



FIAP

Engenharia de Software

EDGE COMPUTING & COMPUTER SYSTEMS

02 – Flasher no Simulador e Hands-On



Prof. Airton Y. C. Toyofuku



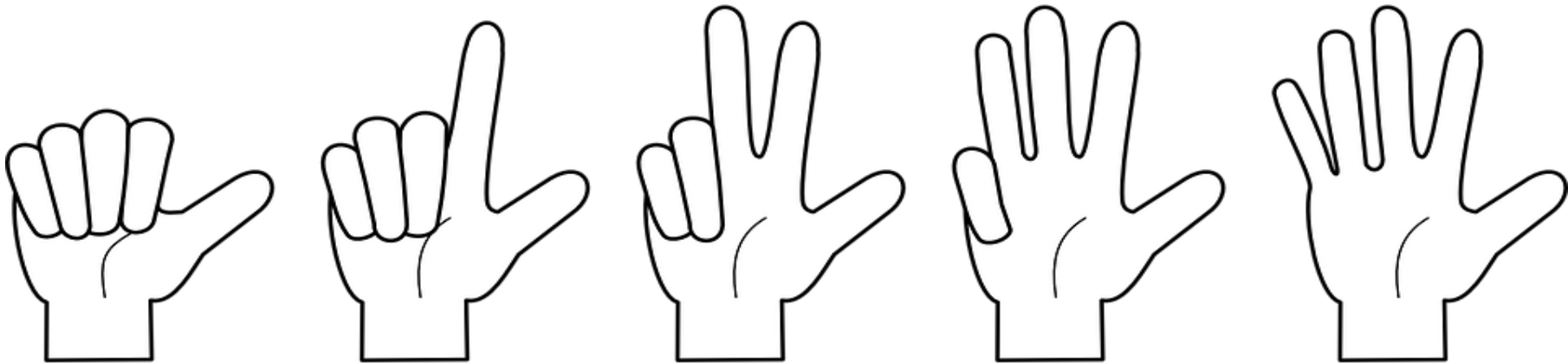
profairton.toyofuku@fiap.com.br

1. Sistemas de Numeração

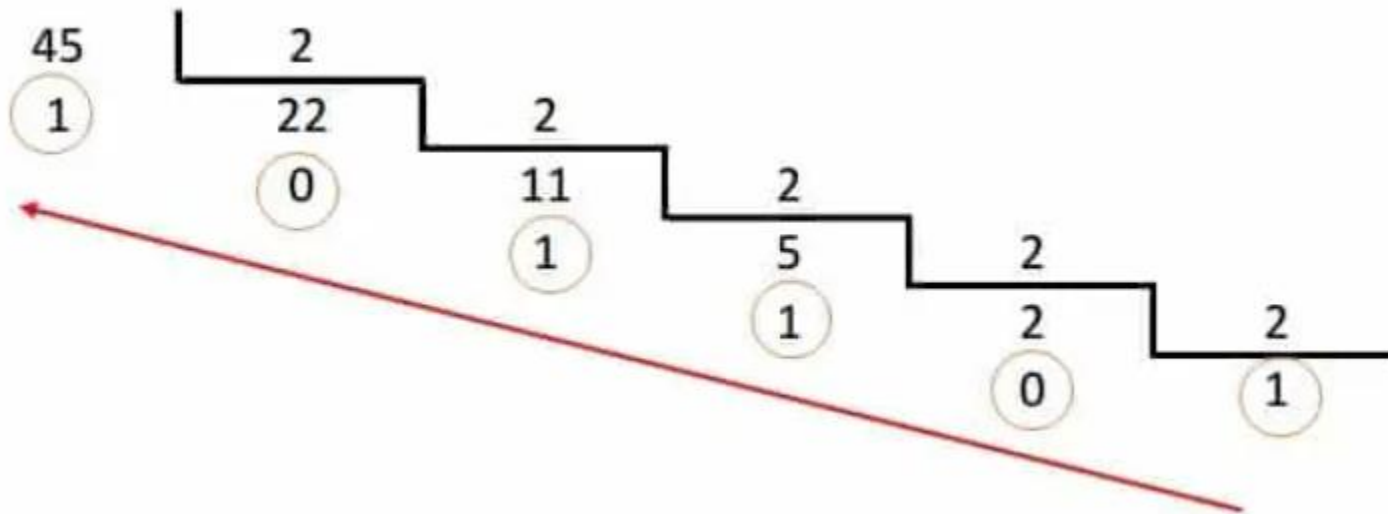
Representação Numérica

Compreender os sistemas digitais requer um entendimento dos sistemas decimal, binário e hexadecimal.

- Decimal—dez símbolos (base10)
- Binário—dois símbolos (base2)
- Hexadecimal—dezesesseis símbolos (base16)



