

INOVAÇÃO TECNOLÓGICA E EMPREENDEDORISMO

ALGORITMOS, ARQUITETURA E ARDUINO

Prof. Dr. Daniel Caetano

2017 - 2

Objetivos

- Rever alguns conceitos iniciais de algoritmos
- Tomar contato com alguns conceitos de arquitetura de computadores
- Tomar primeiro contato com o hardware do Arduino e suas características





**ANTES DE
MAIS NADA...**

Para quem faltou...

Professor	Informações de Contato
Daniel Caetano	prof@caetano.eng.br

- Datas/critérios, apresent., exercícios, bibliog...

<http://www.caetano.eng.br/>



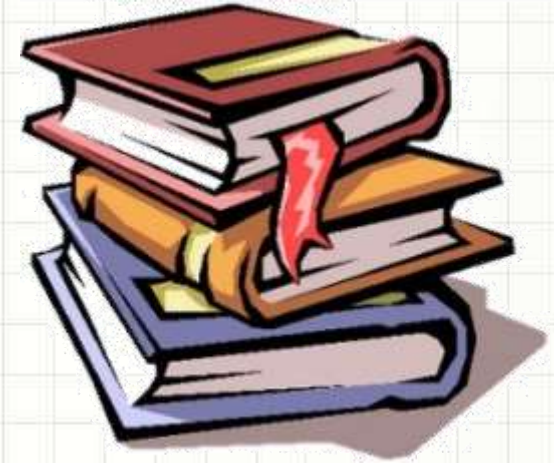
17/07/2012, 10:55
00021924

Prof. Caetano

Home Ensino Pesquisa Publicações Software Pessoal

Nesta seção você encontra acesso ao material didático desenvolvido pelo Prof. Caetano para os cursos já ministrados. O material está dividido por períodos, visto que boa parte do material não está atualizado.

Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Inovação e Empreendedorismo – Aula A)

Material Didático

da disciplina Algoritmos

Biblioteca Virtual

“algoritmos”, “programação”

Material Adicional

Tutorial de Arduino (site do Professor)



CONTEXTUALIZAÇÃO

O que são Algoritmos

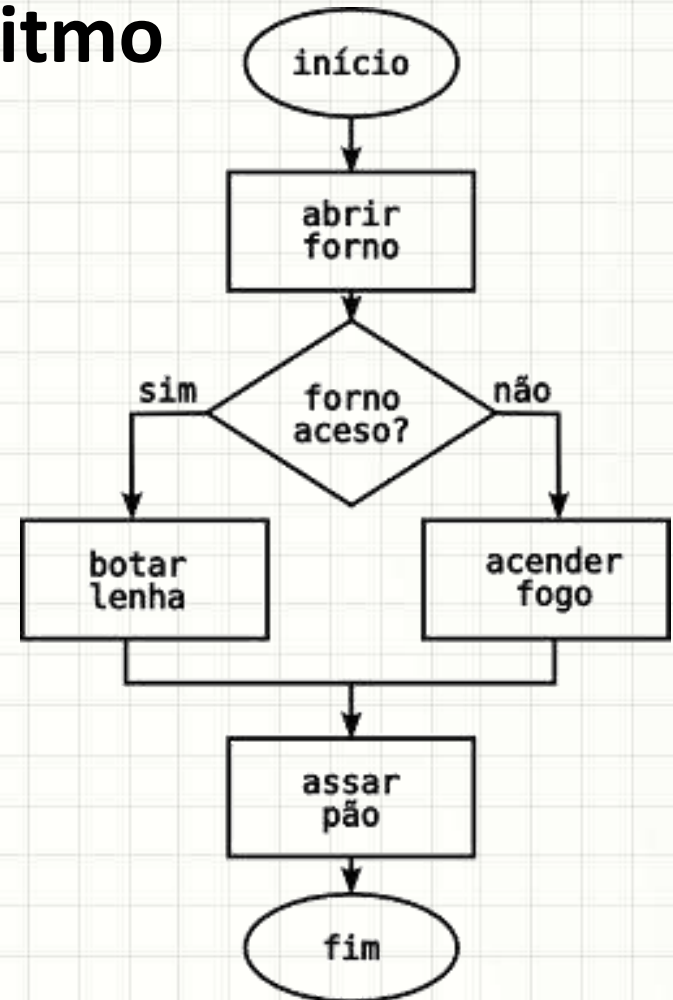
- Toda tarefa complexa pode ser subdividida
 - Tarefas menores e mais simples
- Exemplo: fabricar vinho para venda
 - Plantar a uva
 - Colher a uva
 - Amassar a uva
 - Deixar fermentar
 - Engarrafar
 - Distribuir para a venda



O que são Algoritmos

- O procedimento para realizar uma tarefa complexa chama-se **algoritmo**

- Um algoritmo envolve:
 - **Tarefas/Processos**
 - **Decisões**



Fluxograma

- Forma gráfica tradicional



Início e fim de algoritmo



Processo (trabalho do computador)



Entrada de dados (leitura)



Saída de dados (impressão)



Tomada de decisão



Sentido do fluxo de dados

Fluxograma

- Forma gráfica tradicional



Início e fim de algoritmo



Processo (trabalho do computador)



Entrada de dados (leitura)



Saída de dados (impressão)



