

# Mini Curso Introdução ao Arduino

*"A ciência é um trabalho de artesanato intelectual." W. Mills*

Instrutor  
Walderlan Sena



# \$ \_ Ementa



## MODULO I

Apresentações gerais/Introdução

Breve História

O que é o arduino (Uno)

Tipos de Arduinos

Datasheet

Pinos G.P.I.O

Arduino IDE (Gnu/Linux, Windows, MacOs)

Estrutura/Padrões



## MODULO II

Wiring: Linguagem Baseada em C++

Documentação Principal

Criando o primeiro sketch: Blink

Tipos de dados

Sintaxe básica

Controle de fluxo

Principais bibliotecas

Funções da biblioteca padrão





# \$ \_ Ementa



## MODULO III

Introdução a Protoboard  
Mantendo o padrão de cores dos fios  
Encontrando valores de resistência com o código de cores  
Saídas digitais/Analógicas do Arduino  
Como acender e apagar um LED  
Lendo o estado de um botão  
Fazendo barulho com a função buzzer



## MODULO IV

Serial/USB com o PC  
O que são sensores  
O que são shields  
Utilizando sensor de luminosidade  
Shield Ponte H  
Como funciona Sensor Ultrasonico  
Sensor de toque (Touch)  
Arduino + Relé  
Extra: Projeto Final



\$ \_ Whoami

Walderlan Sena

Programador

Técnico em TI

Acadêmico de Análise e Desenvolvimento de Sistemas

Acadêmico de Advanced Penetration Test

Entusiasta IoT (Robótica, Automação) e de Segurança da Informação, E da Análise da Micro-Expressões Faciais (FACS)

Palestrante

Escritor nas horas vagas (Que não existem kkk)

Ceo e Fundador da empresa Mentres Virtuais Sena



*"Estamos aqui para fazer  
alguma diferença no  
universo, se não porque  
está aqui ?" - Steve Jobs*





# MÓDULO I





# \$ \_ Uma breve História

"O projeto iniciou-se na cidade de Ivrea, Itália, em 2005, com o intuito de interagir em projetos escolares de forma a ter um orçamento menor que outros sistemas de prototipagem disponíveis naquela época."

"Seu sucesso foi sinalizado com o recebimento de uma menção honrosa na categoria Comunidades Digitais em 2006, pela Prix Ars Electronica, além da marca de mais de 50.000 placas vendidas até outubro de 2008."



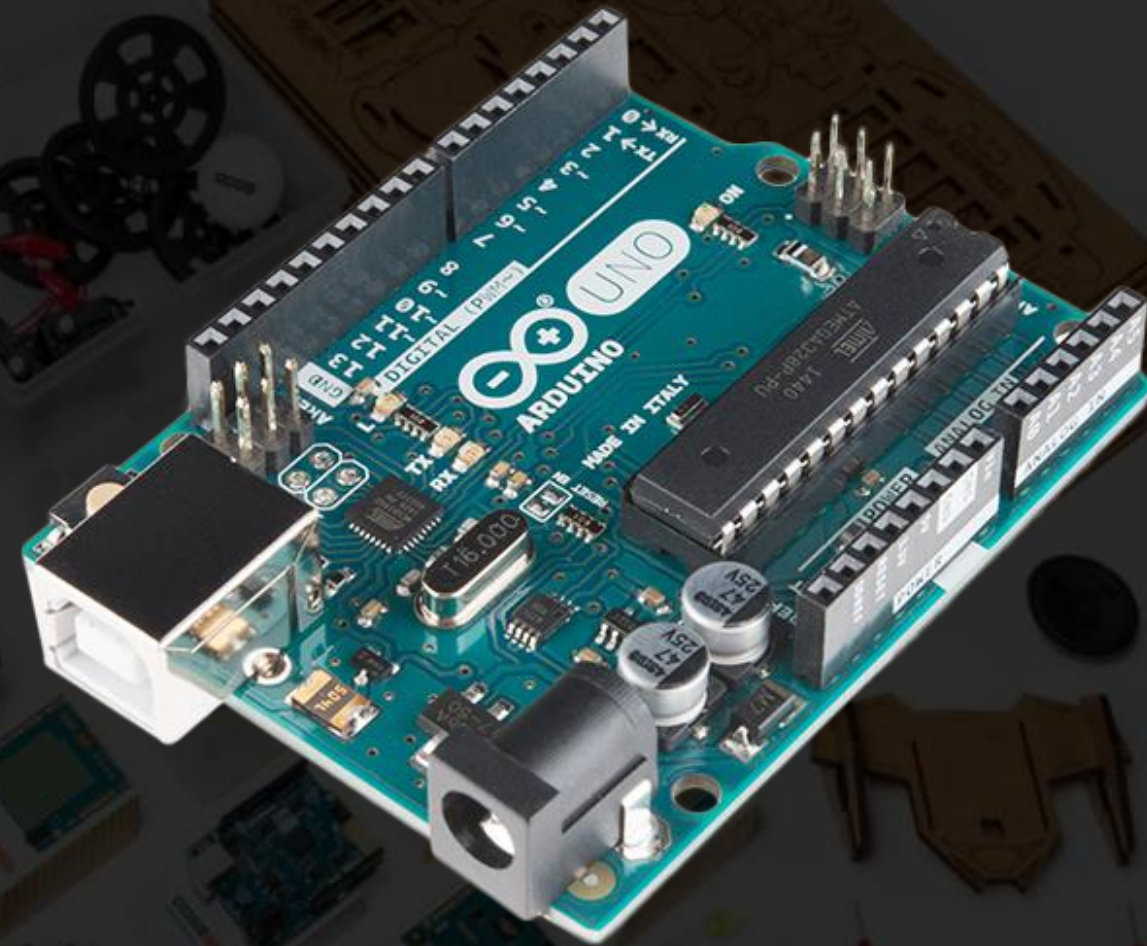
# \$ Uma breve História



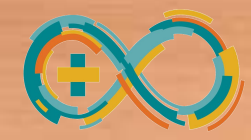
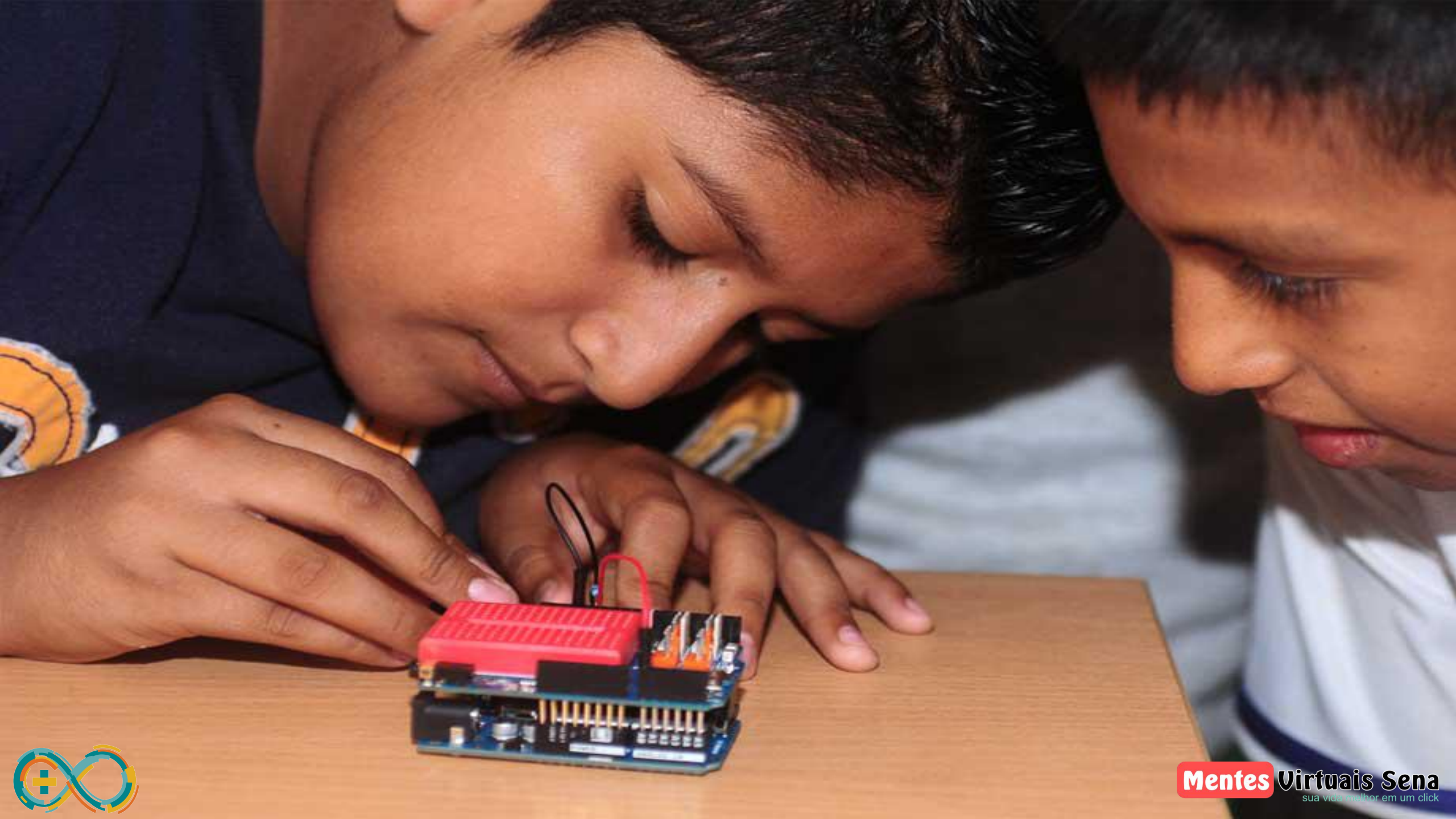


# \$ \_ O que é o arduino?

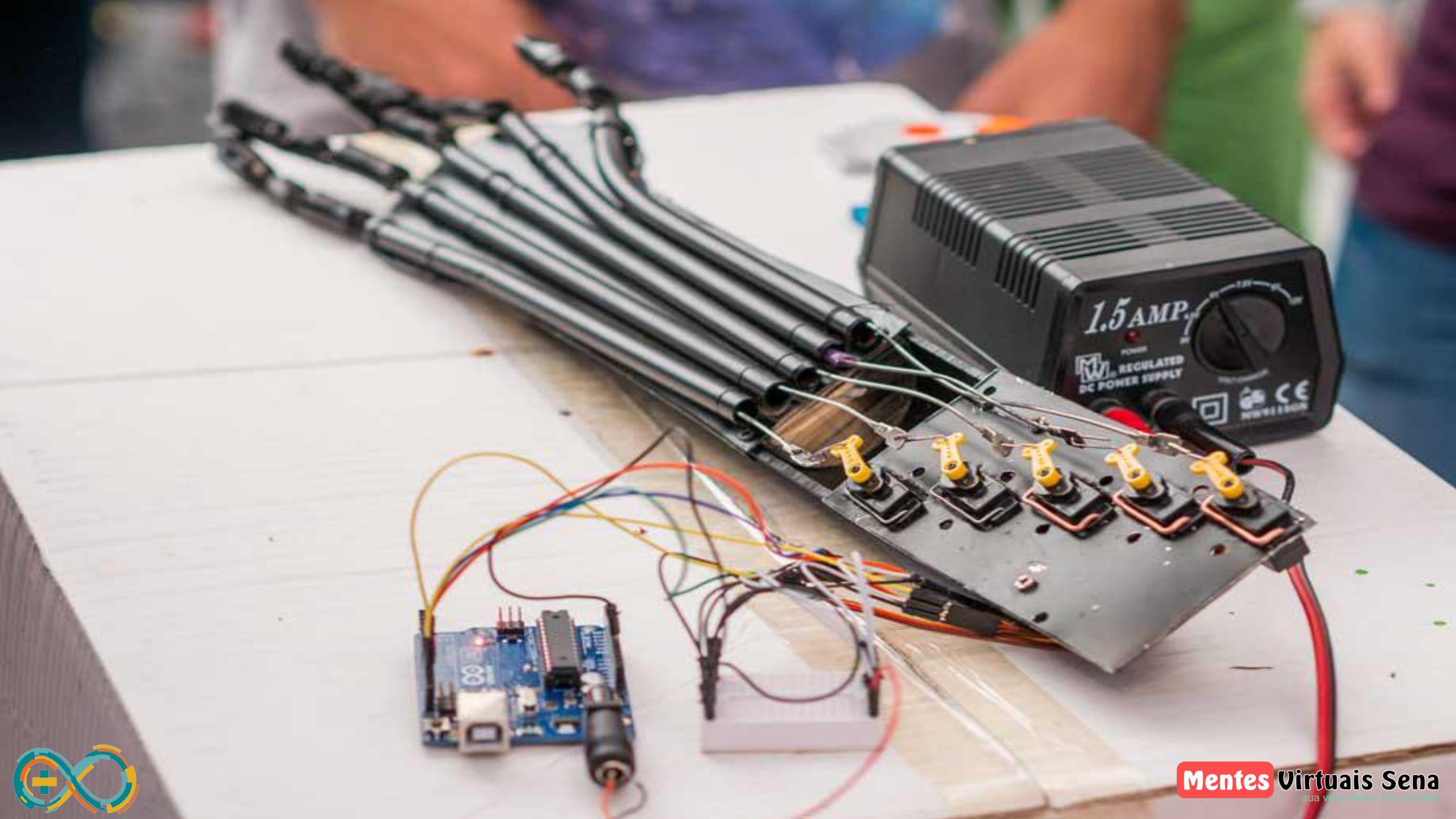
"O Arduino é uma plataforma de prototipagem eletrônica open-source que se baseia em hardware e software flexíveis e fáceis de usar. É destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos." *Fonte: [arduino.cc](http://arduino.cc)*











