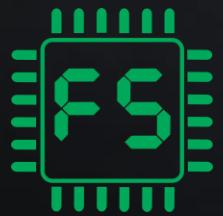


WORKSHOP “CONTROLE SUA CASA REMOTAMENTE ATRAVÉS DA INTERNET”

Por Fábio Souza



Fábio Souza

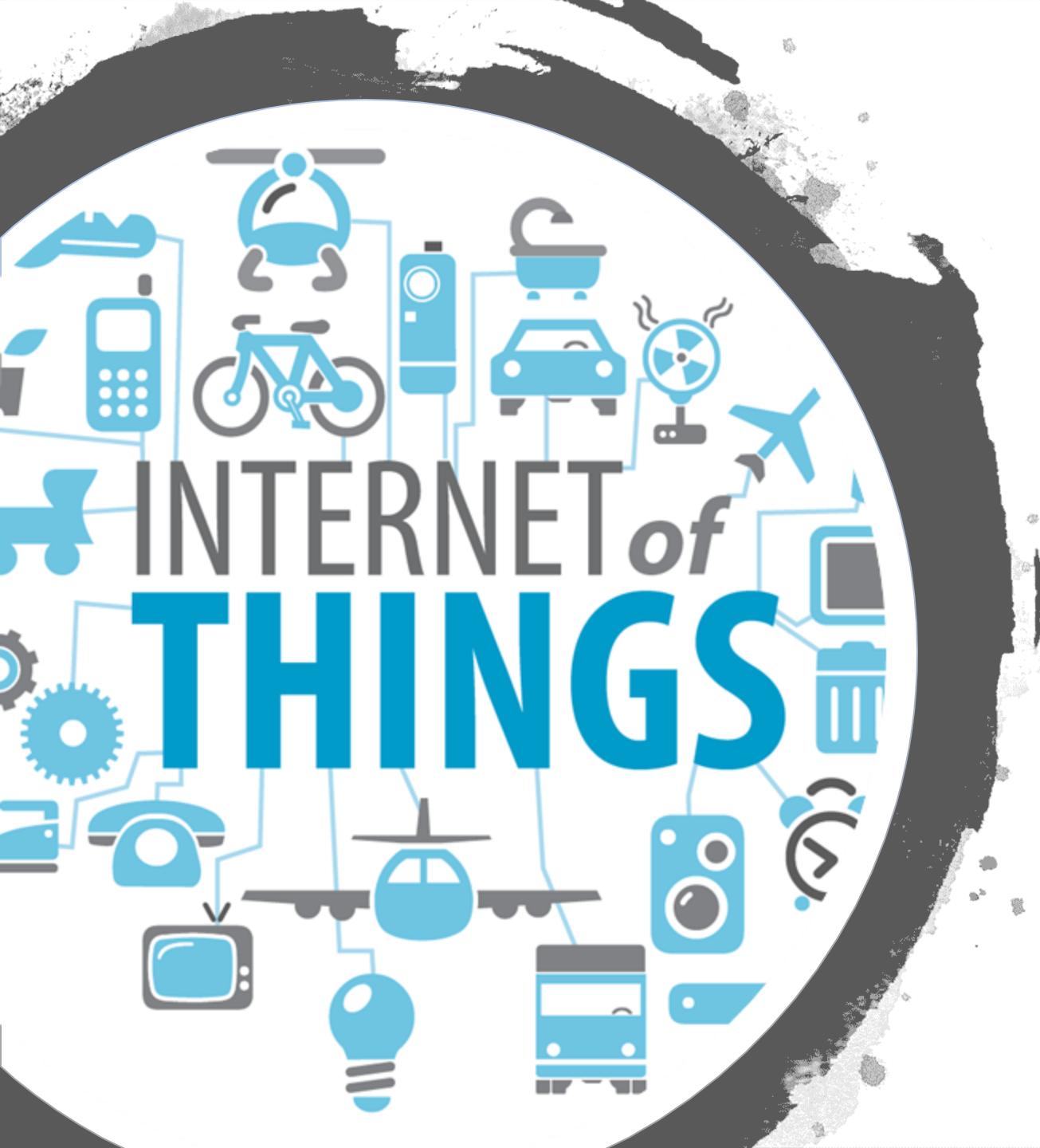
Engenheiro com experiência no desenvolvimento de projetos eletrônicos embarcados. Hoje é diretor de operações do portal Embarcados, onde trabalha para levar conteúdos de eletrônica, sistemas embarcados e IoT para o Brasil. Também atua no ensino eletrônica e programação pelo Brasil. É entusiasta do movimento maker, da cultura DIY e do compartilhamento de conhecimento, publica diversos artigos sobre eletrônica e projetos open hardware, como o projeto Franzininho. Participou da residência hacker 2018 no Red bull Basement. Quando não está ministrando palestras, cursos ou workshops, dedica seu tempo “escovando bits” ou projetando placas eletrônicas.



Agenda

- IoT
- Eletronica Básica
- ESP8266
- IDE Arduino
- ESP8266 web server
- MQTT
- Aplicação com MQTT





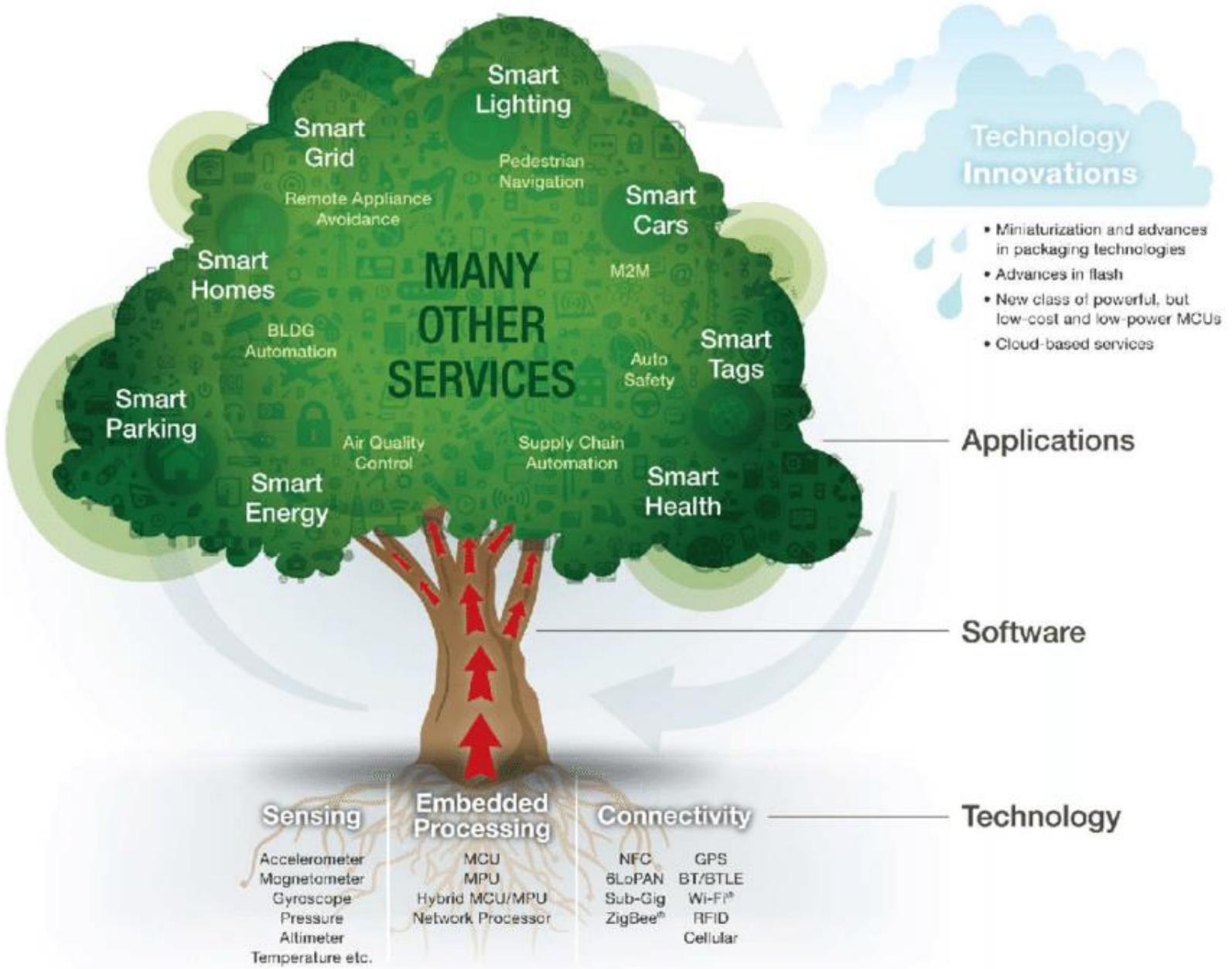
Internet das coisas

É uma rede de objetos físicos que possuem tecnologia embarcada para comunicar, captar sinais e interagir consigo mesmos ou com o ambiente externo.

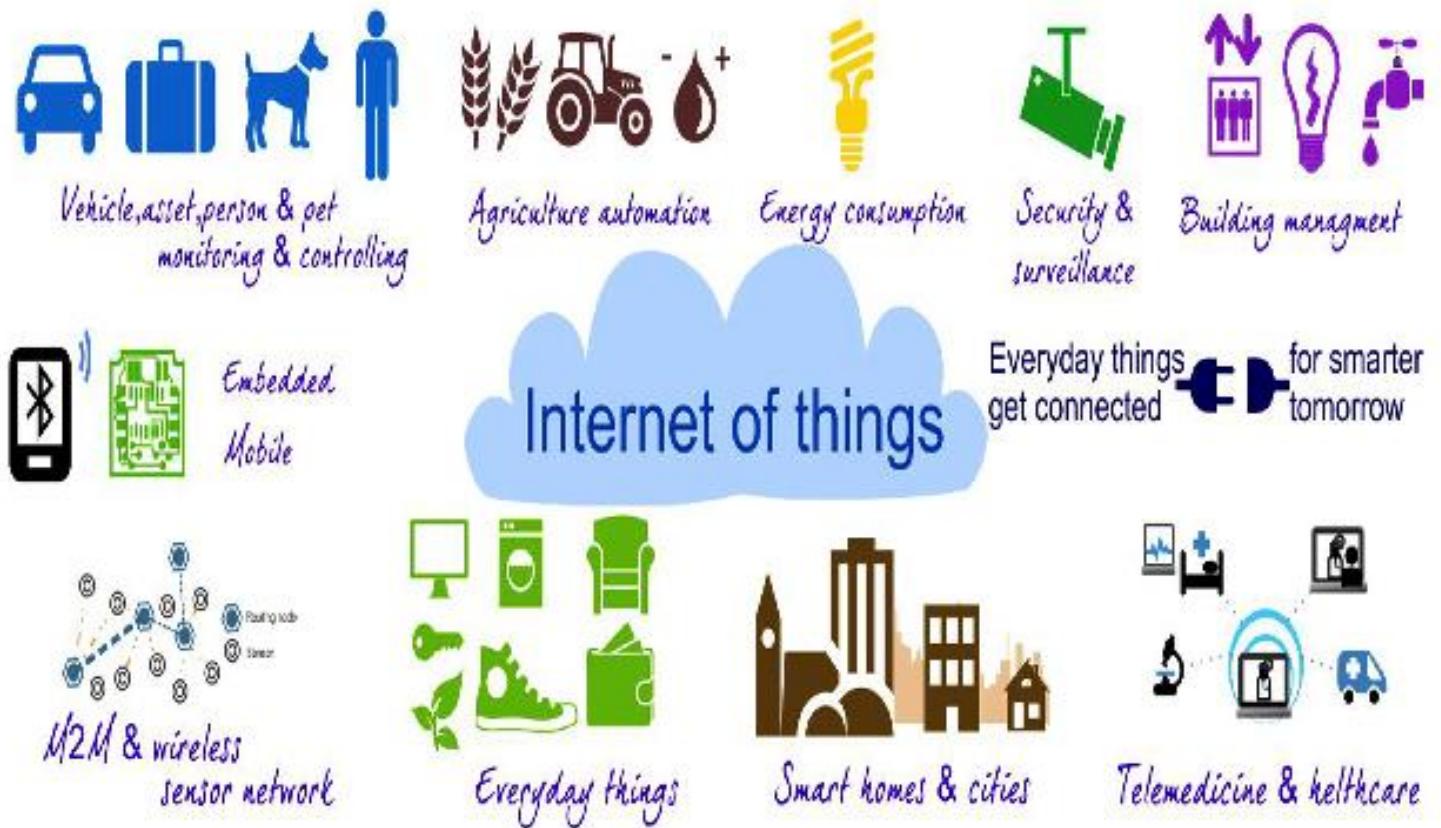
Gartner

IoT

Estrutura de IoT



Aplicações

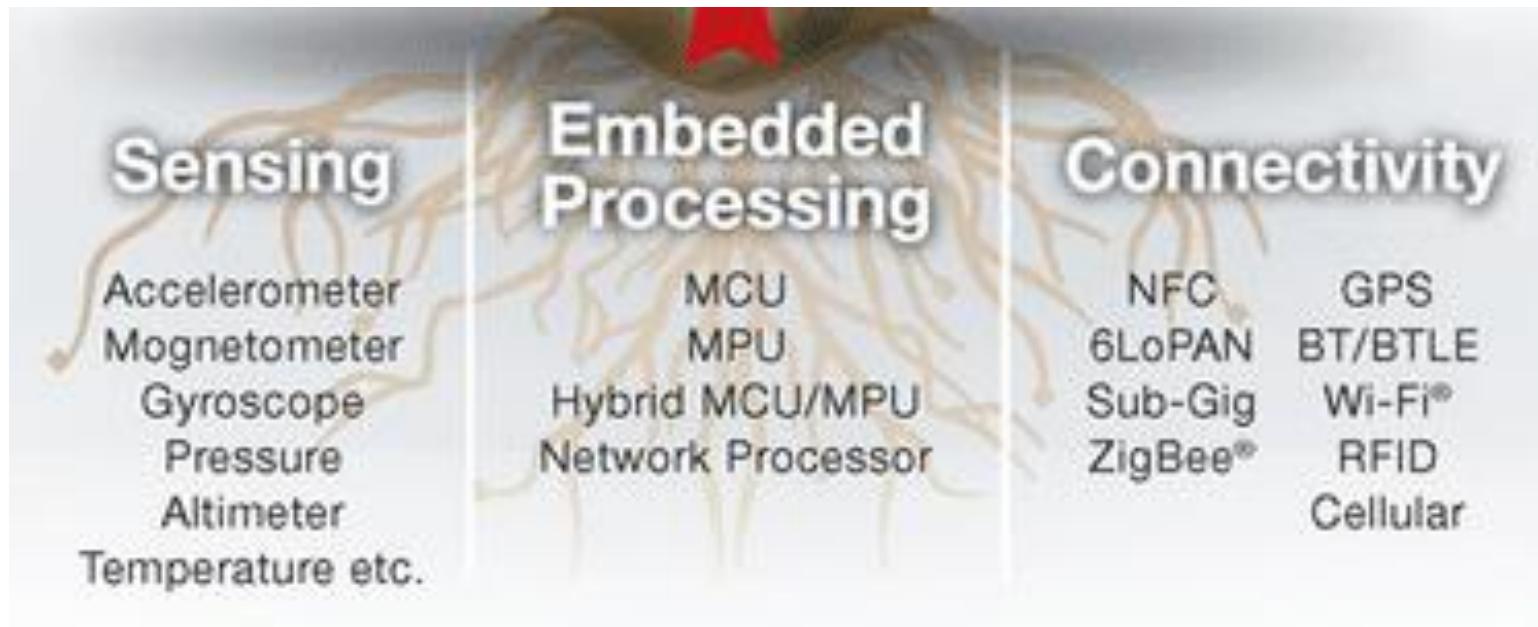


IoT em
funcionamento



Hardware

Sensores + MCU/MPU + Conectividade



Empresas de Hardware

HARDWARE

PROCESSORS / CHIPS

SENSORS

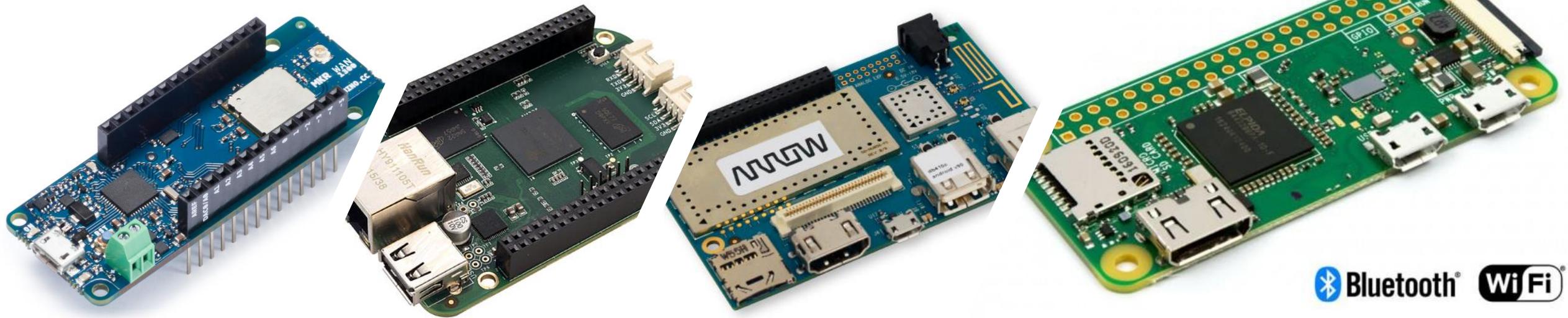
         
         
    

PARTS / KITS

CHARGING

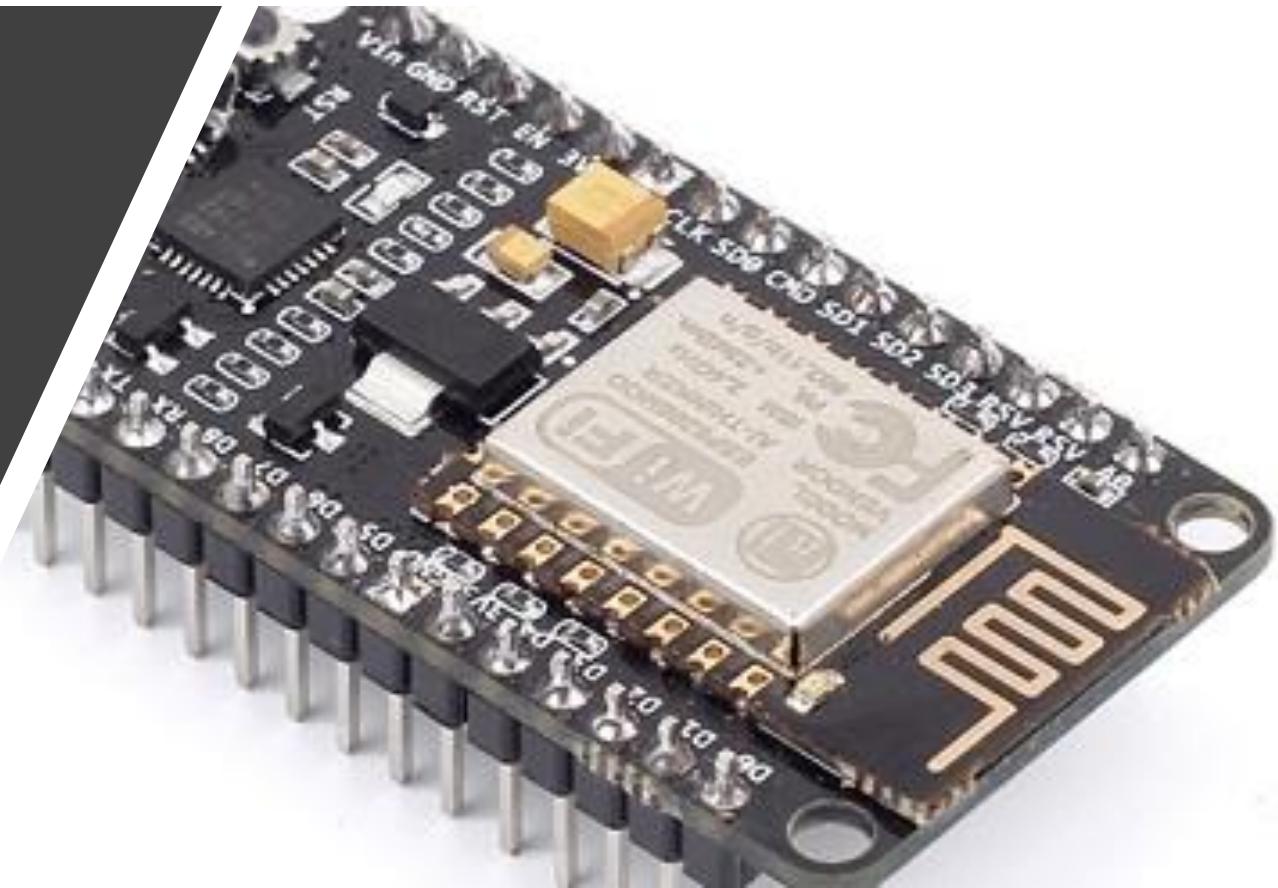


Bluetooth® WiFi®

Placas

Diversas opções para prototipagem usando as tecnologias mais atuais

Saiba Mais: [IoT Hardware Guide](#)



a few Internet of Things network protocols and its evolution

2007

IETF releases **6LoWPAN** network and **RPL**

application *CoAP, MQTT-SN, etc.*

transport *UDP, TCP*

mac *IEEE 802.15.4*

physical
(TCP averse)

Constrained Application Protocol

IPv6 Routing Protocol for Low-Power

- Handles node mobility
- Redundant instances
- Multiple routing metrics
- Seamless internet

RE
for smart
devices

IPv6

Internet

LoWPAN Adaption Layer

Full UDP/IPv6

IP

vs

Minimal UDP & LoWPAN

IEEE 802.15.4

Zigbee PRO

- Defines application services for interoperability
- Gateway has to translate user protocols



Protocolos

Infográfico de alguns protocolos de rede para IoT

2007

Zigbee goes IP: it runs over 6LoWPAN and RPL, supporting UDP, MQTT, CoAP, etc.

application *UDP, 6LoWPAN, IPSec*

Profiles

mac *IEEE 802.15.4*

physical *IEEE 802.15.1*

2013

ARM, Samsung, NXP, Nest Labs, OSRAM and others, release **Thread**, a 6LoWPAN networking protocol, full-mesh, with proprietary routing algorithm for 802.15.4 PHYs.

application *User Protocol*

mac *LoRa MAC: Class A, B or C*

2015

Zigbee PRO

- Defines application services for interoperability
- Gateway has to translate user protocols

Home Automation



Industrial & City



Smart Lighting



Industry 4.0



Bluetooth Smart 2015 version

- Piconet: one master to 7 slave
- Scatternet: multiple piconet in a tree
- Integrated in every modern smartphone

- Support 6LoWPAN by the IPS Profile

Internet

Gateway optional

2015

LoRaWAN: Low Power Wide Area Network

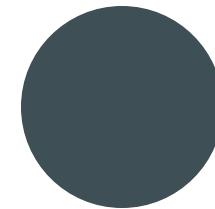
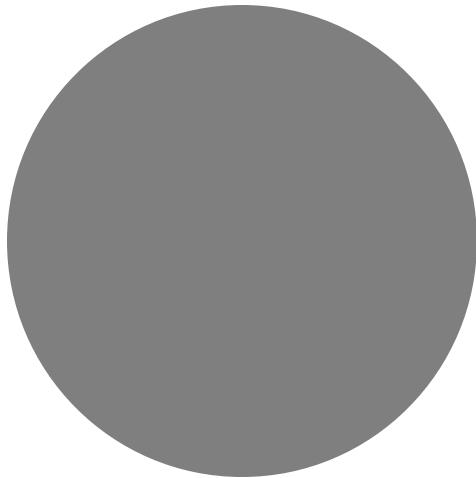
- Long range, home gateway stations