Tomada de decisão

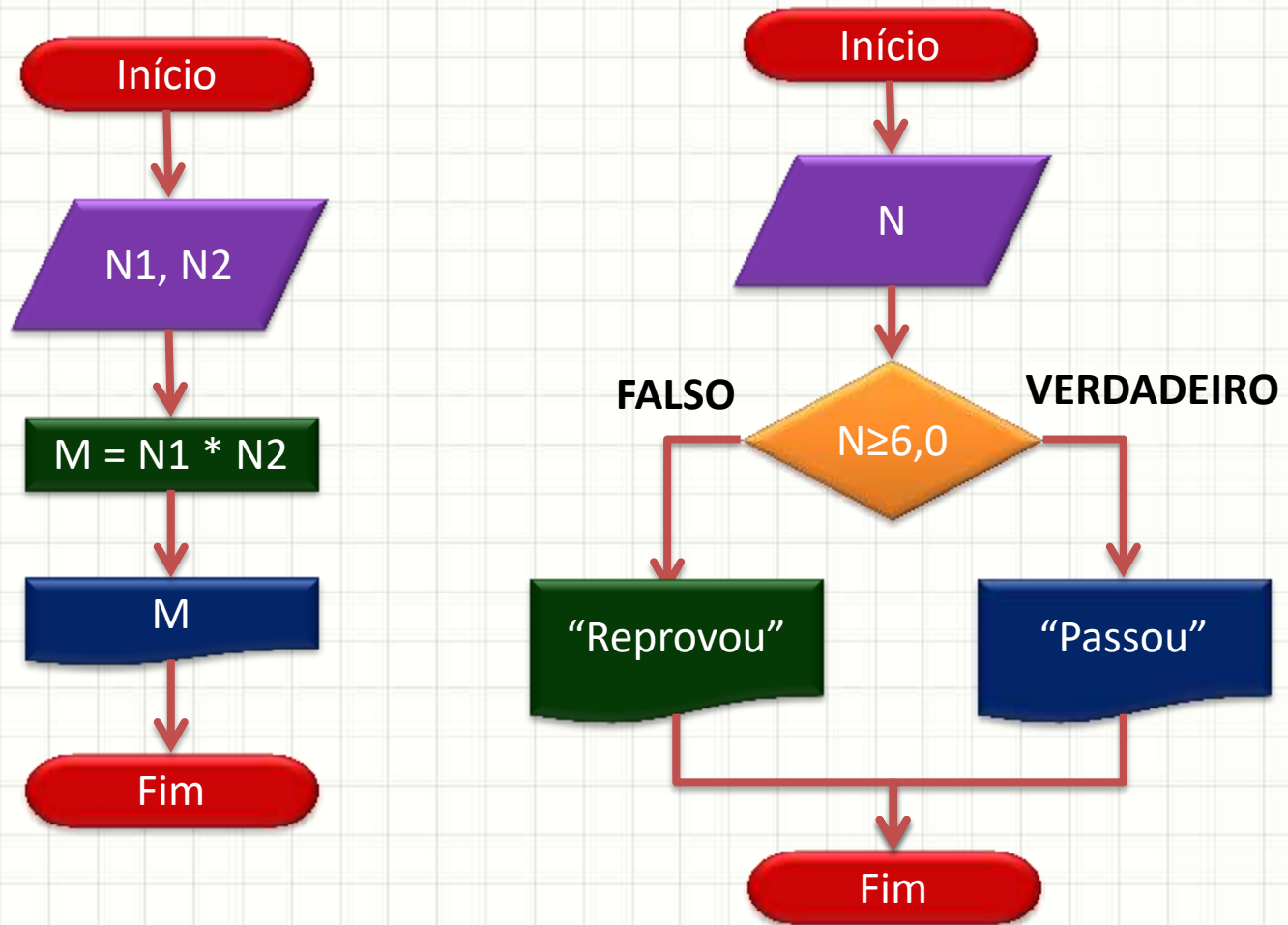


Sentido do fluxo de dados

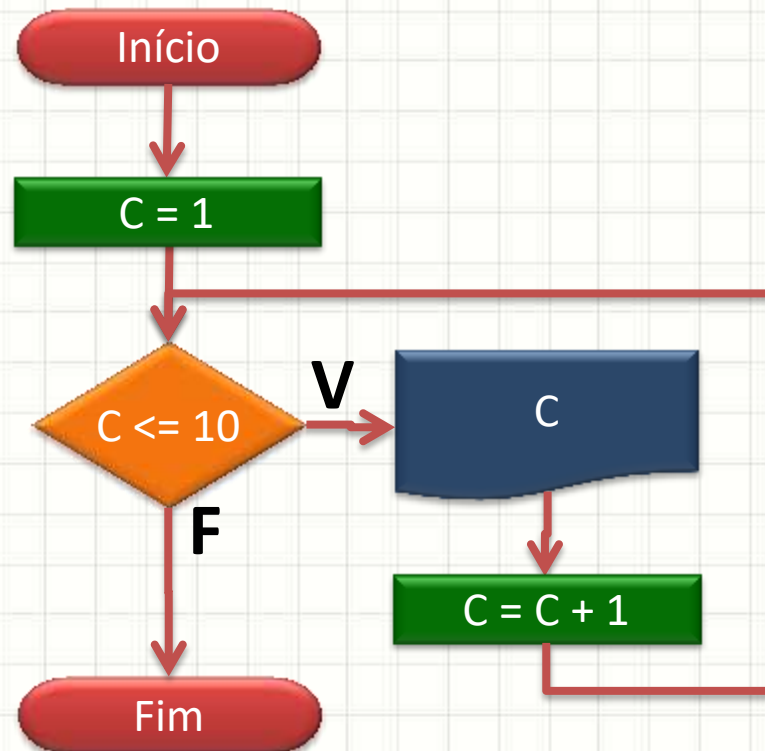
ALTERNATIVOS



Exemplos de Fluxograma



Exemplos de Fluxograma



Exemplos de Linguagem C++

```
#include <iostream>
using namespace std;
main()
{
    int C;
    C = 0;
    while ( C < 1000 )
    {
        cout << "Sei programar!";
        cout << endl;
        C = C + 1;
    }
}
```

Exemplos de Linguagem C++

```
#include <iostream>
using namespace std;
main() // Imprime contador
{
    int C;
    C = 0;
    while ( C < 1000 )
    {
        cout << C << " - ";
        cout << "Sei programar!";
        cout << endl;
        C = C + 1;
    }
}
```

Exemplos de Linguagem C++

```
#include <iostream>
using namespace std;
main()
{
    int N1, N2, R, C;
    cout << "Digite um No.:";
    cin >> N1;
    cout << "Digite outro No.:";
    cin >> N2;
    C = 0;
    while ( C <= N1 )
    {
        R = C * N2;
        cout << C << "*" << N2 << "=" << R << endl;
        C = C + 1;
    }
}
```

Compilação

- Processo de Compilação



Programador



```
#include <io...  
int main(void)  
{  
    cout << "Oi";  
}
```

Código Fonte



Compilador



```
001010101010  
101010101010  
110111011011  
111110010101
```

Código Binário
de PC



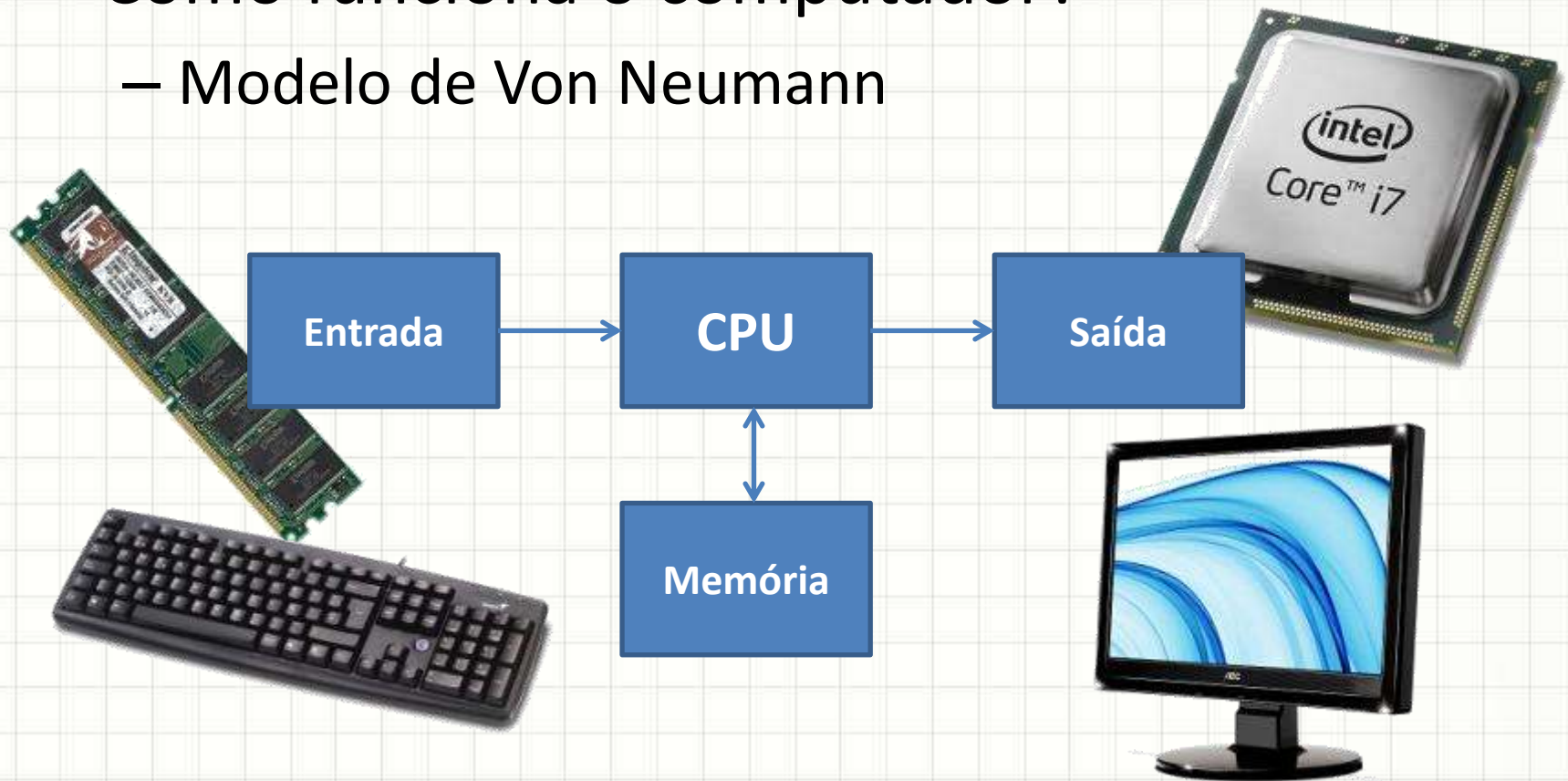
Computador PC



O COMPUTADOR

Entendendo o Computador

- Usar ferramenta: entender a ferramenta
- Como funciona o computador?
 - Modelo de Von Neumann



Entendendo o Computador

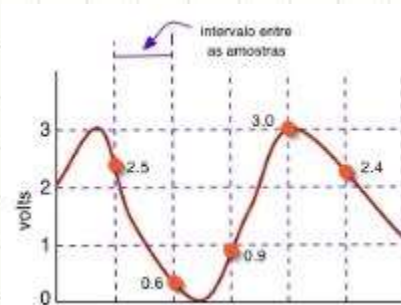
- **CPU**: Coordena todo o funcionamento do computador e realiza cálculos numéricos
- **Unidade de Entrada**: Recebe dados (números) externos para processamento
- **Unidade de Memória**: Armazena dados (números) para uso posterior
- **Unidade de Saída**: Exibe dados (números) para o usuário, após processamento

Entendendo o Computador

- Números...
- Números...
- Números...!?!?
- O computador só entende números!

Dispositivos de Entrada e Saída

- **Dispositivos de Entrada:** convertem informações externas (usualmente fornecidas pelo usuário) em números para o computador

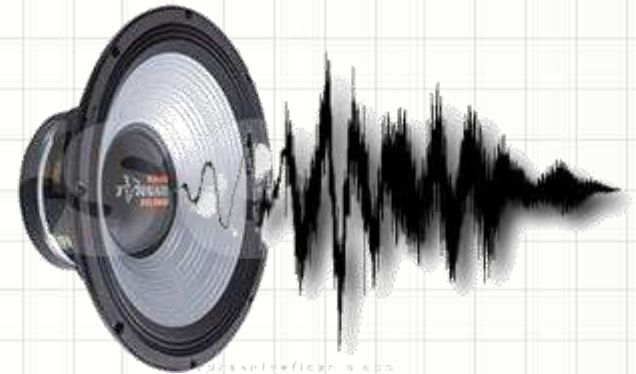
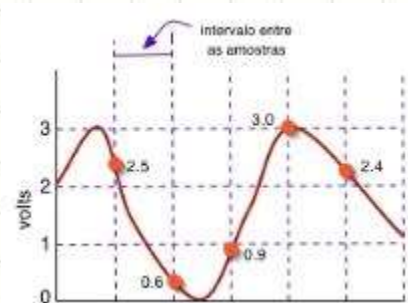


