

Aula prática – Novos componentes e revisão

Componentes:



Display LCD 16x2



Teclado matricial 4x4

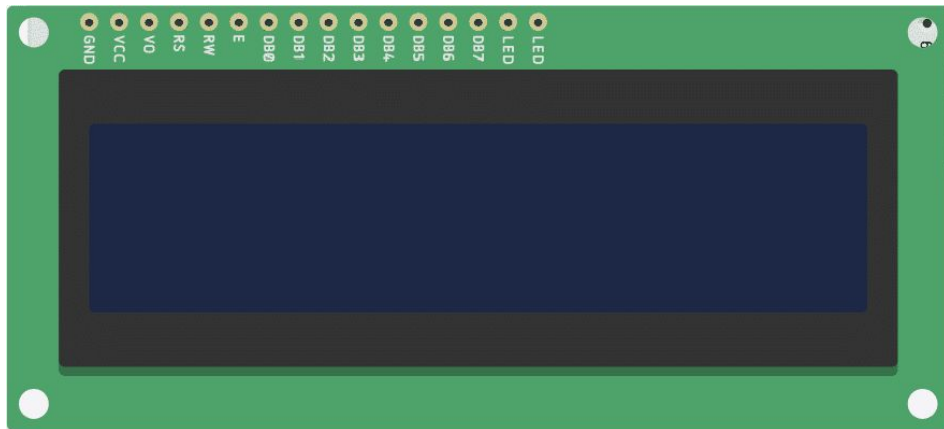


LDR



Potenciômetro

Display 16x2:



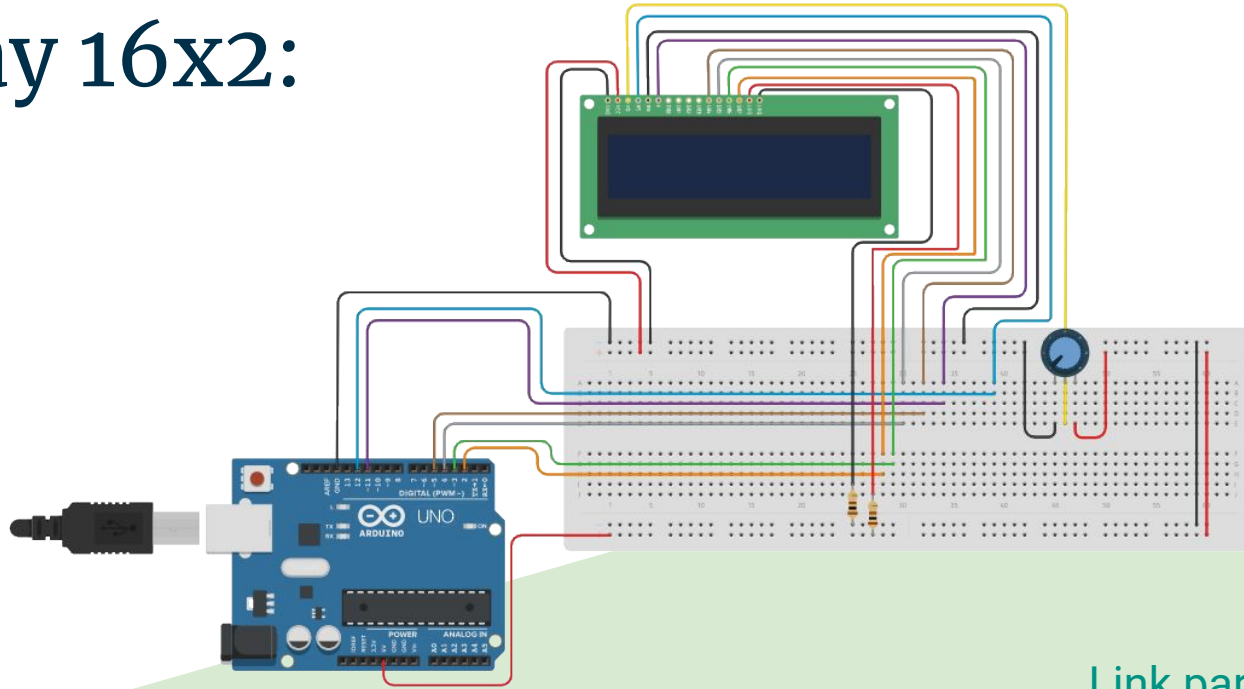
Display LCD 16x2 no Tinkercad

Conexões LCD 16x2 - HD44780		
Pino LCD	Função	Ligação
1	Vss	GND
2	Vdd	Vcc 5V
3	V0	Pino central potenciômetro
4	RS	Pino 12 Arduino
5	RW	GND
6	E	Pino 11 Arduino
7	D0	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 4 Arduino
13	D6	Pino 3 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

Ligações do display

Feito por Ronaldo M. Zica para o Enxurrada de bits

Display 16x2:



[Link para o circuito no tinker cad](#)

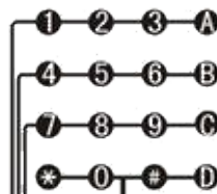
Teclado 4x4:



O teclado é equivalente a 16 push buttons

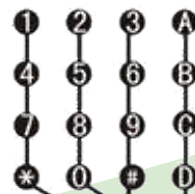


Linhas



1 4

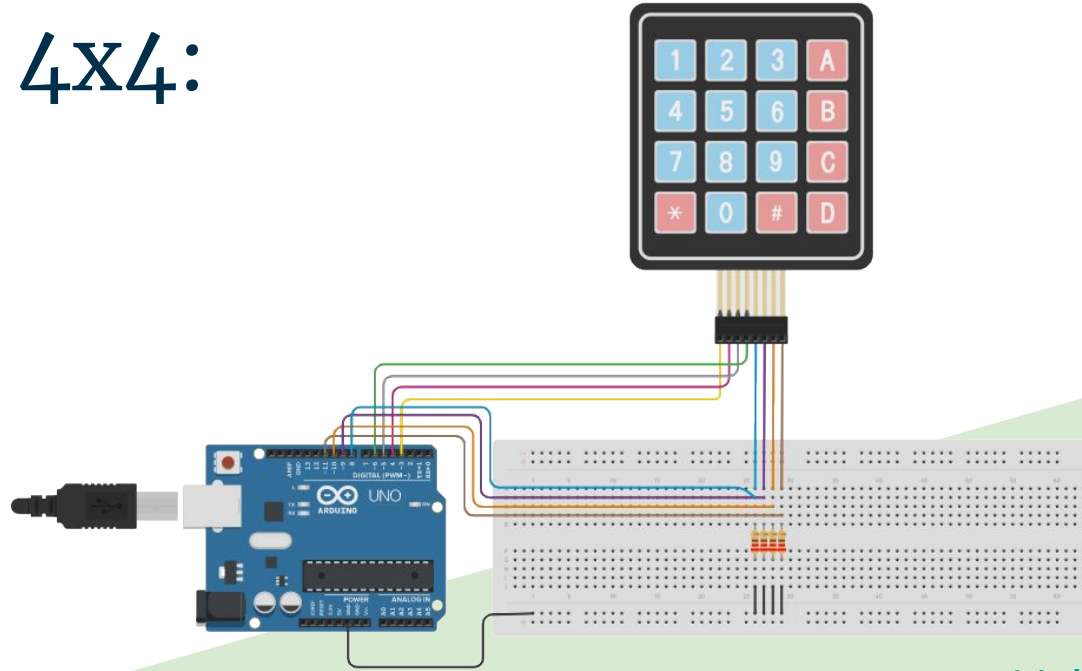
Colunas



5 8

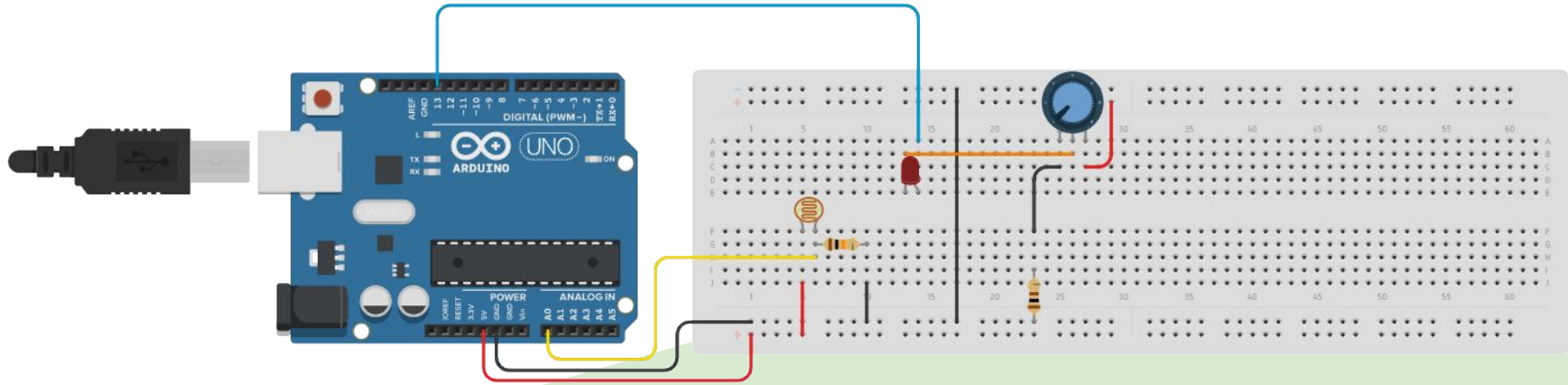
Funcionamento do teclado

Teclado 4x4:



[Link para o circuito
no tinker cad](#)

LDR e potenciômetro:

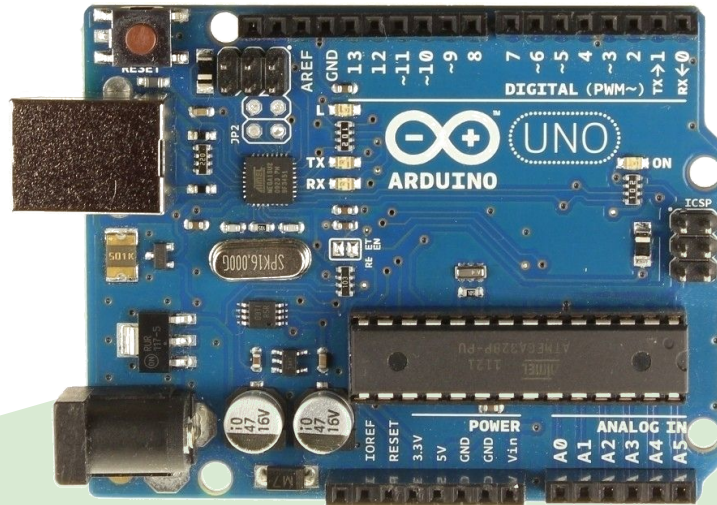


[Link para o circuito no tinkercad](#)

Referências:

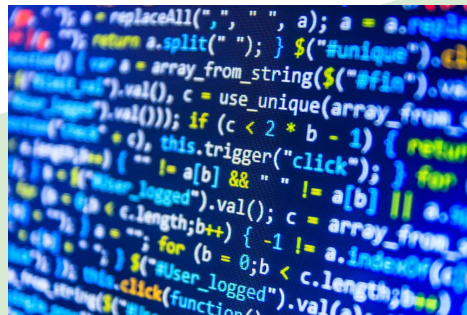
- [FilipeFlop - Controlando um display LCD com arduino.](#)
[Acesso em 26/09/2020;](#)
- [FilipeFlop - Teclado matricial 4x4 com arduino.](#)
[Acesso em 26/09/2020;](#)
- [FilipeFlop - Projeto 10: Sensor de luz de ambiente.](#)
[Acesso em 26/09/2020.](#)