- Cheap stations and nodes

- 15 km range

Internet

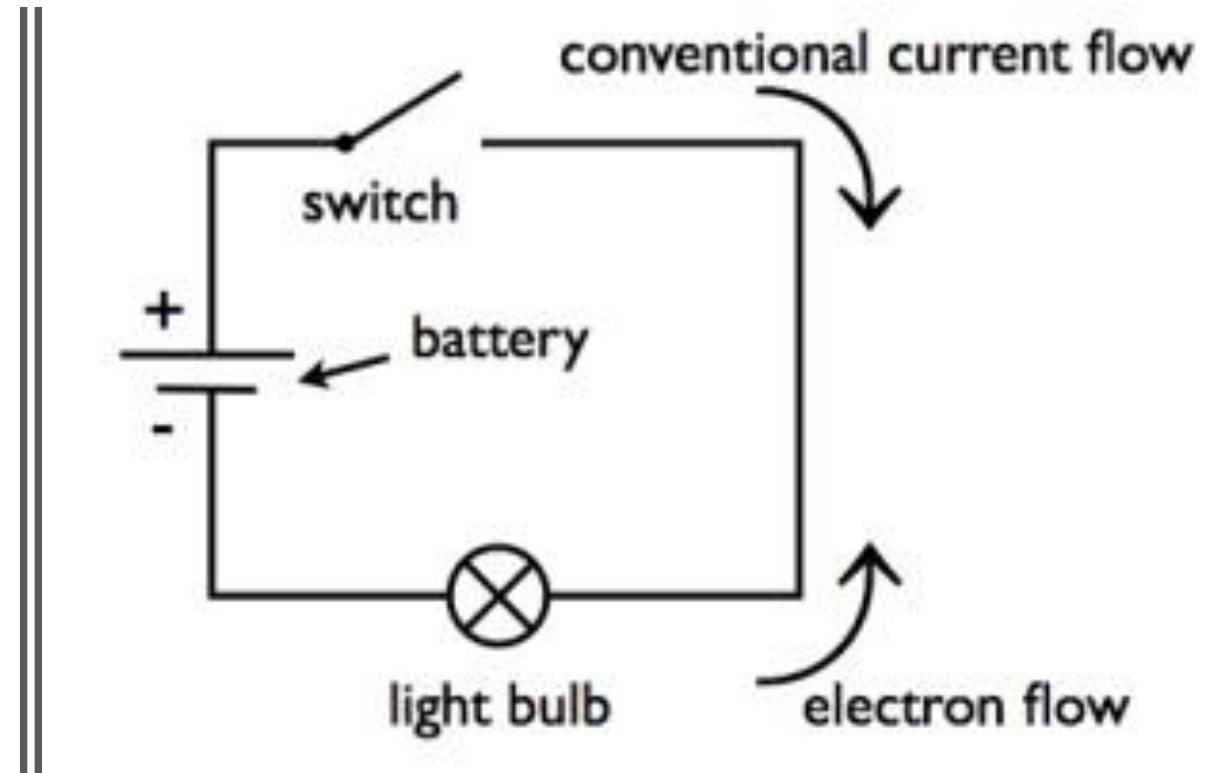
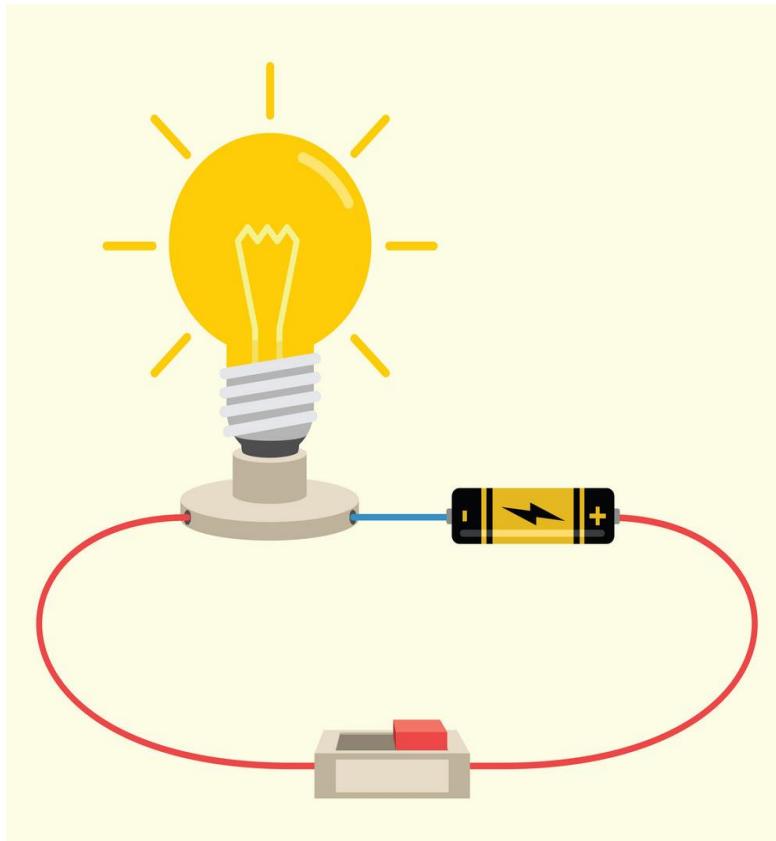
Gateway optional



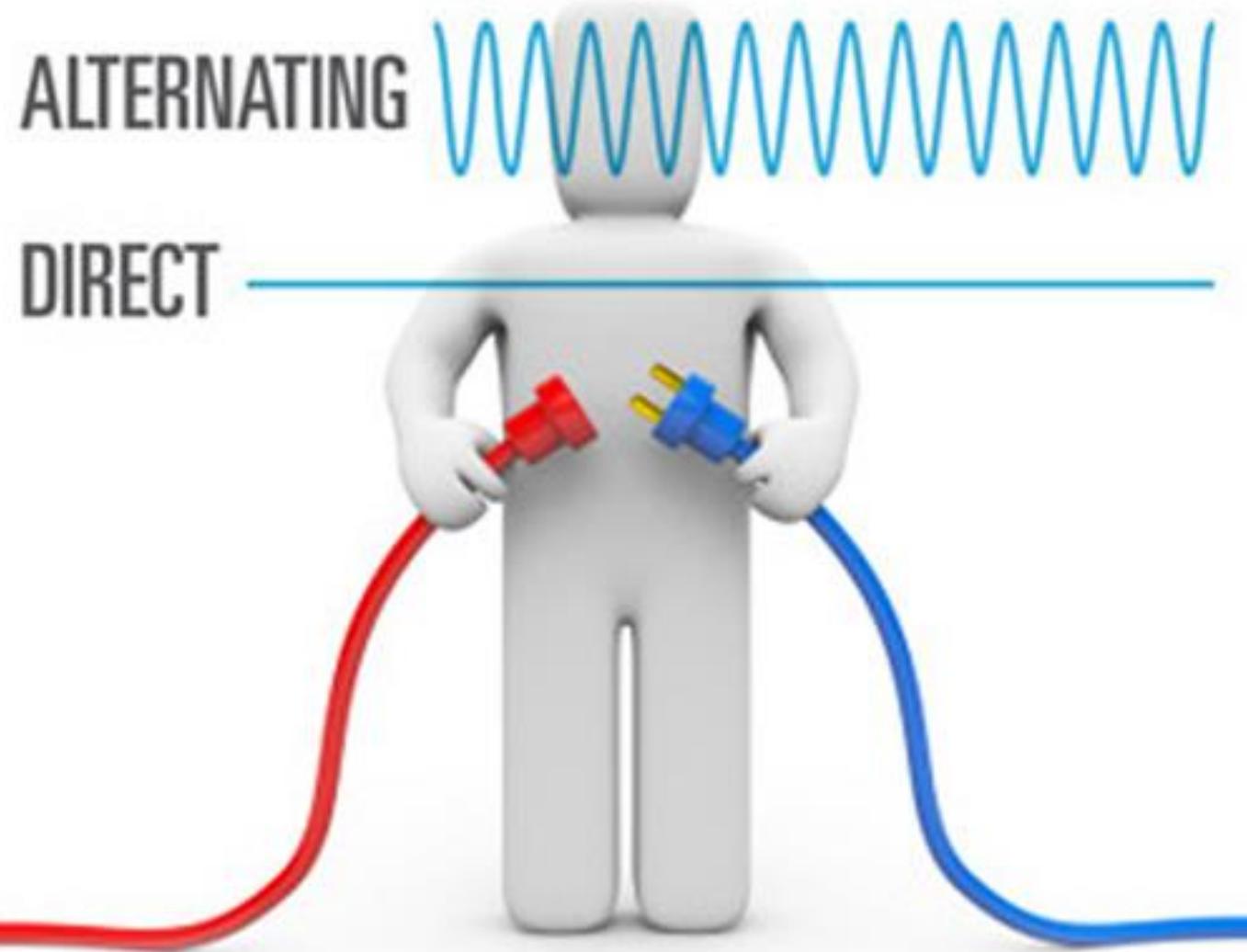
Eletrônica Básica



Circuito Elétrico

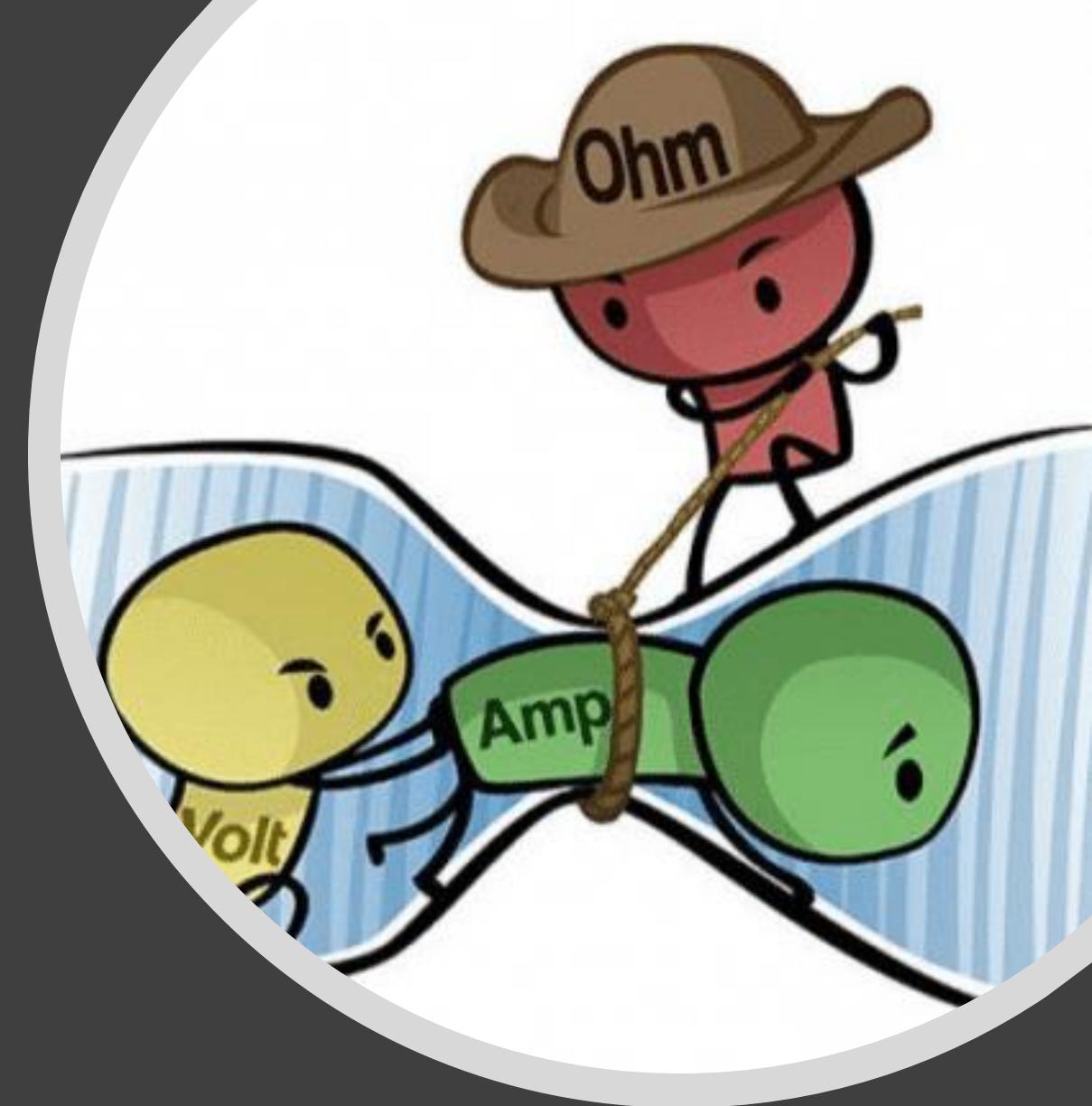


Corrente
Alternada x
Corrente
Contínua
(AC/DC)



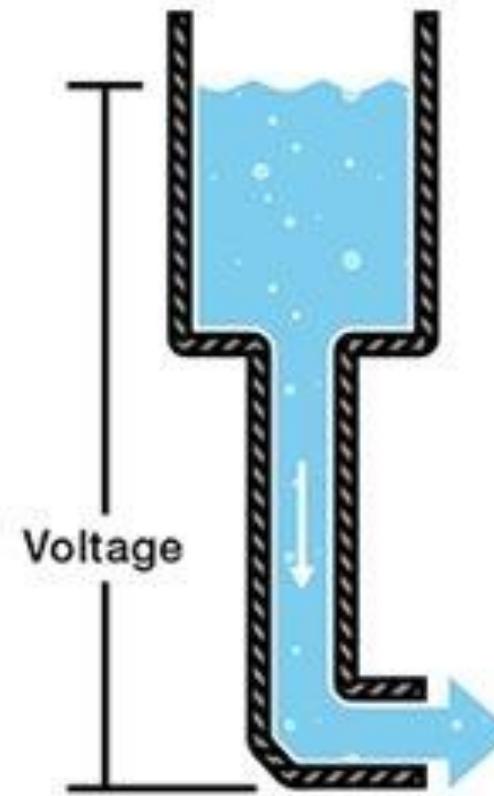
Tensão, Corrente e Resistência

- Tensão: unidade volt (V) em homenagem a Alessandro Volta;
- Corrente: Unidade ampere (A), em homenagem a André Marie Ampère;
- Resistência: Unidade Ohm (Ω), em homenagem a Georg Simon Ohm.



Tensão

Resumidamente a tensão elétrica é a diferença de potencial entre dois pontos. Sua unidade é o volt (V) e é representada nas equações e circuitos geralmente pelas letras U e V.

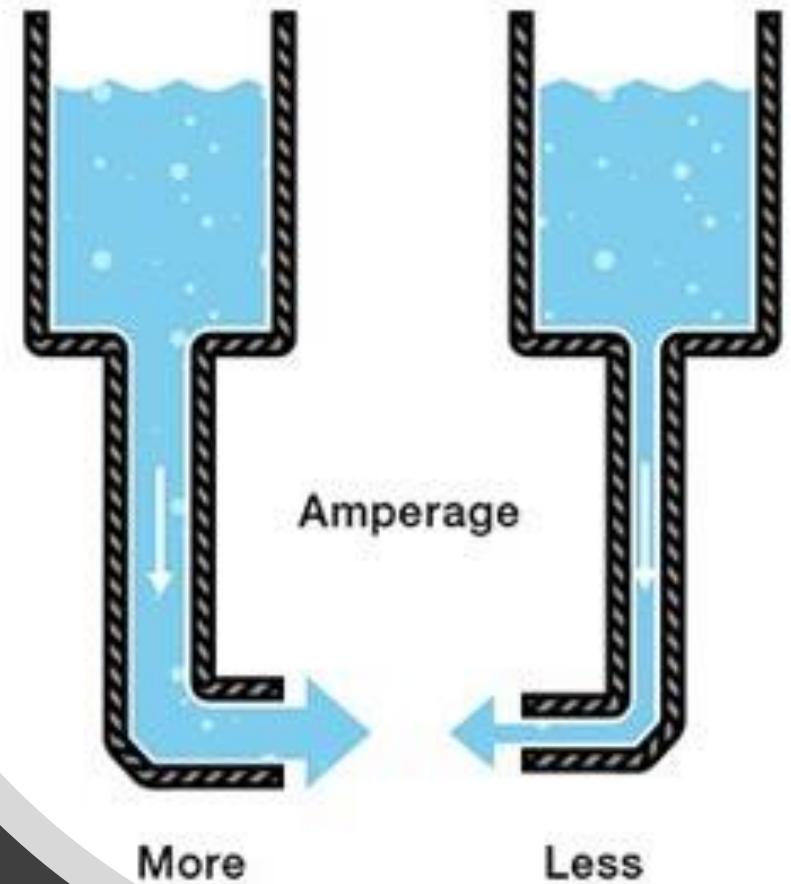


Corrente

É o fluxo de elétrons em um condutor quando submetido a uma diferença de potencial.

A corrente elétrica pode causar alguns efeitos, por exemplo, o efeito térmico e efeito luminoso, que usamos em nosso dia a dia.

A unidade de corrente elétrica é o ampère (A) e é geralmente representada em equações e circuitos pela letra I .

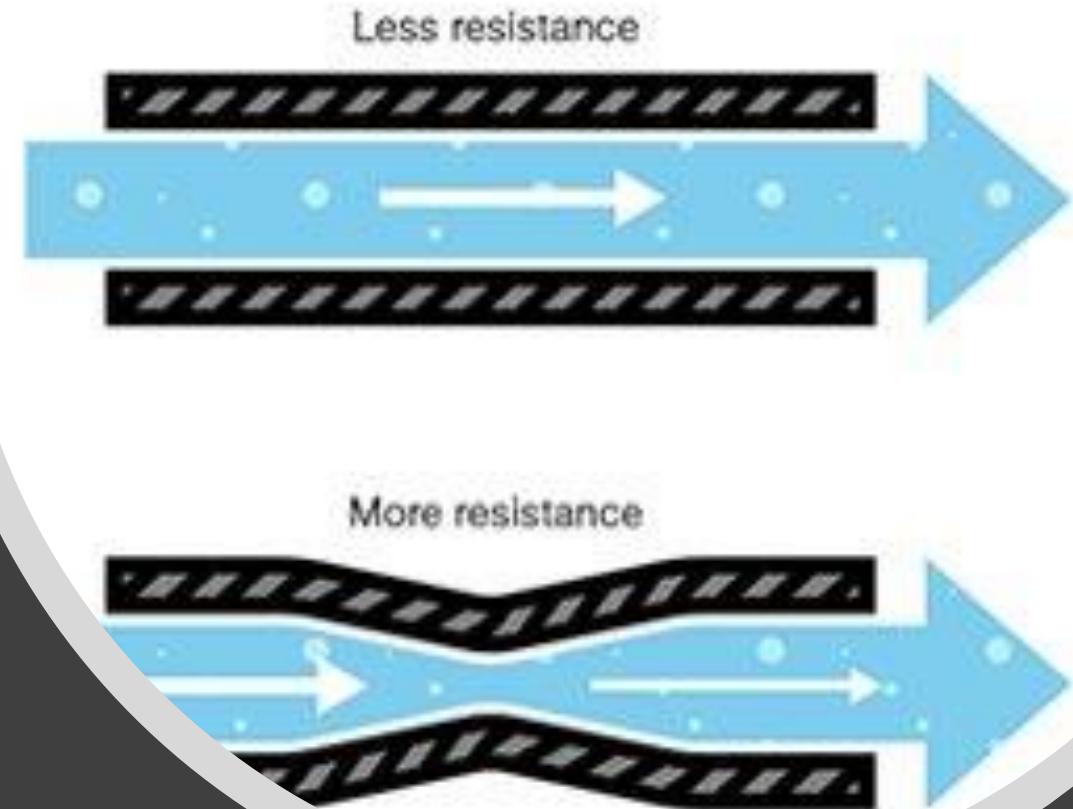


Resistance

Resistência

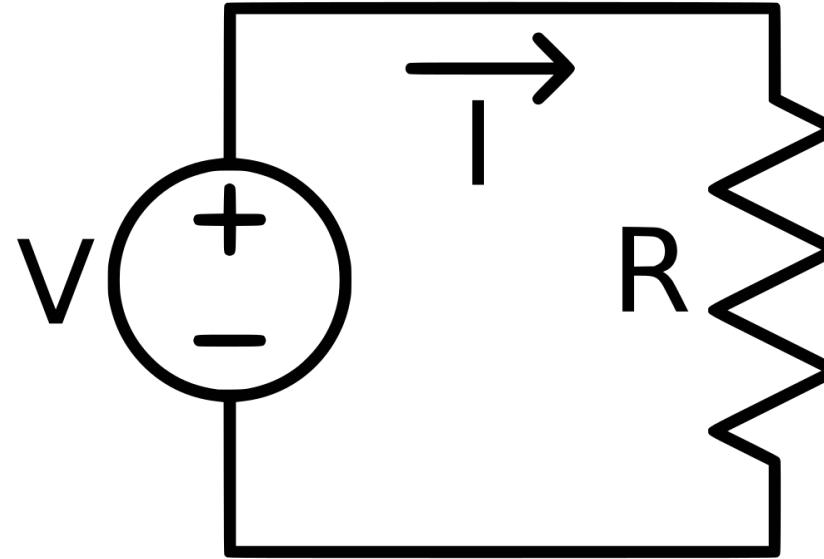
A oposição à passagem de corrente é chamada de resistência elétrica.

A unidade de Resistência elétrica é o Ohm cujo o símbolo é representado pela letra grega “Omega” (Ω). Ela é representada geralmente pela letra R em equações e circuitos.