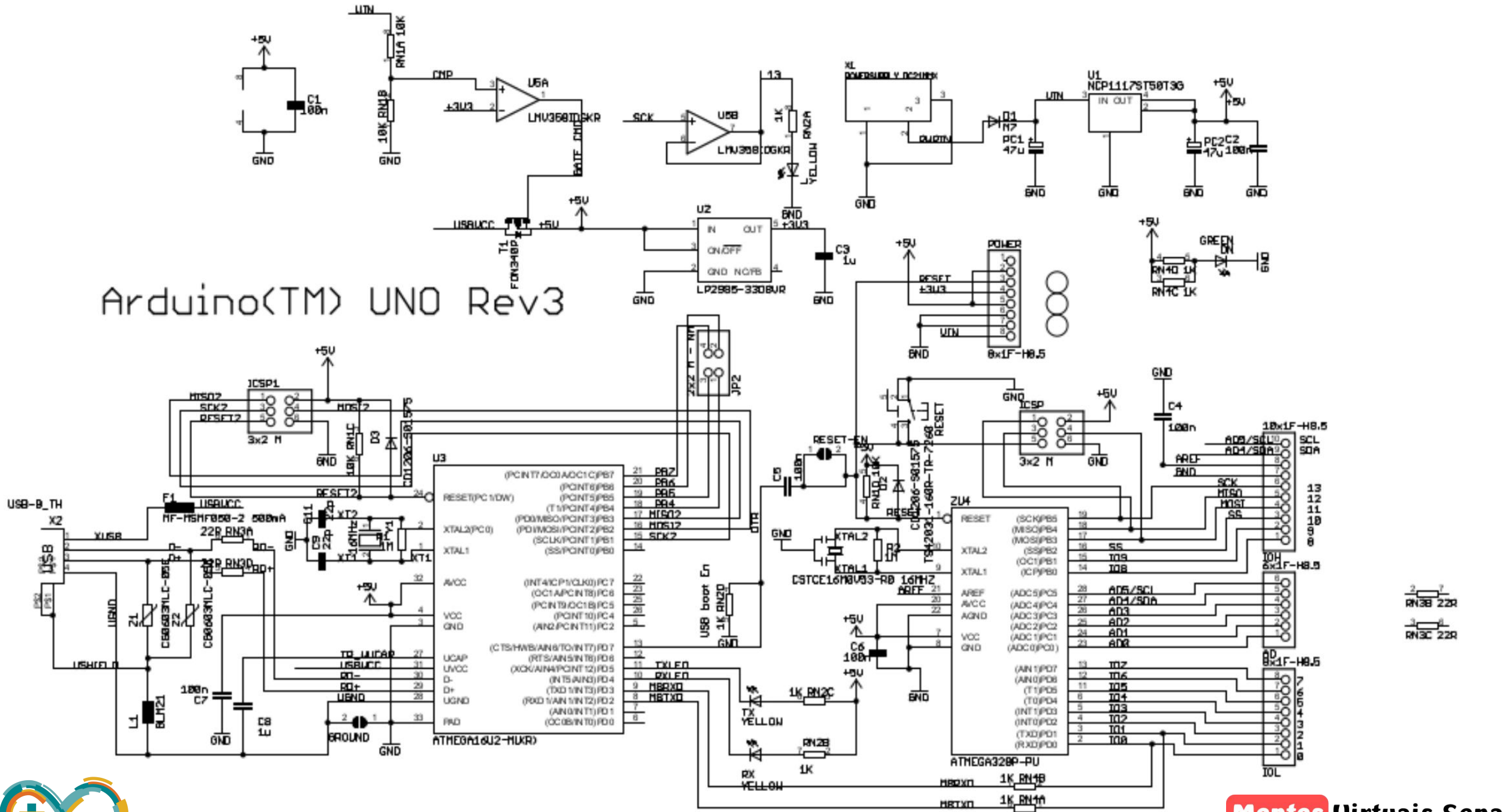


ENTRY LEVEL	<div>UNO</div> <div>LEONARDO</div> <div>101</div> <div>ESPLORA</div> <div>MICRO</div> <div>NANO</div> <div>MINI</div> <div>MKR2UNO ADAPTER</div> <div>STARTER KIT</div> <div>LCD SCREEN</div>
ENHANCED FEATURES	<div>MEGA</div> <div>ZERO</div> <div>DUE</div> <div>MEGA ADK</div> <div>MO</div> <div>MO PRO</div> <div>MKR ZERO</div> <div>MOTOR SHIELD</div> <div>USB HOST SHIELD</div> <div>PROTO SHIELD</div> <div>MKR PROTO SHIELD</div> <div>4 RELAYS SHIELD</div> <div>MEGA PROTO SHIELD</div> <div>MKR RELAY PROTO SHIELD</div> <div>ISP</div> <div>USB2SERIAL MICRO</div> <div>USB2SERIAL CONVERTER</div>
INTERNET OF THINGS	<div>YÚN</div> <div>ETHERNET</div> <div>TIAN</div> <div>INDUSTRIAL 101</div> <div>LEONARDO ETH</div> <div>MKR FOX 1200</div> <div>MKR WAN 1300</div> <div>MKR GSM 1400</div> <div>MKR1000</div> <div>YÚN MINI</div> <div>YÚN SHIELD</div> <div>WIRELESS SD SHIELD</div> <div>WIRELESS PROTO SHIELD</div> <div>ETHERNET SHIELD V2</div> <div>GSM SHIELD V2</div> <div>MKR IoT BUNDLE</div>
EDUCATION	<div>CTC 101</div>
WEARABLE	<div>GEMMA</div> <div>LILYPAD ARDUINO USB</div> <div>LILYPAD ARDUINO MAIN BOARD</div> <div>LILYPAD ARDUINO SIMPLE</div> <div>LILYPAD ARDUINO SIMPLE SNAP</div>
3D PRINTING	<div>MATERIA 101</div>

BOARDS
MODULES
SHIELDS
KITS
ACCESSORIES
COMING NEXT

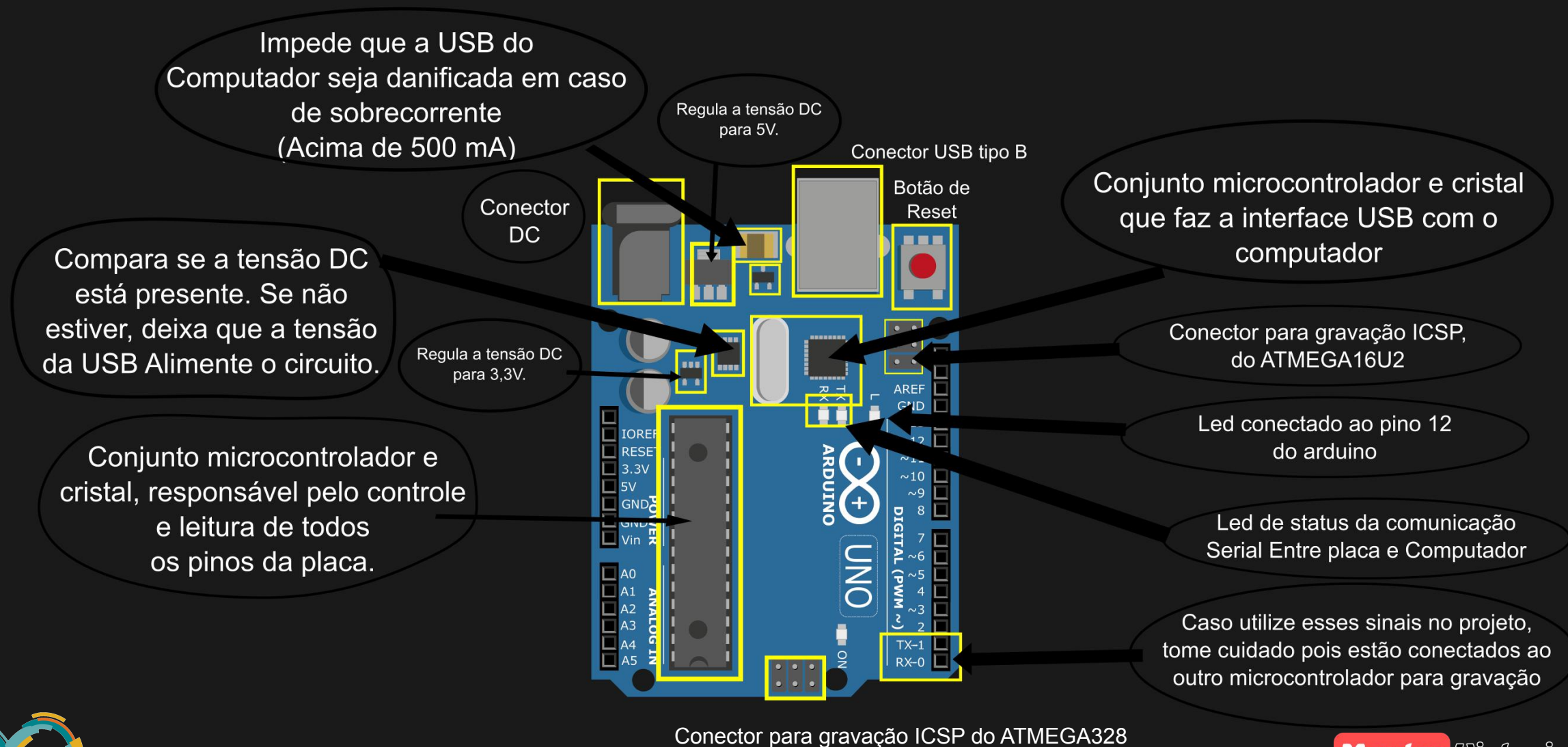
\$ - Quantos arduinos existem?