Representação Numérica – Conversão Decimal - Binária



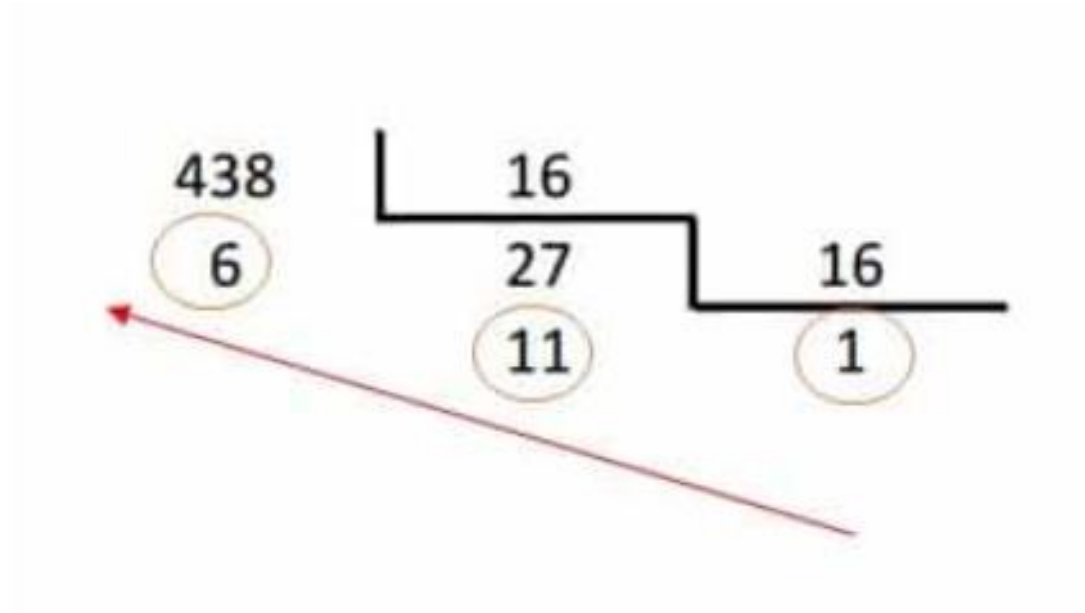
$$45_D = 101101_B$$

Representação Numérica – Conversão Binária - Decimal

1	0	1	1	0	1
2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1

$$101101_B = 1 \times 1 + 0 \times 2 + 1 \times 4 + 1 \times 8 + 0 \times 16 + 1 \times 32 = 45_D$$

Representação Numérica – Conversão Decimal - Hexadecimal



$$438_D = 1B6_H$$

Representação Numérica – Conversão Hexadecimal - Decimal

0	0	0	1	B	6
16^5	16^4	16^3	16^2	16^1	16^0
1.048.576	65.536	4.096	256	16	1

$$1B6_H = 6 \times 1 + 11 \times 16 + 1 \times 256 = 438_D$$

Representação Numérica

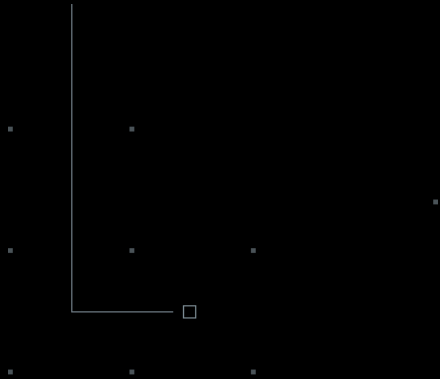
Hexadecimal	Decimal	Binário
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Hexadecimal	Decimal	Binário
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

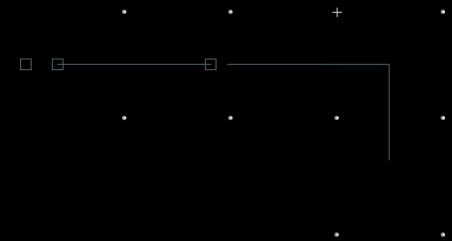
Tabela ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Fonte: <https://pt.wikipedia.org/wiki/Ficheiro:ASCII-Table-wide.svg>



2. Introdução a lógica de programação



Lógica de programação

É a técnica de encadear pensamentos para atingir um determinado objetivo.

Como fazer isso? → Algoritmo

“Sequência de ações que permite solucionar um determinado problema”



Quais são os passos necessários para se trocar uma lâmpada queimada?

- 1 – Comprar uma lâmpada nova;
- 2 – Pegar uma escada;
- 3 – Desligar o interruptor;
- 4 – Pegar a lâmpada nova;
- 5 – Subir na escada;
- 6 – Remover a lâmpada queimada;
- 7 – Instalar a lâmpada nova;
- 8 – Descer da escada;
- 9 – Descartar a lâmpada queimada;
- 10 – Acionar o interruptor;



A linguagem de programação é um método padronizado, com a finalidade de converter um algoritmo em instruções para execução do comportamento desejado em um computador.

Fonte: <https://www.freebsdbrasil.com.br/empresa/dennis-ritchie.html>

Porque linguagem C?

- Eficiência;
- Portabilidade;
- Flexibilidade;
- Orientado ao Programador;
(Fácil de converter em programas os algoritmos)