Lei de Ohm

“A corrente que flui por um resistor é proporcional à tensão aplicada e inversamente proporcional ao valor de sua resistência”



Representada matematicamente por:

$$I = V/R$$

(Corrente é igual ao valor da tensão dividida pela resistência)

Lei de Ohm



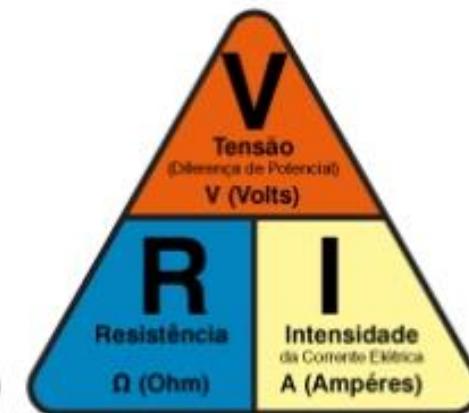
Calcular a Intensidade
da Corrente

$$I = \frac{V}{R}$$



Calcular a Tensão

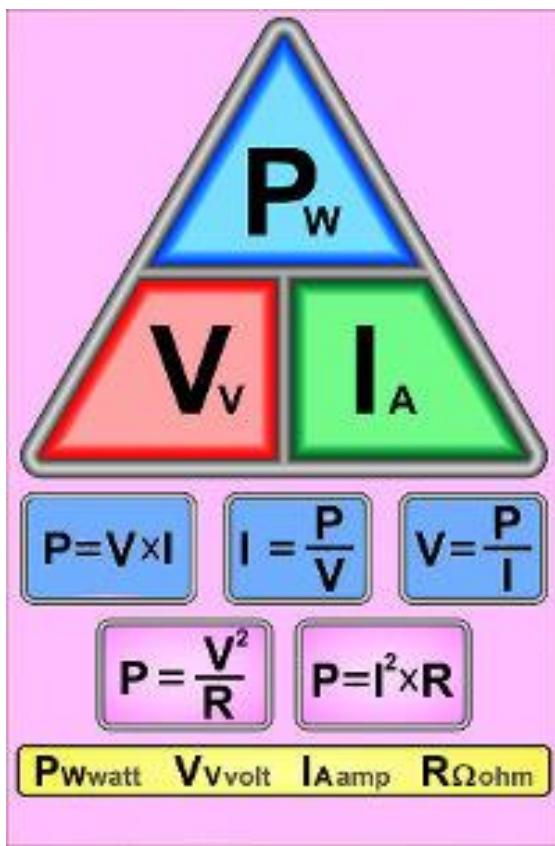
$$V = R \times I$$



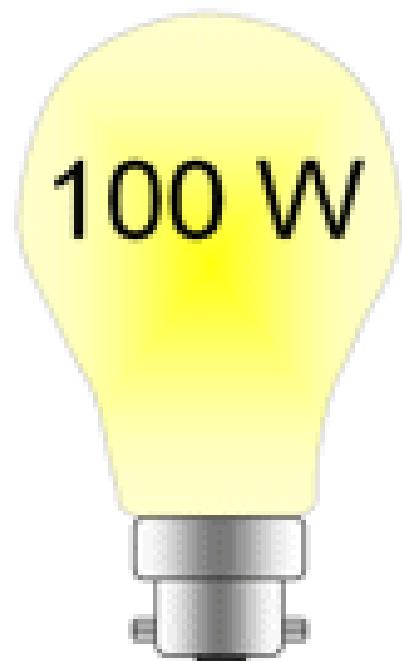
Calcular a Resistência

$$R = \frac{V}{I}$$

Potência

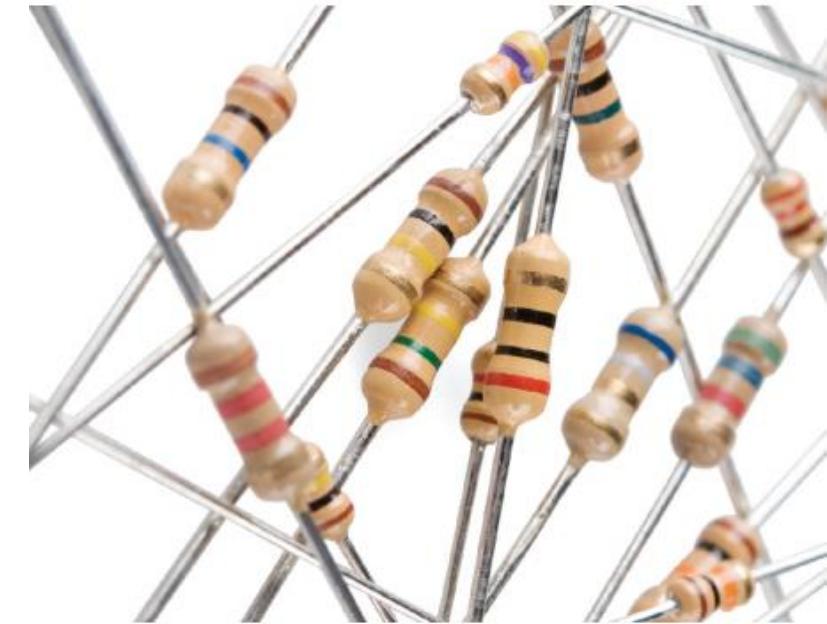


||



Prefixos do sistema internacional

Prefixo	Símbolo	potencia base 10
tera	T	10^{12}
giga	G	10^9
mega	M	10^6
kilo	k	10^3
-	-	10^0
mini	m	10^{-3}
micro	μ	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}



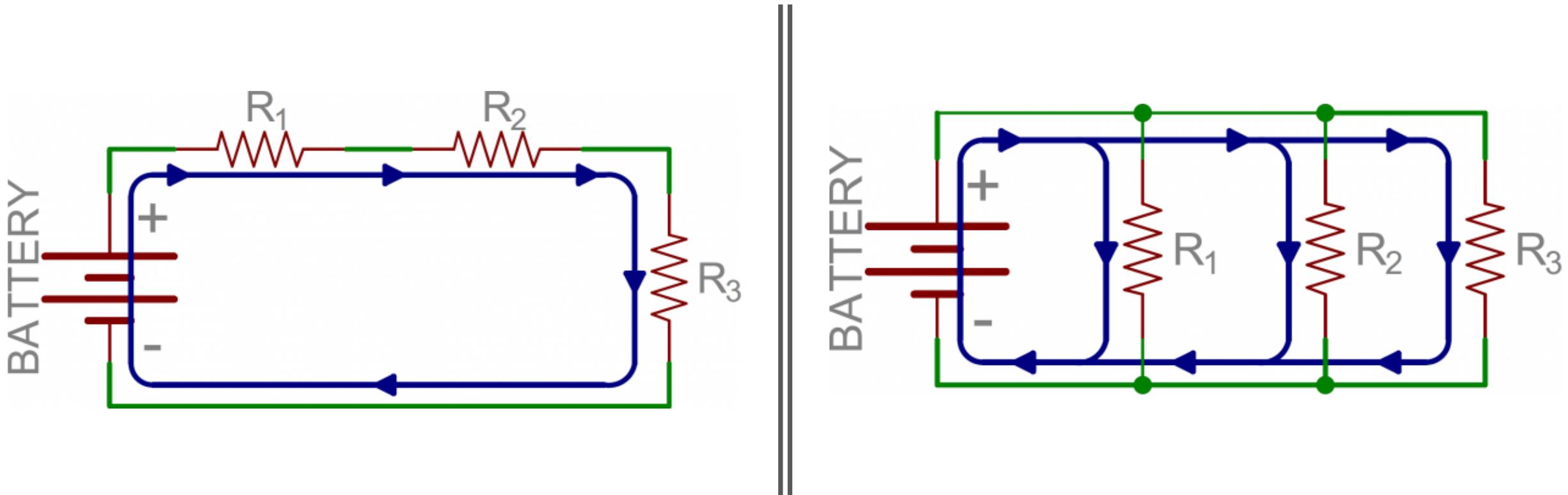
Resistores

Código de cores

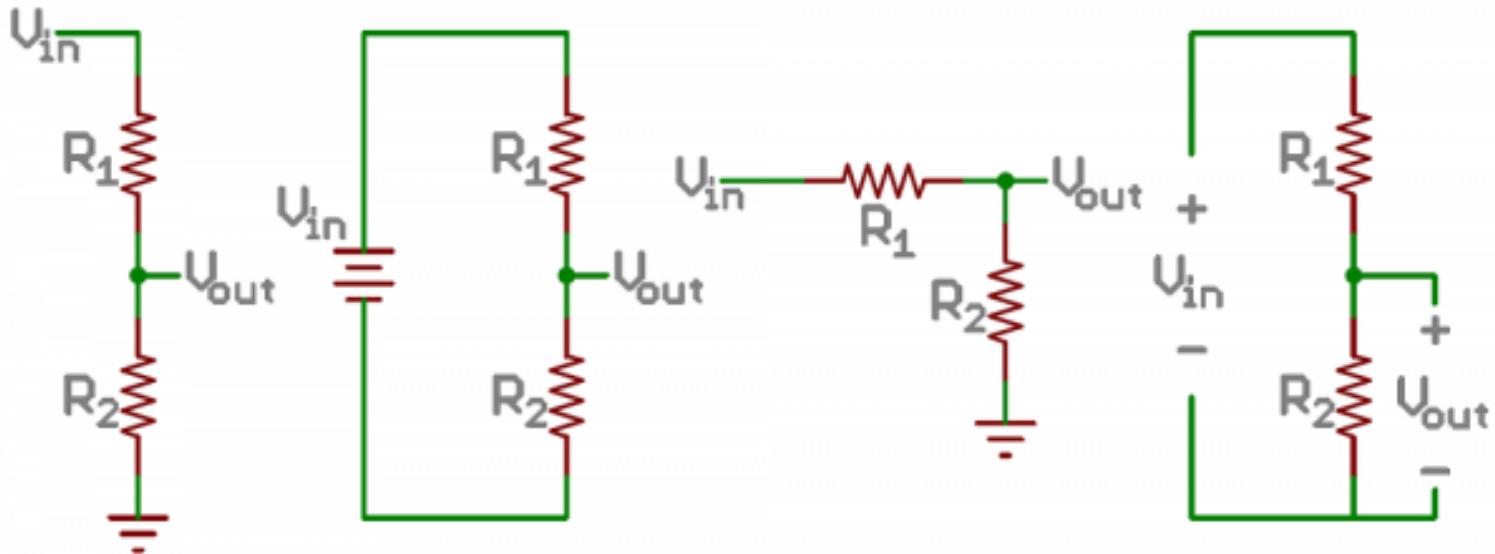
Cor	1ª Faixa	2ª Faixa	Nº de zeros/multiplicador	Tolerância
Preto	0	0	0	
Marrom	1	1	1	± 1%
Vermelho	2	2	2	± 2%
Laranja	3	3	3	
Amarelo	4	4	4	
Verde	5	5	5	± 0,5%
Azul	6	6	6	± 0,25%
Violeta	7	7	7	± 0,1%
Cinza	8	8	8	± 0,05%
Branco	9	9	9	
Dourado			x0,1	± 5%
Prata			x0,01	± 10%



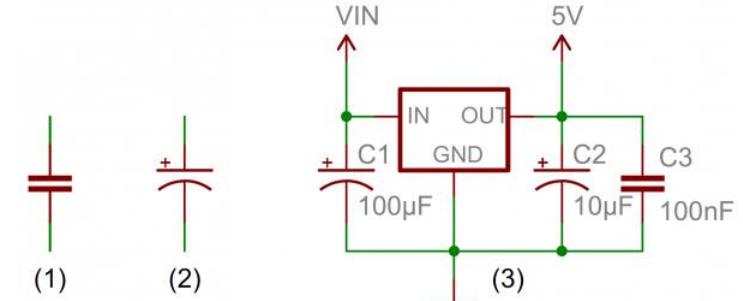
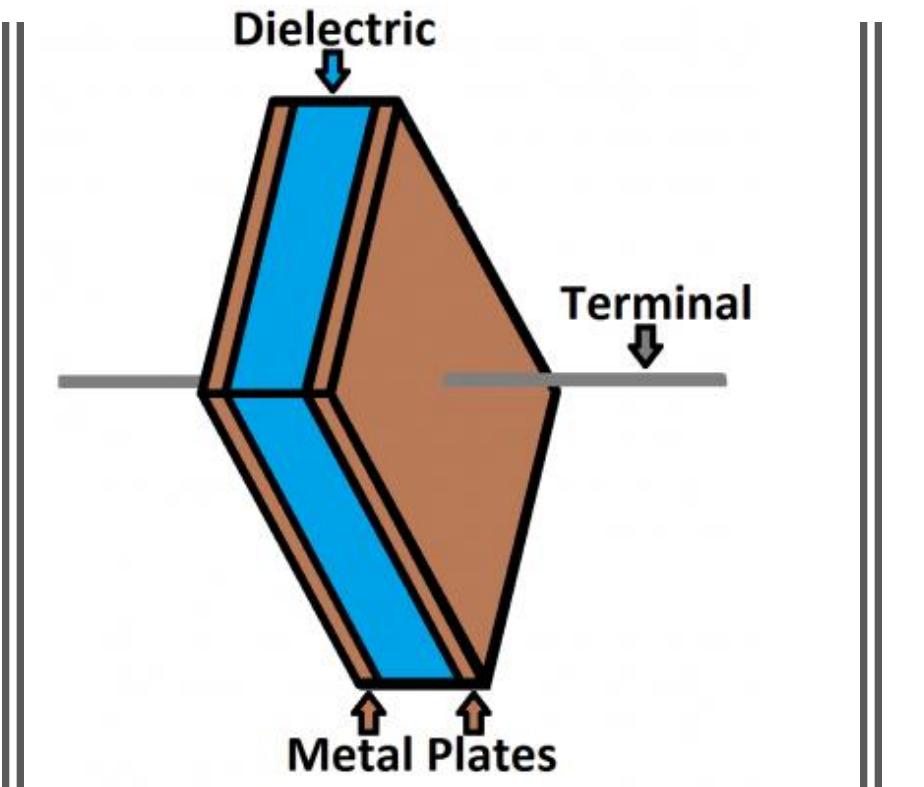
Circuitos: série e paralelo



Divisor de tensão



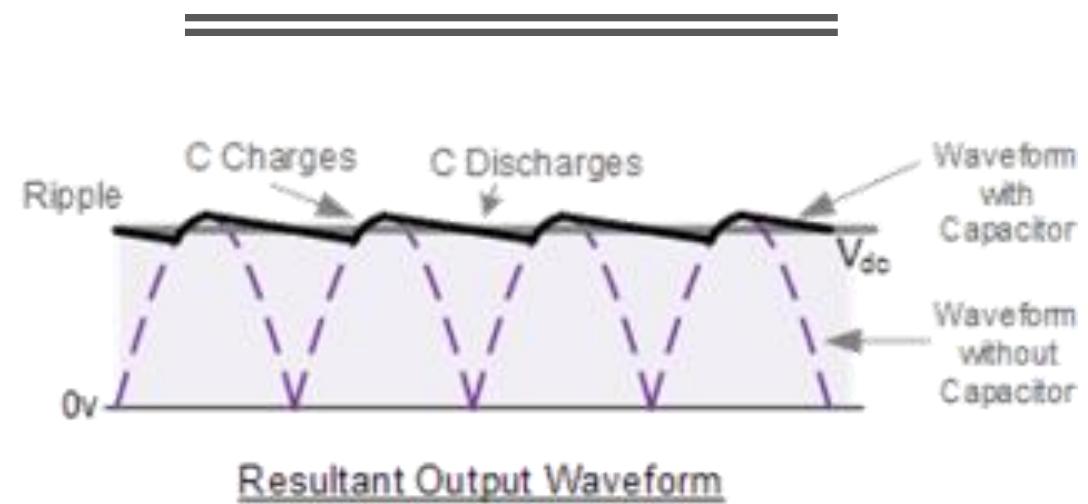
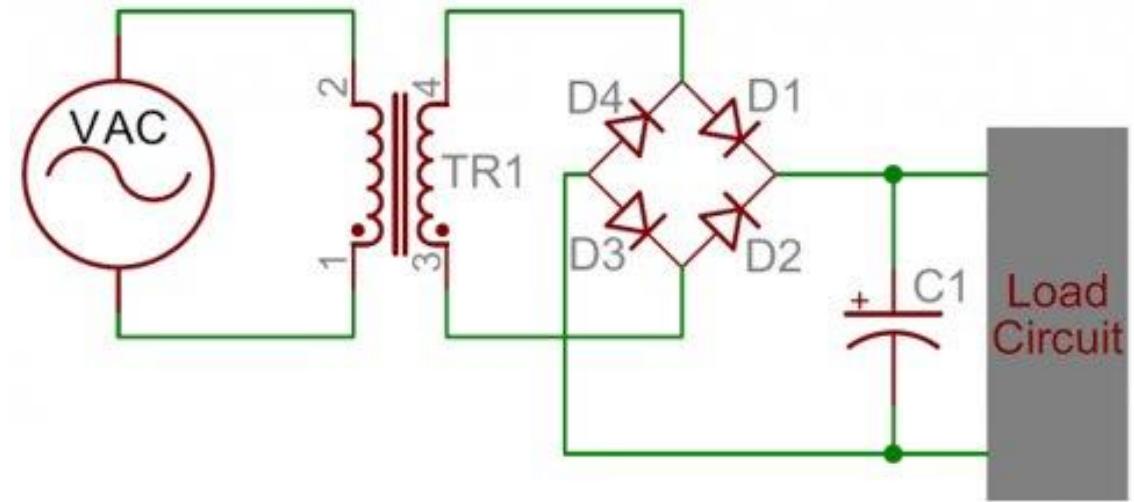
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$



Capacitores

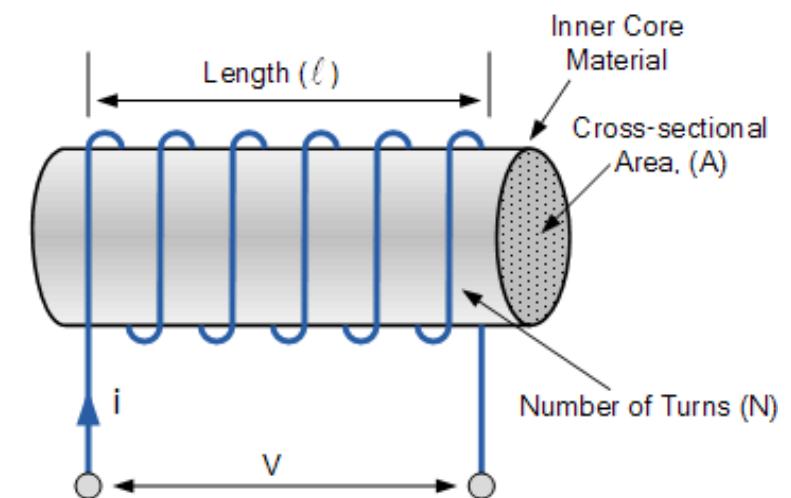
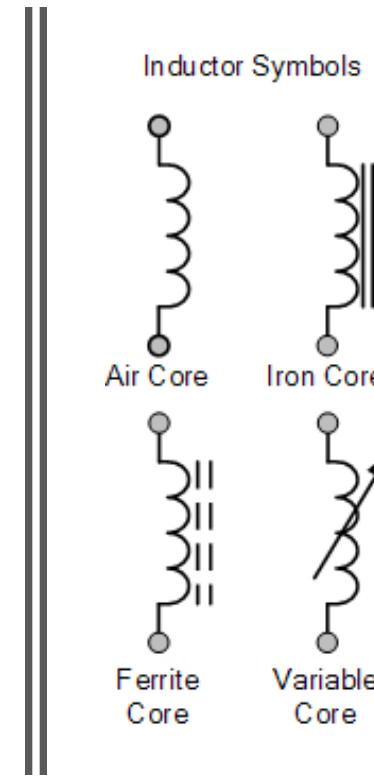
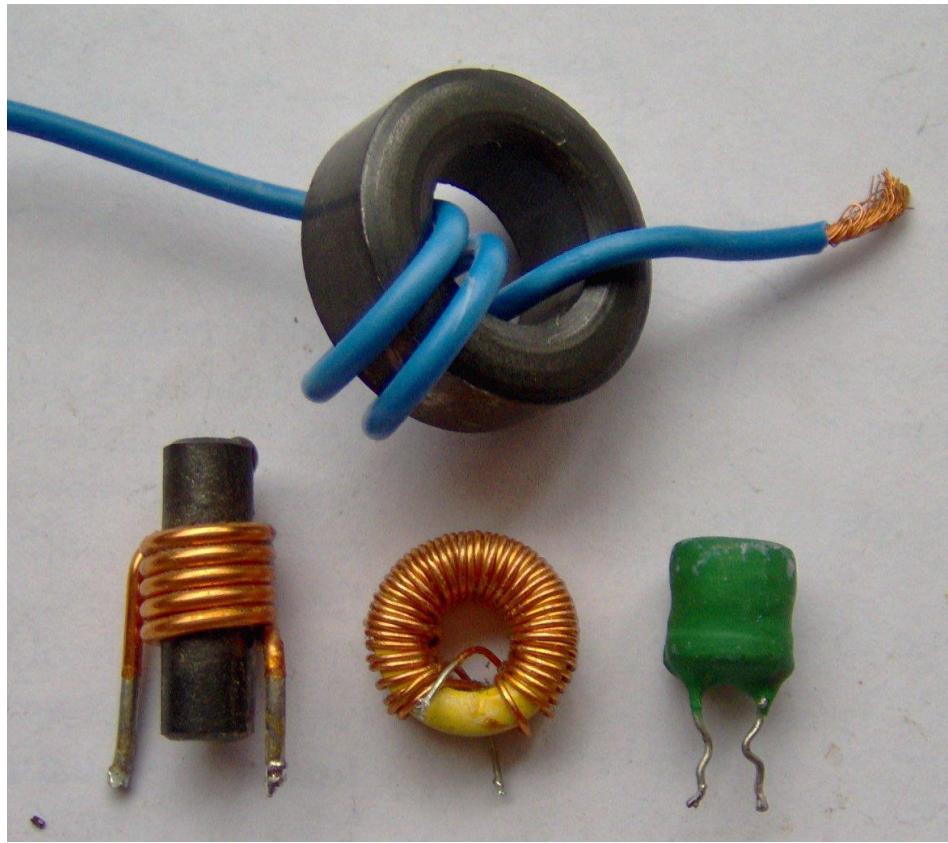
<https://learn.sparkfun.com/tutorials/capacitors>

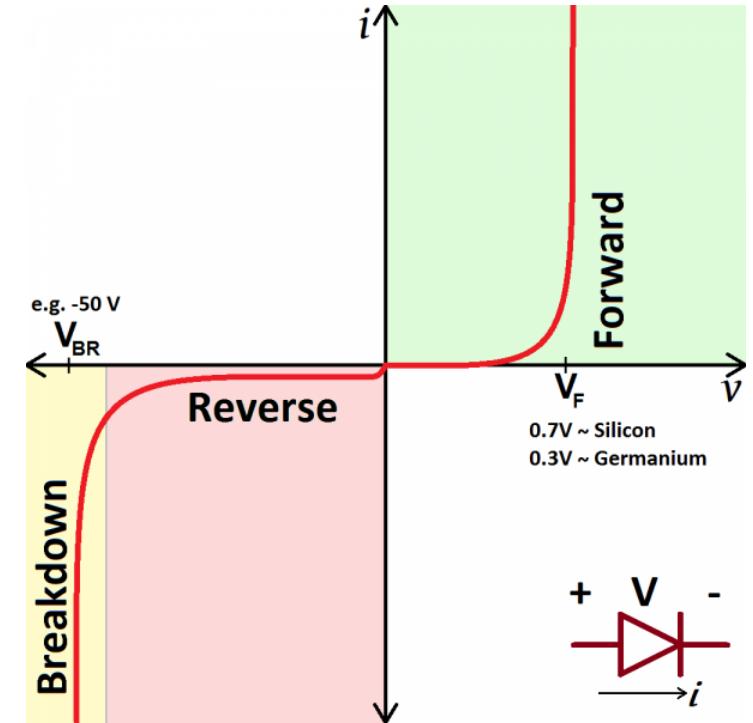
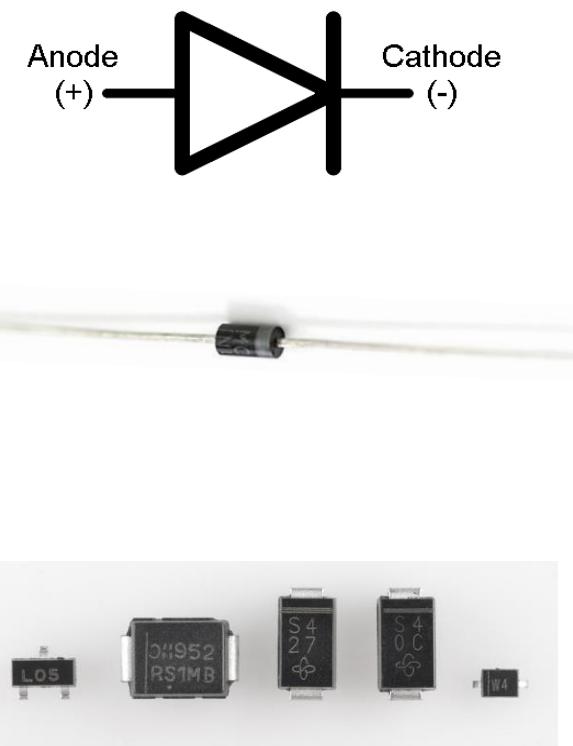
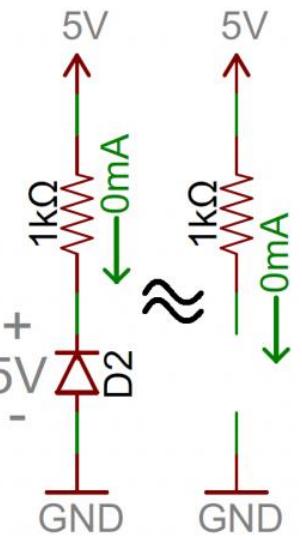
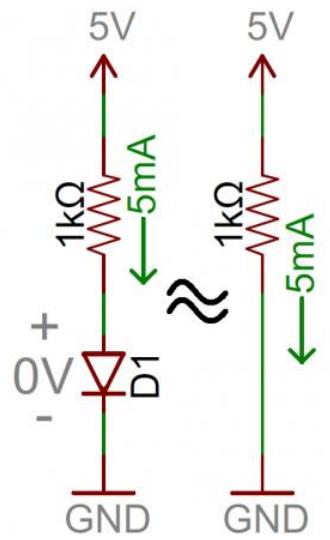
Aplicação