# Arduino(TM) UNO Rev3



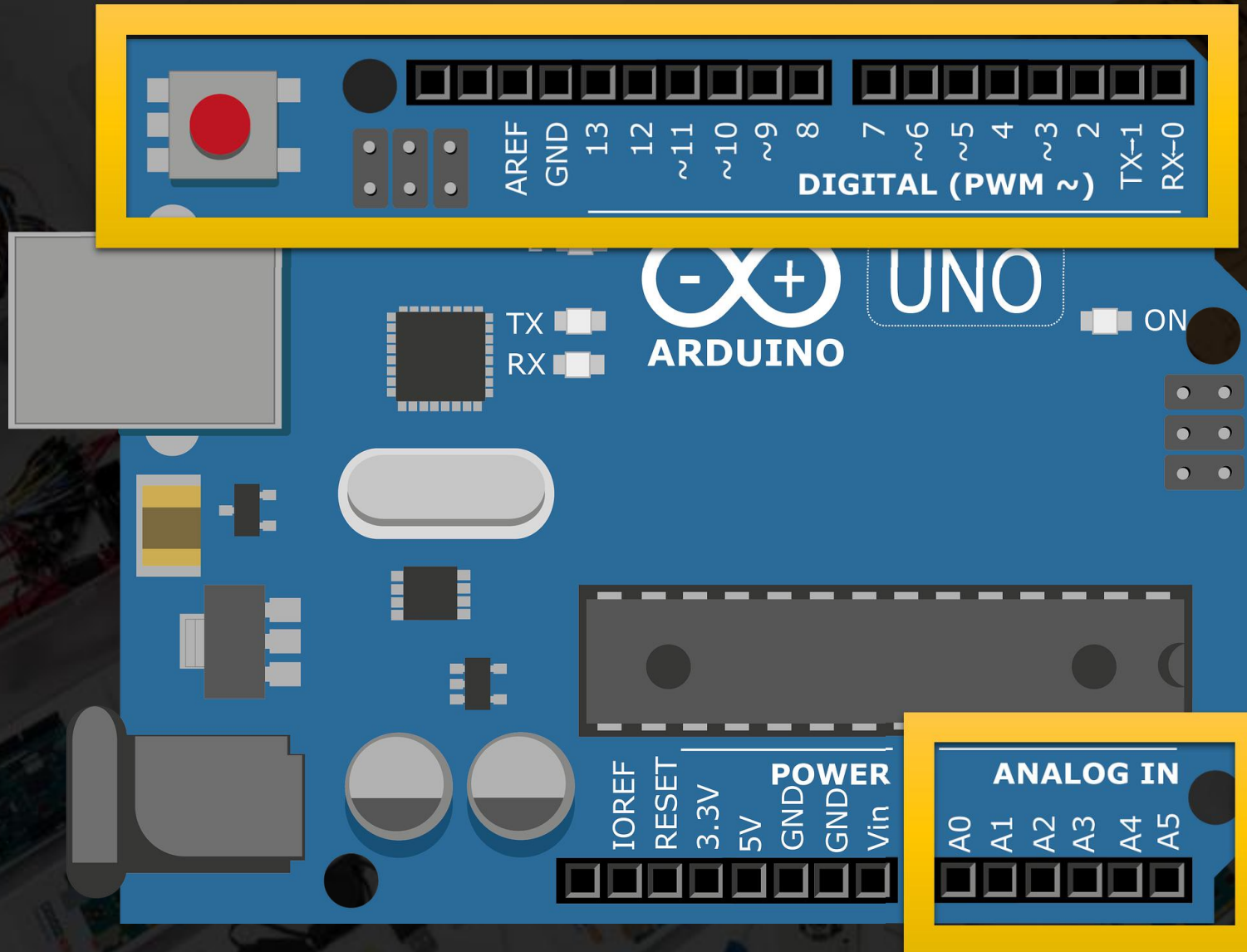


# \$ Componentes





# Esse é o Arduino UNO



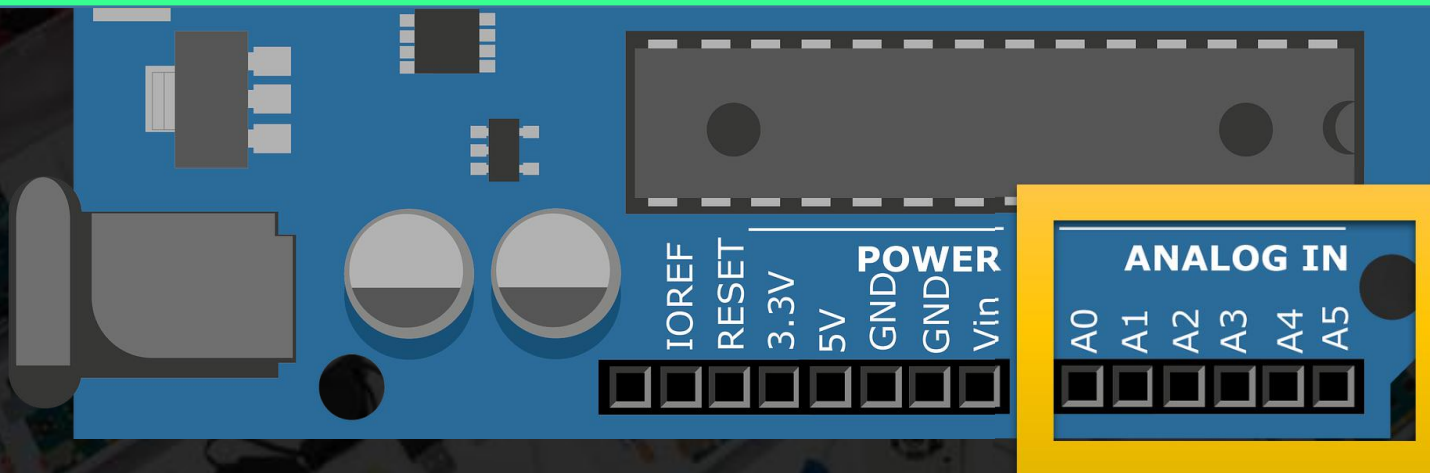




# Esse é o Arduino UNO:



## General Purpose Input/Output



**Mentes**

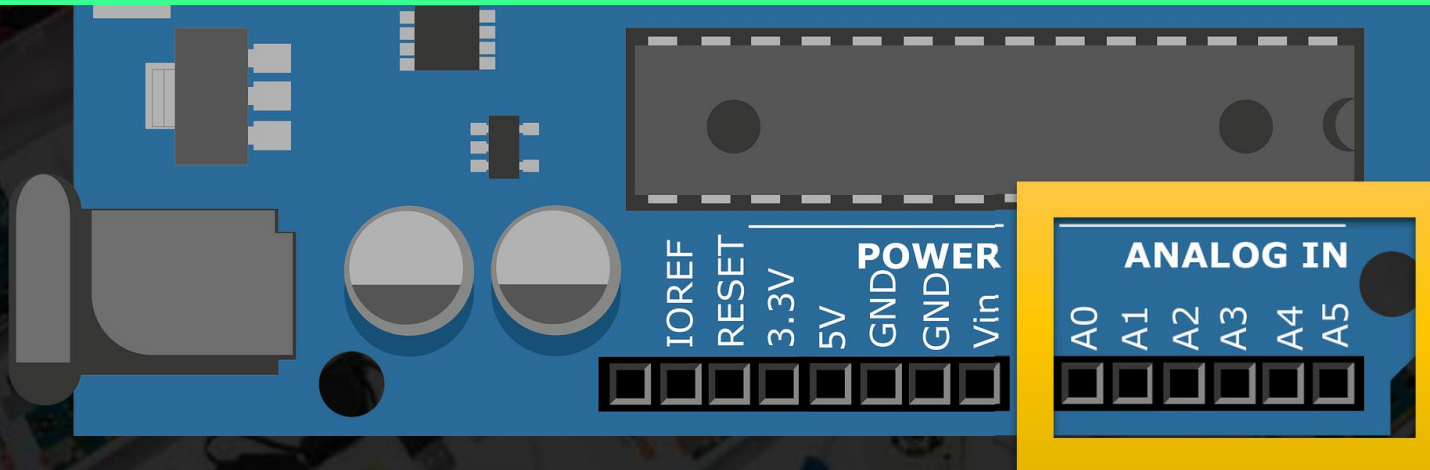
**Virtuais Sena**  
sua vida melhor em um click