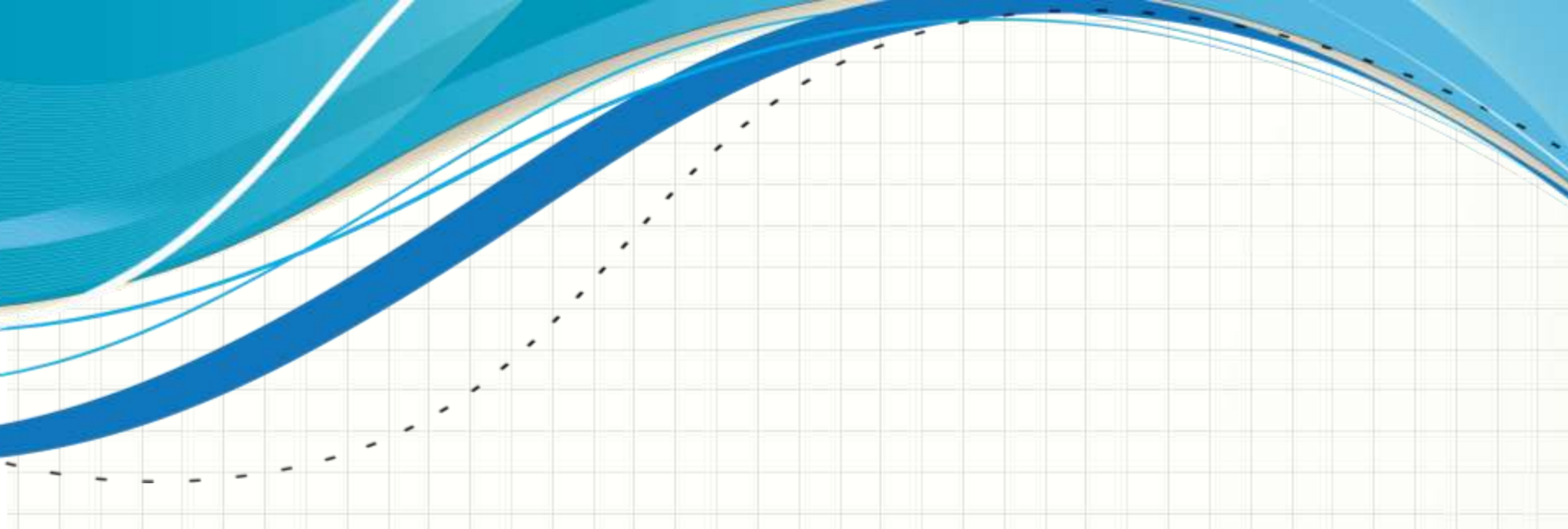
10011001
10011110
10101100
10111001
11001010
11001111
11010011
10111101

Dispositivos de Entrada e Saída

- **Dispositivos de Saída:** convertem números fornecidos pelo computador em informações para o usuário

10011001
10011110
10101100
10111001
11001010
11001111
11010011
10111101

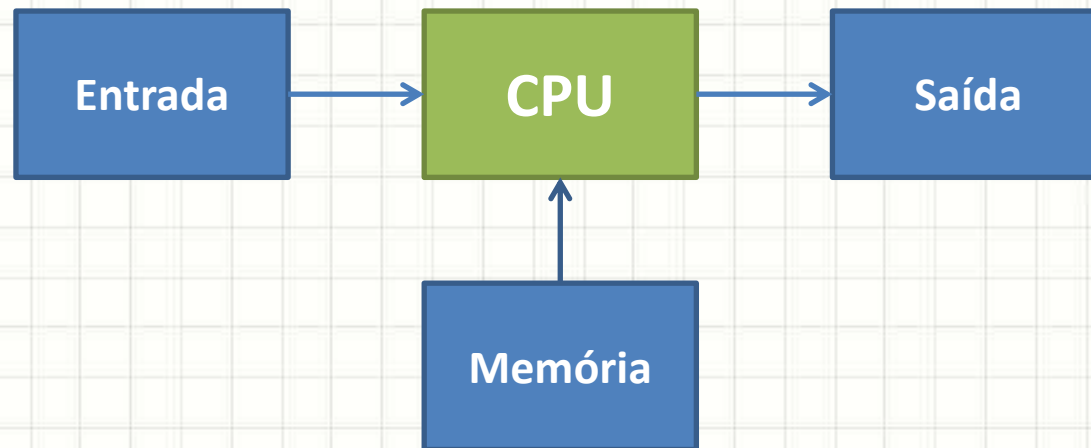




FUNCIONAMENTO DA CPU

Funcionamento da CPU

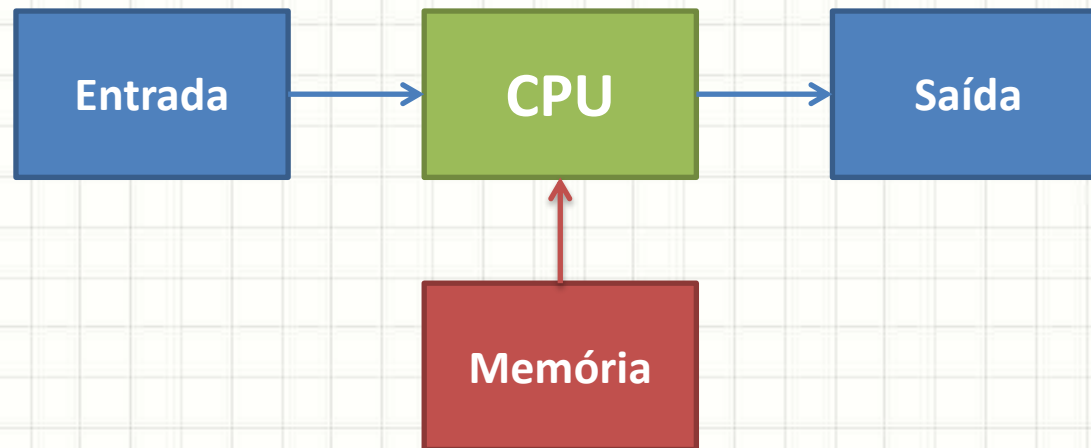
- Como a CPU coordena essas partes para produzir resultado útil?



Funcionamento da CPU

- **1. Busca Instrução**

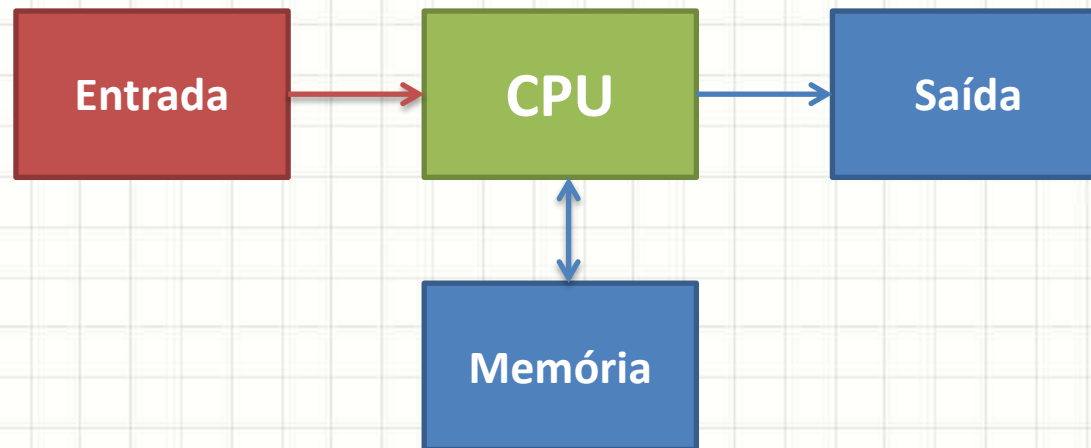
- CPU lê a memória em busca do que deve fazer
- Instrução lida: **leia entrada**



Funcionamento da CPU

- **2. Lê entrada**

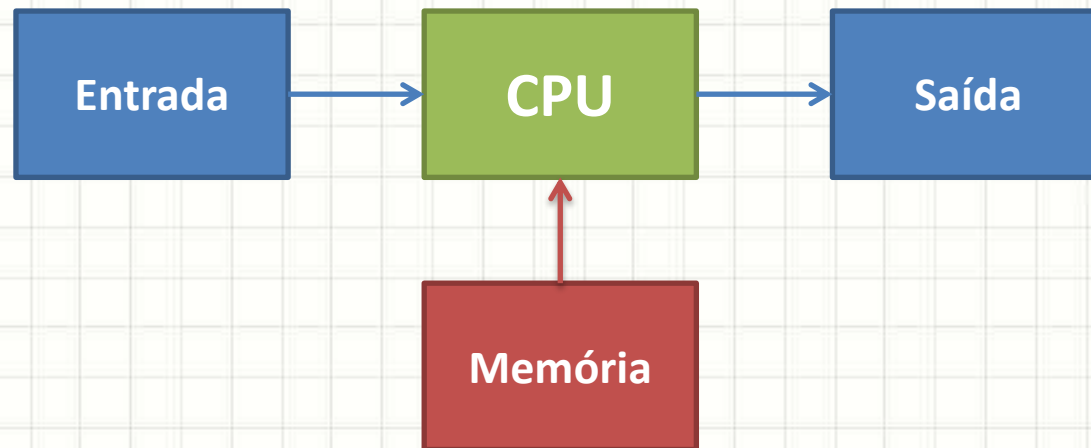
- CPU lê entrada, recebendo um dado numérico
- Dado lido: 33