Resultant Output Waveform

Indutores

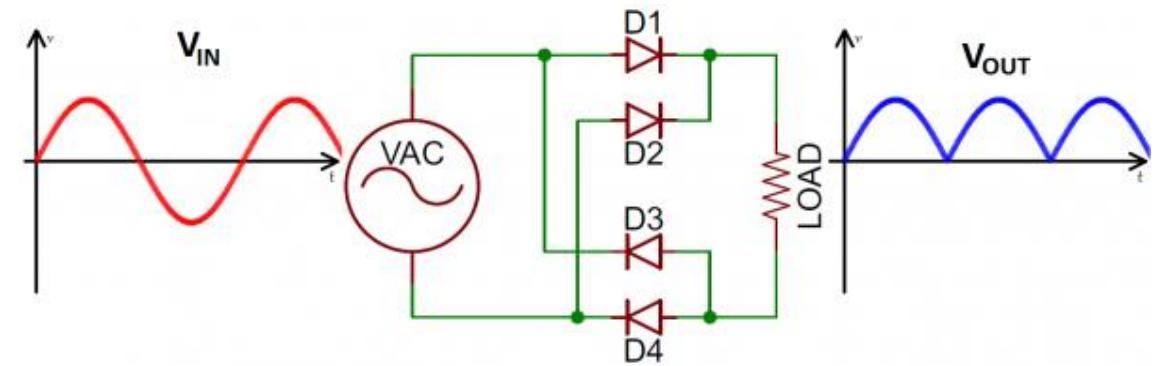
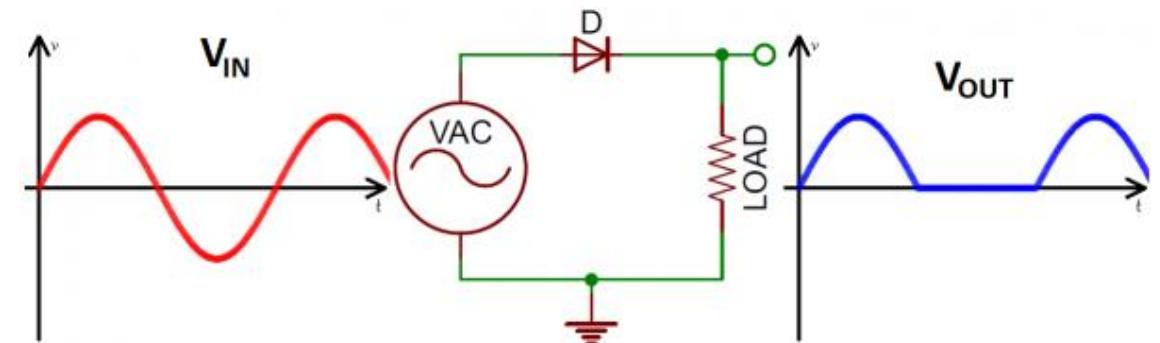


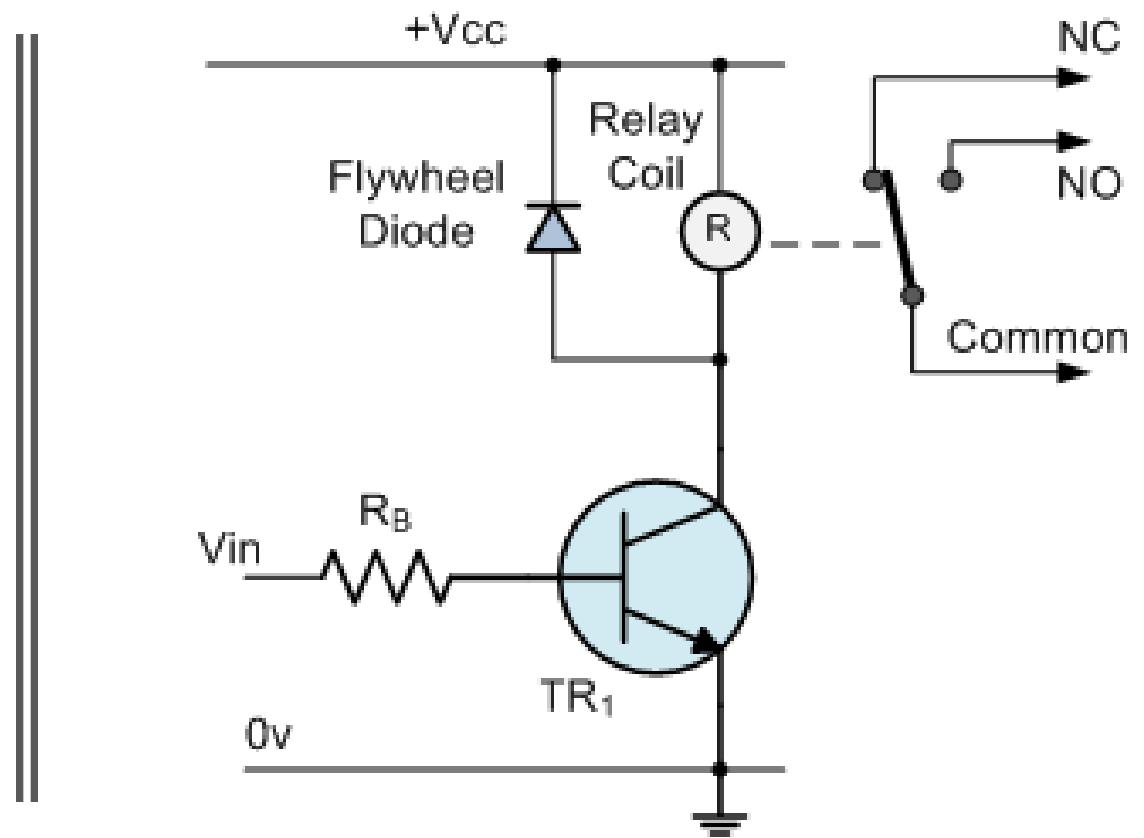
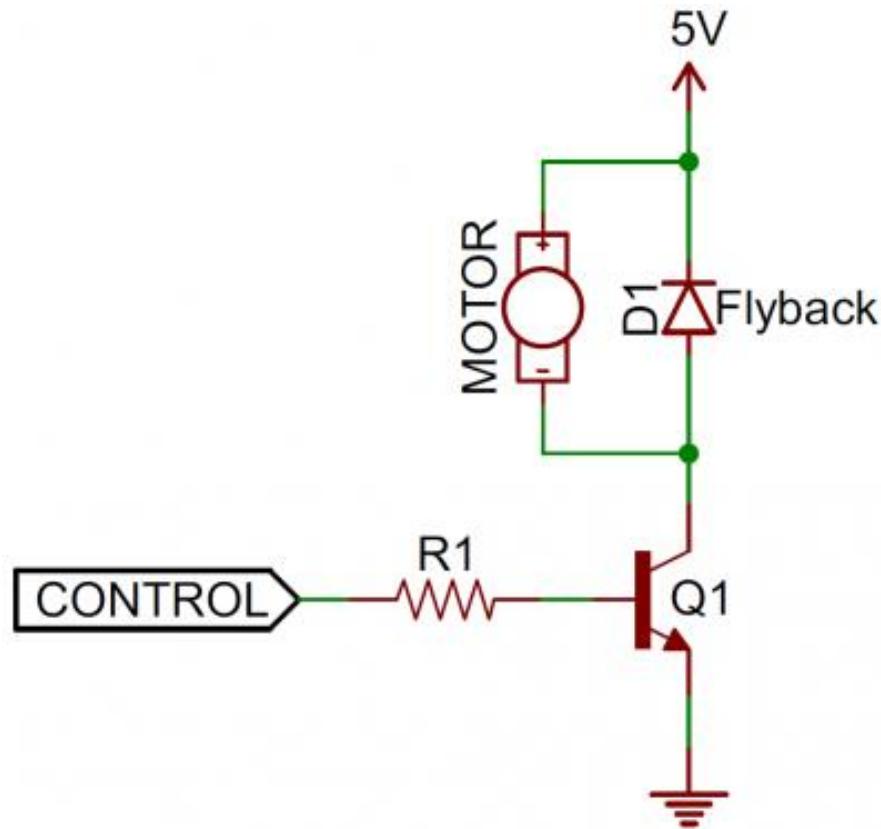


Diodos

<https://learn.sparkfun.com/tutorials/diodes>

Retificador

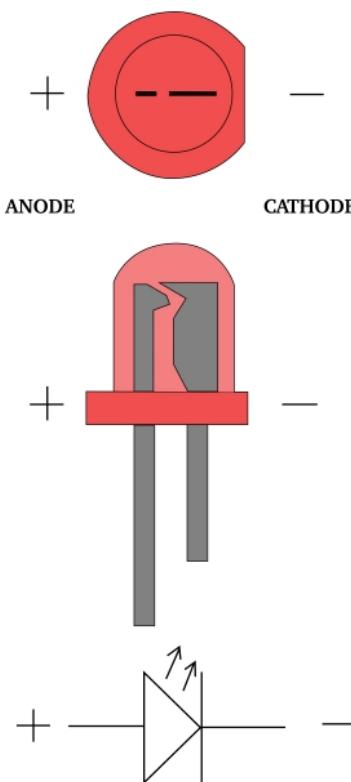




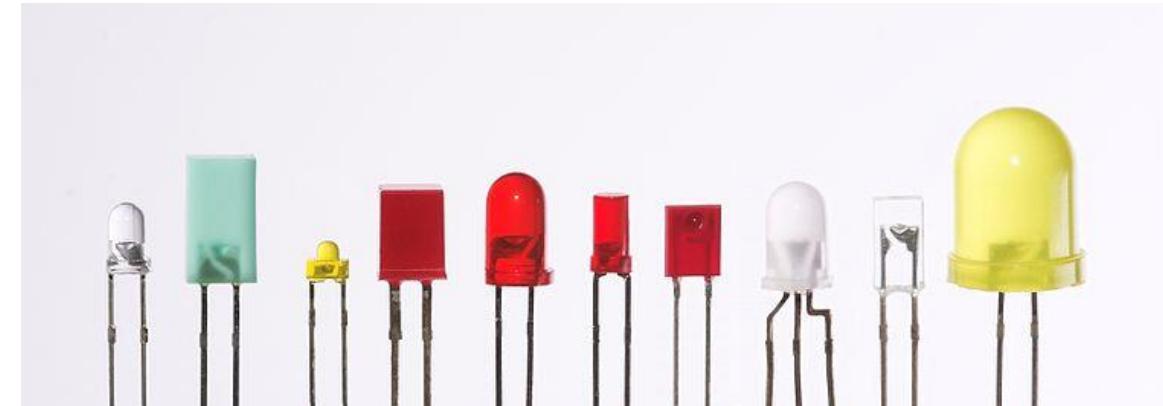
Diodo flyback

<https://learn.sparkfun.com/tutorials/diodes>

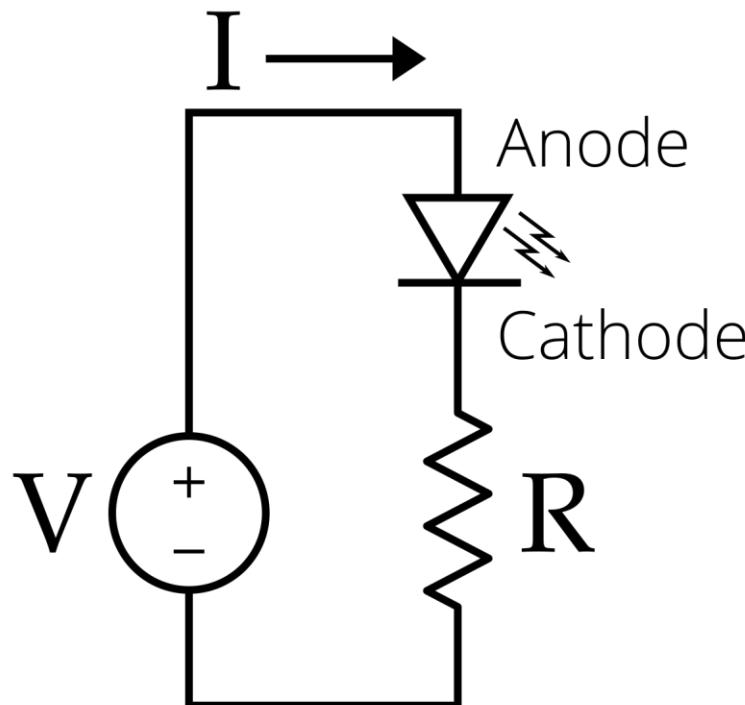
LEDs



||



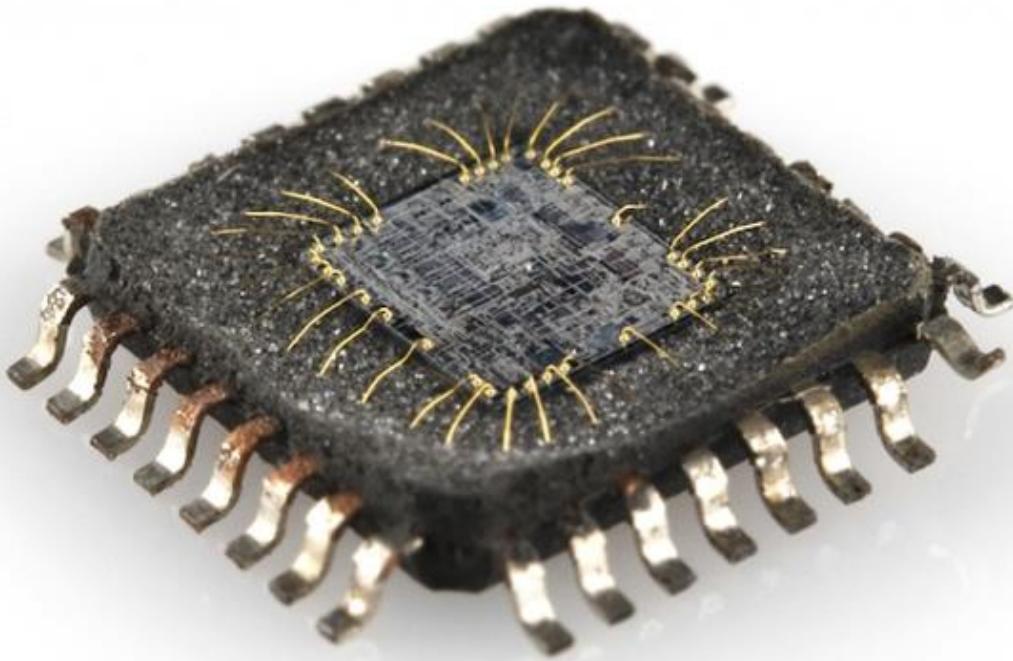
LEDs



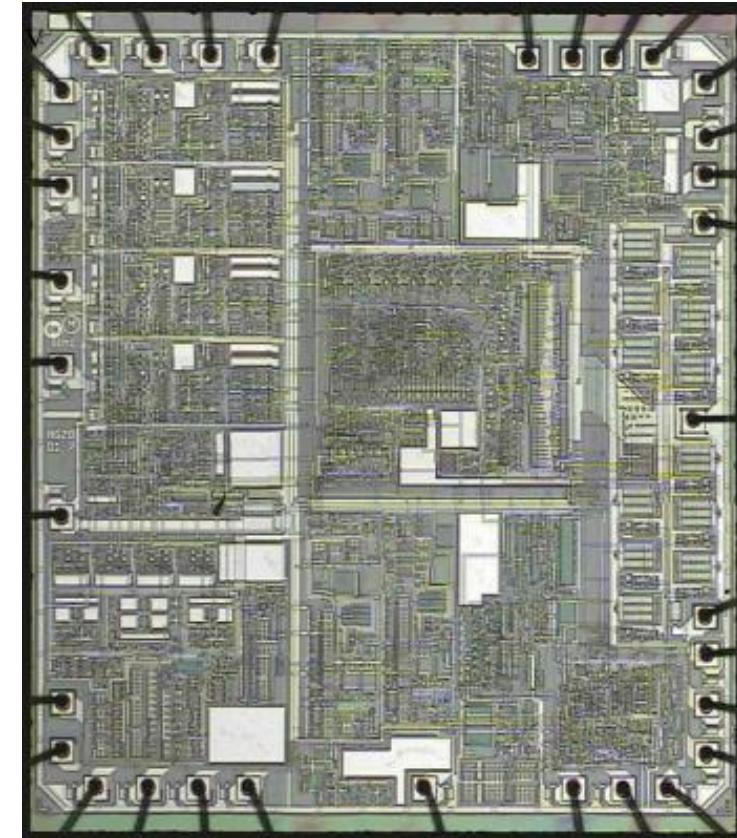
ITEMS	Symbol	Test condition	Min.	Typ.	Max.	Unit
Forward Voltage	V_F	$I_F=20\text{mA}$	1.8	---	2.2	V
Wavelength (nm) or TC(k)	$\Delta \lambda$	$I_F=20\text{mA}$	620	---	625	nm
*Luminous intensity	I_v	$I_F=20\text{mA}$	150	---	200	mcd

ITEMS	Symbol	Absolute Maximum Rating	Unit
Forward Current	I_F	20	mA
Peak Forward Current	I_{FP}	30	mA
Suggestion Using Current	I_{su}	16-18	mA
Reverse Voltage ($V_R=5\text{V}$)	I_R	10	uA
Power Dissipation	P_D	105	mW
Operation Temperature	T_{OPR}	-40 ~ 85	°C
Storage Temperature	T_{STG}	-40 ~ 100	°C
Lead Soldering Temperature	T_{SOL}	Max. 260°C for 3 Sec. Max. (3mm from the base of the epoxy bulb)	