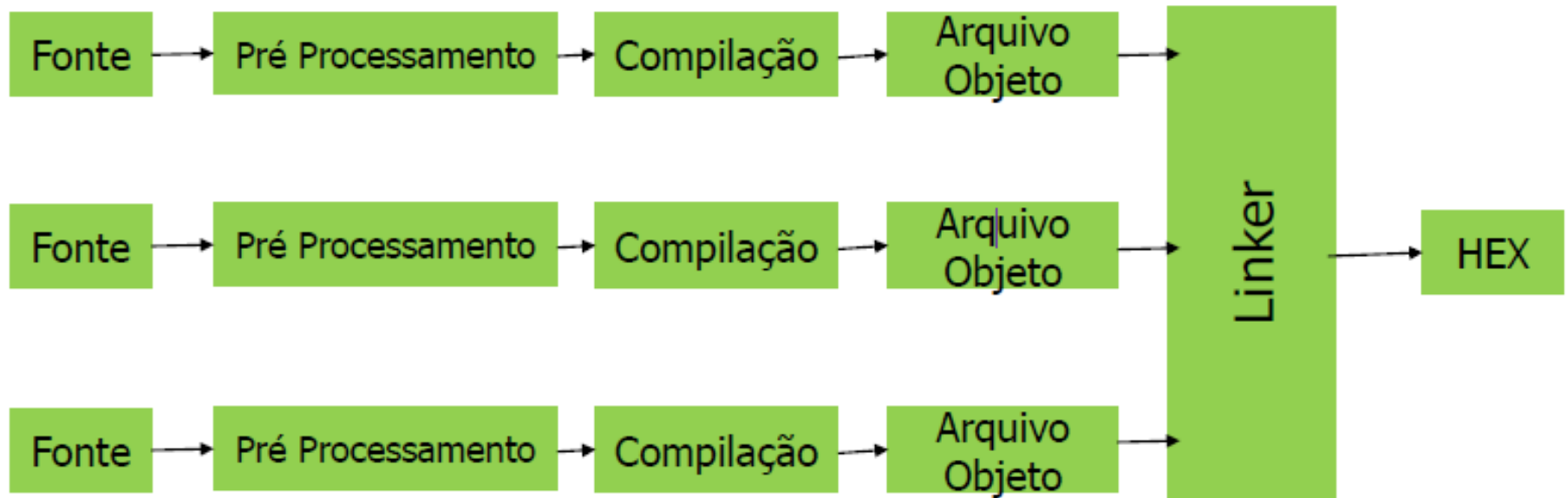


Dennis Ritchie

Criador do Unix e da
Linguagem C

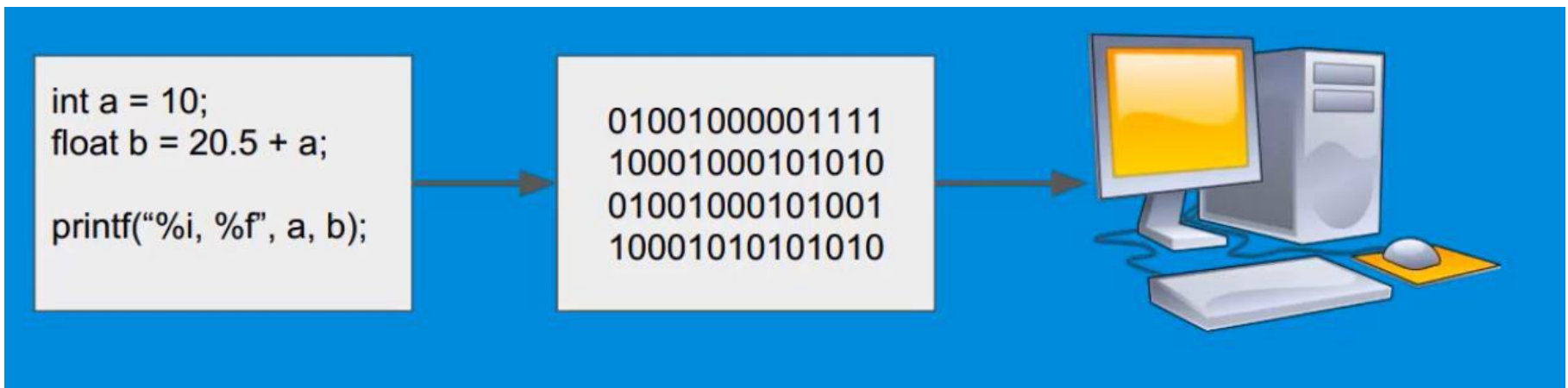
Arquivos Objeto, Executáveis e Bibliotecas

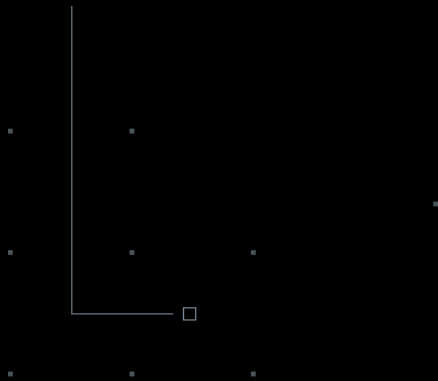
O processo de conversão dos arquivos fonte em executável, é dividido em Pré-Processamento, Compilação e Linkagem.



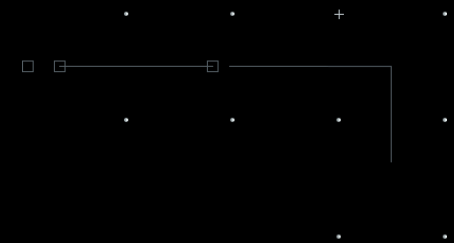
Arquivos Objeto, Executáveis e Bibliotecas

O resultado final é um arquivo em hexadecimal com os comandos em hexadecimal para o dispositivo executar o algoritmo desejado





3. Arduino



O Arduino

Basicamente o Arduino é uma plataforma de prototipagem “**Open Source**” de eletrônica que foi desenvolvida para fins educacionais, para projetistas amadores (Makers) e facilitar o desenvolvimento de provas de conceitos (POCs).

Pequeno computador com hardware limitado, livre e de placa única



Fonte: <https://www.arduino.cc/>

O Arduino



Fonte: <https://www.arduino.cc/>

- 1 - Conector USB para o cabo tipo AB
- 2 - Botão de reset
- 3 - Pinos de entrada e saída digital e PWM
- 4 - LED verde de placa ligada
- 5 - LED laranja conectado ao pin13
- 6 - ATmega encarregado da comunicação com o computador
- 7 - LED TX (transmissor) e RX (receptor) da comunicação serial
- 8 - Porta ICSP para programação serial
- 9 - Microcontrolador ATmega 328, cérebro do Arduino
- 10 - Cristal de quartzo 16Mhz
- 11 - Regulador de tensão
- 12 - Conector Jack fêmea 2,1mm com centro positivo
- 13 - Pinos de tensão e terra
- 14 - Pinos de entrada analógica