Funcionamento da CPU

- **3. Busca Instrução**

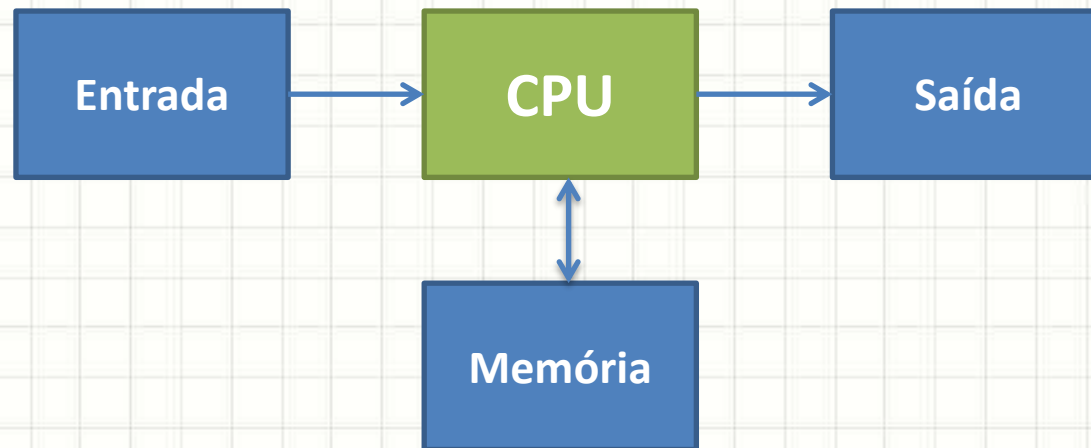
- CPU lê a memória em busca do que deve fazer
- Instrução lida: **multiplique por dois**



Funcionamento da CPU

- **4. Cálculo**

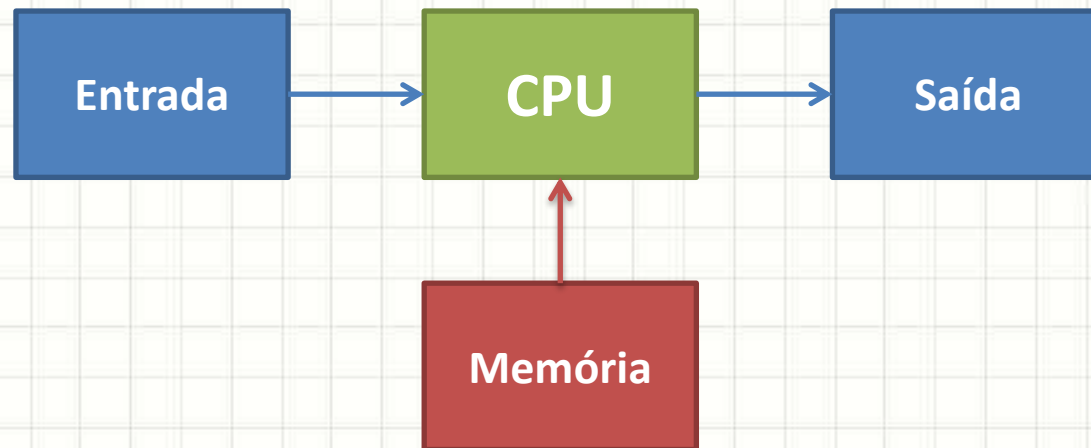
- CPU realiza operação de cálculo
- $33 * 2 = 66$



Funcionamento da CPU

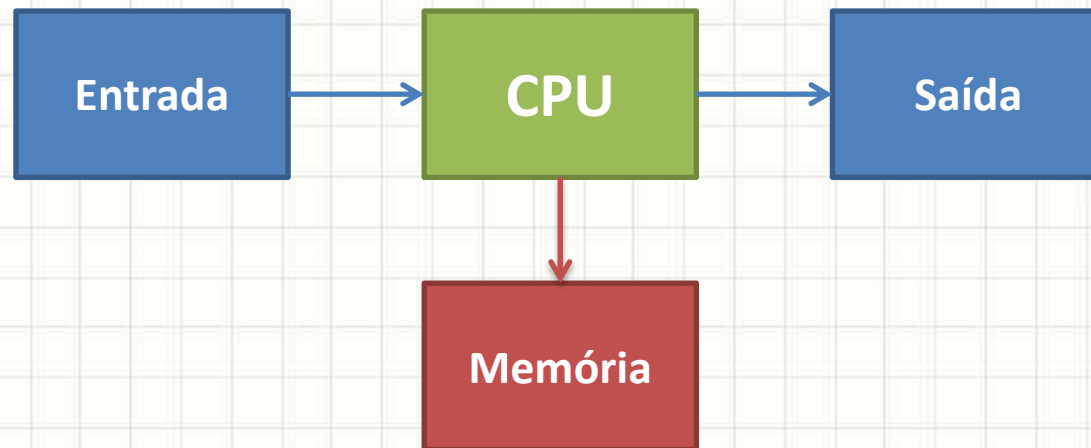
- **5. Busca Instrução**

- CPU lê a memória em busca do que deve fazer
- Instrução lida: **armazenar resultado**



Funcionamento da CPU

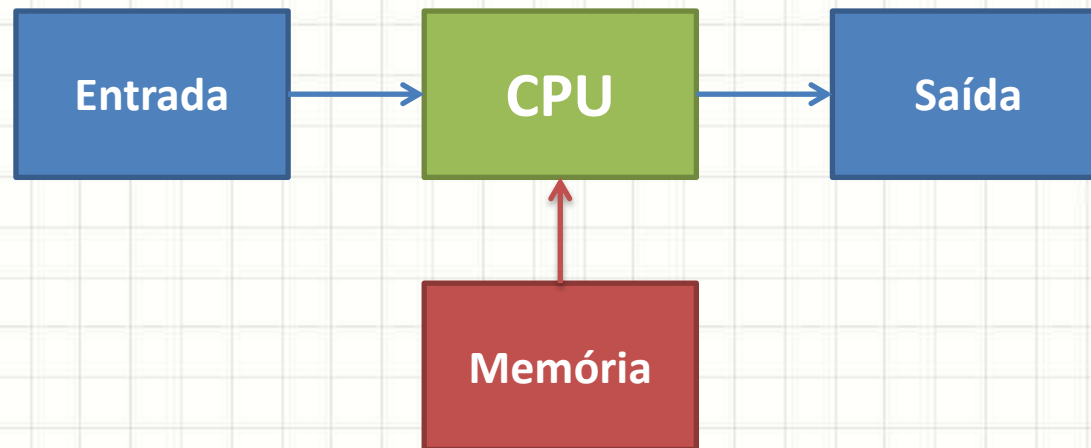
- **6. Armazena dado**
 - Guarda o dado na memória
 - Dado armazenado: 66