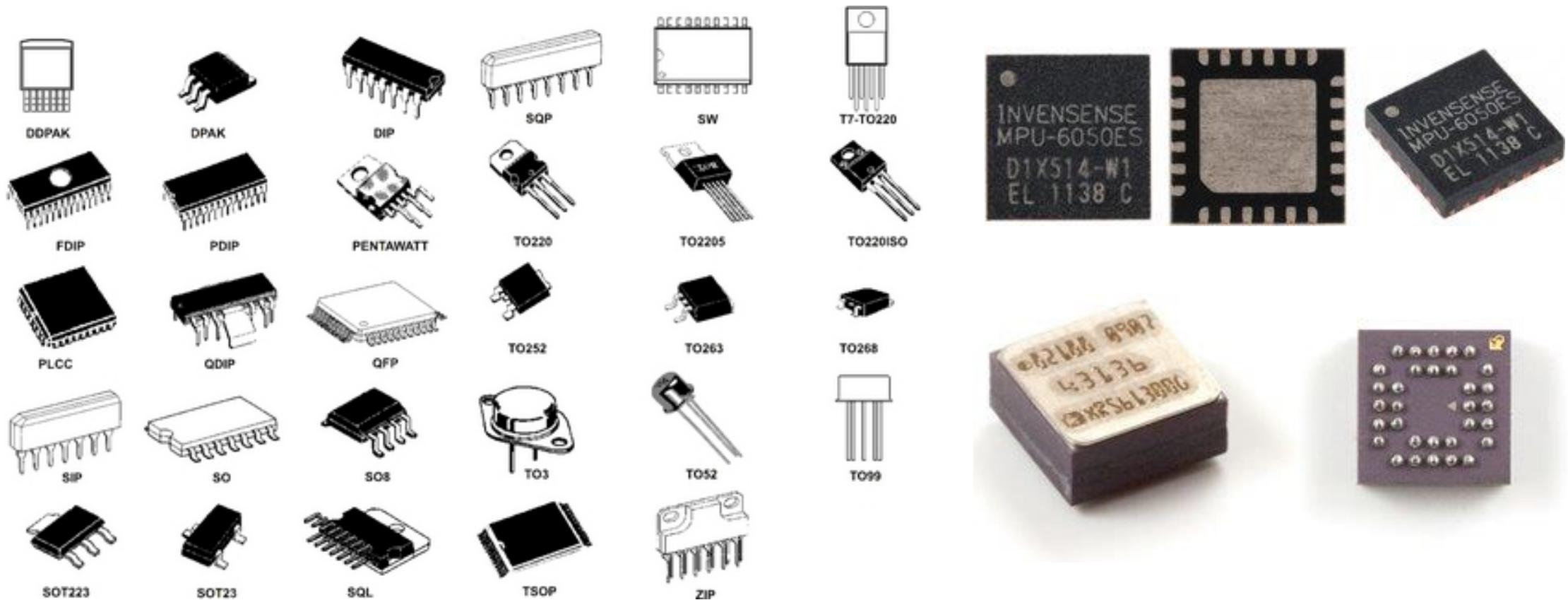
Circuitos integrados



||

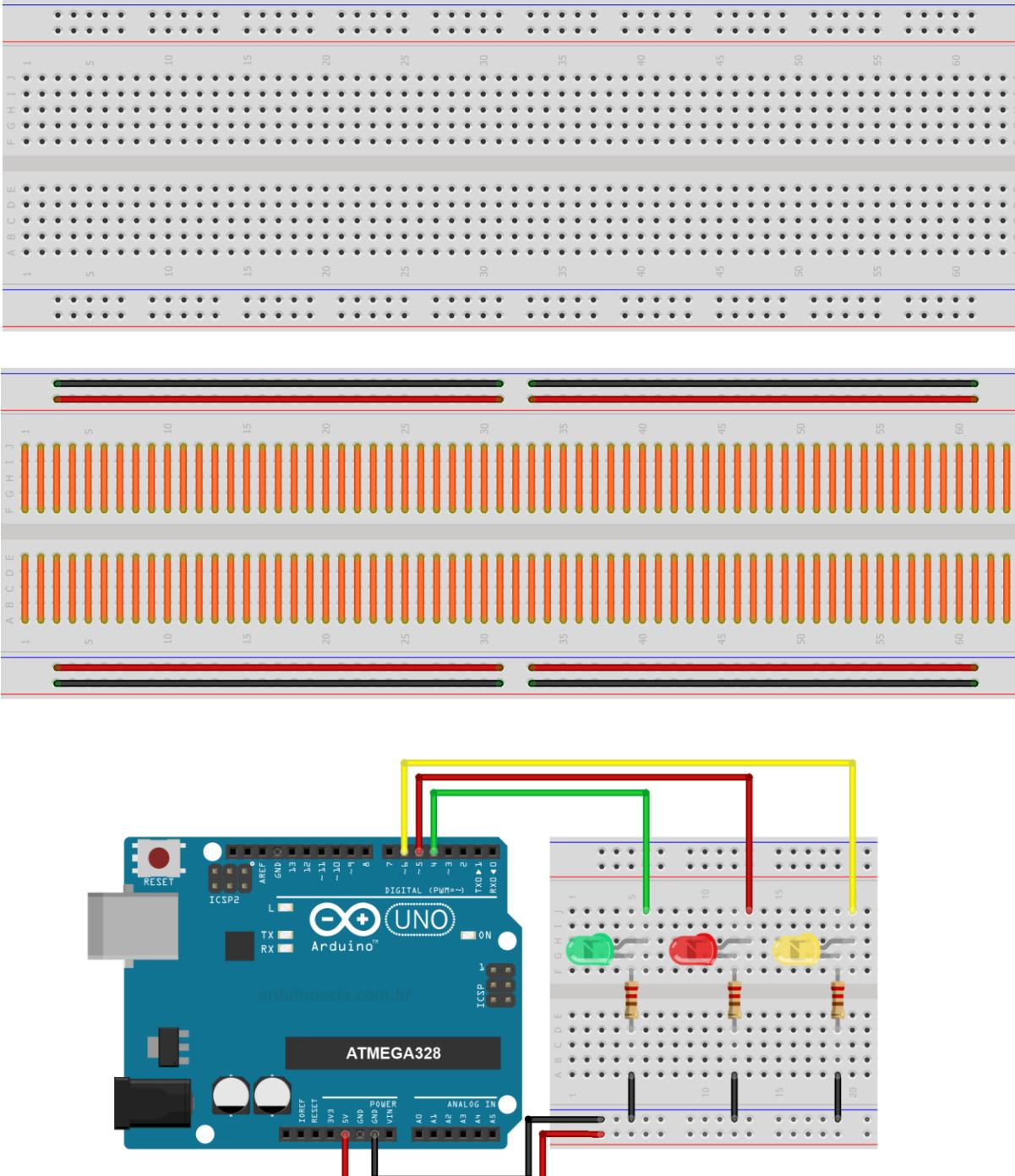


Circuitos integrados

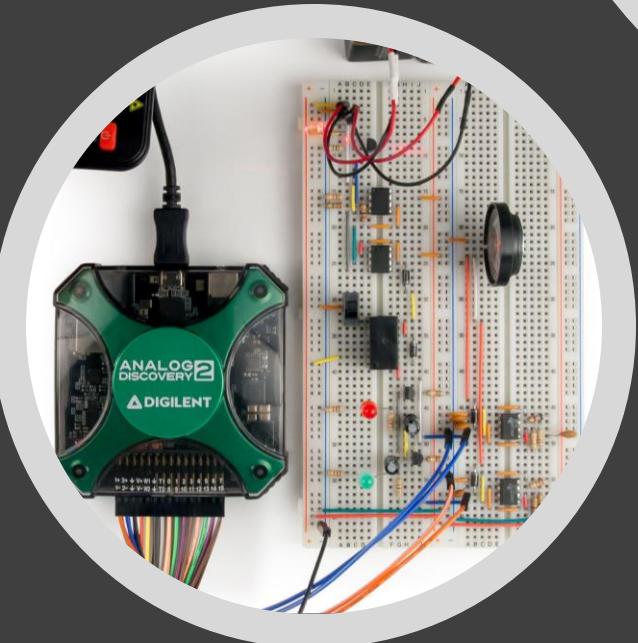
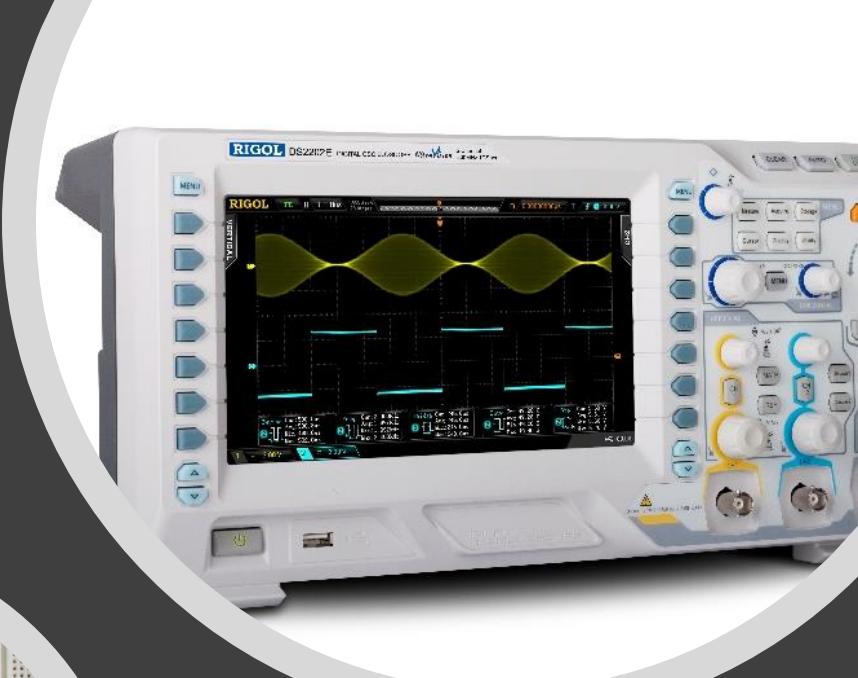


Protoboard

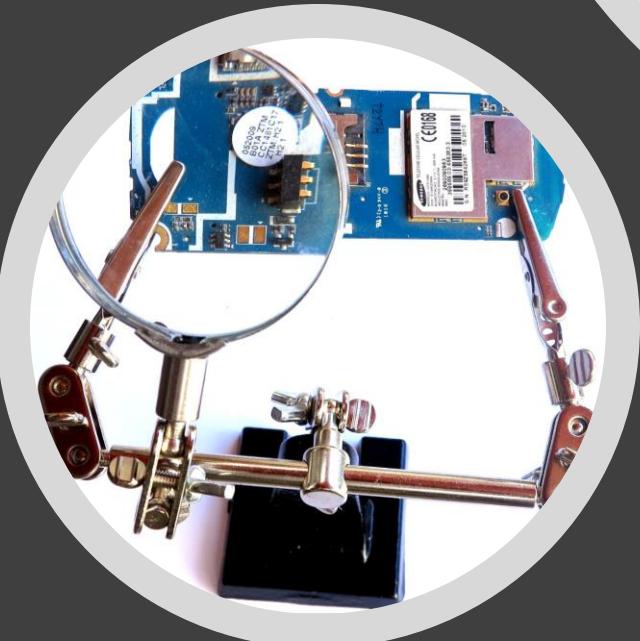
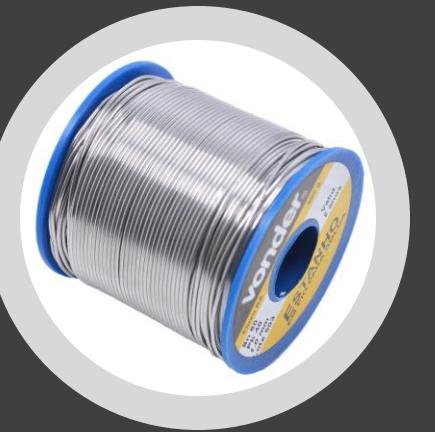
Matriz de contatos



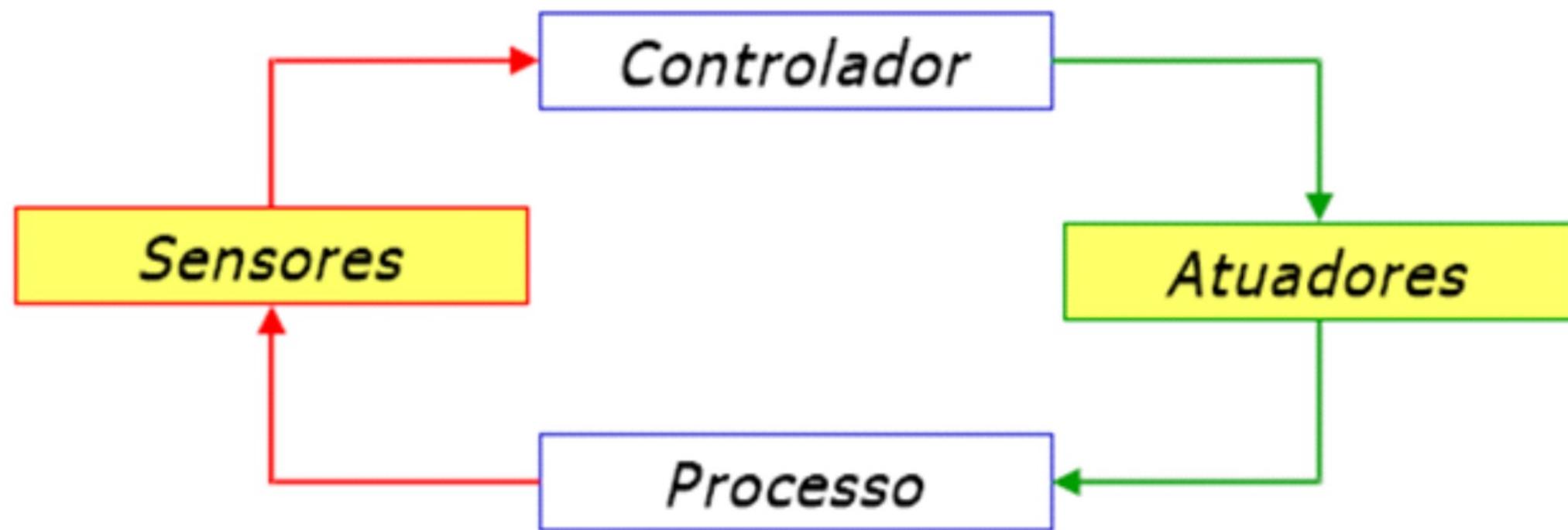
Instrumentos



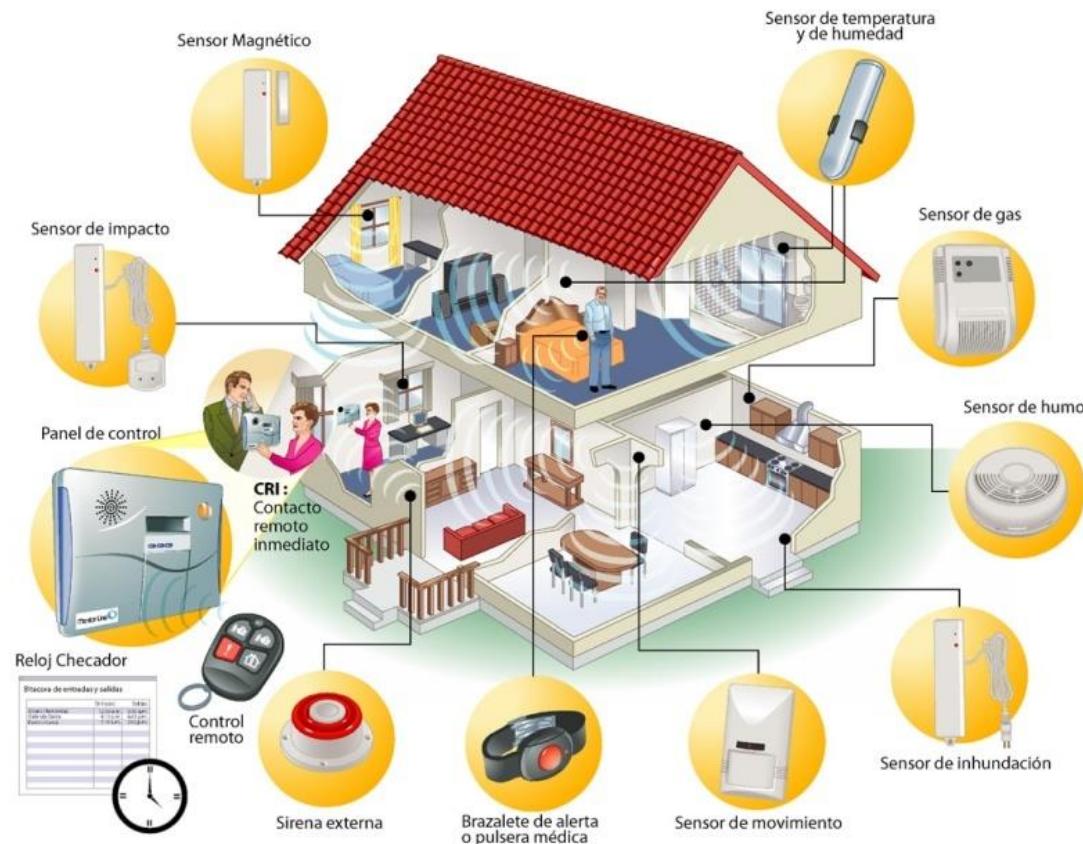
Ferramentas

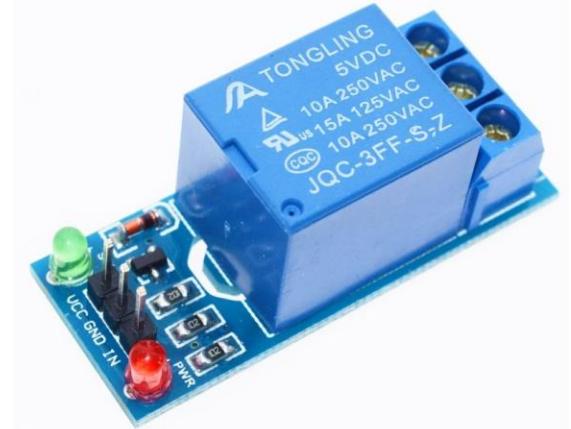
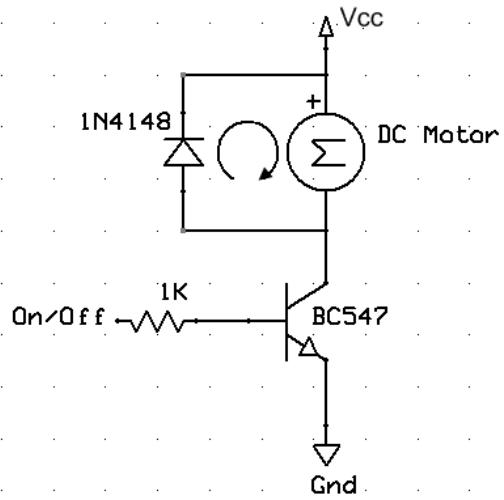


Sensores e Atuadores



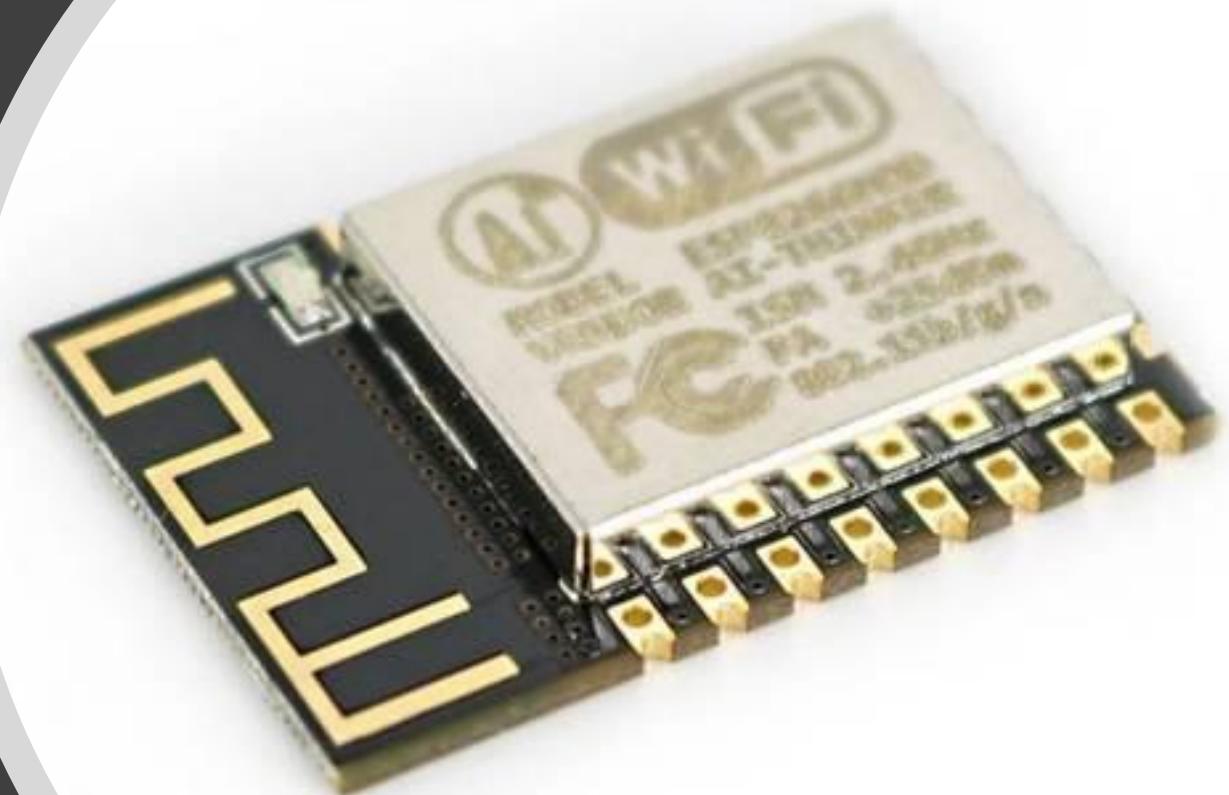
Sensores



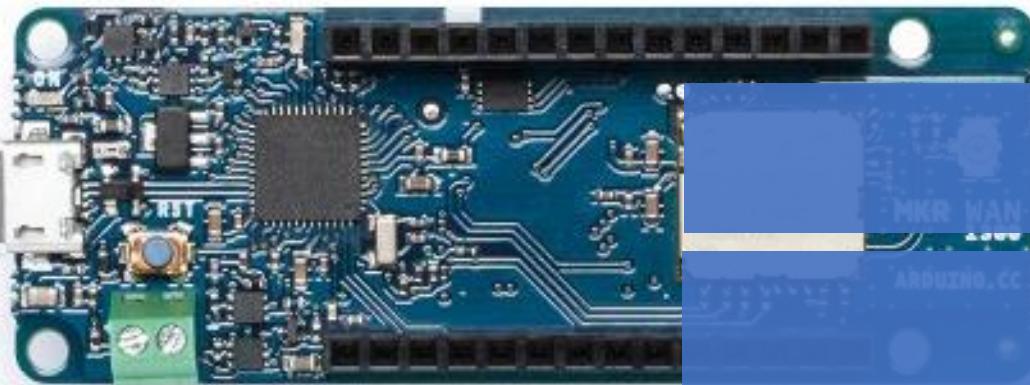


Atuadores

Placas para IoT

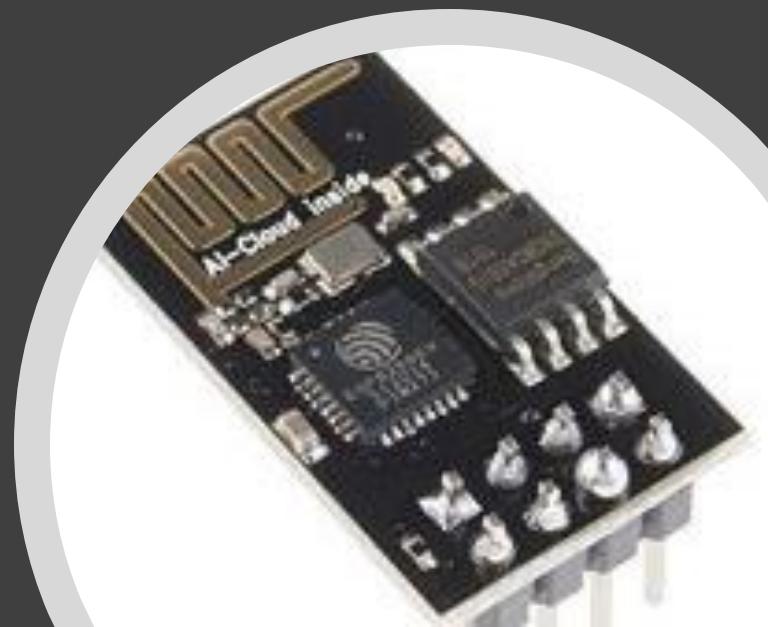
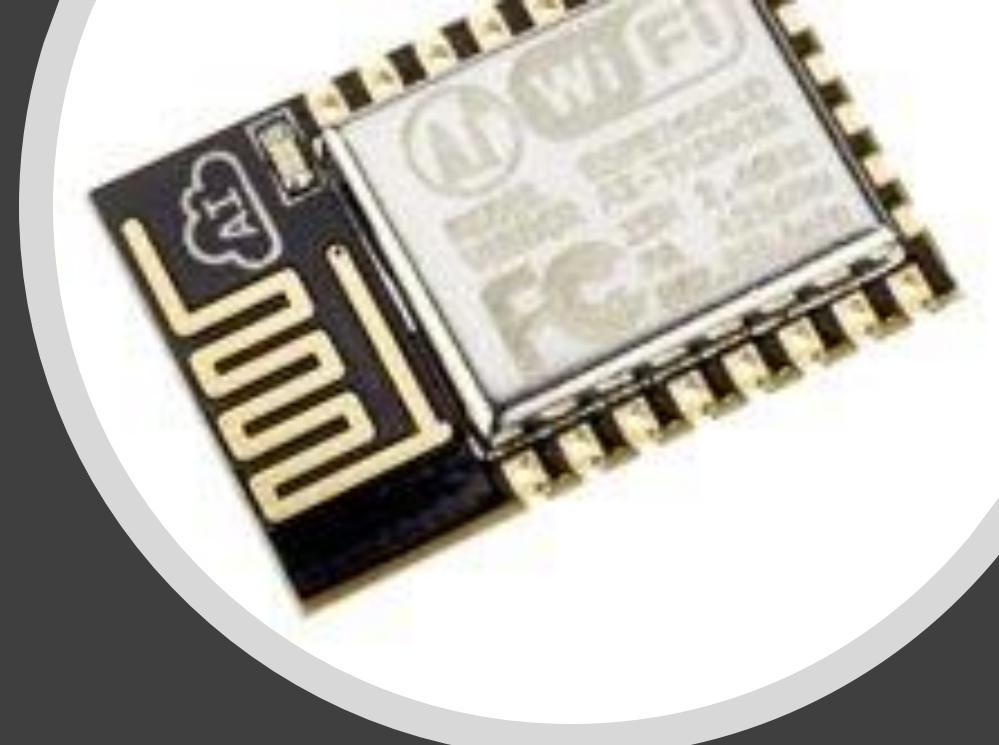