Revisão de programação no Arduino



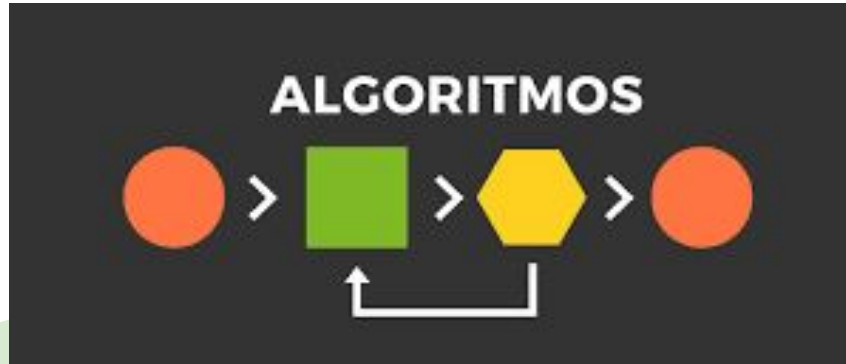
O que é programar um microcontrolador?

Programação de microcontroladores é o processo de desenvolver um **código**, que irá realizar instruções de acordo com o **algoritmo** que lhe foi programado.



Tá, mas o que é um algoritmo?

Um algoritmo é uma sequência de instruções tomada para resolver determinado problema.



Tá, mas o que é um algoritmo?

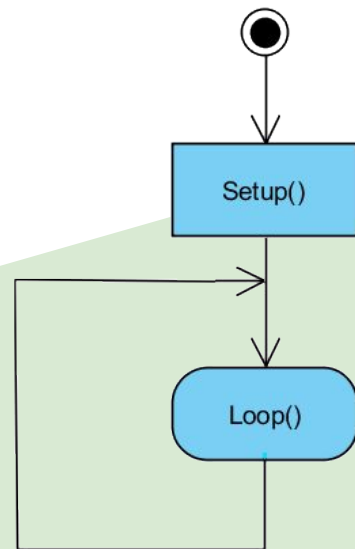
Aí vai um exemplo cotidiano de um algoritmo básico para você fazer café em uma cafeteira:



Como serão meus algoritmos no arduino?

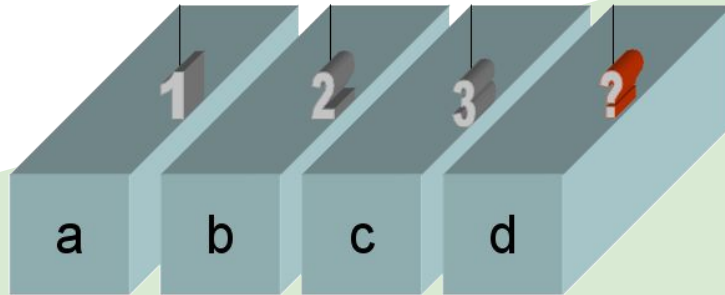
Seus algoritmos no arduino serão compostos por duas partes principais:

- **Setup:** Inicializa as portas do arduino e as **variáveis**.
- **Loop:** Repete as instruções contidas aqui até o fim da execução do programa.



Variáveis

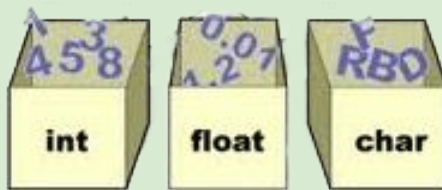
Variáveis são espaços na memória que podemos utilizar para armazenar algum valor, seja ele um número, uma letra ou até mesmo uma frase.



Variáveis

As variáveis devem atender a algumas regras:

- Possuir um tipo (número inteiro - *int*, número real - *float*, caractere - *char*, frase - *string...*)
- Possuir um nome que comece com uma letra e não contenha caracteres especiais, como acentos, operadores e valores que não estão na tabela ascii.
- Possuir um nome que não seja uma palavra reservada (lista de palavras reservadas no Anexo 1).