Funcionamento da CPU

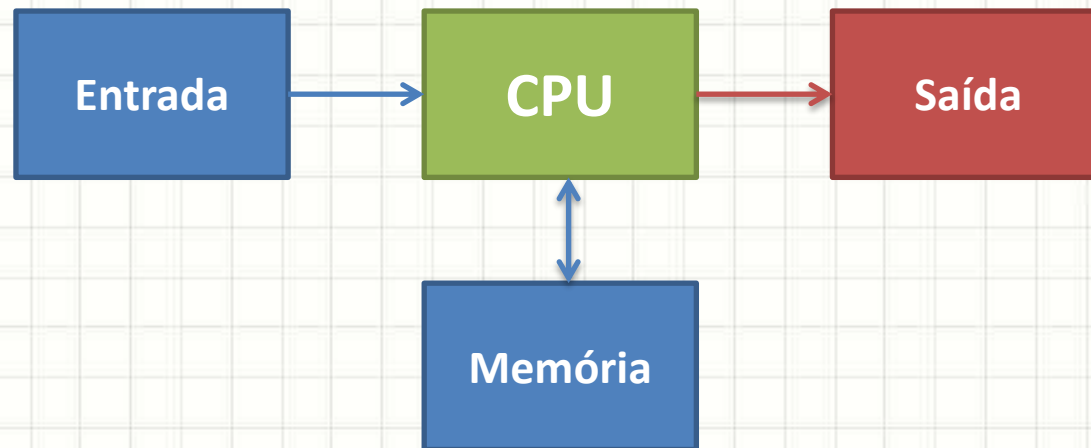
- **7. Busca Instrução**

- CPU lê a memória em busca do que deve fazer
- Instrução lida: **imprime dado**



Funcionamento da CPU

- **8. Saída de Dados**
 - CPU escreve na saída
 - O número 66 é impresso

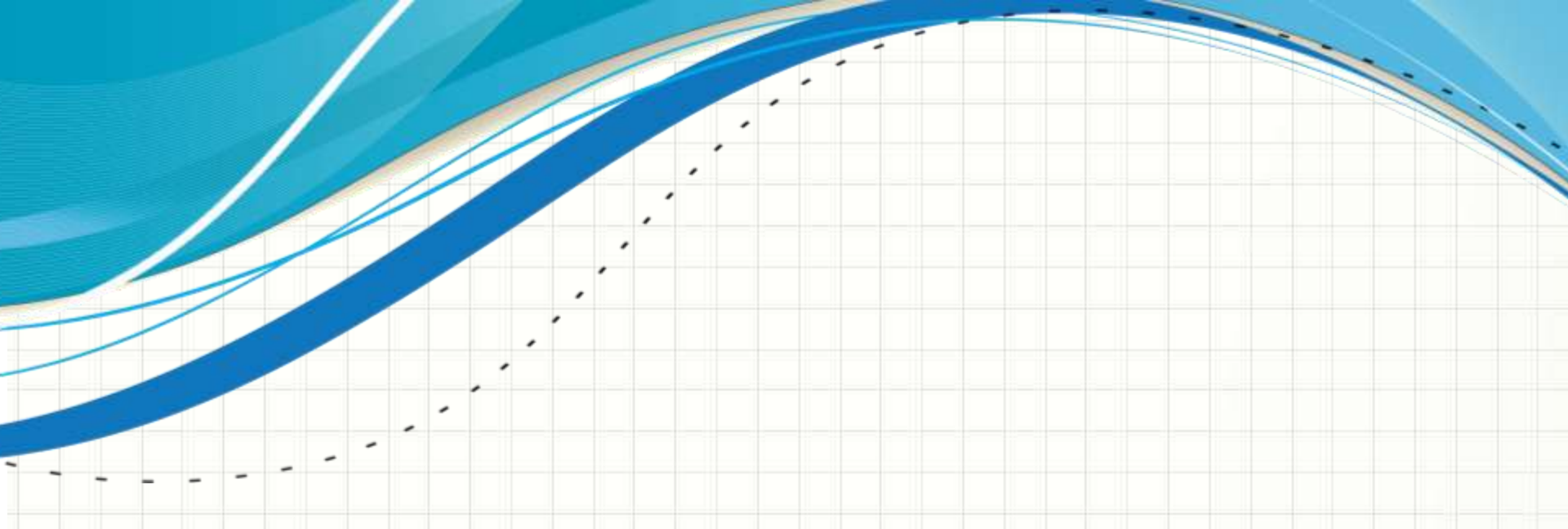


Funcionamento da CPU

- **8. Saída de Dados**
 - CPU escreve na saída

E assim sucessivamente...

Memória



FUNCIONAMENTO DA MEMÓRIA PRINCIPAL

A Memória Principal

- Cada gaveta é chamada **posição de memória**
- Cada gaveta possui um número que a identifica, chamado **endereço de memória**
- Em cada uma das gavetas, podemos guardar um único número



A Memória Principal

- A memória principal (RAM) é...



A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>								

A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>								

- Armazenemos o valor **255** na posição de memória cujo endereço é **3**

A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>								

- Armazenemos o valor **255** na posição de memória cujo endereço é **3**

A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>				255				

- Armazenemos o valor **255** na posição de memória cujo endereço é **3**

A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>				255				

- Agora, armazenemos o valor **7** na posição de memória cujo endereço é **5**

A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>				255				

- Agora, armazenemos o valor **7** na posição de memória cujo endereço é **5**

A Memória Principal

- Quando queremos guardar um número na memória, temos dizer em qual **posição de memória** ele deve ser armazenado, usando para isso o **endereço de memória**

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>				255		7		

- Agora, armazenemos o valor **7** na posição de memória cujo endereço é **5**

A Memória Principal

- Tomemos, agora, uma memória cheia

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>	10	57	0	255	100	7	10	2

A Memória Principal

- Tomemos, agora, uma memória cheia

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>	10	57	0	255	100	7	10	2

- Qual é o valor na posição de memória cujo endereço é **7**?

A Memória Principal

- Tomemos, agora, uma memória cheia

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>	10	57	0	255	100	7	10	2

- Qual é o valor na posição de memória cujo endereço é **7**?

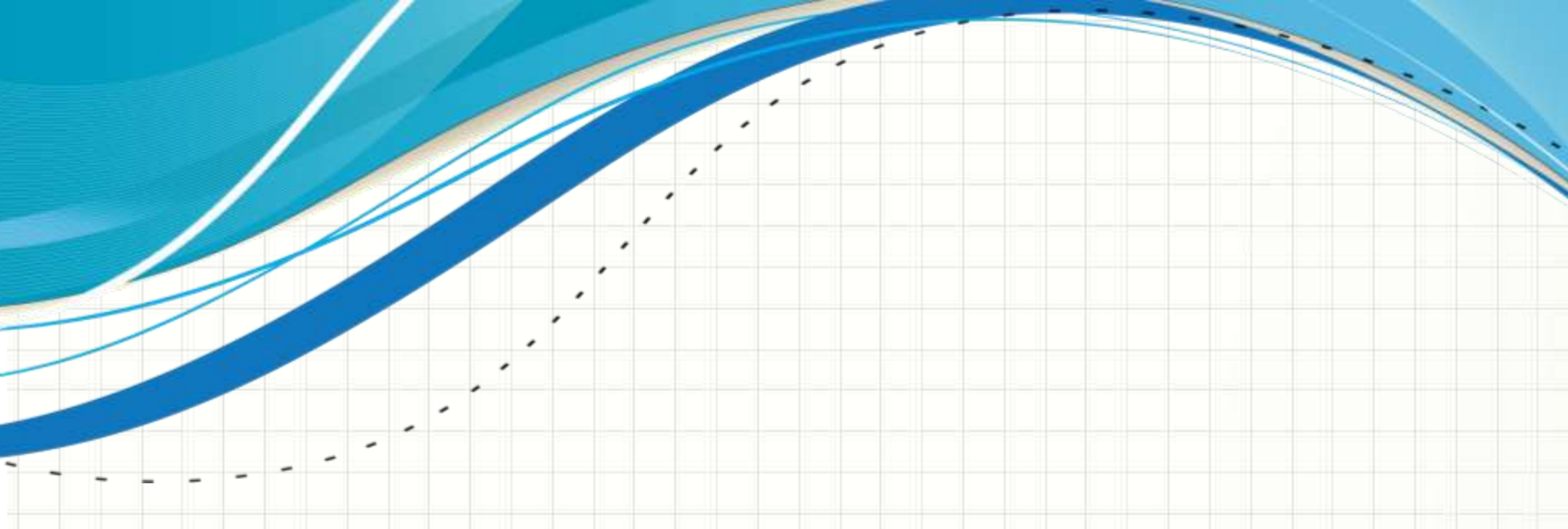
A Memória Principal

- Tomemos, agora, uma memória cheia

<u>Endereço</u>	0	1	2	3	4	5	6	7
<u>Valor</u>	10	57	0	255	100	7	10	2

- Qual é o valor na posição de memória cujo endereço é **7**?
- O valor é **2**!

- Atenção! O computador **NÃO** apaga a memória sozinho!



**COMO OS NÚMEROS SÃO
ARMAZENADOS: A LINGUAGEM
DO COMPUTADOR**

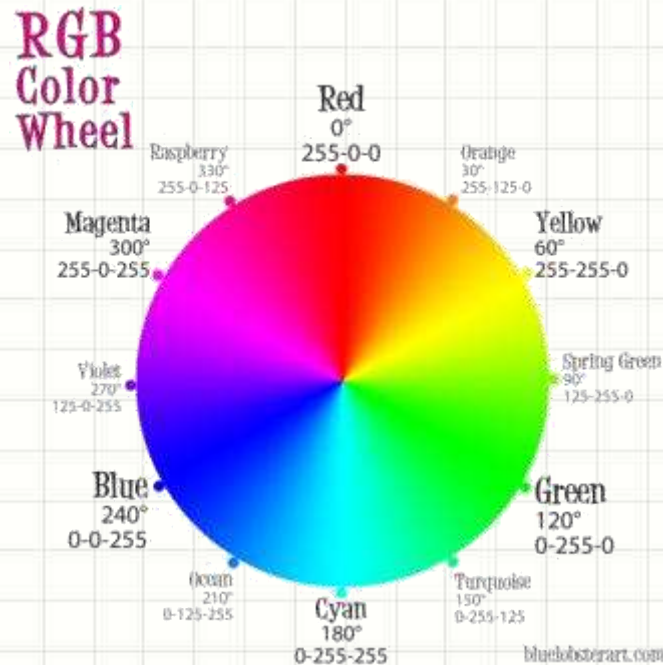
O Que o Computador Entende?

- Já vimos que o computador entende apenas números...
- ...mas ele entende os nossos números?
- Infelizmente... não.
- O computador um *dialeto* chamado “binário”:
0101001010111b
- Mas o que isso significa?

O Que o Computador Entende?

0101001010111b

- Isso pode significar várias coisas...
 - Música, imagem, quantidades...



Humanos x Processadores

- Desde muito cedo aprendemos a contar com nossos dedos;
- Como temos **DEZ** dedos nas mãos, usamos naturalmente os números DECIMAIS
- Isso significa que cada “casa” do nosso número será preenchida com um de 10 símbolos diferentes:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?

FIOS



Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?



Humanos x Processadores

- Como indicar números decimais para um processador?



94.614

Como saber isso?

- Imaginemos que cada fio representa um “dígito” numérico, chamado **bit**
- Esse fio pode estar **desligado ou ligado**
- Associando o símbolo **0** ao fio “desligado” e **1** ao fio “ligado”, representa-se o “número” em um formato que o computador entende:

10111000110010110

94.614



Como saber isso?

- Imaginemos que cada fio representa um “dígito” numérico, chamado **bit**
- Esse fio pode estar **desligado** ou **ligado**

10111000110010110b = 94.614

10111000110010110

94.614



Os Números Binários

- Como cada **bit** pode ser apenas 0 ou 1...
...o nome dessa representação é “**binária**”.
- Um único bit armazena pouca informação
- Usualmente, os bits aparecem agrupados






COMO GUARDAR OUTROS DADOS NA MEMÓRIA?