# Esse é o Arduino UNO:



## G.P.I.O



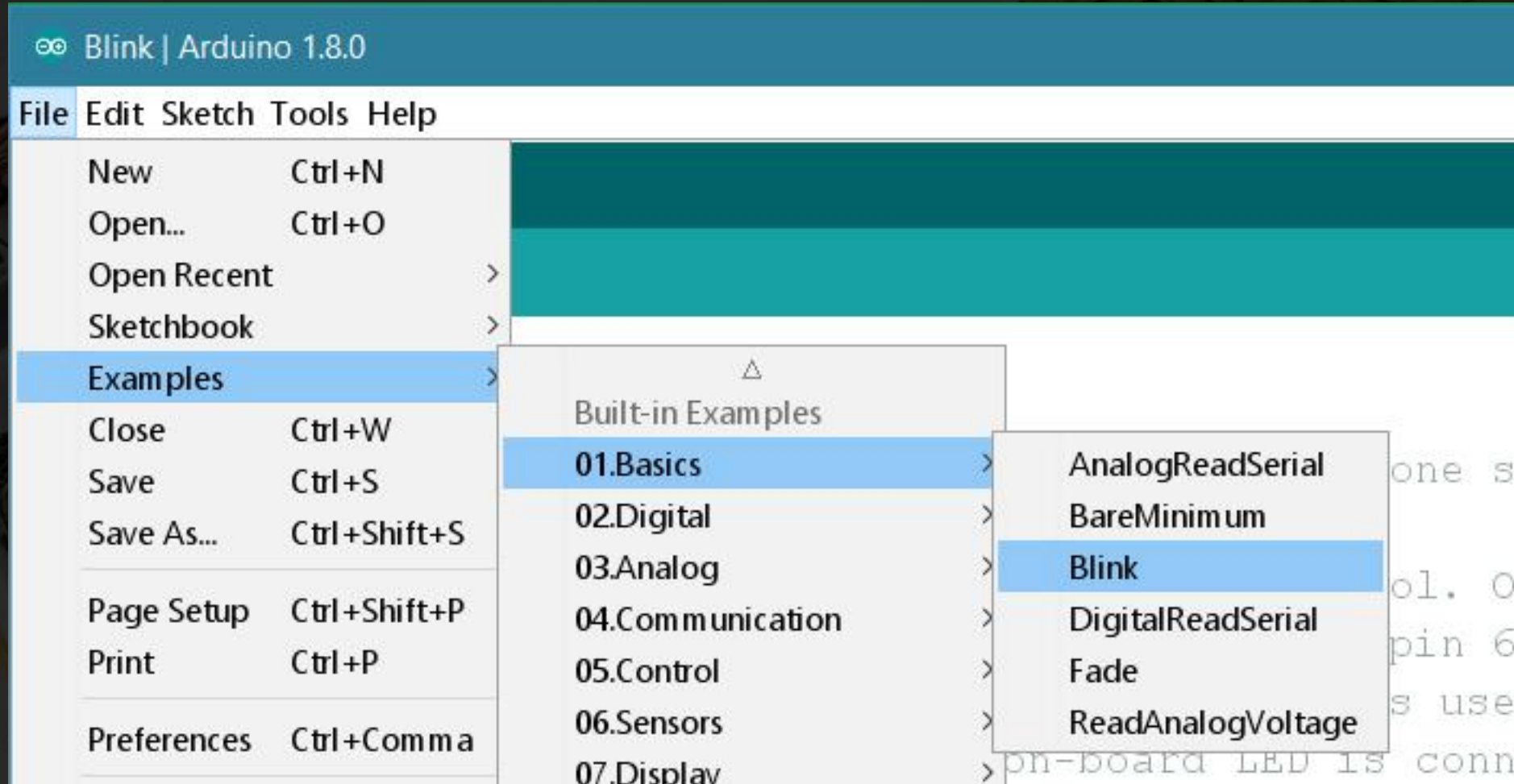


# \$ \_ Arduino IDE - Interface Inicial



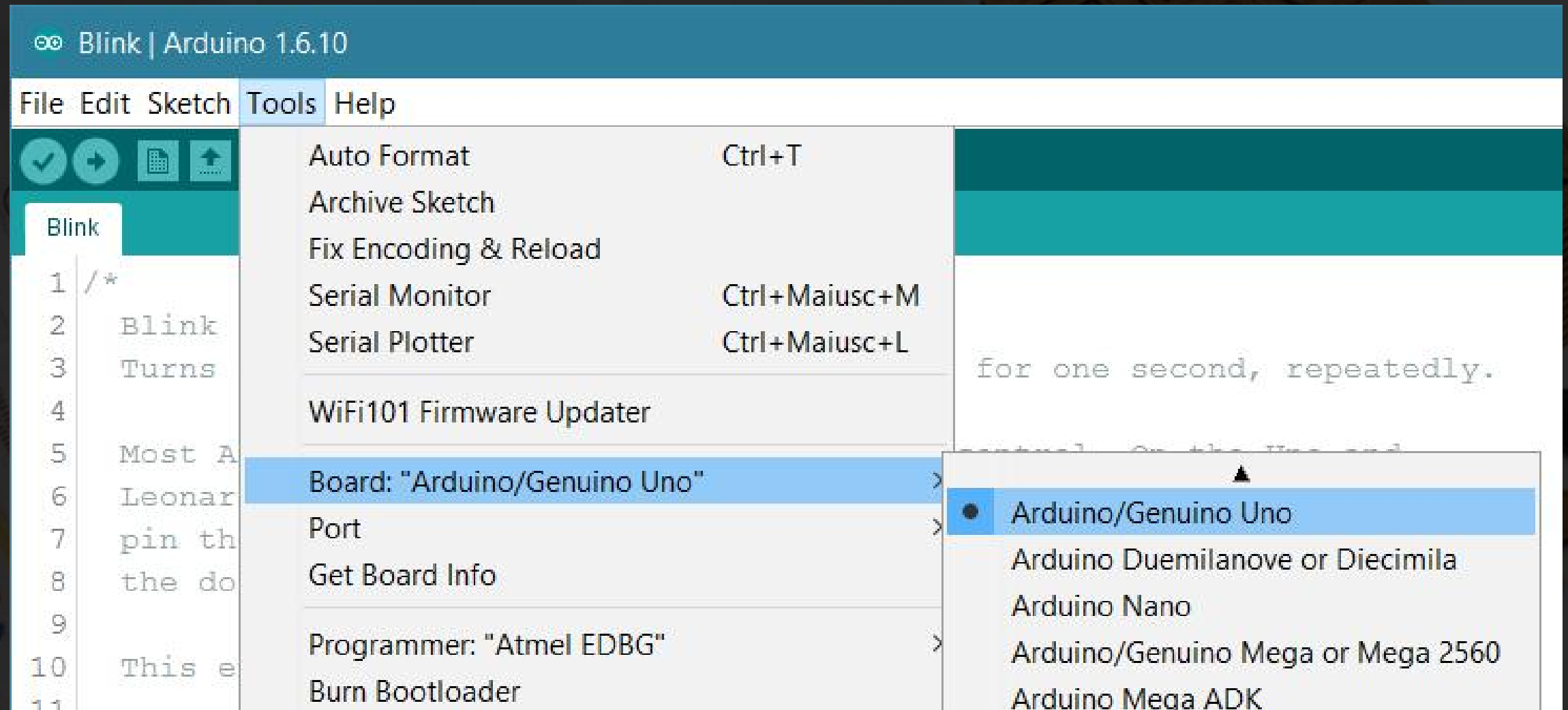


# Arduino IDE - Exemplos

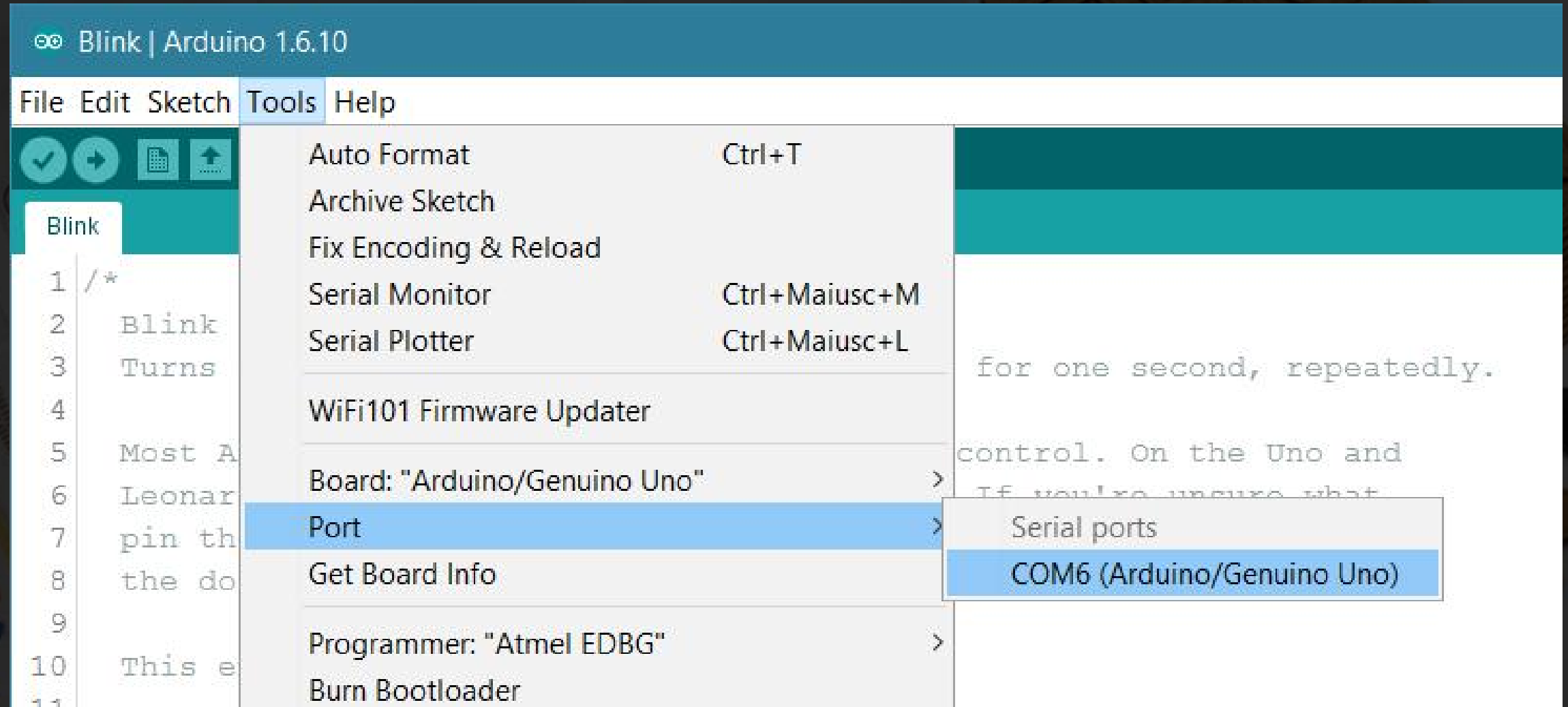




# \$ \_ Arduino IDE - Setando a Placa

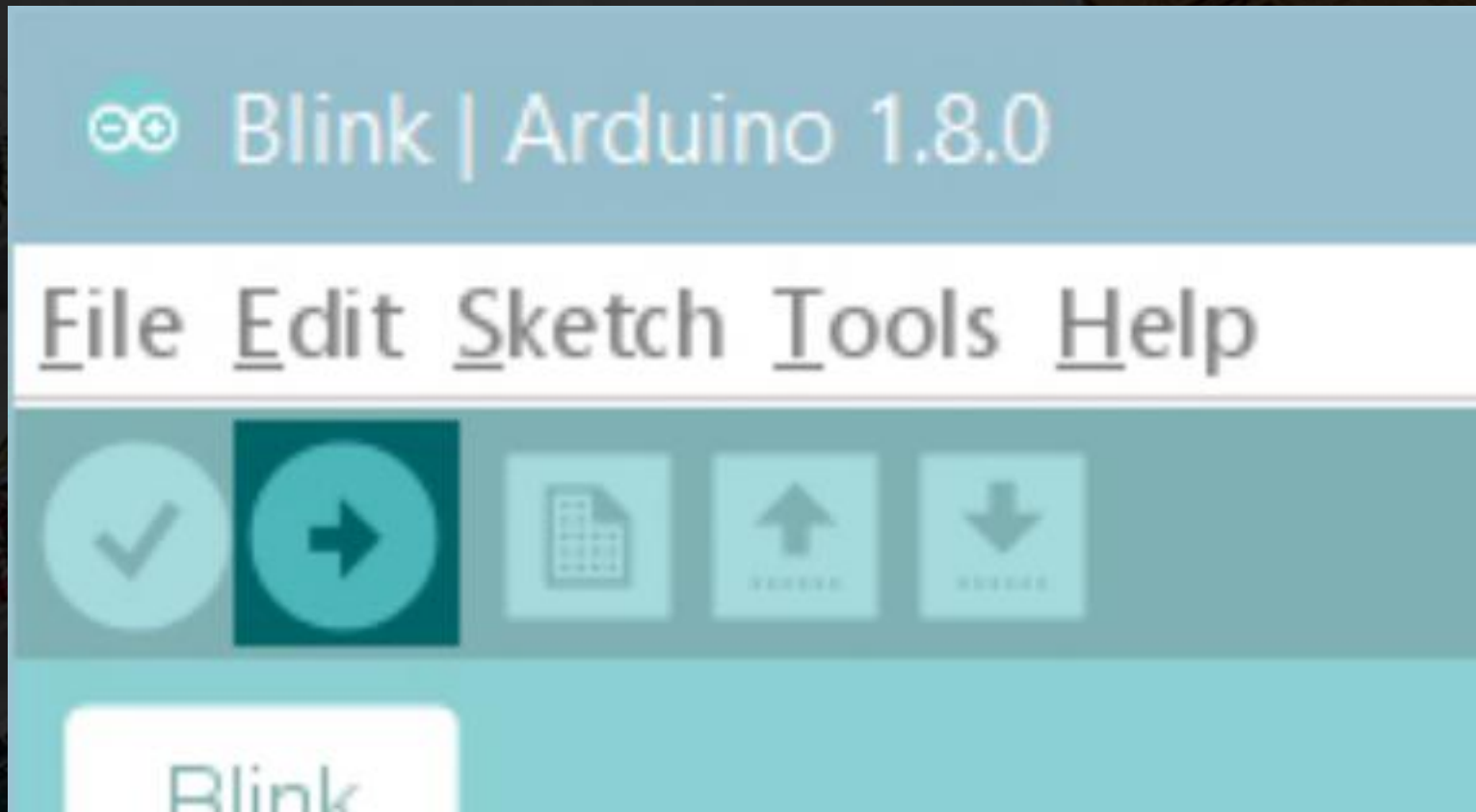


# \$ \_ Arduino IDE - Definindo a Porta





# \$ \_ Arduino IDE - Upload code





# Estrutura e padrões

```
sketch_oct24a | Arduino 1.8.3
File Edit Sketch Tools Help

sketch_oct24a

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly
8
9 }
```

Arduino/Genuino Uno on /dev/ttyACM1







# MÓDULO II




# \$ \_ Linguagem baseada na Wiring - C/C++

A Wiring é uma estrutura de programação de fonte aberta para microcontroladores. A Wiring permite a escrita de software multiplataforma para controlar dispositivos conectados a uma ampla gama de placas de microcontroladores para criar todos os tipos de **codificação criativa, objetos interativos, espaços ou experiências físicas.**



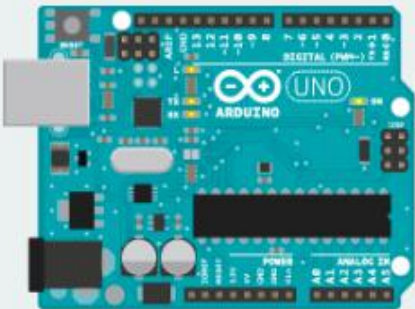


# \$ \_ Documentação principal



ARDUINO


HOME BUY SOFTWARE PRODUCTS LEARNING COMMUNITY SUPPORT



WHAT IS ARDUINO?