Variáveis

Quais caracteres estão na tabela ascii?

Caracteres numéricos, letras do alfabeto em maiúsculo e minúsculo (sem acento e sem o “ç”) e mais alguns caracteres especiais.

Via de regra, para o nome de suas variáveis use letras, números e, se necessário, *underline* “_”.

Ascii	Char	Ascii	Char	Ascii	Char
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	Forv

Variáveis

Exemplo de variáveis:

Conteúdo da variável	Nome da variável	Tipo da variável
5	quantidadeLeds	int
3.45	valorLuzLDR	float
"Hello World!"	fraseDisplay	string
true	ledVermelho	boolean
R	letraDisplay	char

Estruturas de controle – if


Estrutura “se”:

Expression is true.

```
int test = 5;
```

```
if (test < 10)
{
    // codes
}

// codes after if
```




Expression is false.

```
int test = 5;
```

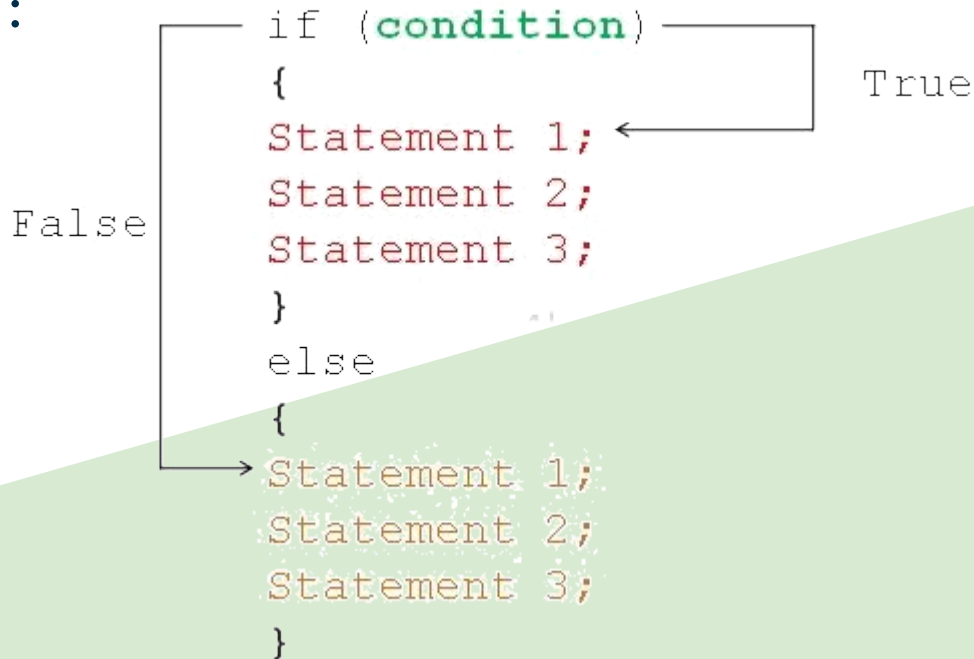
```
if (test > 10)
{
    // codes
}

// codes after if
```



Estruturas de controle – if / else

Estrutura “se/ou”:



Estruturas de repetição – while

Estrutura “enquanto”:

```
→ while (test Expression)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

Estruturas de repetição – for

Estrutura “para cada”:

```
→ for (init, condition, update)
{
    // codes
    if (condition for continue)
    {
        continue;
    }
    // codes
}
```

Anexo 1 – Lista de palavras reservadas:

- [Linguagem C - Lista de palavras reservadas.](#)
[Acesso em 02/10/2020.](#)

Referências:

- [tntBasic - Ascii code explained.](#)
[Acesso em 02/10/2020;](#)
- [Wikipedia - Computer programming.](#)
[Acesso em 02/10/2020;](#)
- [Wikipédia - Programação de computadores.](#)
[Acesso em 02/10/2020.](#)