O Arduino – O projeto

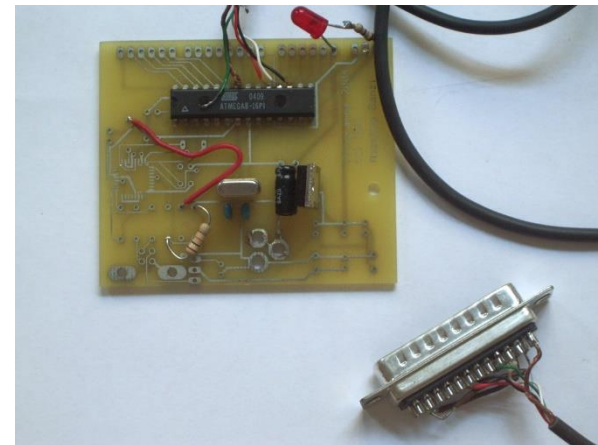
O **Arduino** foi criado em 2005 por um grupo de 5 pesquisadores : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis.

O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores.



David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi

Fonte: <https://linuxhint.com/who-invented-arduino/>



Primeiro protótipo 2005

Fonte: <https://embarcados.com.br/construindo-sua-placa-arduino/>

O Arduino – Modelos de Placas



Microcontrolador	ATmega328P	ATmega32u4	Intel Curie	ATmega32u4
Tensão de operação	5V	5V	3.3V (5V tolerant I/O)	5V
Tensão de alimentação	7-12V	7-12V	7-12V	
Pinos I/O digital	14 (of which 6 provide PWM output)	20	14 (of which 4 provide PWM output)	
Pinos I/O PWM digital	6	7	4	
Pinos analógicos	6	12	6	
Corrente DC por pino I/O	20mA	40mA	20mA	
Corrente DC por pino I/O de 3,3V	50mA	50mA		

Fonte: <https://www.arduino.cc/>

O Arduino – Modelos de Placas



Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader	32 KB (ATmega32u4) of which 4 KB used by bootloader	196 kB	32 KB of which 4 KB used by bootloader
SRAM	2 KB (ATmega328P)	2.5 KB (ATmega32u4)	24KB	2.5 KB
EEPROM	1 KB (ATmega328P)	1 KB (ATmega32u4)		1 KB
Clock Speed	16 MHz	16 MHz	32Mhz	16 MHz
Peso	25g	20g	34g	53g
Features			Bluetooth LE, 6-axis accelerometer/gyro	Analog joystick; Microphone; Light sensor; Temperature sensor ; three-axis accelerometer; Buzzer

Fonte: <https://www.arduino.cc/>

O Arduino – Modelos de Placas



Microcontrolador	ATmega32U4	ATmega328
Tensão de operação	5V	5V
Tensão de alimentação	7-12V	
Pinos I/O digital	20	22
Pinos I/O PWM digital	7	6
Pinos analógicos	12	8
Corrente DC por pino I/O	20mA	40mA
Corrente DC por pino I/O de 3,3V	50mA	

Fonte: <https://www.arduino.cc/>

O Arduino – Modelos de Placas



Flash Memory	256 KB of which 8 KB used by bootloader	256 KB	512 KB
SRAM	8 KB	32 KB	96 KB
EEPROM	4 KB		
Clock Speed	16 MHz	48 MHz	84 MHz
Peso	37 g	12g	36g

Fonte: <https://www.arduino.cc/>

O Arduino – Ambiente de programação

- Ambiente integrado de Desenvolvimento (IDE)

Pode ser gratuitamente baixado do site www.arduino.cc

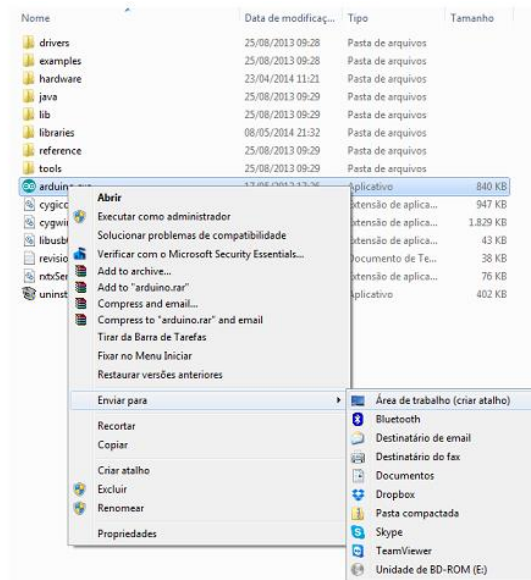


Fonte: <https://www.arduino.cc/>

O Arduino – Ambiente de programação

- Ambiente integrado de Desenvolvimento (IDE)