Arduino



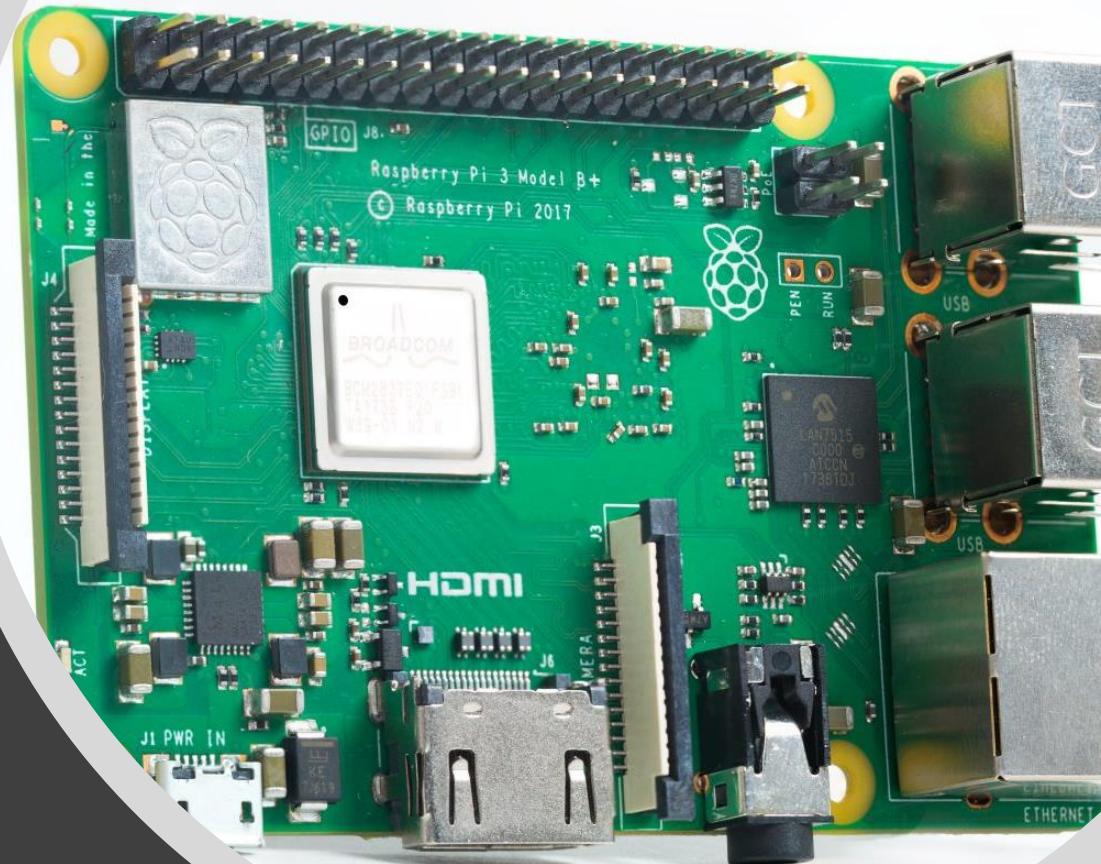
ESP8266

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz†
- 64 KiB of instruction RAM, 96 KiB of data RAM
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi
 - Integrated TR switch, balun, LNA, power amplifier and matching network
 - WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI
- I²C (software implementation)
- I²S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC)



Raspberry Pi

- SoC: Broadcom BCM2837
- CPU: 4x ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Storage: microSD
- GPIO: 40-pin header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)



LinkIt ONE

All-in-One connectivity: Supports GSM, GPRS, Wi-Fi, Bluetooth (2.1 SPP and 4.0 GATT profiles), GNSS (GPS, GLONASS and BeiDou), Audio out and SD card connectors

Pin-out similar to Arduino UNO, including Digital IO, Analog IO, PWM, I2C, SPI and UART

Open hardware board reference design including schematic, layouts and pin details



A stylized illustration of a man with dark hair and glasses, wearing a grey suit jacket over a green shirt. He is sitting at a desk, looking down at a black computer keyboard. The background is a solid green.

Hands ON

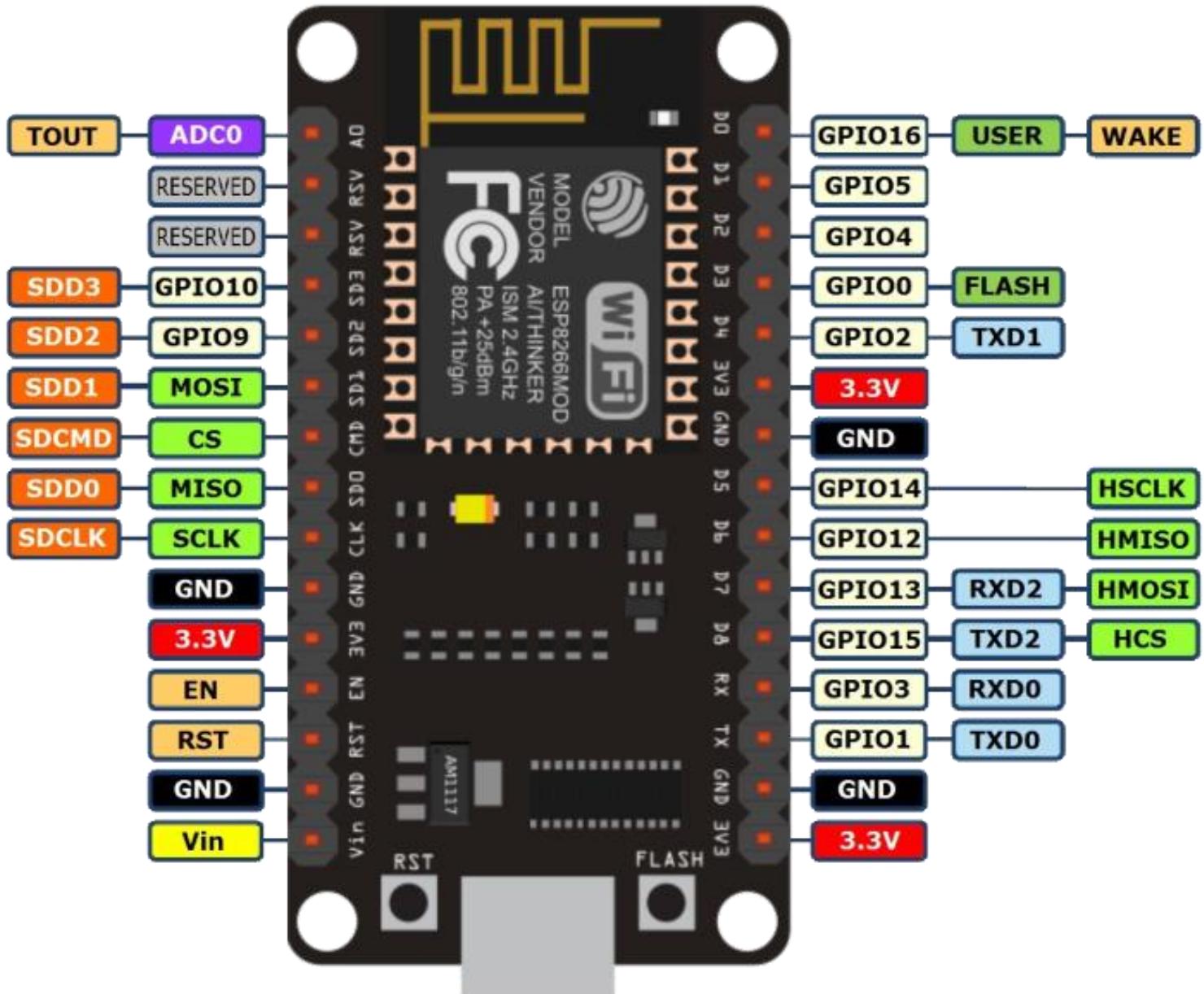


nodeMCU

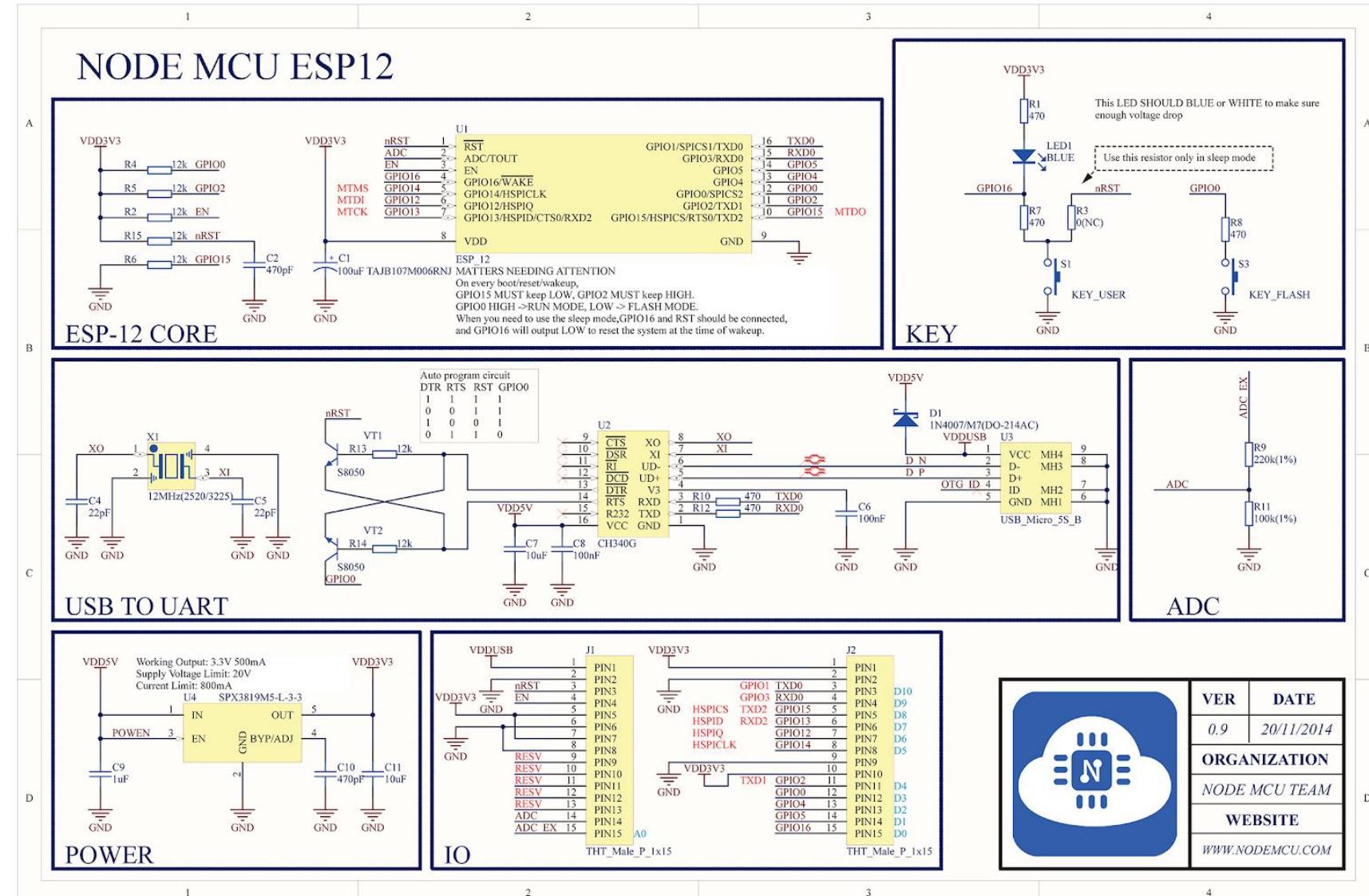
A nodeMCU criada com base no módulo ESP 12E, facilita o processo de programação do ESP8266 por já possuir onboard, o conversor USB serial, regulador de tensão e pino de I/O para conexão em protoboard. Hoje você programa uma placa com ESP8266 diretamente na IDE do Arduino, como se fosse um.

pinout

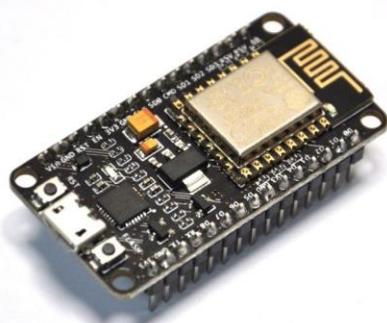
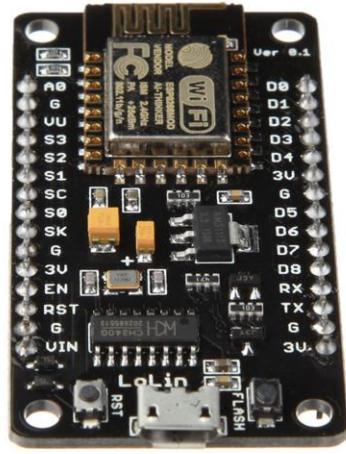
- A placa ainda possui um LED de uso geral conectado ao GPIO_16, um LED no módulo ESP 12E (GPIO_2), e botões RST (Reset) e FLASH (gravação do programa).



Esquemático



O projeto é open hardware e você pode acessar os arquivos no github: <https://github.com/nodemcu/nodemcu-devkit-v1.0>



Conversor USB Serial

Existem versões da placa que usam diferentes chips para comunicação serial. A seguir você acessa os drivers para os dois chips mais comuns:

- [CH340](#)
- [CP2102](#)

Instalar a Arduino IDE

A IDE pode ser baixada gratuitamente no [site do Arduino](#), onde pode ser escolhida a melhor opção de download conforme o sistema operacional que você utiliza.



ARDUINO 1.8.7
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

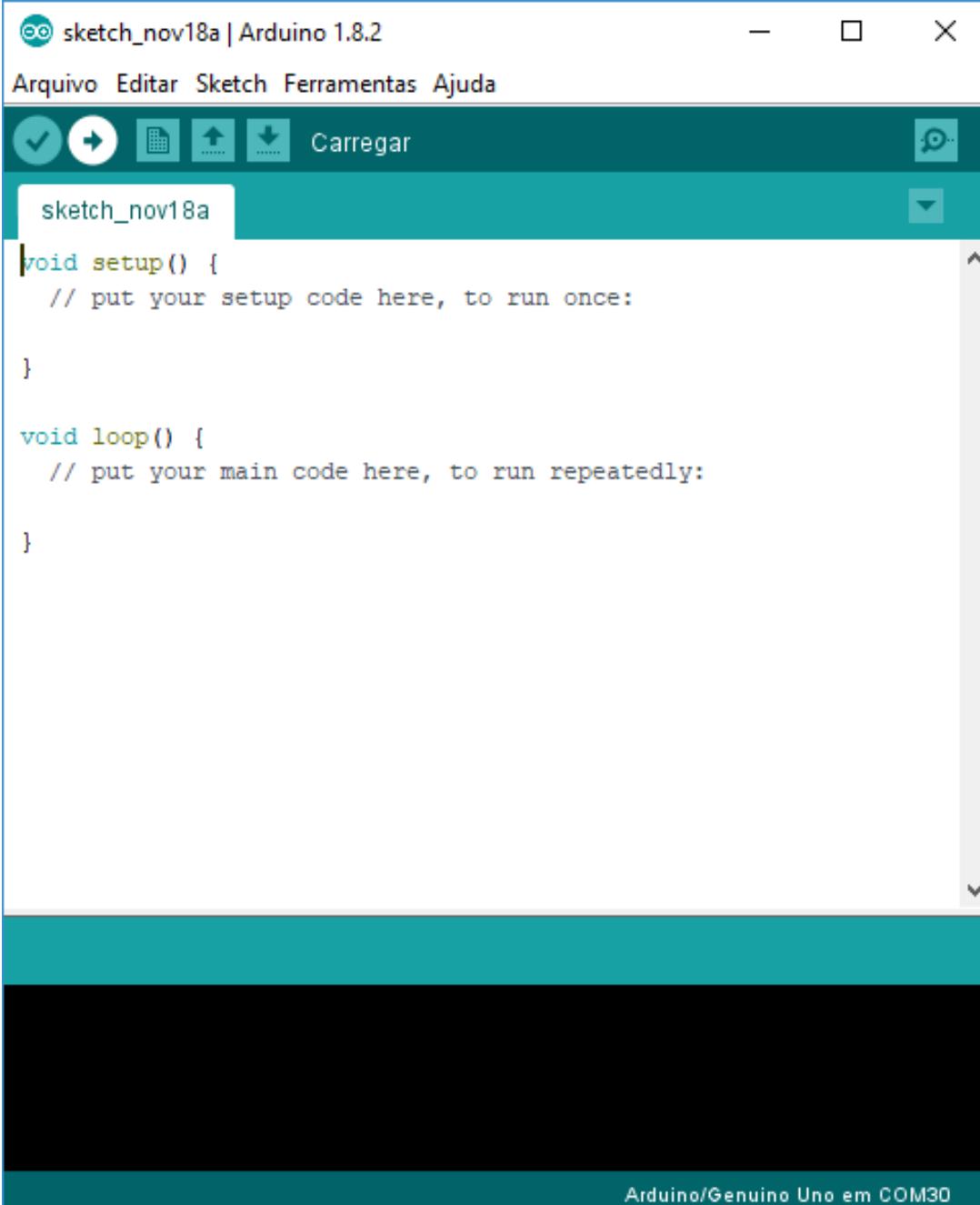
Windows app Requires Win 8.1 or 10
[Get !\[\]\(c4aaed3b5c356fb84b11eeae3fb16d4c_img.jpg\)](#)

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Abra a IDE do
Arduino



The image shows a screenshot of the Arduino IDE version 1.8.2. The window title is "sketch_nov18a | Arduino 1.8.2". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The toolbar contains icons for saving, loading, and uploading sketches, along with a "Carregar" button. A dropdown menu is open, showing "sketch_nov18a" as the selected item. The main code editor area displays the following code:

```
void setup() {
  // put your setup code here, to run once:

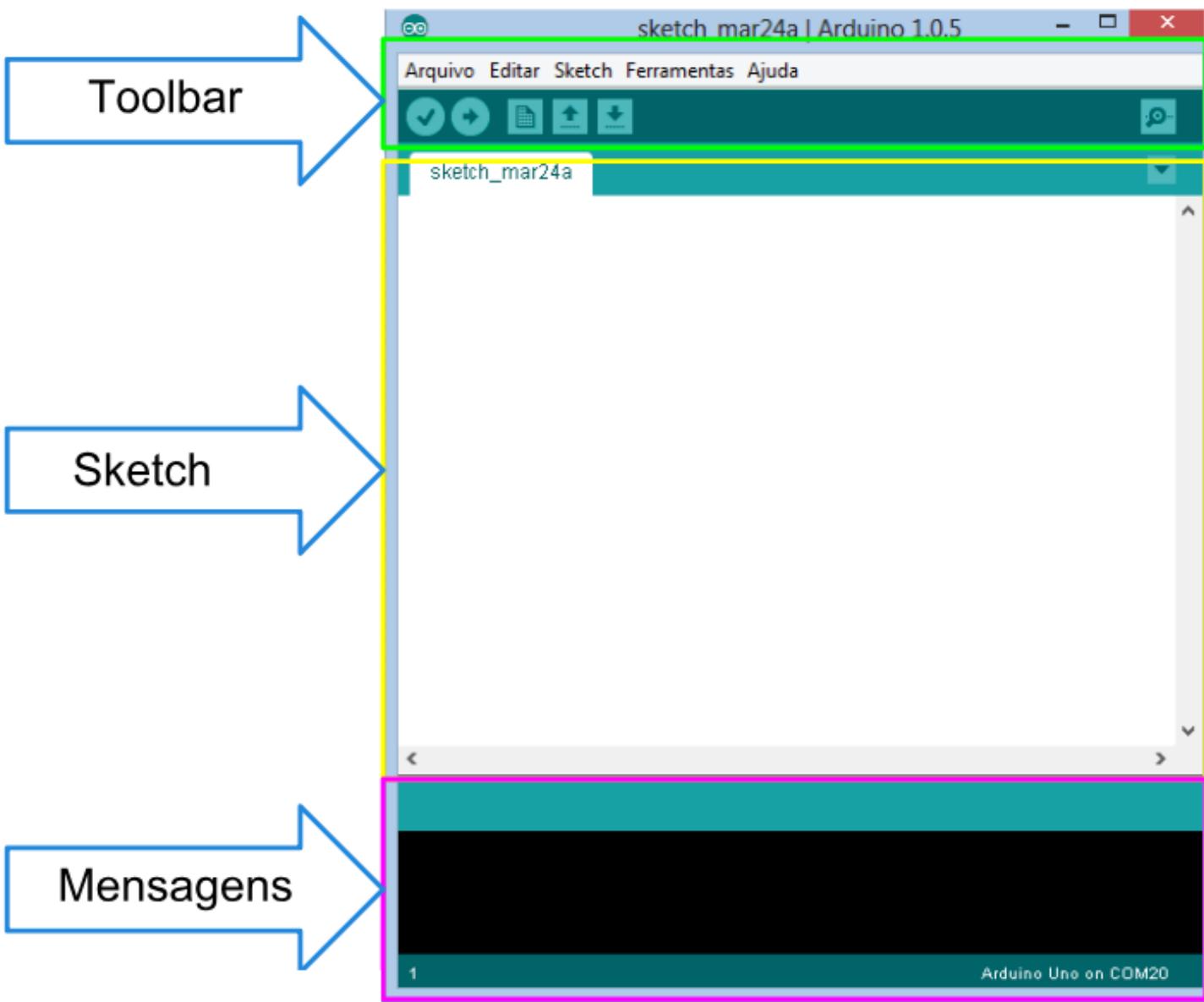
}

void loop() {
  // put your main code here, to run repeatedly:

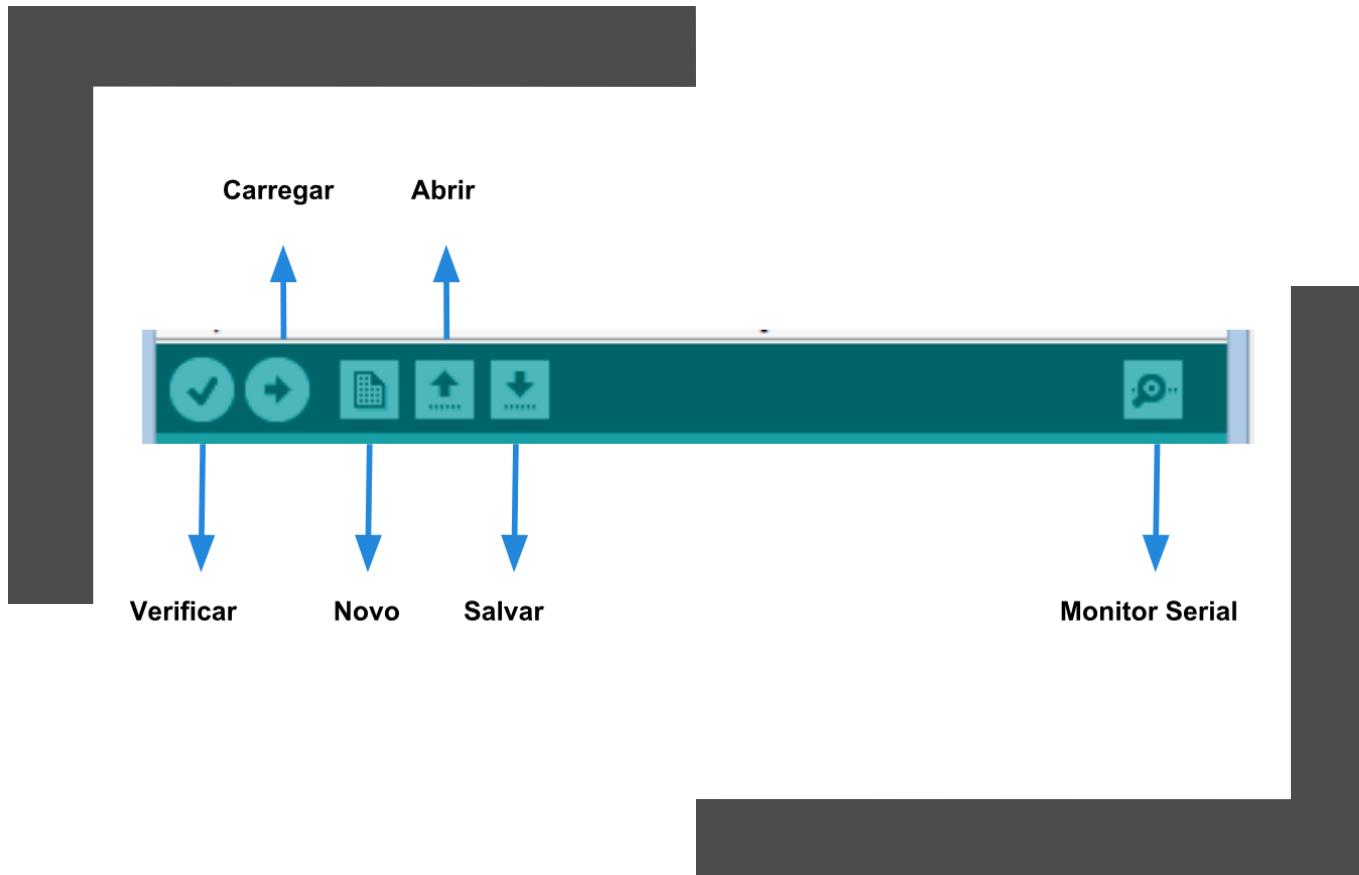
}
```

At the bottom of the IDE window, there is a status bar with the text "Arduino/Genuino Uno em COM30".