BUY AN ARDUINO

LEARN ARDUINO



BLOG

STUDENTS INVENT A LOW-COST ELECTRIC WHEELCHAIR KIT WITH ARDUINO



HAPPY HALLOWUINO!

SHARE YOUR SPOOKTACULAR PROJECTS USING #HALLOWUINO!



# \$ Criando o primeiro sketch: Blink

```
Blink $
1 #define LED 13
2
3 void setup()
4 {
5   pinMode(LED, OUTPUT);    // Setando o pino 13 como saída
6 }
7
8 void loop()
9 {
10  digitalWrite(LED, HIGH); // Ligando o LED (ALTO é o nível de tensão)
11  delay(1000);             // esperando um segundo
12  digitalWrite(LED, LOW);  // desligando o LED fazendo a tensão BAIXO
13  delay(1000);             // esperando um segundo
14 }
```

Done uploading.

Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.



# \$ \_Tipos primitivos

exemplo02

```
12 unsigned int | int | long
13 unsigned long | short | float
14 double
15 string - char array | String - object
16 array
17 */
18 //Exemplo de algumas declarações
19 int valor1 = 0;
20 float valor2 = 0;
21 bool valor6 = true;
22 byte valor3 = 0;
23 char valor4 = '0';
24 String valor5 = "0";
25
26
27 void setup ()
28 {
29
30 } //end setup
31
32 void loop ()
33 {
34
35 } //end loop
```