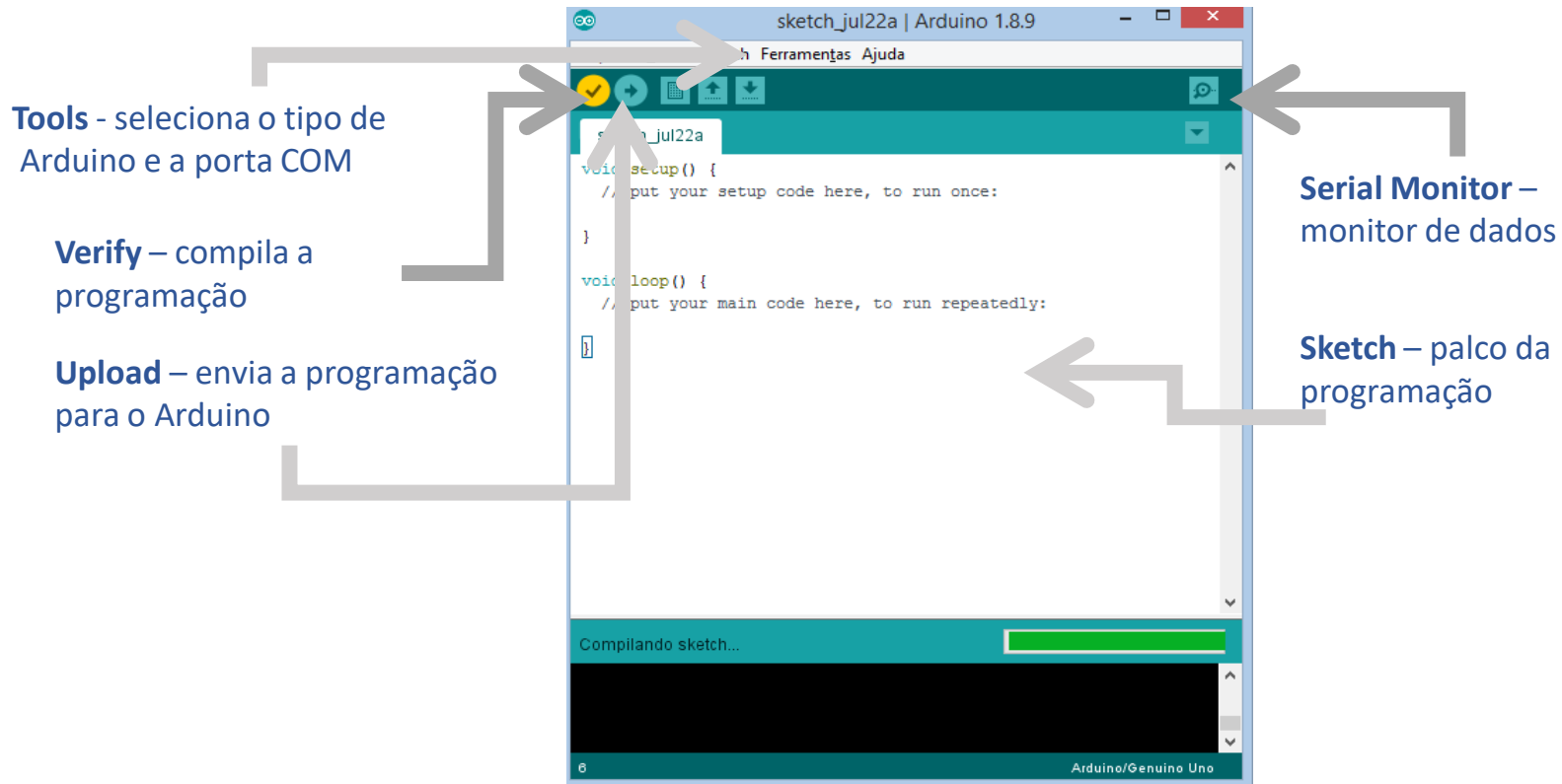
Agora basta criar um atalho da IDE na área de trabalho e você já poderá programar sua placa!



Fonte: Autor

O Arduino – Ambiente de programação

- Ambiente integrado de Desenvolvimento (IDE)

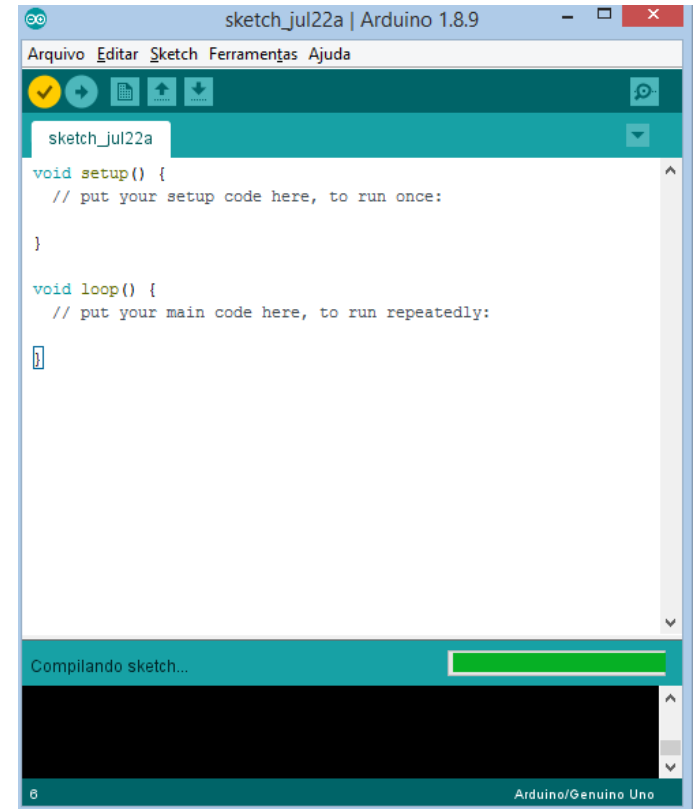


Fonte: Autor

O Arduino – Ambiente de programação

- O IDE é muito simples e intuitivo. Um programa, que no Arduino é chamado de sketch, apresenta duas funções básicas: `setup()` e `loop()`.
- A função **`setup()`** deverá conter o código que irá executar apenas uma vez, quando o sketch iniciar. Normalmente colocamos nesta função as definições iniciais do programa.

```
void setup() {  
    // initialize the LED pin as an output:  
    pinMode(ledPin, OUTPUT);  
    // initialize the pushbutton pin as an input:  
    pinMode(buttonPin, INPUT);  
}
```

