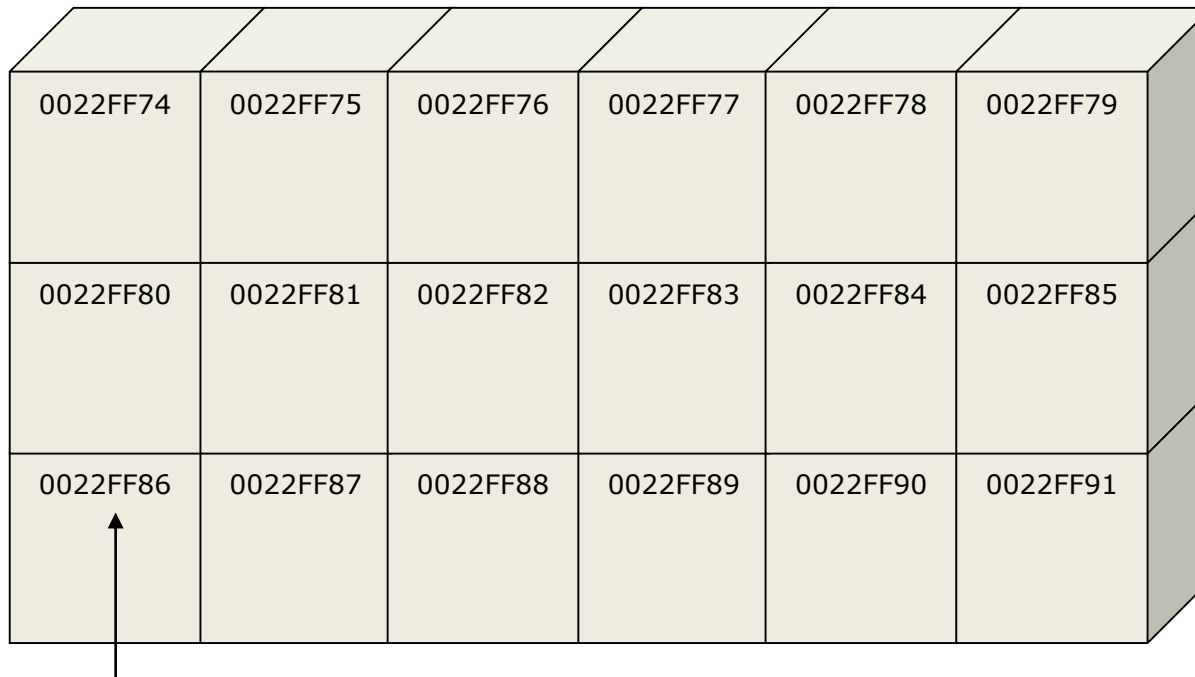
- A memória RAM (***Random Access Memory***) é constituída por um número finito de localizações (ou células) nas quais são armazenados os dados.
- É a área de trabalho do computador.
- A memória é organizada em posições, como os elementos de uma matriz. Um sistema de endereçamento permite identificar cada localização (ou célula) da memória. Esta identificação é conhecida como **endereço** e possui representação numérica em [hexadecimal](#).
- Para escrever ou ler algum dado na memória do computador é necessário fazê-lo **byte a byte***.

Considerando que a palavra do computador possui 1 byte de tamanho ou mais, especificamente fazê-lo **palavra a **palavra**.*

Endereço	Conteúdo								Localização
A013545D	0	1	0	0	1	1	0	1	
A013545E	0	1	1	0	1	0	1	1	
A013545F	0	1	1	1	1	1	1	1	
A0135460	0	0	0	0	0	0	0	0	
A0135461	0	1	0	1	1	1	0	1	
A0135462	1	0	1	1	1	0	1	1	
A0135463	1	0	1	0	0	1	0	1	

Hardware: memória

- **Memória** – armazenar bytes em tempo de execução. Esses bytes compõem os programas e os dados manipulados.
- Representação esquemática da memória (RAM):

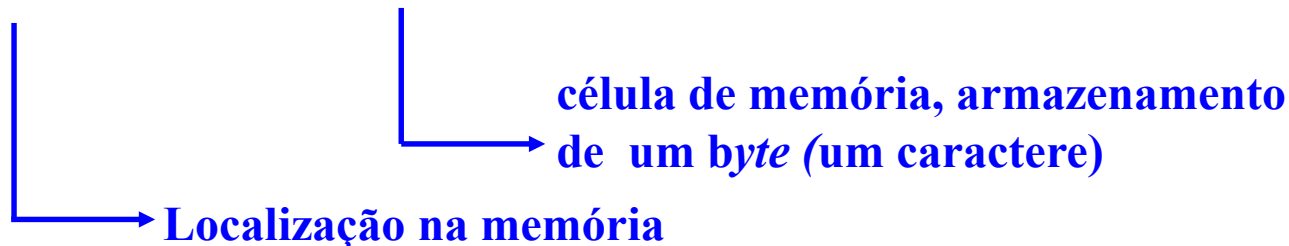


Célula de memória (endereço sequencial)–local de armazenamento

Exemplo: palavra "aluno" armazenada na memória de um computador

Endereço	Conteúdo (representação binária)
0040180E	01100001
0040180F	01101100
00401810	01110101
00401811	01101110
00401812	01101111

código ASCII	Caractere
97	a
108	l
117	u
110	n
111	o



Memória: variáveis

- **Declarar uma variável é reservar células** (espaço, endereço) **de memória** denominando-as por um identificador (nome) para armazenar valores e permitir a sua recuperação posterior.
 - A quantidade de células necessárias, reservadas, depende do tipo de dado a ser armazenado.
 - **Para declarar uma variável é necessário definir um nome** (identificador) para o endereço reservado. É por meio desse identificador que dados são armazenados e recuperados nesse e desse “espaço” de memória.

Leituras complementares

- Unicode
<http://unicode.org/standard/translations/portuguese.html>
<https://pt.wikipedia.org/wiki/Unicode>
- Supercomputadores
<https://www.tecmundo.com.br/supercomputadores>
- Memória flash
<http://www.hardware.com.br/tutoriais/entendendo-ssd/memoria-flash.html>

Referências

- **Arquitetura Harvard.** Disponível em https://pt.wikipedia.org/wiki/Arquitetura_Harvard. Acesso em 01 mar. 2022.
- **Byte.** Disponível em <https://pt.wikipedia.org/wiki/Byte>. Acesso em 01 mar. 2022.
- ANDRADE, Eduardo M. **Unicode: conceitos básicos.** Disponível em: <http://www.devmedia.com.br/unicode-conceitos-basicos/25169>. Acesso em 01 mar. 2022.
- **ARDUINO.** Disponível em: <https://www.arduino.cc/en/Main/Products>. Acesso em: 01 mar. 2022.
- BORATTI, I. C., OLIVEIRA, A. B. de. **Introdução à programação de algoritmos**, 3ª edição, Visual Books: Florianópolis, 2007.
- **Significado de Bit e Byte.** Disponível em: <http://www.significados.com.br/bit-e-byte>. Acesso em 01 mar. 2022.
- Alecrim, Emerson. **O que são bits, bytes, megabits, megabytes e afins?** Disponível em: <http://www.infowester.com/bit.php>. Acesso em 01 mar. 2022.
- Conteúdo baseado no material disponibilizado pela professora Beatriz Borsoi.

Dúvidas

- ???