Outros Dados na Memória

- Só números binários... 0101001010111b
- Seu significado depende da interpretação!
 - Já vimos números inteiros “sem sinal”
- E outras coisas?
 - Números inteiros com sinal
 - Letras...
 - Imagens...
- Interpretação depende de uma **convenção**

Representação de Caracteres

- Letras: Padrões de codificação
 - ASCII
 - UTF-8
 - UTF-16



Código	Caracteres	Código	Caracteres	Código	Caracteres
32	[space]	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(72	H	104	h
41)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[123	{
60	<	92	\	124	
61	=	93]	125	}
62	>	94	^	126	~
63	?	95	_	127	[backspace]

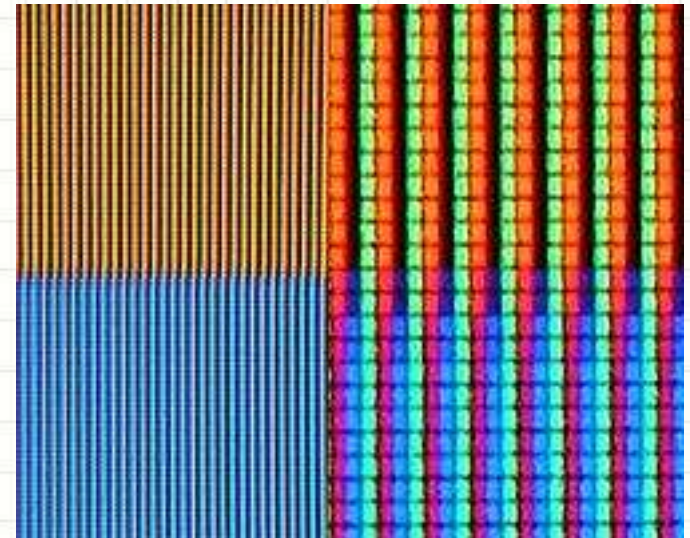
Representação de Imagens

- Pontos: RGB (Vermelho, Verde, Azul)



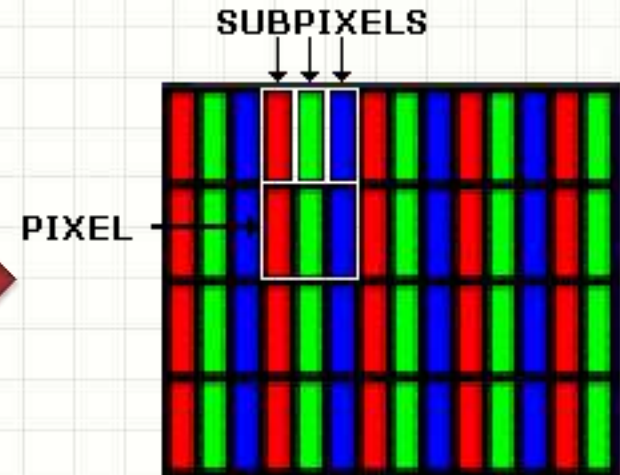
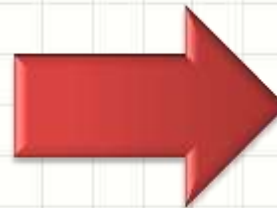
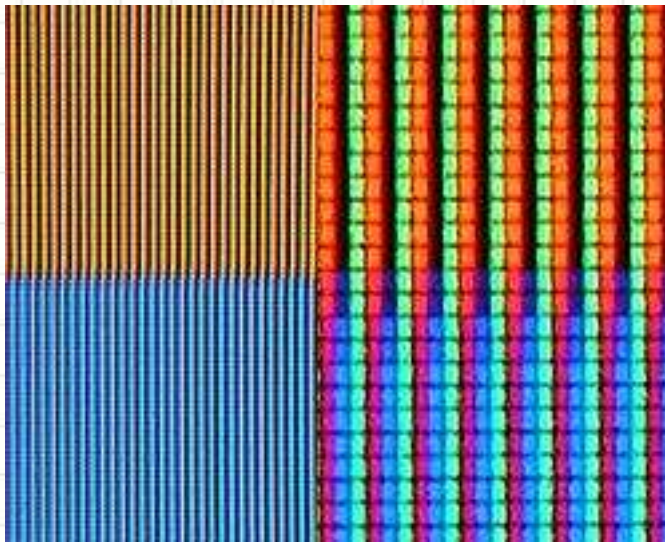
Representação de Imagens

- Pontos: RGB (Vermelho, Verde, Azul)



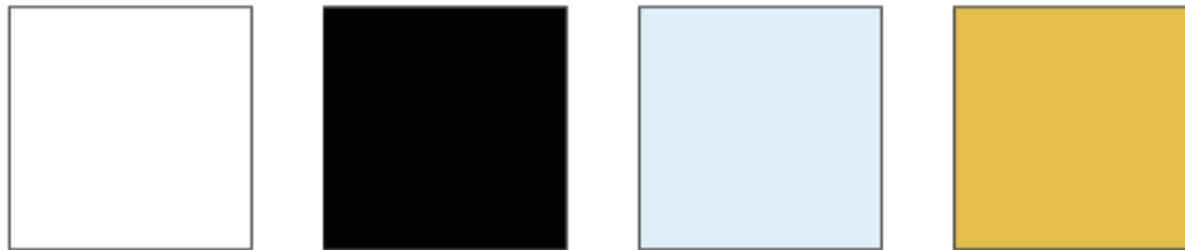
Representação de Imagens

- Pontos: RGB (Vermelho, Verde, Azul)



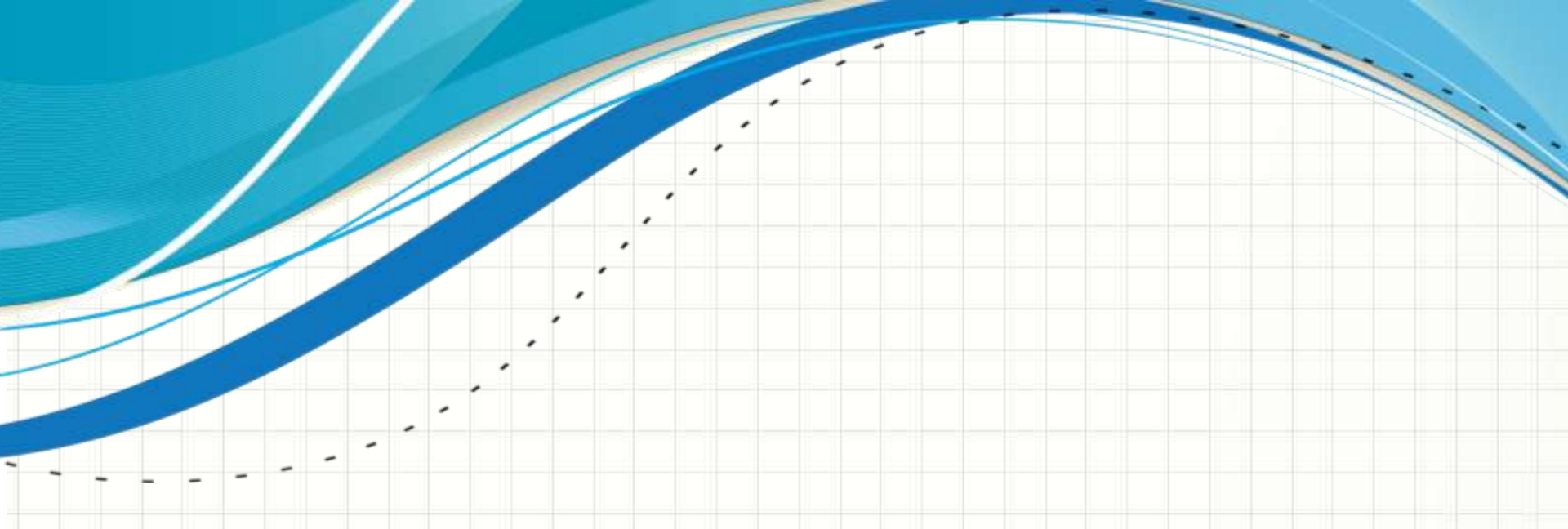
Representação de Imagens

- Pontos: RGB (Vermelho, Verde, Azul)