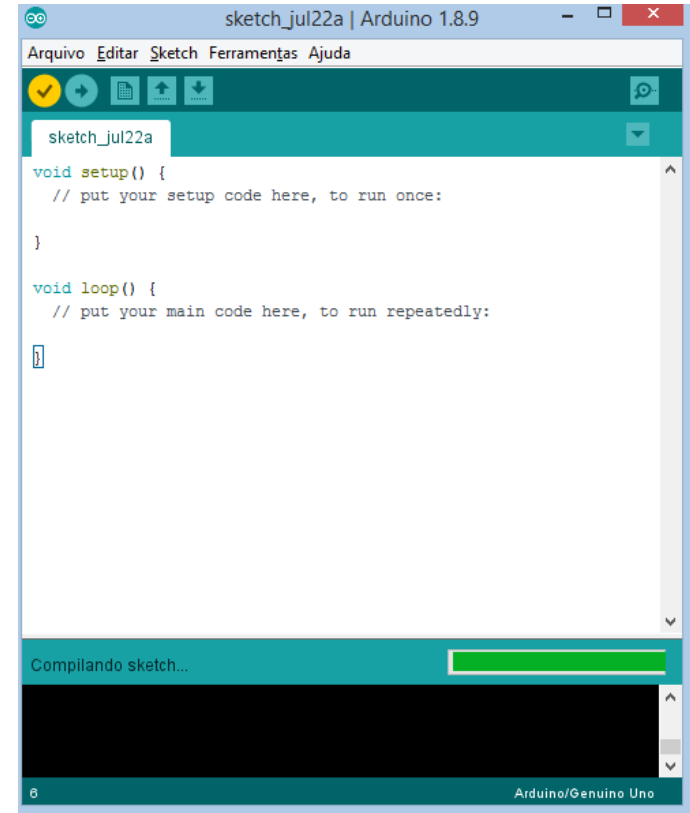


Fonte: Autor

O Arduino – Ambiente de programação

- A função **loop()** irá executar continuamente as instruções que estão lá até que outro sketch seja carregado na memória “flash” do Arduino.
- É importante notar que no Arduino é possível armazenar e executar um sketch por vez, desta forma, sempre quando transferimos um sketch esse irá substituir o programa que estava anteriormente carregado na memória.

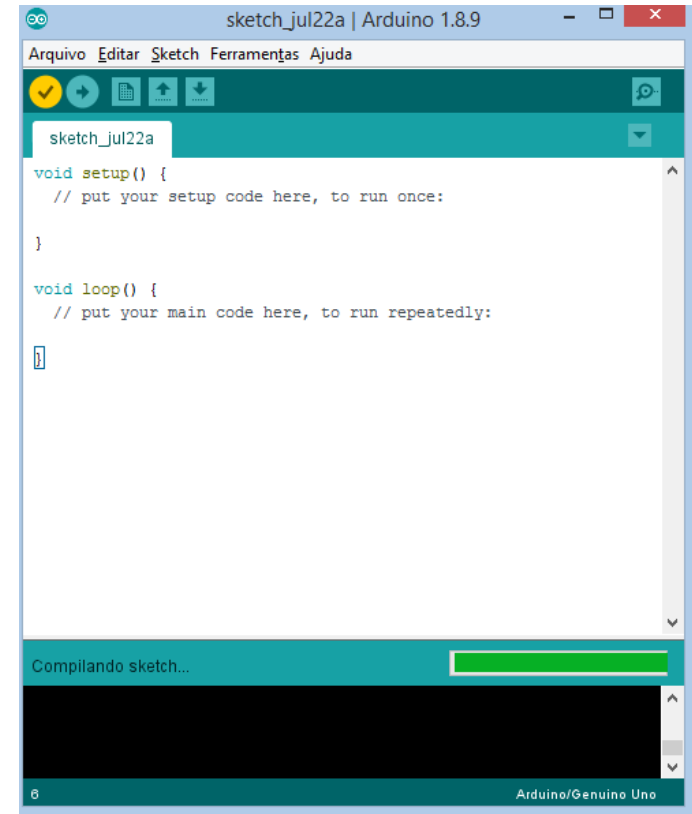
```
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
  } else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}
```



Fonte: Autor

O Arduino – Ambiente de programação

- Também observe que como o sketch fica armazenado na memória “flash”, que é permanente, mesmo quando desligamos o Arduino, o programa continua armazenado e irá entrar novamente em execução quando o Arduino for ligado novamente.
- Note também que, nestas duas funções, a palavra reservada **void** indica que as funções não apresentam um valor de retorno, sendo usadas exclusivamente para realizar a execução de um conjunto de instruções.