# \$ \_Sintaxe básica e Controle fluxo



The screenshot shows the Arduino IDE interface with a sketch named 'sketch\_oct24a'. The code is written in C++ and follows the standard Arduino template structure. The 'setup()' function is intended for initialization code that runs once, and the 'loop()' function is intended for the main code that runs repeatedly. The background of the slide features a collage of various electronic components and tools, including breadboards, wires, and microcontrollers.

```
sketch_oct24a | Arduino 1.8.3
File Edit Sketch Tools Help

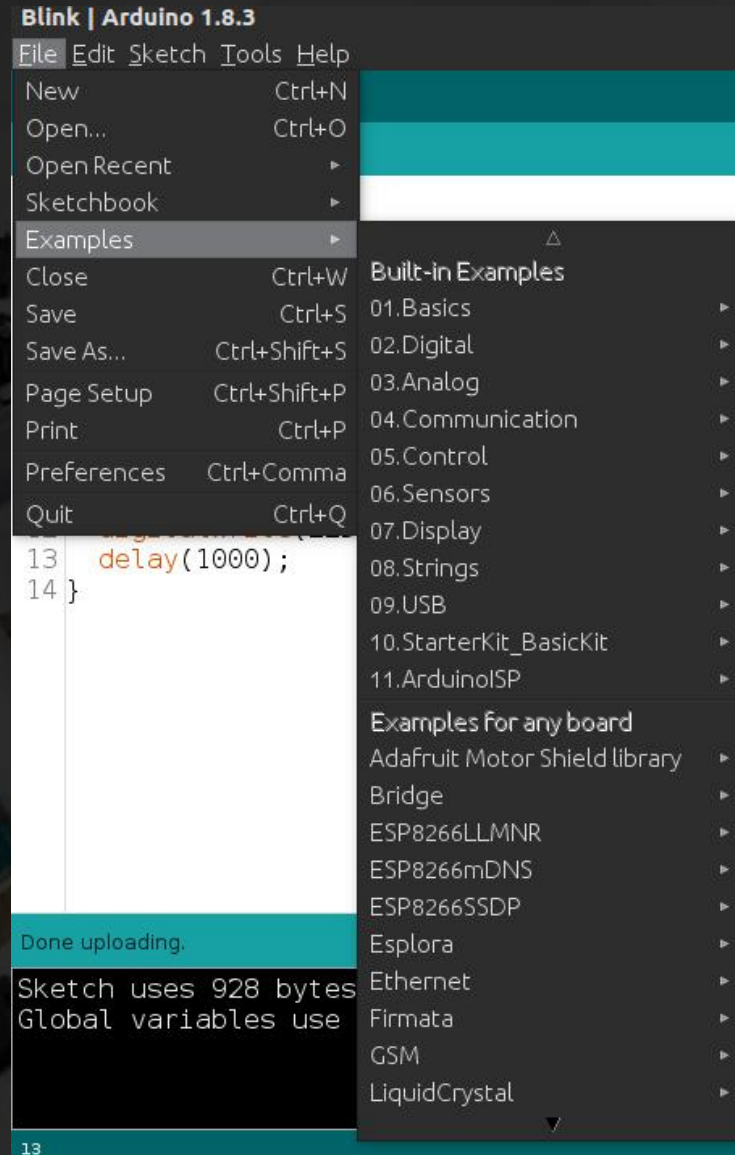
sketch_oct24a

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino/Genuino Uno on /dev/ttyACM1



# \$ \_ Algumas das principais bibliotecas



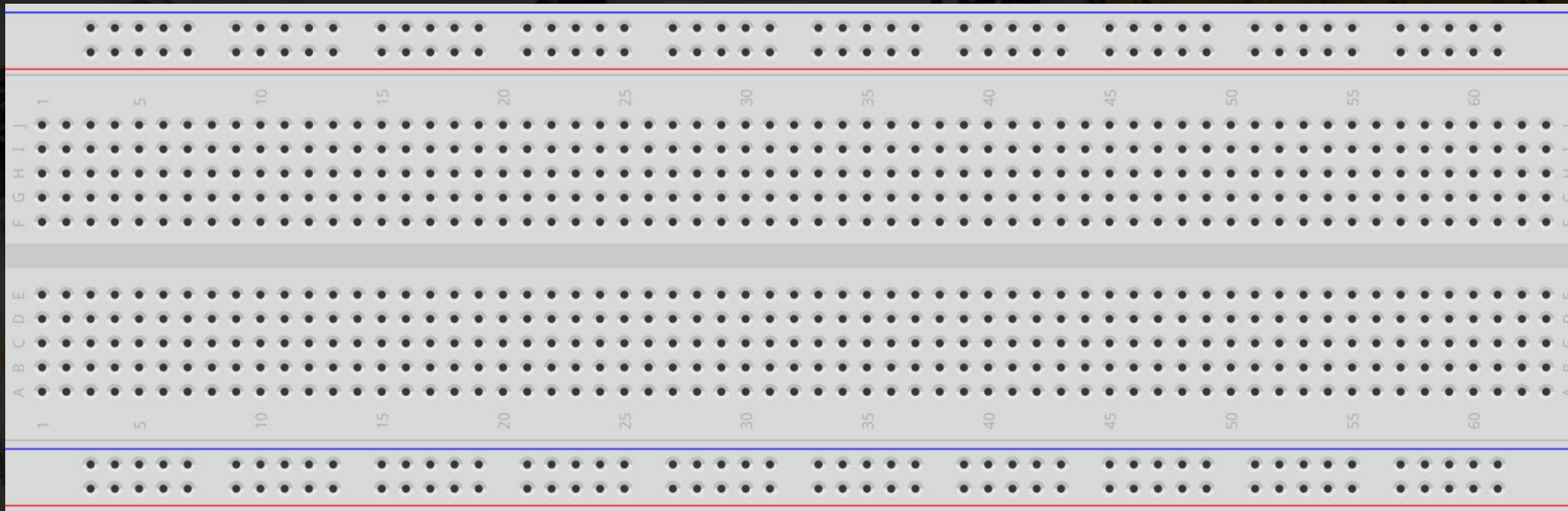


# MÓDULO III

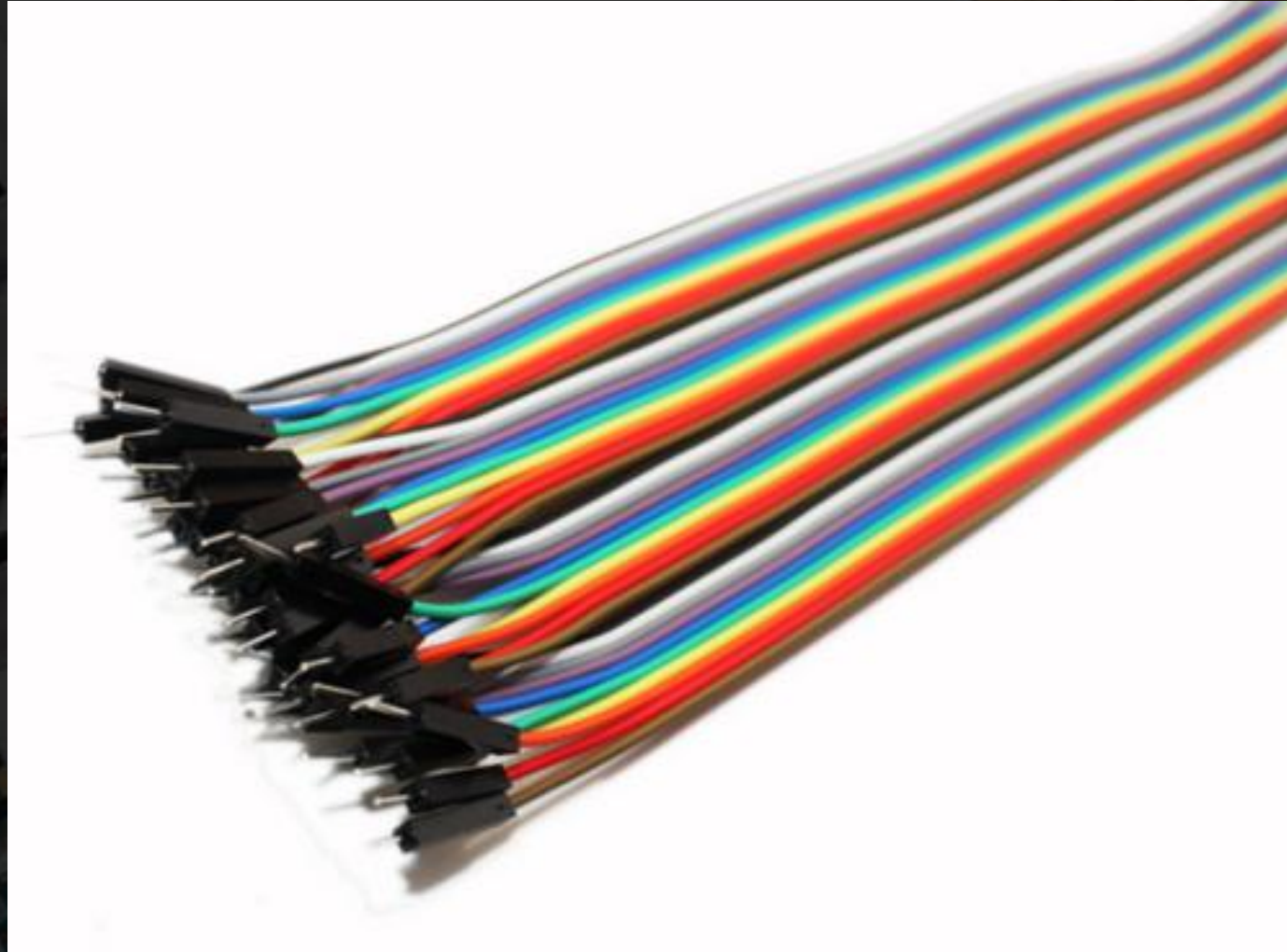




# \$ \_ Introdução a Protoboard

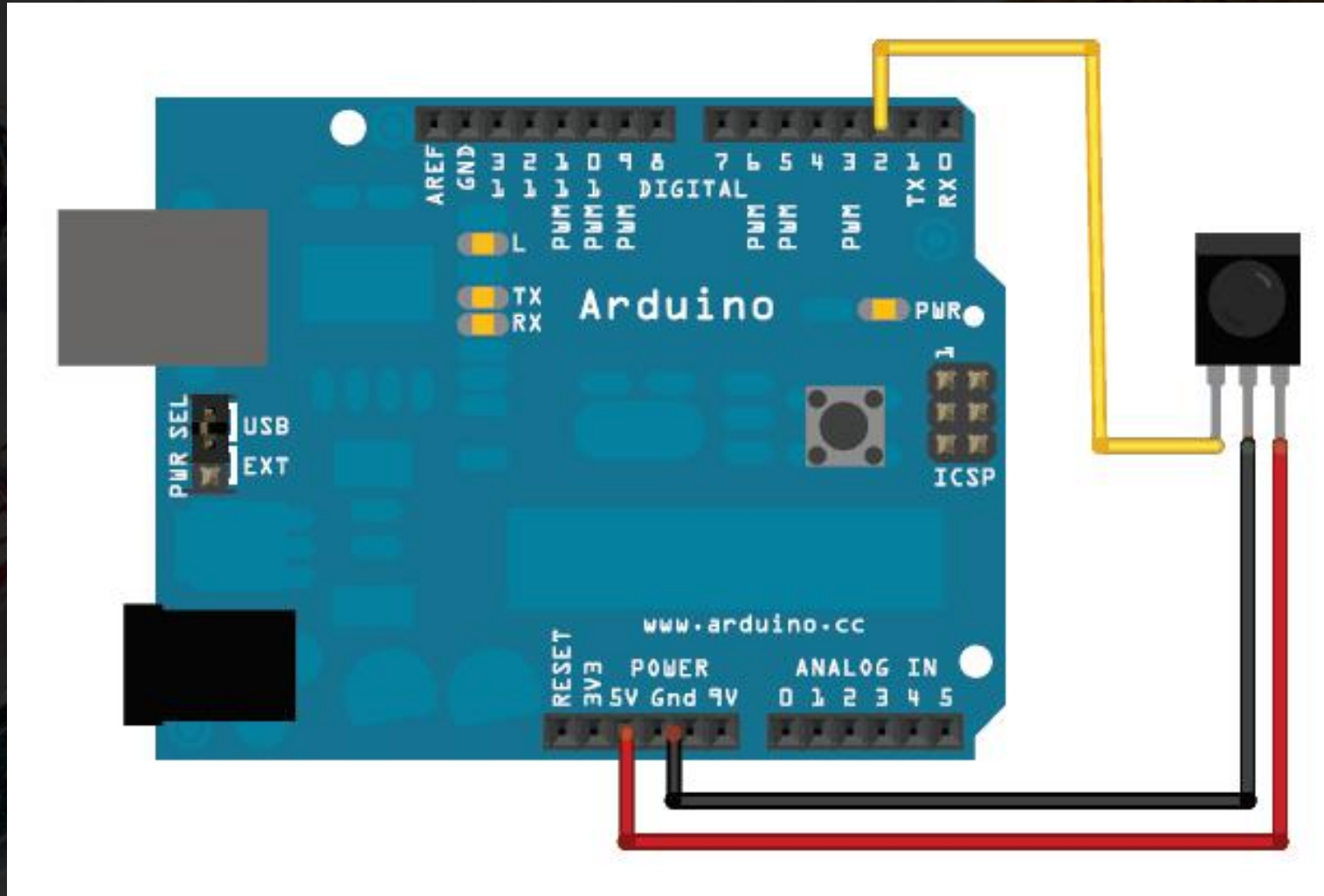


# \$\_ Mantendo os padrões dos Fios



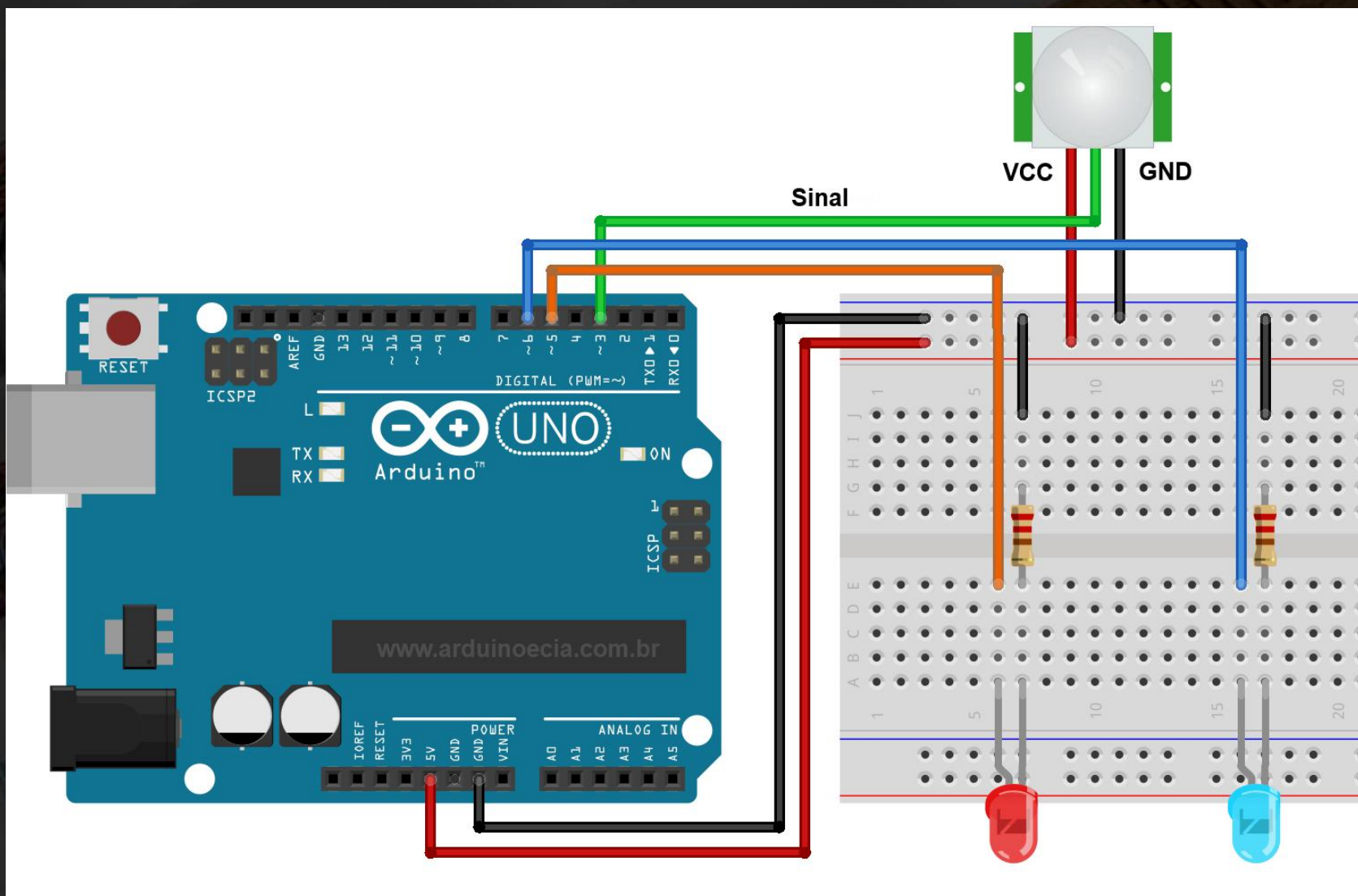


# \$\_ Mantendo os padrões dos Fios





# \$\_ Mantendo os padrões dos Fios



# \$ \_ Encontrando valor da resistência com código de cores

Os resistores possuem um código de cores que nos permite identificar qual sua resistência. Para isso, mapeamos as cores das diversas faixas do resistor e utilizamos a seguinte fórmula:

VALOR NOMINAL										
COR	PRETO	MARROM	VERMELHO	LARANJA	AMARELO	VERDE	AZUL	VIOLETA	CINZA	BRANCO
VALOR	0	1	2	3	4	5	6	7	8	9

VALOR DA TOLERÂNCIA			
COR	DOURADO	PRATA	SEM COR
VALOR	±5%	±10%	±20%

$$R = (10a + b) \cdot 10^c \pm t,$$

onde a, b e c são as primeiras faixas e t a última faixa (geralmente prata ou dourada), que representa a tolerância.



# \$ \_ Encontrando valor da resistência com código de cores

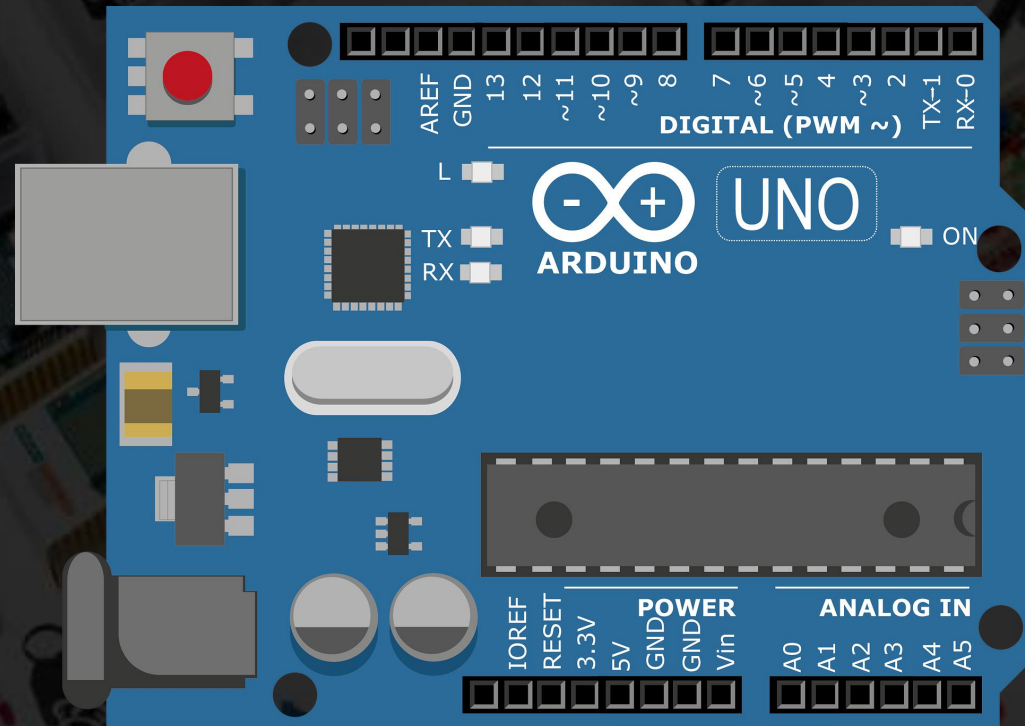
Color	Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Black	Preto	0	0	0	x 1 $\Omega$	
Brown	Marrom	1	1	1	x 10 $\Omega$	+/- 1%
Red	Vermelho	2	2	2	x 100 $\Omega$	+/- 2%
Orange	Laranja	3	3	3	x 1K $\Omega$	
Yellow	Amarelo	4	4	4	x 10K $\Omega$	
Green	Verde	5	5	5	x 100K $\Omega$	+/- .5%
Blue	Azul	6	6	6	x 1M $\Omega$	+/- .25%
Violet	Violeta	7	7	7	x 10M $\Omega$	+/- .1%
Gray	Cinza	8	8	8		+/- .05%
White	Branco	9	9	9		
Gold	Dourado				x .1 $\Omega$	+/- 5%
Silver	Prateado				x .01 $\Omega$	+/- 10%