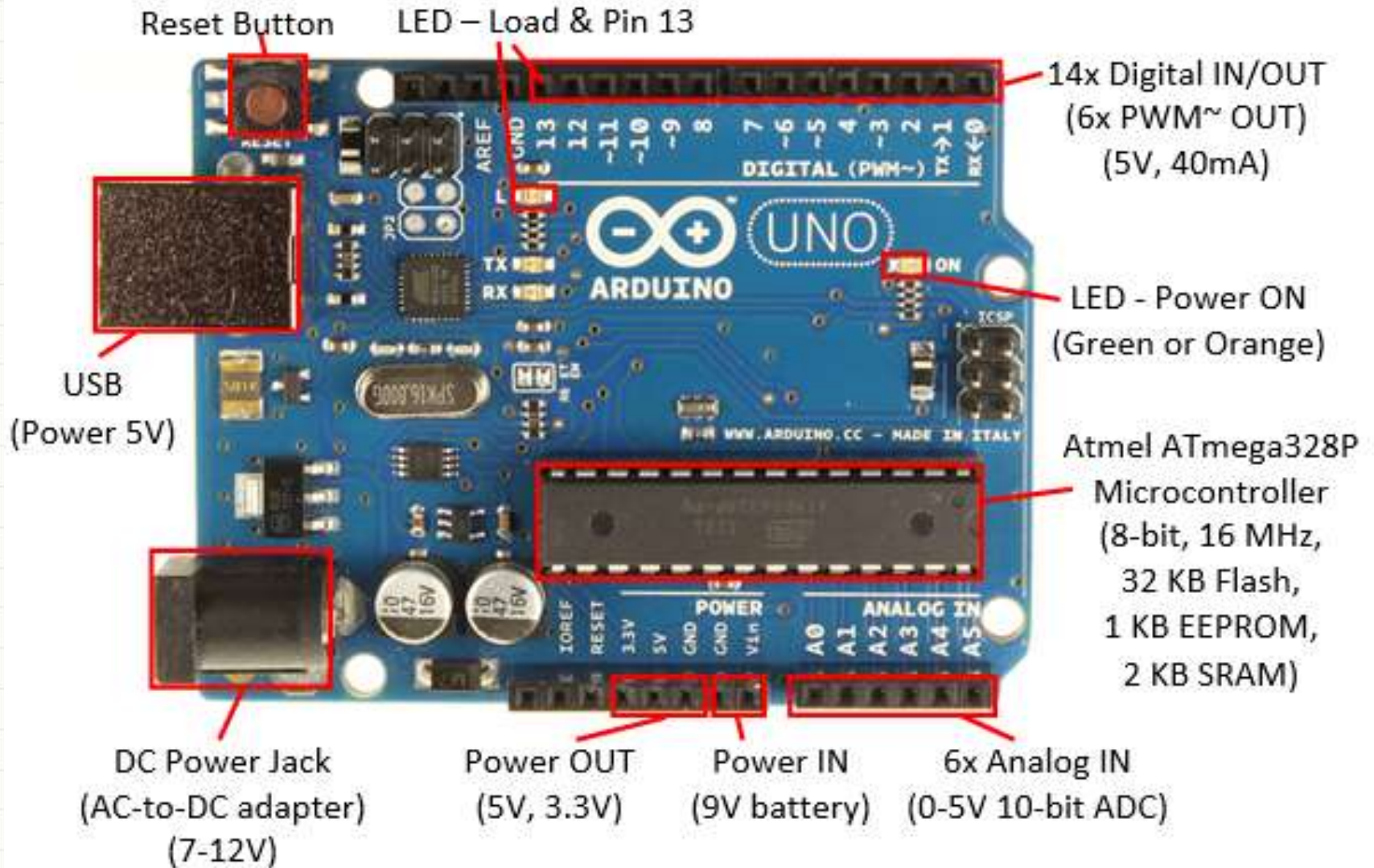
RGB 255, 255, 255 RGB 0, 0, 0 RGB 255, 239, 248 RGB 228, 189, 79





O HARDWARE DO ARDUINO UNO

Hardware Arduino UNO



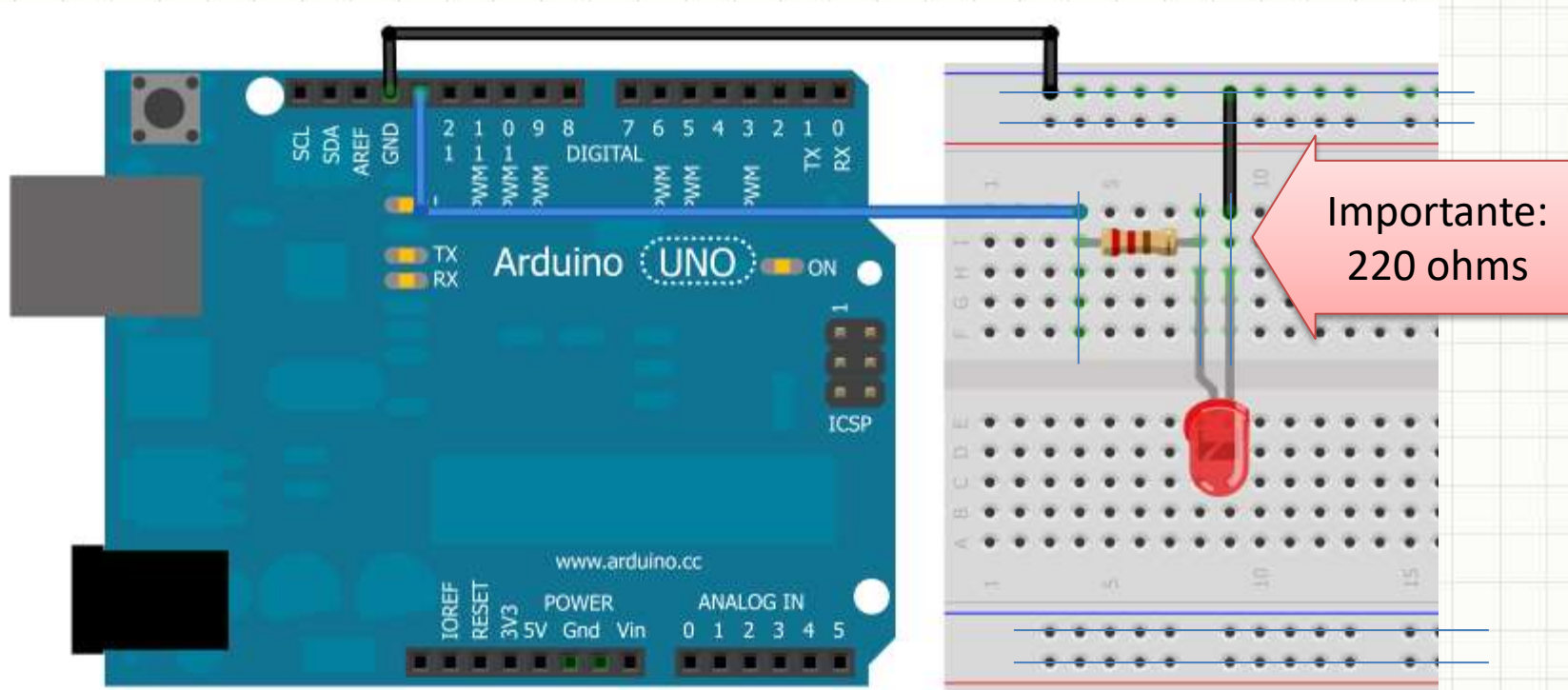
Como ligar um LED no UnoArduSim

The image shows the UnoArduSim software interface, which is a virtual representation of an Arduino Uno board. The central component is the ATMEGA328 microcontroller. Surrounding it are several peripheral modules:

- Serial (01):** TX and RX characters, Baud rate: 9600.
- SFTSER (00):** TX and RX characters, Baud rate: 300.
- SPISLV (11):** MOSI, MISO, DATA, Recv, SS*, Mode, SCK.
- I2CSLV (A4):** SDA, SCL, Send, Addr, Recv, Clock, 100 kHz.
- Digital Pins:** A row of pins from 0 to 13. Pin 13 is highlighted with a red box.
- Analog Pins:** A row of pins from A0 to A5.
- Resistors:** Five 1K resistors are shown on the left side.
- PIEZO:** Two piezo sensors are shown on the left side.
- SERVO:** A servo motor module.
- MOTOR:** A motor module with Pwm and Enc options.
- PULSER:** A pulse generator module with Pulse (1000) and Period (10000) settings.
- FUNCGEN:** A function generator module with Period (10000) settings.
- LEDs:** Five LED modules are shown on the right side, each with R and YG pins.
- Power:** Two +5V power sources and two 0V ground sources are shown at the bottom right.



Como ligar um LED no Arduino



images.google.com

Como ligar um led no arduino

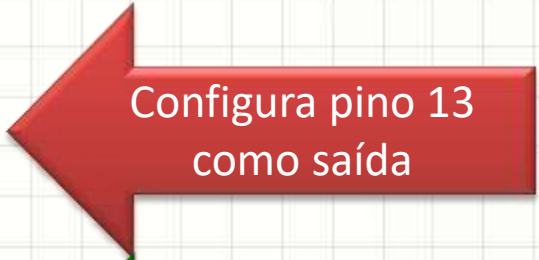
Calculadora de resistores

http://www.audioacustica.com.br/exemplos/Valores_Resistores/Calculadora_Ohms_Resistor.html

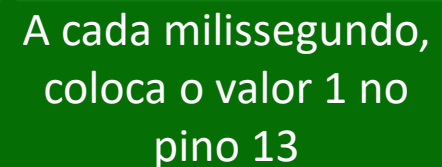
Hardware Arduíno UNO

- Como um programa pode acender o LED?
- Programa Arduino tem duas partes principais:
 - Função **setup()**: configurações
 - Função **loop()**: código repetitivo
- Exemplo

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(13, HIGH);  
}
```



Configura pino 13
como saída



A cada milissegundo,
coloca o valor 1 no
pino 13

Hardware Arduino UNO

- Como testar?
- Crie o arquivo **acende.ino** com esse conteúdo

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
}
```



- Carregue o arquivo no UnoArduSim

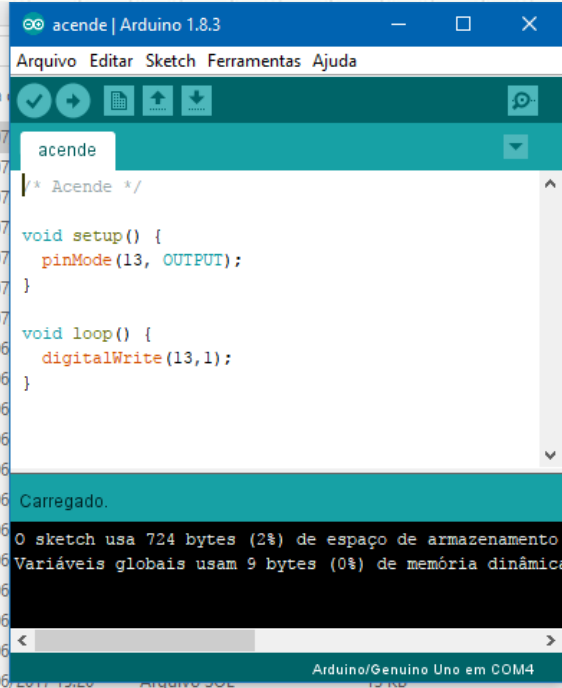


- Execute o programa



Hardware Arduíno UNO

- Como testar no hardware real?
- Abra a IDE do Arduino
- Carregue o arquivo **acende.ino**
- Configure a saída (**Ferramentas > Porta:**)
- Teste o programa 
- Envie o programa para o hardware 



```
acende
/* Acende */

void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13,1);
}
```

Carregado.