Fonte: Autor

Project 1

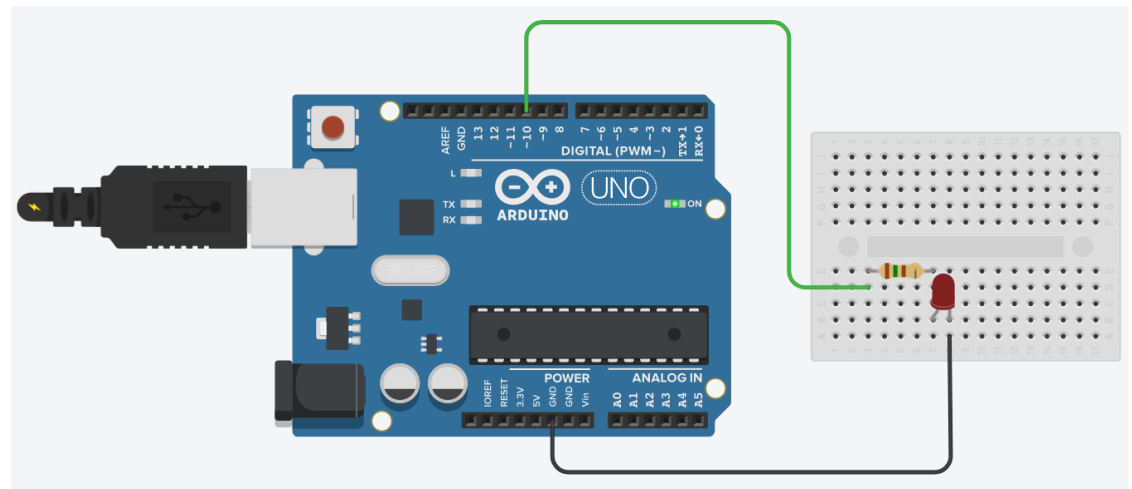
LED Flasher

Led Flasher

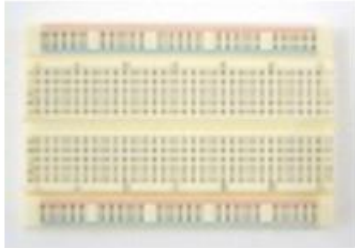



Um nível 1 (HIGH) colocado no pino irá acender o LED durante 1s, enquanto um nível 0 (LOW) vai apagar o LED por 1s.

Material necessário:

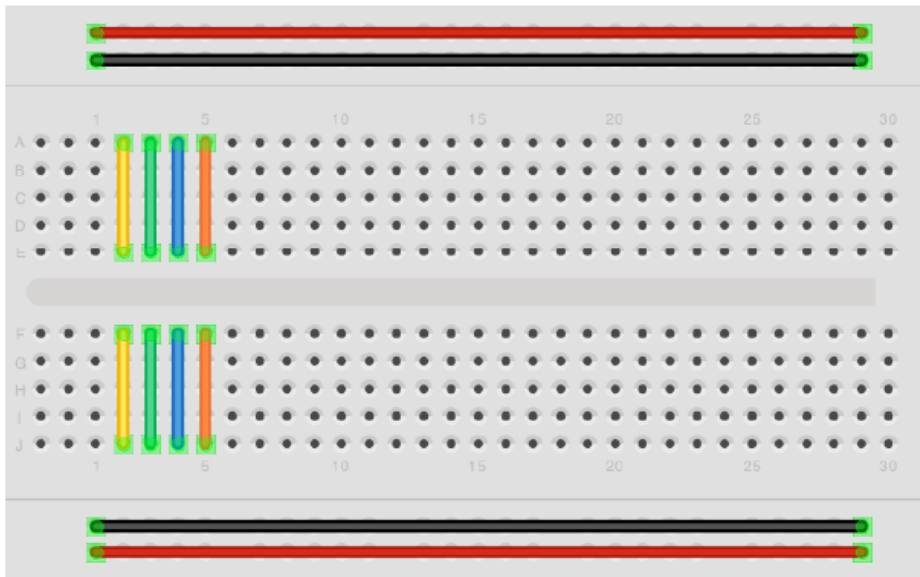
- 1 Arduino;
- 1 Resistor de 150 ohms (marrom, verde, marrom);
- 1 Led (qualquer cor);
- 1 Protoboard;
- Jumpers cables.



Conhecendo o Hardware

Breadboard	
Red LED	
150 Ω Resistor	
Jumper Wires	

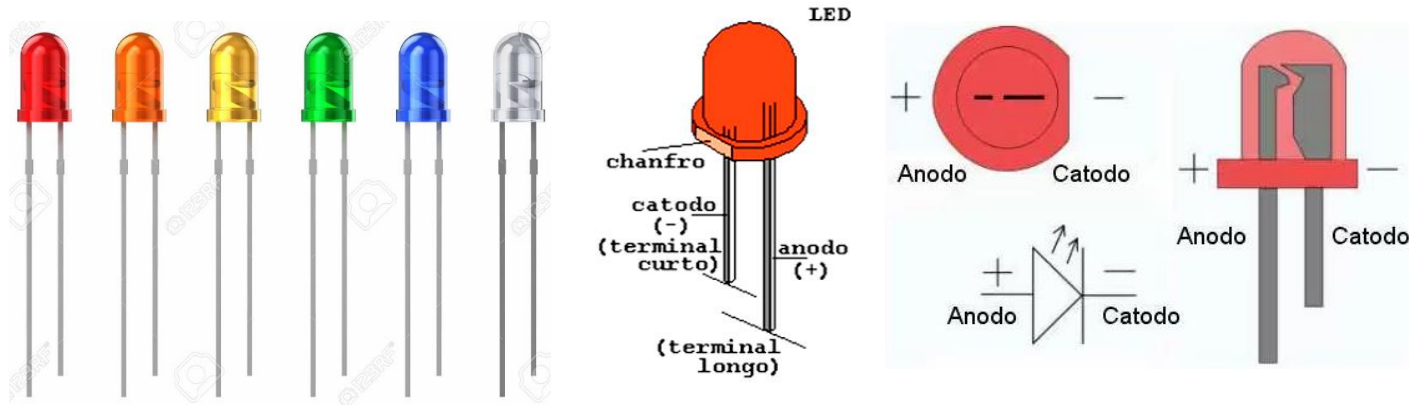
Conhecendo o Hardware – Protoboard



- Matriz de contatos elétricos;
- As Extremidades são ligadas na horizontal e servem como pontos de alimentação (VCC e GND);
- O interior segue uma matriz vertical, ligando 5 pontos.
- A matriz de cima não fala com a matriz de baixo!