# \$\_ Saídas Digitais e Analógicas

O Arduino possui dois tipos de portas de entrada: analógicas e digitais. Além disso, as portas digitais também servem como portas de saída, funcionando com dois tipos básicos de saída: saída digital comum e saída PWM o PWM pode ser utilizado para simular uma saída analógica, dentre outras coisas.



# \$ \_ Saídas Digitais e Analógicas

Utilizamos as portas digitais quando precisamos trabalhar com valores bem definidos de tensão. Apesar de nem sempre ser verdade, geralmente trabalhamos com valores digitais binários, ou seja, projetamos sistemas que utilizam apenas dois valores bem definidos de tensão. Existem sistemas ternários, quaternários, mas focaremos no binário, já que é esse o utilizado pelo Arduino. Como o sistema é binário, temos que ter apenas duas tensões. São elas: 0V e 5V. Dessa forma, as portas digitais do Arduino podem trabalhar apenas com essas duas tensões e o software que desenvolvermos poderá requisitar ao microcontrolador do Arduino que:

- Coloque uma determinada porta em 0V;
- Coloque uma determinada porta em 5V;
- Leia o valor de uma determinada porta (terá 0V ou 5V como resposta).





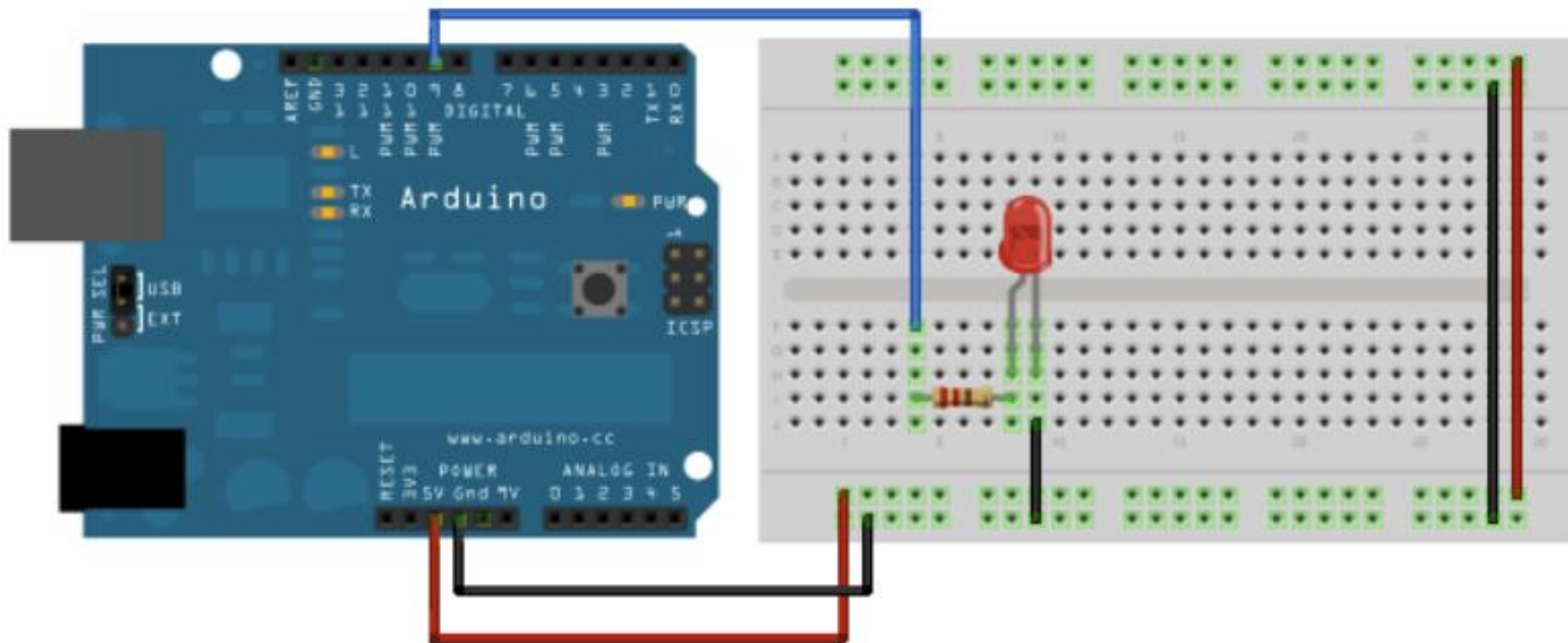
# \$\_ Saídas Digitais e Analógicas

Além das portas digitais o Arduino possui as portas analógicas. Ao contrário das portas digitais, as portas analógicas são apenas de entrada e nelas podemos ter como entrada infinitos valores de tensão (delimitados na faixa de 0V a 5V). Como os conversores analógico-digitais (ADC – analog-digital converter, do Inglês) do Arduino possuem 10 bits de precisão, a precisão das medições de tensão no Arduino é de por volta de 0,005V ou 5mV.

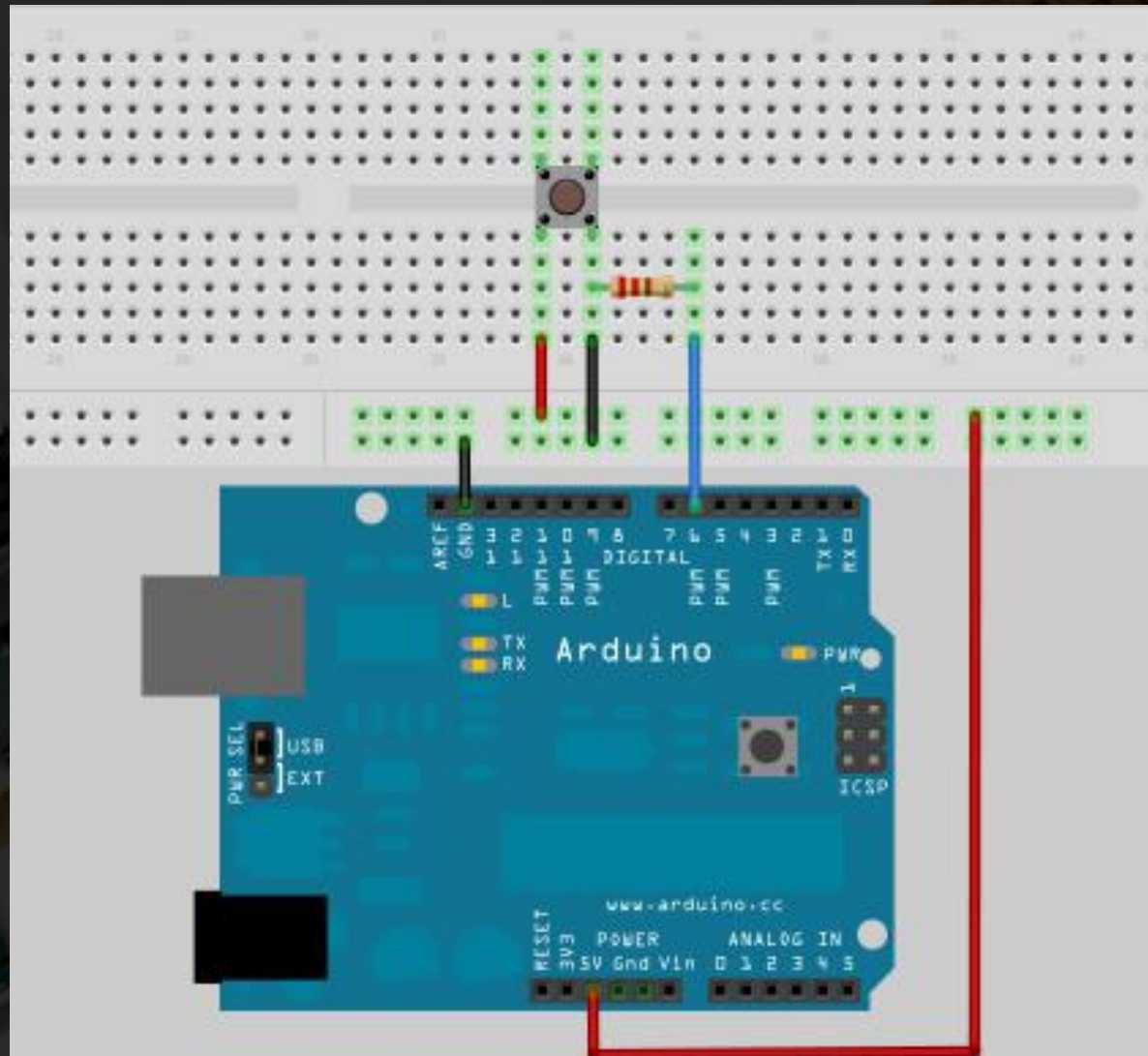




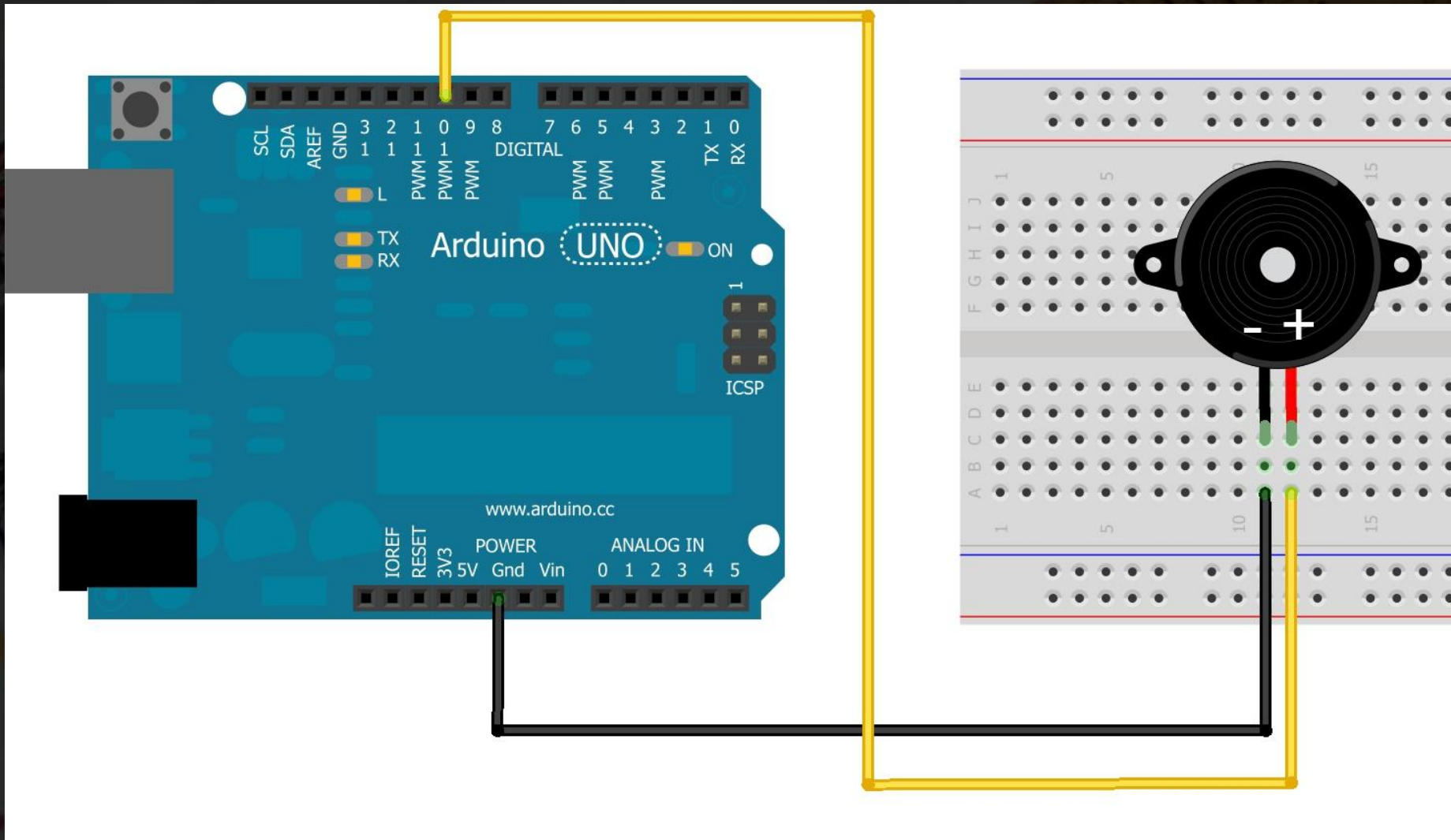
# \$ \_ Como acender e apagar um led



## \$ \_Lendo estado de um botão



# \$\_ Fazendo barulho com o Buzzer



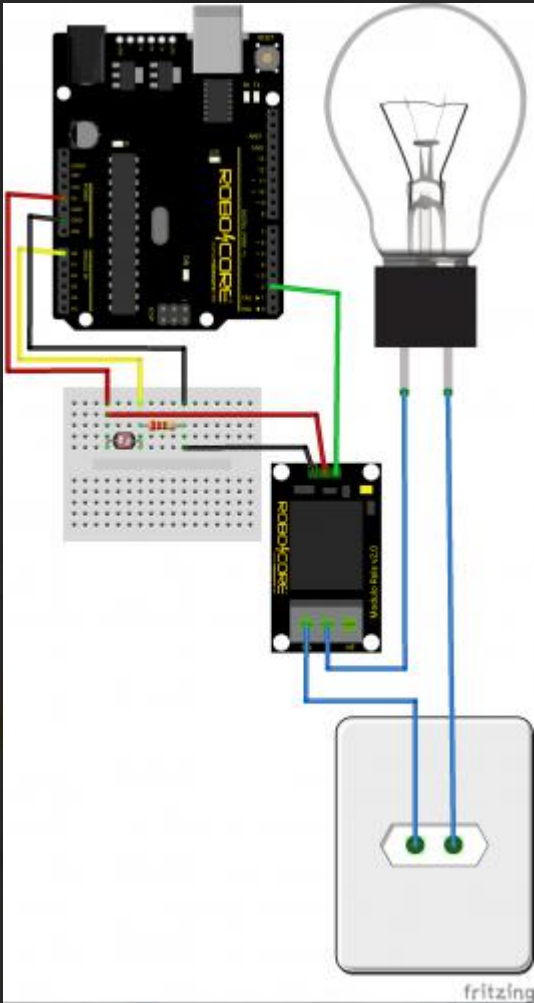




# MÓDULO IV



# \$\_ Porta Serial, sensore e shields, Relé

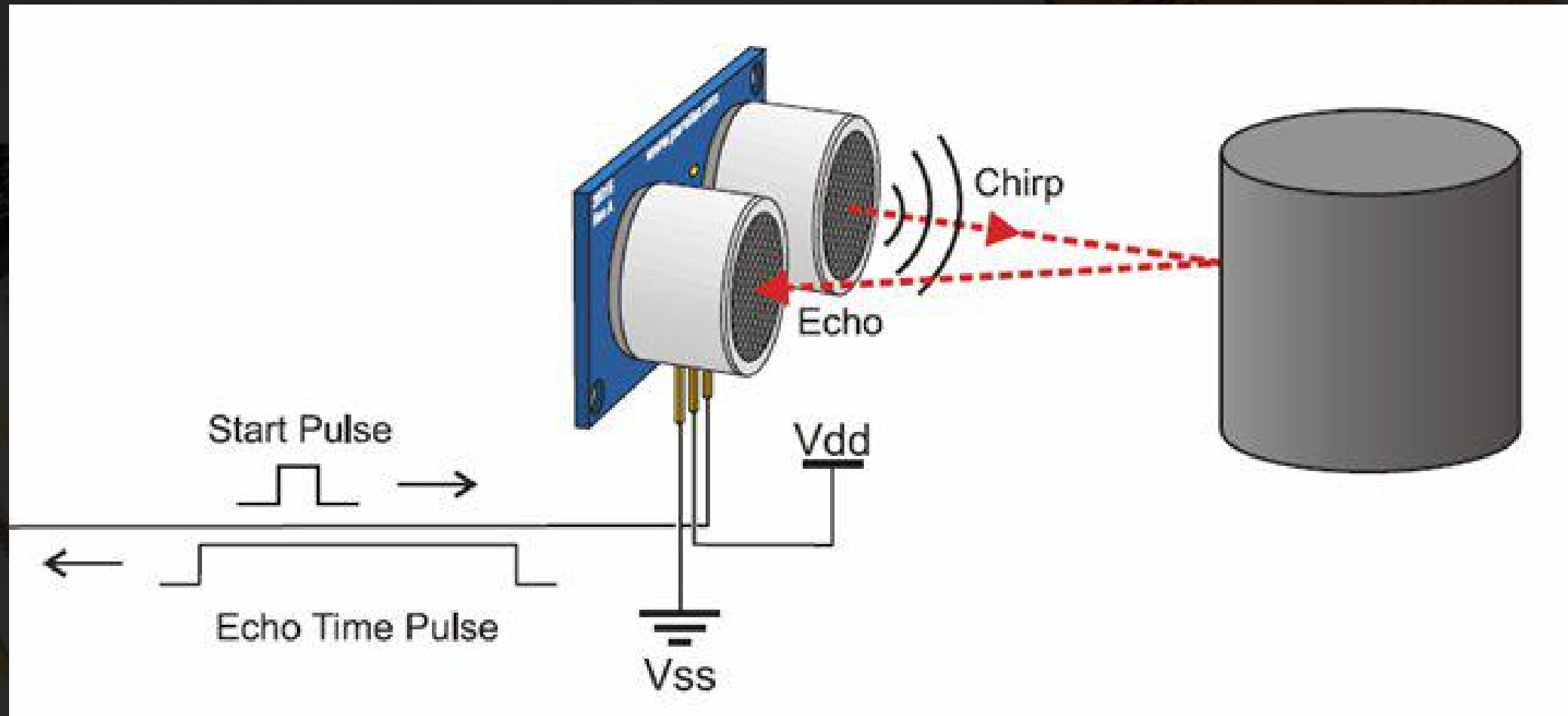


Primeiramente precisamos saber o que é um LDR. LDR, do inglês Light Dependent Resistor ou Resistor Dependente de Luz, é um sensor que varia sua resistência conforme a intensidade de luz. Quanto mais luz, menor sua resistência. Porém o LDR não fornece uma leitura da quantidade de LUX em um ambiente, somente consegue fornecer a informação se está claro ou escuro.

Como o LDR varia sua resistência de acordo com a intensidade da luz, iremos utilizar uma porta analógica do Arduino para ler esta variação. Para fazermos isso, montaremos um divisor de tensão com um resistor de 10kΩ.