O sketch usa 724 bytes (2%) de espaço de armazenamento.
Variáveis globais usam 9 bytes (0%) de memória dinâmica.

Arduino/Genuino Uno em COM4

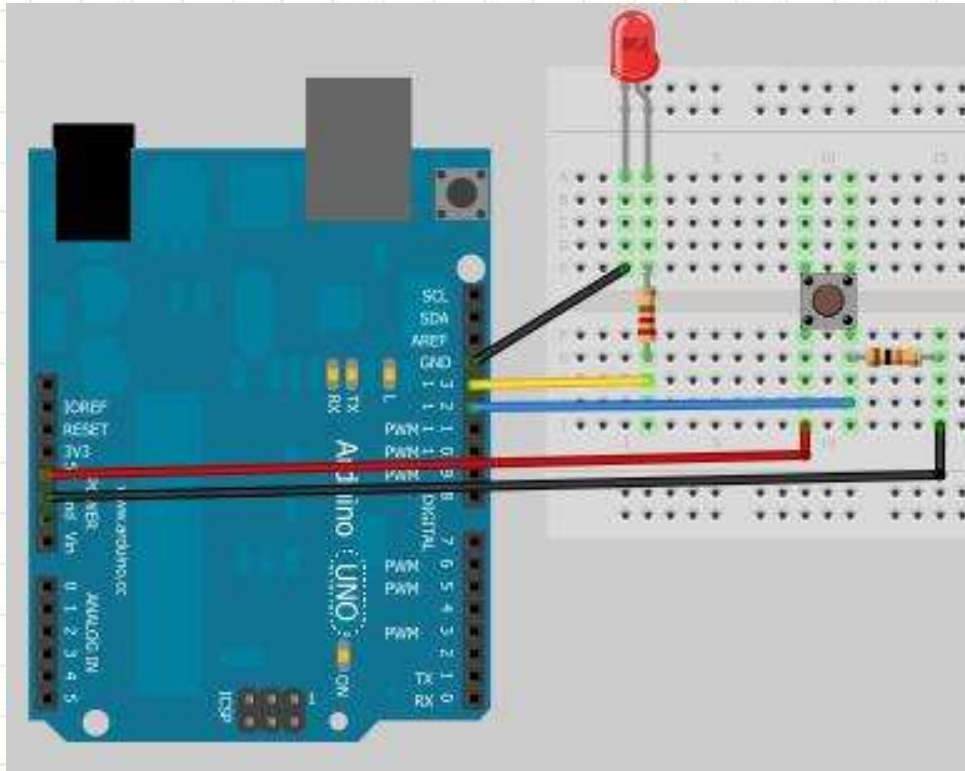
Como ligar um Botão no UnoArduSim



The image shows the UnoArduSim software interface. A central virtual Arduino Uno board is displayed, labeled "UNO ARDUINO" and "ATMEGA328". The board's pins are visible, with a red box highlighting pin 12. Surrounding the board are several peripheral modules:

- Serial:** A purple module with "TX chars" and "RX chars" fields, and a "Baud" rate set to 9600.
- SFTSER:** A teal module with "TX chars" and "RX chars" fields, and a "Baud" rate set to 300.
- SPISLV:** A yellow module for SPI communication, showing "MOSI", "MISO", "DATA", "Recv", and "SS*" fields.
- I2CSLV:** A red module for I2C communication, showing "SDA", "SCL", "Send", "Addr", "Recv", and "Clock" fields.
- Buttons:** Two "PUSH" buttons on the left, each with a "latch" checkbox.
- Resistors:** Four "R=1K" resistor modules on the left.
- PIEZO:** Two piezo sensor modules on the left.
- SERVO:** A servo motor module at the bottom left.
- MOTOR:** A motor module at the bottom center, with "Pwm" and "Enc" fields.
- PULSER:** A cyan module for pulse generation, showing a pulse waveform, "Pulse" (1000) and "Period" (10000) fields.
- FUNCGEN:** A green module for function generation, showing a square wave, "Period" (10000) field, and "0V" and "+5V" output indicators.
- LEDs:** Five "LED" modules on the right, each with "R" and "RYG" fields.

Como ligar um botão no Arduino



Importante:
10.000 ohms

images.google.com
Como ligar um botão no arduino

Calculadora de resistores

http://www.audioacustica.com.br/exemplos/Valores_Resistores/Calculadora_Ohms_Resistor.html

Como botão pode acender LED?



- Crie o arquivo **acendebotao.ino**:

```
void setup() {  
    pinMode(12, INPUT);  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    int button = digitalRead(12);  
    digitalWrite(13,button);  
}
```

- Carregue o arquivo no UnoArduSim
- Execute o programa



Hardware Arduíno UNO

- Como testar no hardware real?
- Abra a IDE do Arduino
- Carregue o arquivo **acendebotao.ino**
- Configure a saída (**Ferramentas > Porta:**)
- Teste o programa 
- Envie o programa para o hardware 



PERGUNTAS?



CONCLUSÕES

Resumo

- O que é um algoritmo
 - Linguagem de Programação
- O papel do compilador
- Noções de arquitetura de computadores
 - Como o computador enxerga o mundo
- Noção básica de um software Arduino

-
- Programas sequenciais
 - Estruturas de Decisão



PARTE PRÁTICA

Parte Prática

- Por que realizar uma parte prática?

Eu vejo e eu esqueço.
Eu ouço e eu lembro.
Eu faço e eu compreendo.
- Confúcio



Depois de 2 semanas,
nós lembramos de...

- 10% do que LEMOS
- 20% do que OUVIMOS
- 30% do que VEMOS
- 50% do que VEMOS e OUVIMOS
- 70% do que FALAMOS
- 90% do que FALAMOS e FAZEMOS

PASSIVO

ATIVO

Fonte: Edgar Dale (1969)



FORMAÇÃO DE GRUPOS

Projeto em Grupo

- Por que grupos?
 - Trabalho único no semestre todo
- Quantos alunos?
 - De 5 a 6 alunos
 - Por quê?
- Parte em aula, parte em casa
 - Entrega e apresentações
- Cadastro:
 - No fim da aula



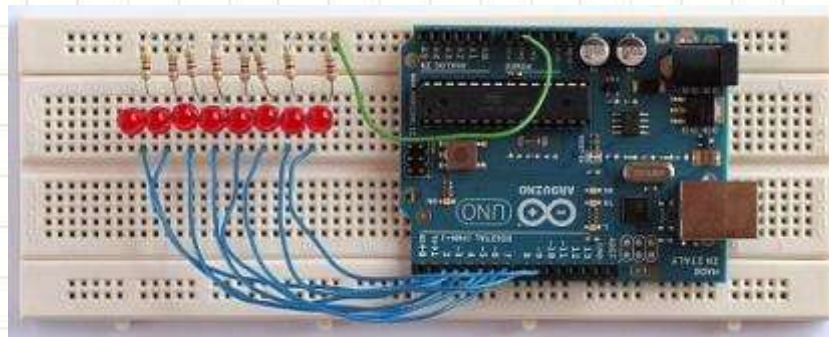
Projeto em Grupo

- O que é o projeto (Parte A)?
 - Conhecer/Aprender a plataforma Arduino UNO/Nano
 - Examinar soluções, elementos e sensores
 - Propor uma solução com Arduino
 - Implementar essa solução
 - Apresentar essa solução
 - Documentada!



Projeto em Grupo

- O que é o projeto (Parte A)?
- Simulador de Arduíno (UnoArduSim)
 - <https://www.sites.google.com/site/unoardusim/>



- Tutoriais
 - <https://www.youtube.com/watch?v=5P62WfQ4ijs>
 - <http://www.comofazerascosas.com.br/projeto-arduino-pisca-led.html>

Projeto em Grupo

- O que é o projeto (Parte B)?
 - Explicitar a oportunidade de negócio
 - Avaliar custos
 - Avaliar público e potencial receita (cenários)
 - Elaborar plano de negócio formal
 - Apresentar a empresa
 - Apresentação curta
 - Focada no resultado
 - Vender para um investidor



Trabalho para Hoje

- Pesquise **arduino & inovação** e veja usos inovadores para a plataforma.
- Responda às questões A e B da lista no site do professor, referentes à Aula 01.
- Se não entregou a lista de alunos do grupo, entregue (lista referente à Aula 00).