Conheça a
IDE



Toolbar



- **Verify**

Verifica se existe erro no código digitado.

- **Upload**

Compila o código e grava na placa Arduino se corretamente conectada;

- **New**

Cria um novo *sketch* em branco.

- **Open**

Abre um *sketch*, presente no sketchbook.

- **Save**

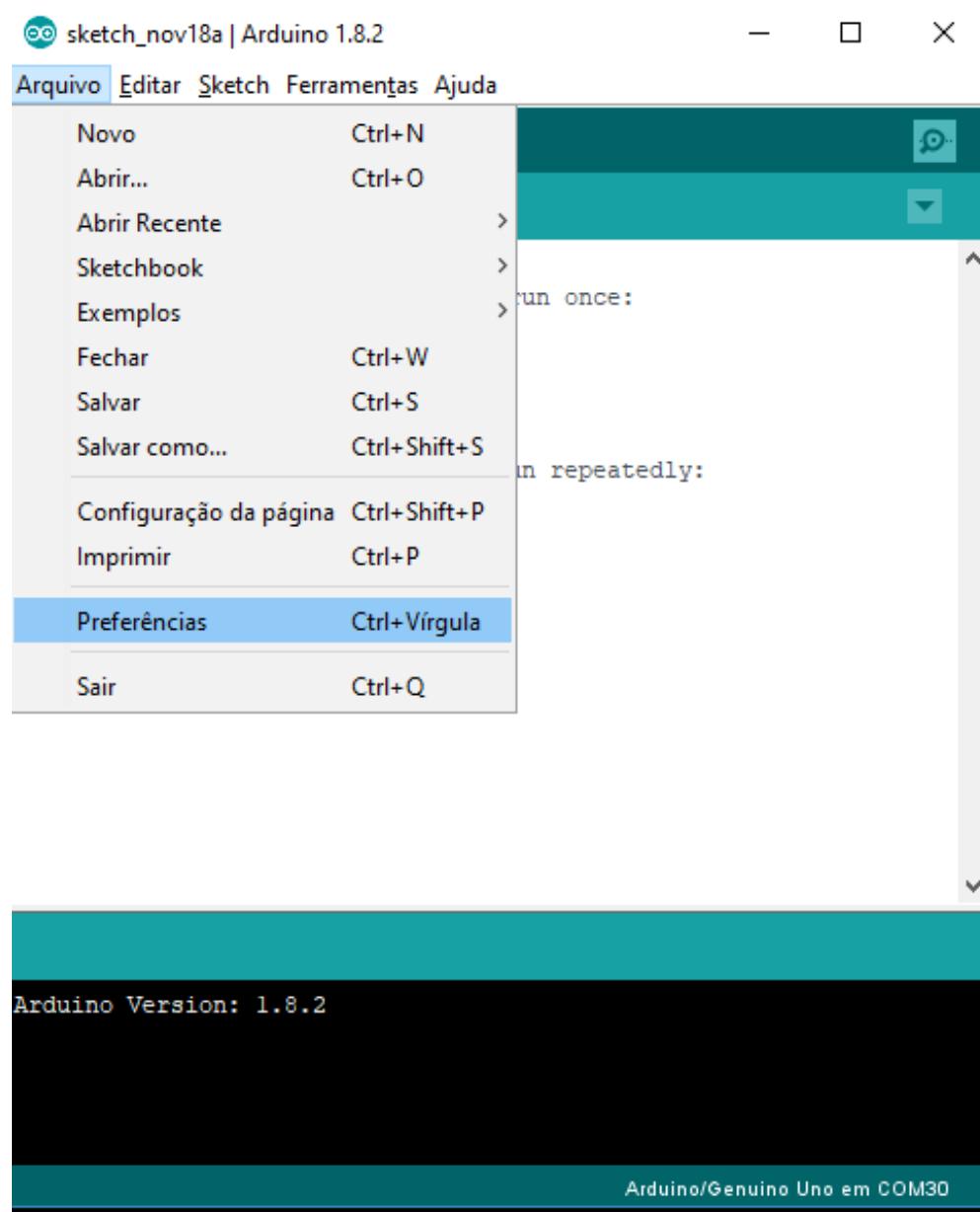
Salva o *sketch* ativo

- **Serial monitor**

Abre o monitor serial.

Configuração da IDE Arduino para programar o ESP8266

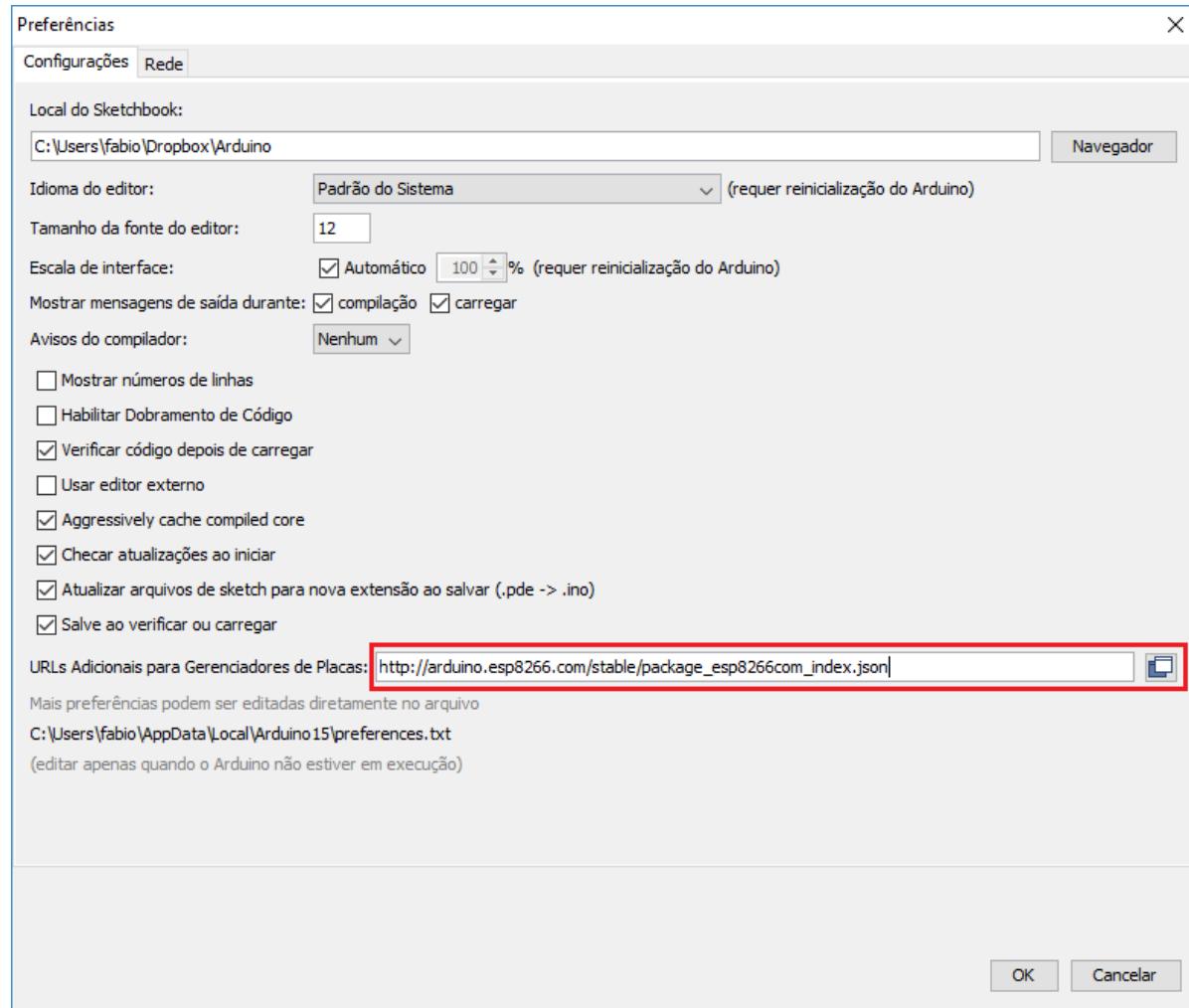
Na IDE acesse **Arquivo > Preferências**



Adicione a URL

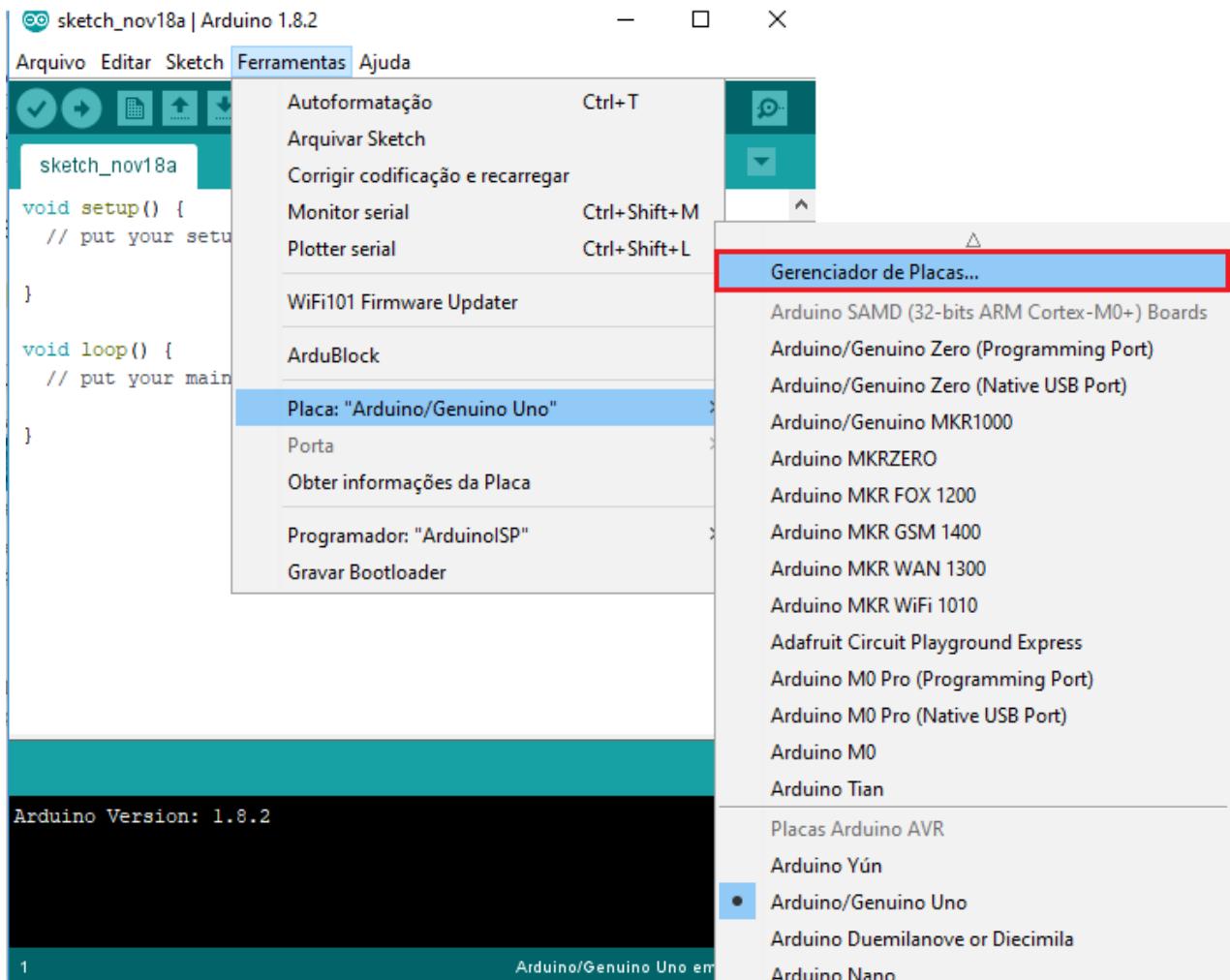
Na janela de configurações de preferências, digite a URL abaixo no campo “**URLs adicionais de Gerenciadores de Placas**”:

http://arduino.esp8266.com/stable/package_esp8266com_index.json



Acesse o gerencidor de placas

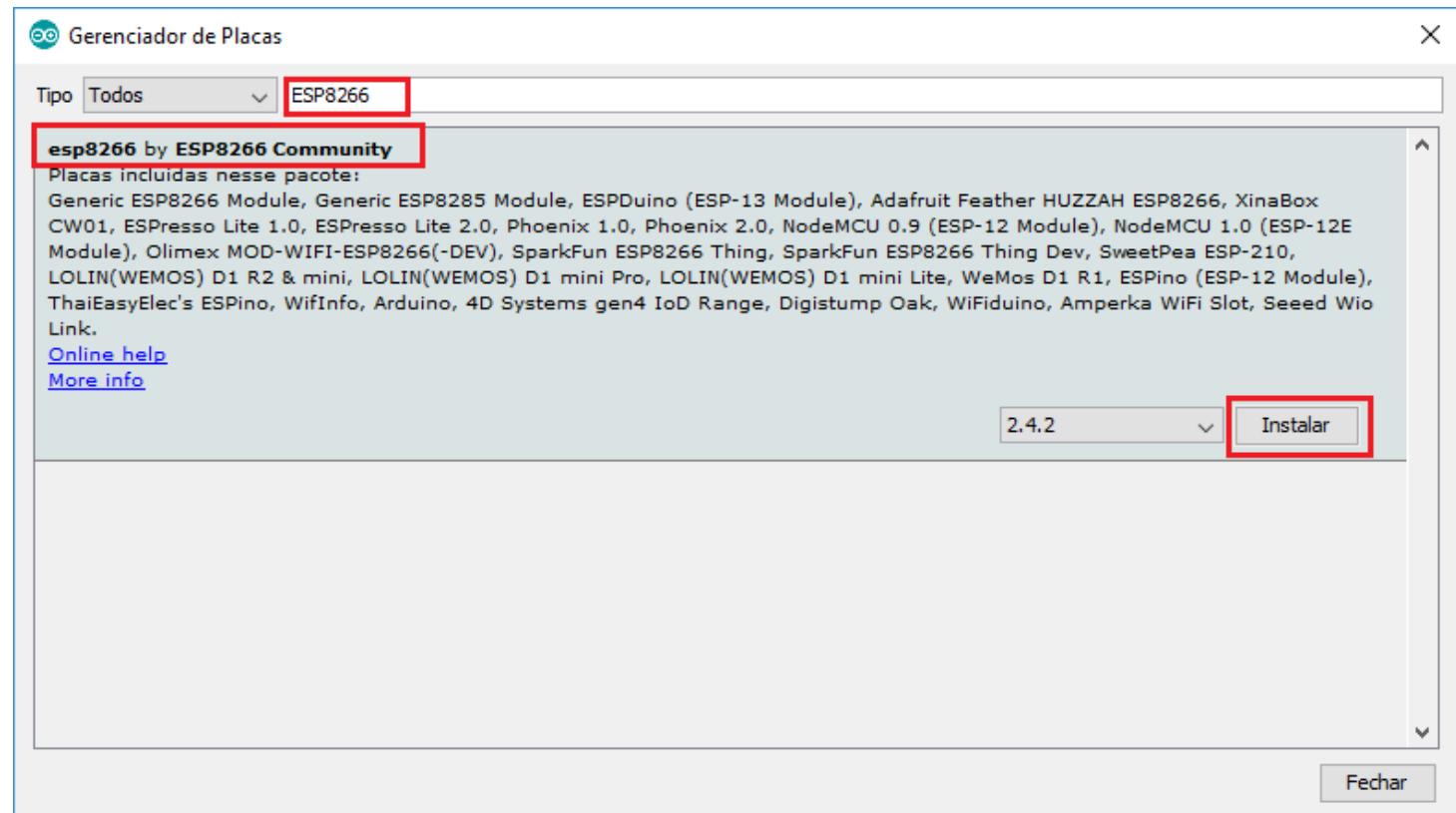
Agora acesse
Ferramentas > placas >
Gerenciadores de
Placas:



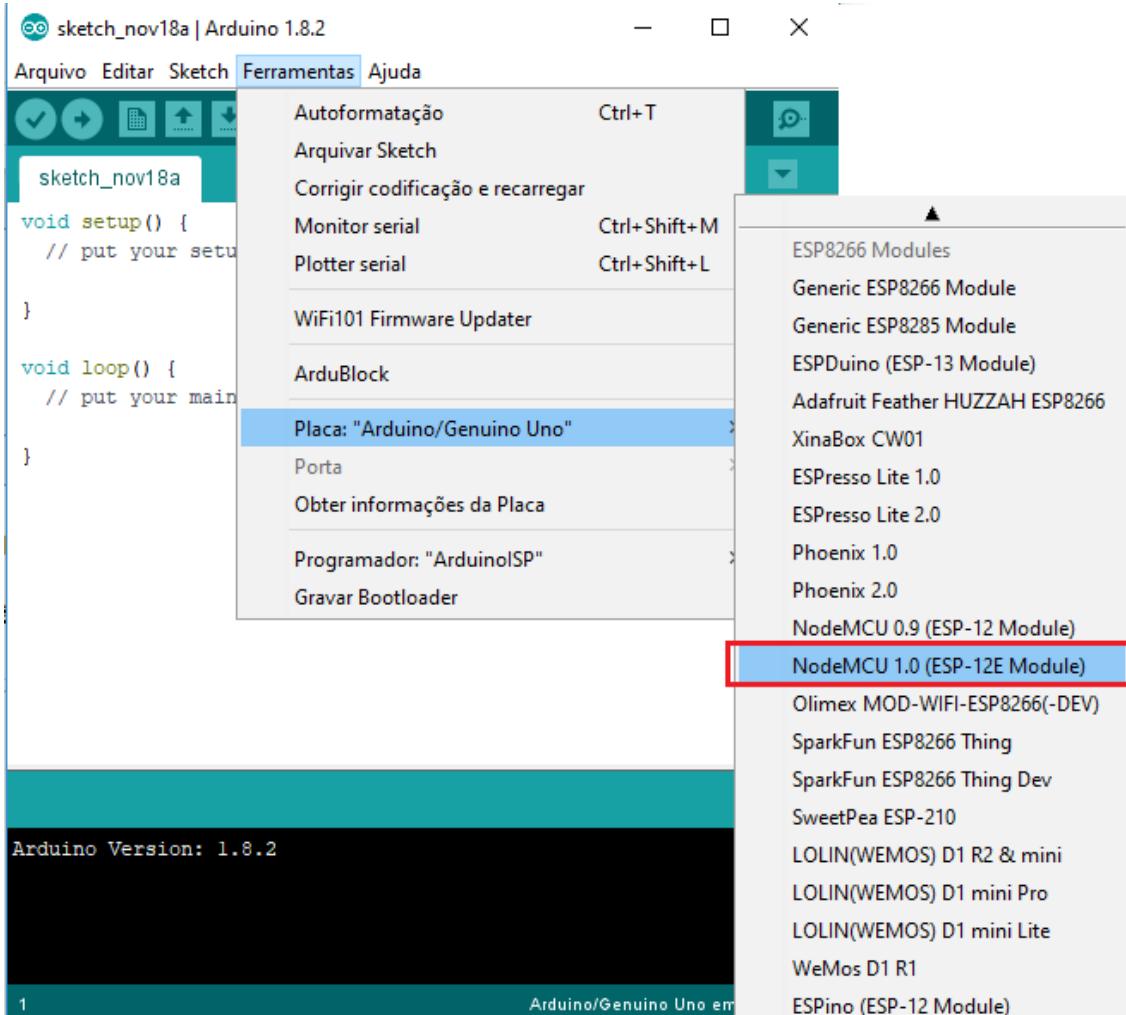
Instale o
pacote

Procure por ESP8266.