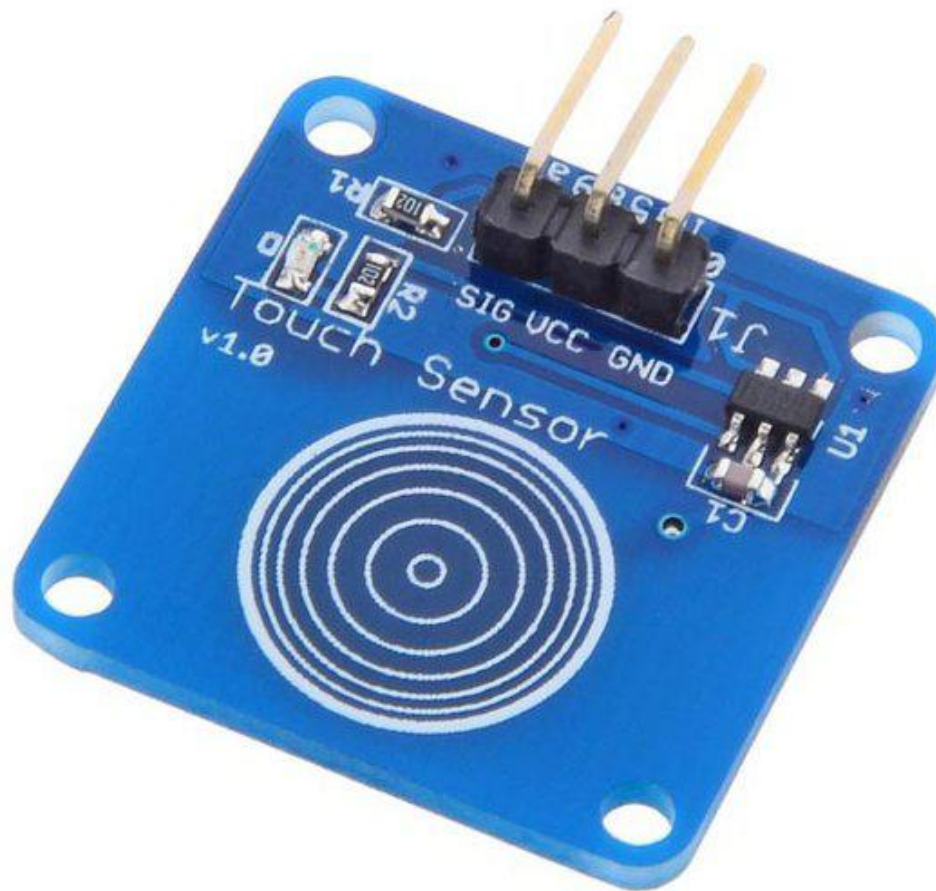


# \$\_ Sensor Ultrasonic





# \$\_ Sensor Touch



# \$\_ Downloads do Códigos do Mini Curso



<https://www.github.com/WalderlanSena/miniCursoArduino>






# "Projeto no GitHub";

← → ↻ [GitHub, Inc. \[US\]](#) | <https://github.com/WalderlanSena/miniCursoArduino> ☆ f? 📄 ⋮

📖 README.md

## Mini Curso de Arduino

*"A ciência é um trabalho de artesanato intelectual." W. Mills*



Códigos desenvolvidos no Mini Curso de Arduino

License [GPL v3](#)

### 📖 Ementa do Mini Curso


📖 MÓDULO I	📖 MÓDULO II	📖 MÓDULO III	📖 MÓDULO IV
Apresentações gerais/Introdução	Wiring: Linguagem Baseada em C++	Introdução a Protoboard	Serial/USB com o PC





# \$ "Projeto no GitHub";

GitHub, Inc. [US] | <https://github.com/WalderlanSena>




Search GitHub


Pull requests


Issues


Marketplace

Explore












Walderlan Sena

WalderlanSena

Programmer, IT Technician, Ac  
System Analysis and Development,  
Ac  
AdvancedPentest,Escritor,Palestrant  
e, Entusiasta IoT & SI, CEO na  
empresa @mentesvirtuaissena

 Mentes Virtuais Sena

 Fortaleza-CE

 [senawalderlan@gmail.com](mailto:senawalderlan@gmail.com)

Overview

Repositories 38

Stars 40


Followers 10

Following 15

Pinned repositories

Customize your pinned repositories


worksena

 O Micro Framework PHP, para projetos especializados em performance de alto desenvolvimento na web.

PHP


★ 1

phplot

 Scripts desenvolvidos em Php para utilização em projetos lot especialmente com Arduino


PHP

dnsdragon


 Pesquise por diretórios e arquivos ocultos em aplicações web

Shell

★ 1


 1

md5crack




 Perform md5 hash break with offline shell script

Shell


★ 1

 2

hashblade

 Encrypted texts and perform Brute Force. Through Hash MD5, SHA1, SHA256, SHA512.. In all operating systems Gnu/Linux , Windows, MacOS 

miniCursoArduino

 Códigos desenvolvidos no Mini Curso de Arduino

 Mentes Virtuais Sena  
sua vida melhor em um click



\$ \_ "the end";

**Mentes** Virtuais Sena  
sua vida melhor em um click

 /walderlan.sena

 @walderlansena

 MentesVrtuaisSena

 [github.com/WalderlanSena](https://github.com/WalderlanSena)

< *Obrigado* />

<https://www.MentesVirtuaisSena.com>    [contato@mentesvirtuaissena.com](mailto:contato@mentesvirtuaissena.com)

<"As pessoas que são loucas o suficiente para achar que podem mudar o mundo, são aquela que realmente o mudam"/>