Conhecendo o Hardware – LED

O “**Led**” são dispositivos emissores de luz



As informações mais importantes são: Polaridade, tensão limite e a corrente máxima;

O Led tem a posição correta de ser ligado, onde tem um chanfro ou terminal menor é o cátodo (Negativo) e o terminal maior é o ânodo (positivo)

Existe em diversos tamanhos e formatos redondo, quadrado e retangular...

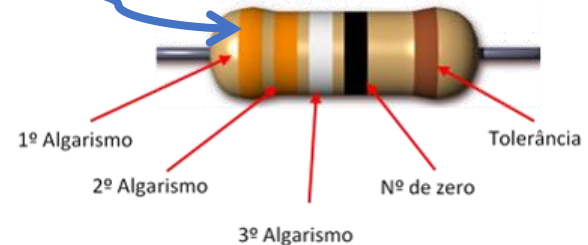
Conhecendo o Hardware – Resistor



- Componente eletrônico usado para limitar a passagem de corrente elétrica;
- Causam uma queda de tensão controlada no circuito eletrônico;
- Sua medida é em Ohms e são regidos pela Lei de Ohm;
- Possuem muitos valores e são identificados por um Código de Cores;
- Também são usados para esquentar alguma coisa (chuveiro);

Conhecendo o Hardware – Resistor

A primeira faixa é sempre a mais próxima do terminal

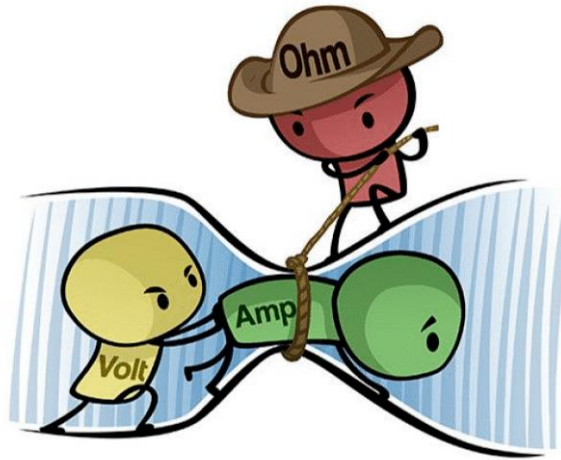
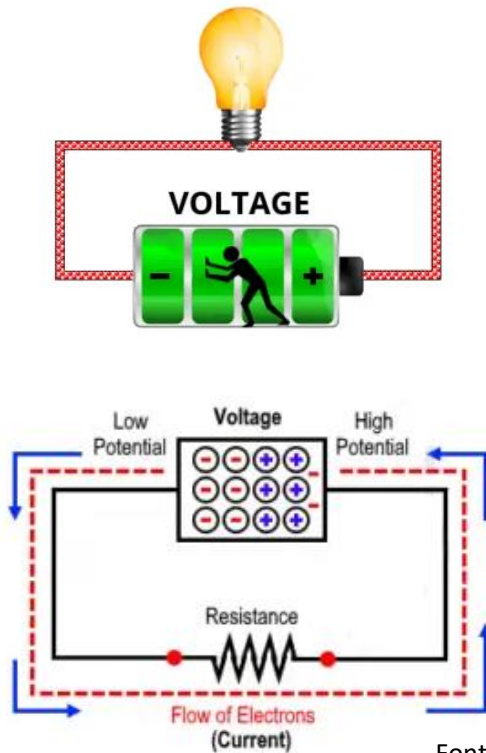


Os “**resistores**” são componentes com a finalidade de oferecer resistência à passagem da corrente elétrica.

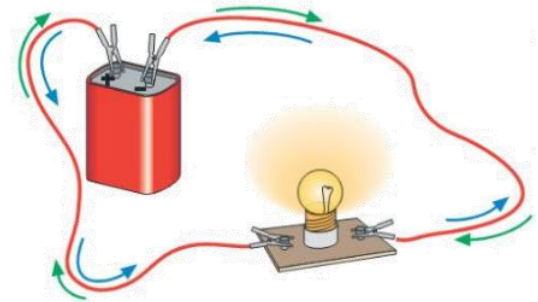
Cores	Valores			Multiplicadores X	Tolerância %
	Faixa 1	Faixa 2	Faixa 3		
Prata	-	-	-	0,01	10%
Ouro	-	-	-	0,1	5%
Preto	-	0	0	1	-
Marrom	1	1	1	10	1%
Vermelho	2	2	2	100	2%
Laranja	3	3	3	1000	-
Amarelo	4	4	4	10000	-
Verde	5	5	5	100000	5%
Azul	6	6	6	1000000	0,25%
Violeta	7	7	7	10000000	0,10%
Cinza	8	8	8	-	-
Branco	9	9	9	-	-
Sem cor	-	-	-	-	20%

Fonte: <https://aprendendoeletrica.com/codigo-de-cores-para-resistores/>

Conhecendo o Hardware – Resistor



Sentido
convencional



Fonte: <https://portald Engenharia.com/instalacao-eletrica/o-que-e-tensao-eletrica/>
Fonte: <https://embarcados.com.br/lei-de-ohm/>

Conhecendo o Hardware – Resistor

$$LEI \text{ de OHM} \rightarrow R = \frac{V}{I}$$

- R = Resistência Elétrica em Ohms(Ω);
 - V = Queda de tensão no resistor em Volts (V);
 - I = Corrente elétrica que passa pelo resistor em Amperes (A);
- ❖ Exemplo: Qual resistor eu devo usar para ligar um LED que consome 20mA a 3,3 Volts?

$$LEI \text{ de OHM} \rightarrow R = \frac{V}{I} \rightarrow R = \frac{3,3}{0,020} \rightarrow R = 165 \text{ Ohms}$$

Copyright © 2023 Prof. Airton Y. C. Toyofuku

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).