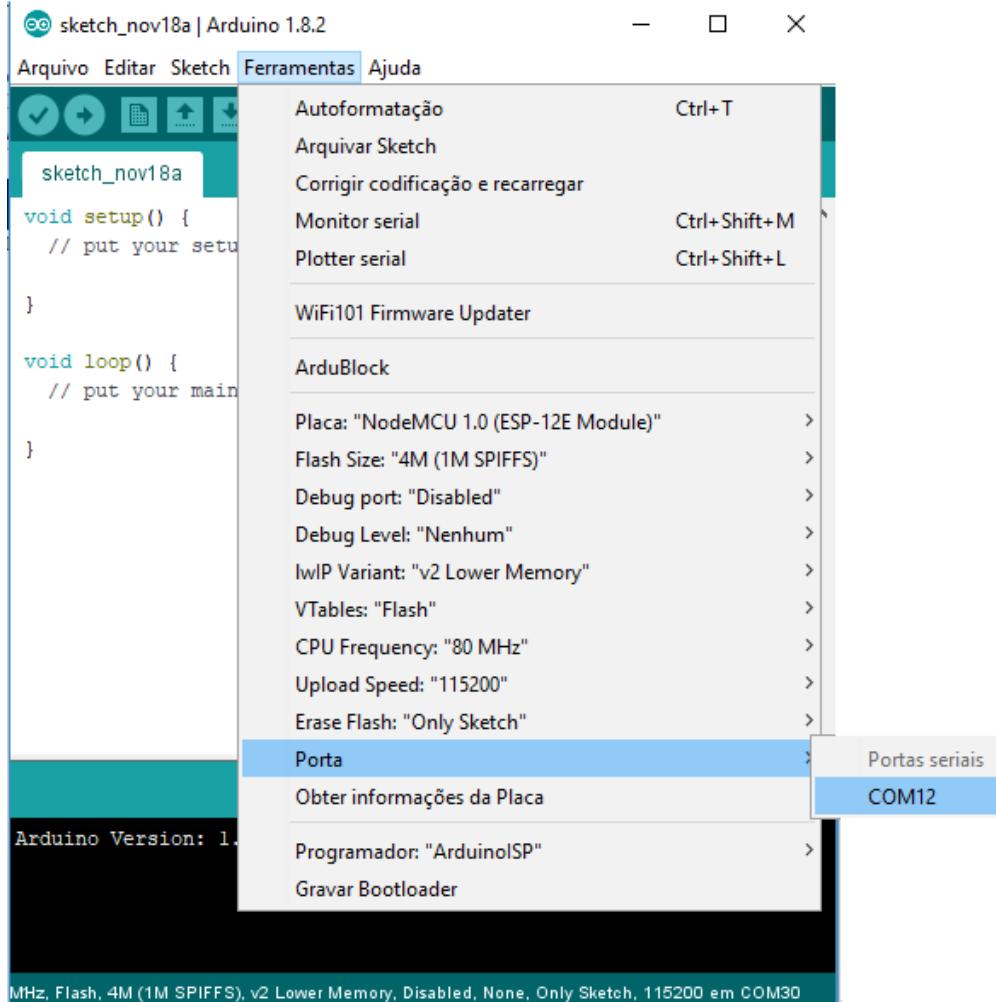
Encontrando a opção “esp8266
by ESP8266 Community”
clique em instalar:



Selecione a nodeMCU



Selecione a
porta



Pronto a IDE está configurada para programar a sua placa NodeMCU

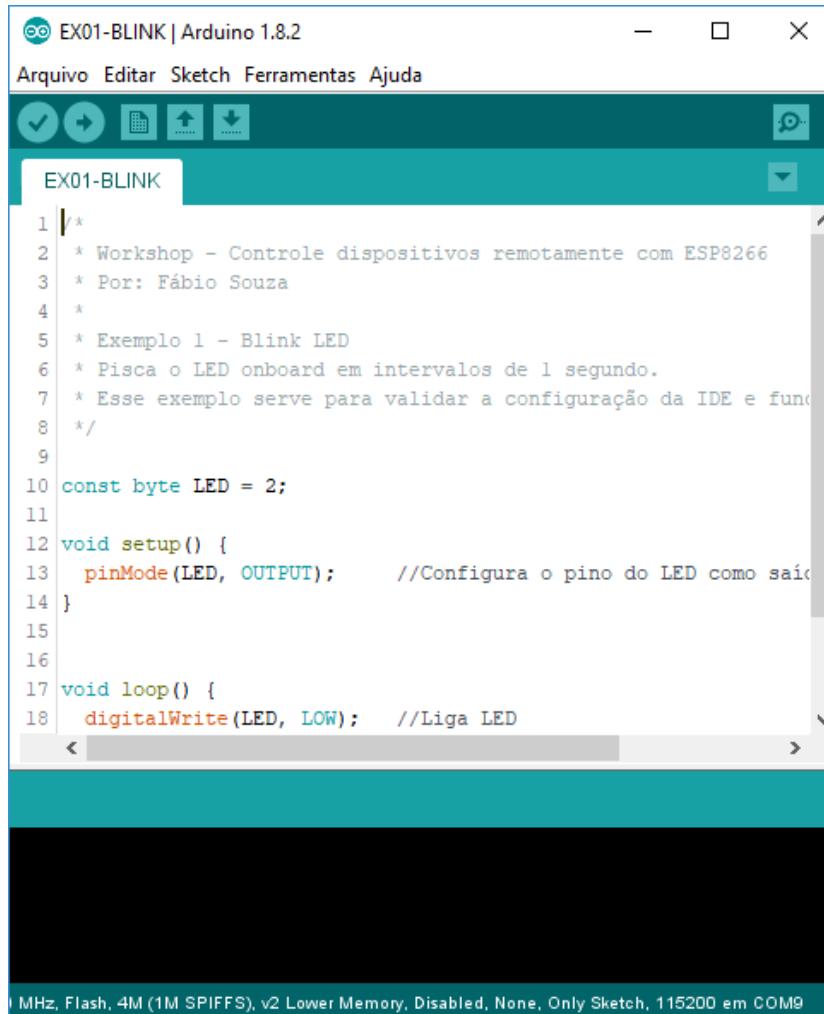
Hello,

World_

Blink

Abra o Exemplo 1

Carregue o código para
a placa e verique se o
LED está piscando.



The screenshot shows the Arduino IDE interface with the title bar "EX01-BLINK | Arduino 1.8.2". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". Below the menu is a toolbar with icons for file operations. The main window displays the code for "EX01-BLINK". The code is as follows:

```
1 // 
2 * Workshop - Controle dispositivos remotamente com ESP8266
3 * Por: Fábio Souza
4 *
5 * Exemplo 1 - Blink LED
6 * Piscia o LED onboard em intervalos de 1 segundo.
7 * Esse exemplo serve para validar a configuração da IDE e função
8 */
9
10 const byte LED = 2;
11
12 void setup() {
13     pinMode(LED, OUTPUT);      //Configura o pino do LED como saída
14 }
15
16
17 void loop() {
18     digitalWrite(LED, LOW);    //Liga LED
19 }
```

At the bottom of the IDE, the status bar shows "MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM9".

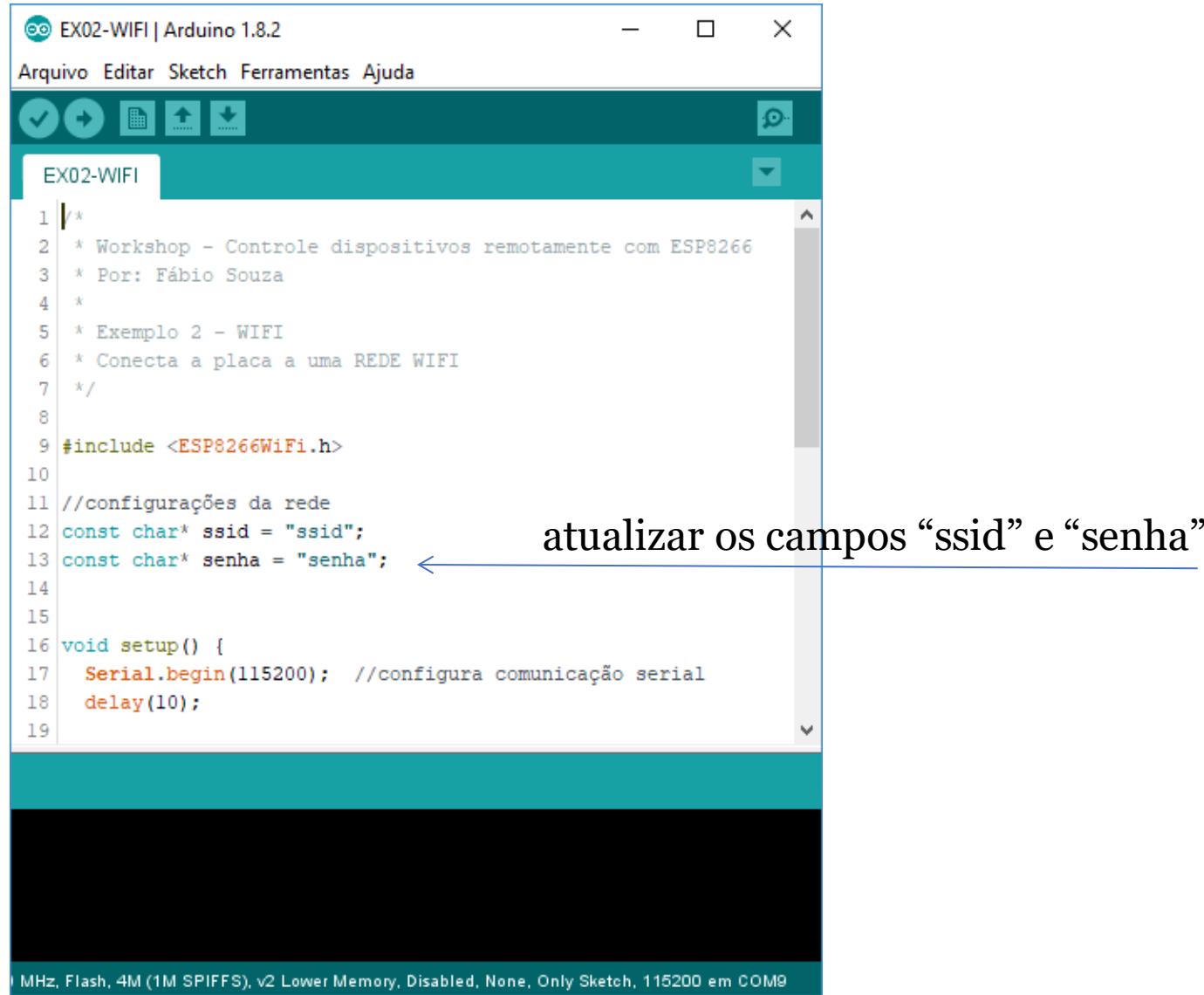


Conexão WIFI

Agora vamos aprender conectar o ESP8266
a rede WIFI

Abra o Exemplo 2

Para conectar nossa placa ao
WIFI vamos usar a
biblioteca ESP8266Wifi.h



EX02-WIFI | Arduino 1.8.2

Arquivo Editar Sketch Ferramentas Ajuda

EX02-WIFI

```
1 // *  
2 * Workshop - Controle dispositivos remotamente com ESP8266  
3 * Por: Fábio Souza  
4 *  
5 * Exemplo 2 - WIFI  
6 * Conecta a placa a uma REDE WIFI  
7 */  
8  
9 #include <ESP8266WiFi.h>  
10  
11 //configurações da rede  
12 const char* ssid = "ssid";  
13 const char* senha = "senha"; ←  
14  
15  
16 void setup() {  
17   Serial.begin(115200); //configura comunicação serial  
18   delay(10);  
19 }
```

MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 em COM9

atualizar os campos “ssid” e “senha”

Veja se sua placa foi conectada a rede

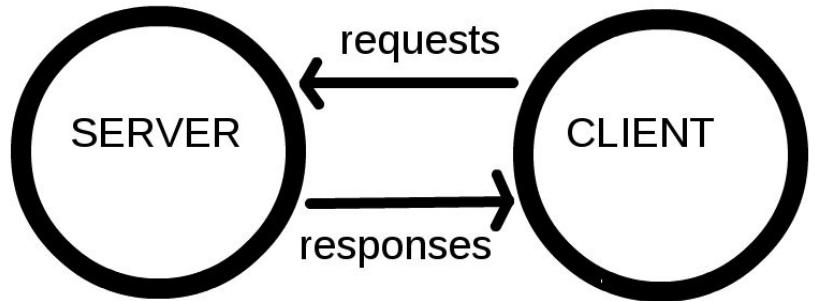
```
COM12
|
| Enviar
| (48 00) | 01 0 | 00000001 00 b | 0000 p 00 | 0" 00 # 000 04 on 00000b 0p 0# 0 (1p 0n 00000001 00000b 0o 0 | 04 0000 # 000 04 001 0000 on 04` 0000 (0_000b 000010 (00 00b 000108 # b 000001 ` 000N00 Conectan
| .....
| WiFi conectado
| Endereço IP:
| 192.168.0.21
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\fabio>ping 192.168.0.21

Disparando 192.168.0.21 com 32 bytes de dados:
Resposta de 192.168.0.21: bytes=32 tempo=51ms TTL=255
Resposta de 192.168.0.21: bytes=32 tempo=74ms TTL=255
Esgotado o tempo limite do pedido.
Resposta de 192.168.0.21: bytes=32 tempo=2ms TTL=255

Estatísticas do Ping para 192.168.0.21:
  Pacotes: Enviados = 4, Recebidos = 3, Perdidos = 1 (25% de
            perda),
Aproximar um número redondo de vezes em milissegundos:
  Mínimo = 2ms, Máximo = 74ms, Média = 42ms

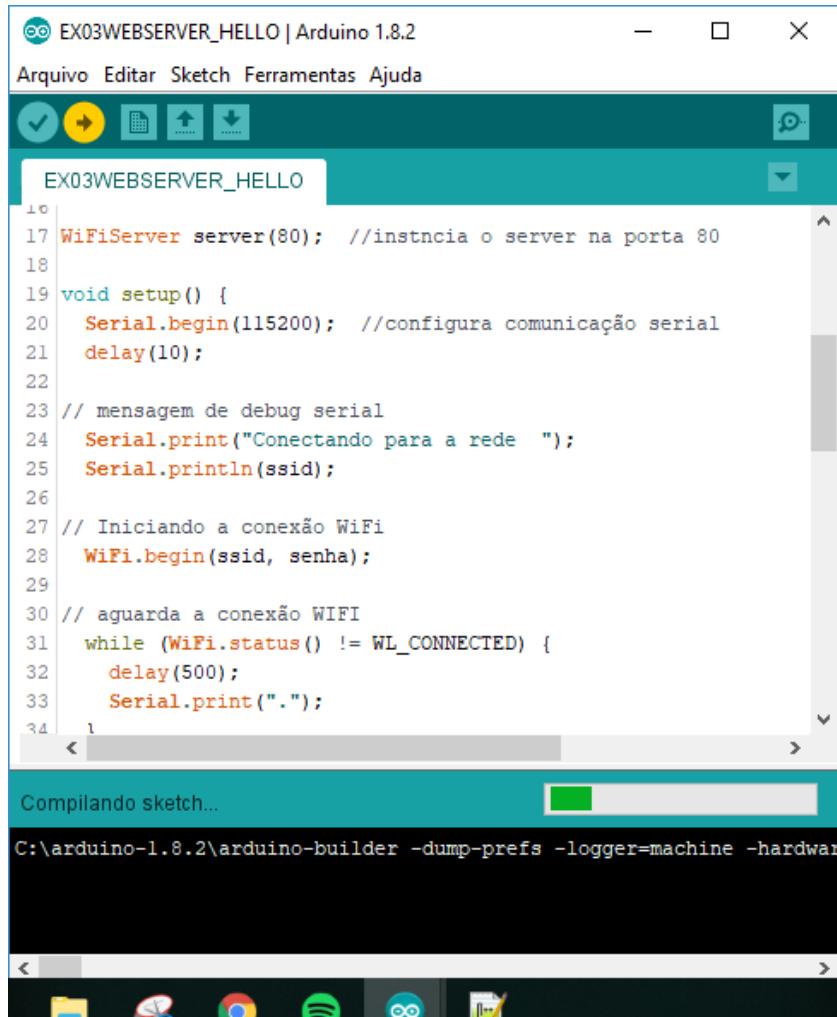
C:\Users\fabio>
```



ESP8266
como
Webserver

Abra o exemplo 3

Faça as configurações do WIFI e carregue para a placa



The screenshot shows the Arduino IDE interface with the title bar "EX03WEB SERVER HELLO | Arduino 1.8.2". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". Below the menu is a toolbar with icons for file operations. The main area displays the code for "EX03WEB SERVER HELLO". The code initializes a WiFi server at port 80, begins serial communication at 115200 baud, and prints debug messages to the serial port while waiting for WiFi connection. A progress bar at the bottom indicates "Compilando sketch..." (Compiling sketch...). The status bar at the bottom shows the command: "C:\arduino-1.8.2\arduino-builder -dump-prefs -logger=machine -hardware".

```
EX03WEB SERVER HELLO | Arduino 1.8.2
Arquivo Editar Sketch Ferramentas Ajuda
EX03WEB SERVER HELLO
16
17 WiFiServer server(80); //instancia o server na porta 80
18
19 void setup() {
20   Serial.begin(115200); //configura comunicação serial
21   delay(10);
22
23 // mensagem de debug serial
24   Serial.print("Conectando para a rede ");
25   Serial.println(ssid);
26
27 // Iniciando a conexão WiFi
28   WiFi.begin(ssid, senha);
29
30 // aguarda a conexão WiFi
31   while (WiFi.status() != WL_CONNECTED) {
32     delay(500);
33     Serial.print(".");
34   }
Compilando sketch...
C:\arduino-1.8.2\arduino-builder -dump-prefs -logger=machine -hardware
```

Abra o terminal
serial e veja qual
o IP atribuído a
placa:



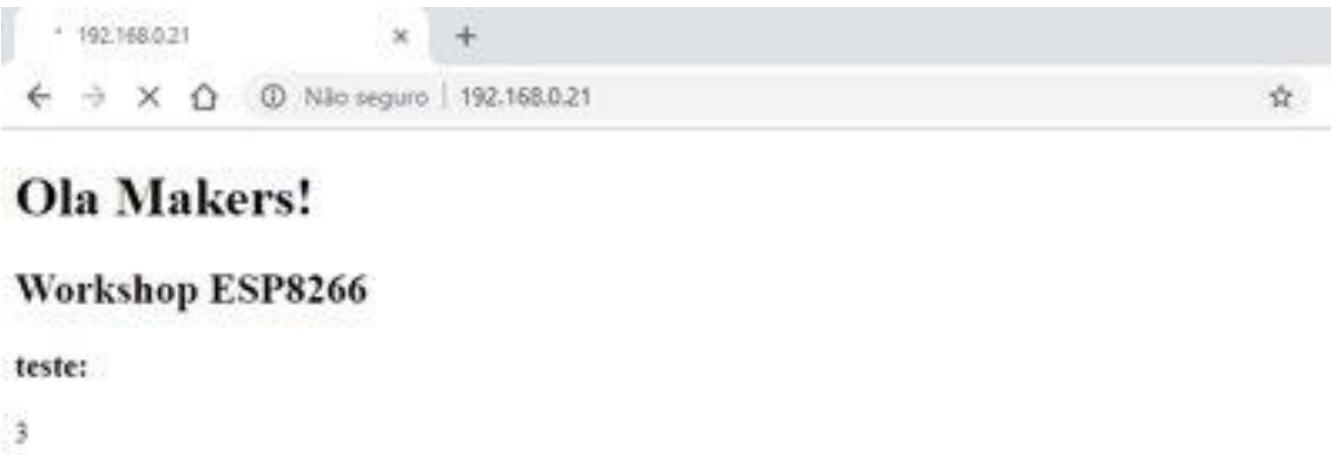
```
COM12
Enviar
r$10000000|000000100p0(00000c00Nn1on000080p0b1 0(lp0000000100000c0n0|0d00000p000no00$00$`00200o$`000or0_00"00010r00000b00010bBb0000d`000o00Conect
.....Servidor inicializado!
WiFi conectado
Endereço IP:
192.168.0.21
```

< >

Auto-rolagem

Nova-linha 115200 velocidade

Digite o IP da
placa em um
Navegador



Pressione F5 e veja o que acontece com
o valor abaixo de teste

Para atualizar o navegador automaticamente

Adicione o seguinte trecho no HTML

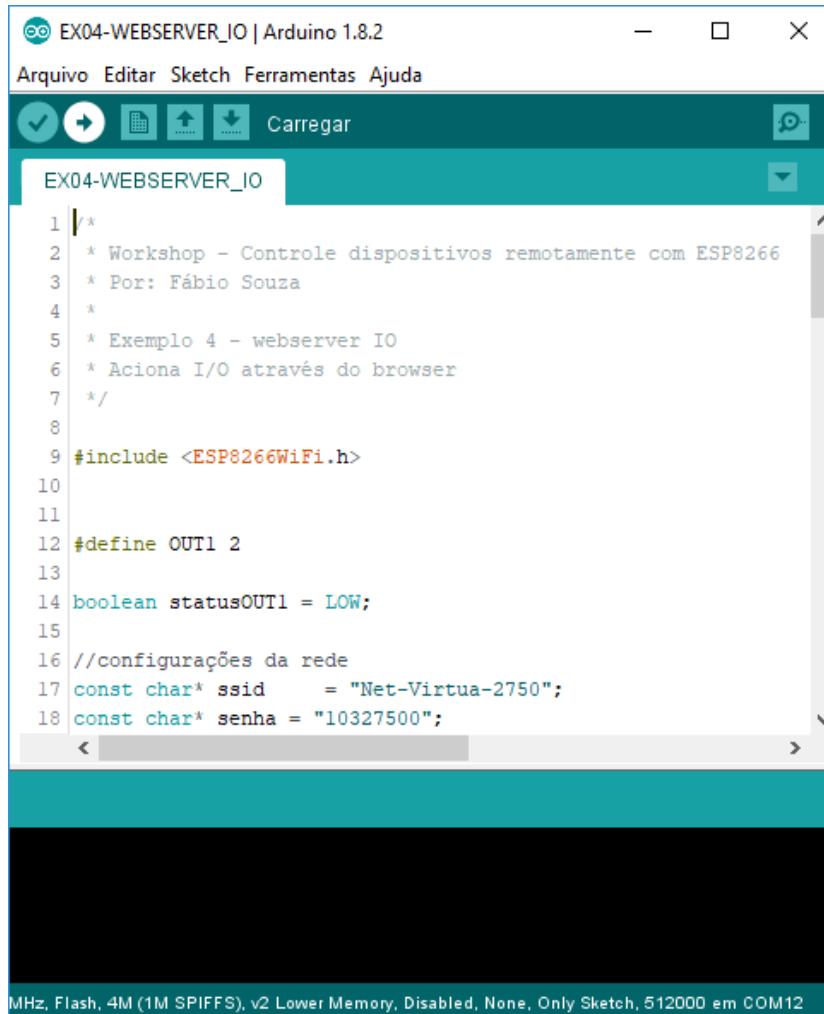
```
"HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE  
HTML>\r\n<html><meta http-equiv='refresh' content='3'>\r\n";
```

Atualize o código na placa e veja o que acontece no navegador.

Abra o Exemplo 4

Agora vamos acionar o LEDs.

Faça as configurações do
WIFI e carregue para a
placa



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** EX04-WEBSERVER_IO | Arduino 1.8.2
- Menu Bar:** Arquivo, Editar, Sketch, Ferramentas, Ajuda
- Toolbar:** Includes icons for Save, Load, Open, Upload, Download, and a refresh button labeled "Carregar".
- Sketch Area:** Displays the code for "EX04-WEBSERVER_IO".
- Code Content:**

```
1 /*  
2  * Workshop - Controle dispositivos remotamente com ESP8266  
3  * Por: Fábio Souza  
4  *  
5  * Exemplo 4 - webserver IO  
6  * Aciona I/O através do browser  
7  */  
8  
9 #include <ESP8266WiFi.h>  
10  
11  
12 #define OUT1 2  
13  
14 boolean statusOUT1 = LOW;  
15  
16 //configurações da rede  
17 const char* ssid      = "Net-Virtua-2750";  
18 const char* senha    = "10327500";
```
- Status Bar:** Shows "MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 512000 em COM12".

Abra o terminal
serial e veja qual
o IP atribuído a
placa:



```
COM12
Enviar
r$10000000|000000100p0(00000c00Nn1on000080p0b1 0(lp0000000100000c0n0|0d00000p000no00$00$`00200o$`000or0_00"00010r00000b00010bBb0000d`000o00Conect
.....Servidor inicializado!
WiFi conectado
Endereço IP:
192.168.0.21
```

< >

Auto-rolagem

Nova-linha 115200 velocidade

Digite o IP da
placa em um
Navegador



Clique no botão e veja se o estado do
LED muda na placa.

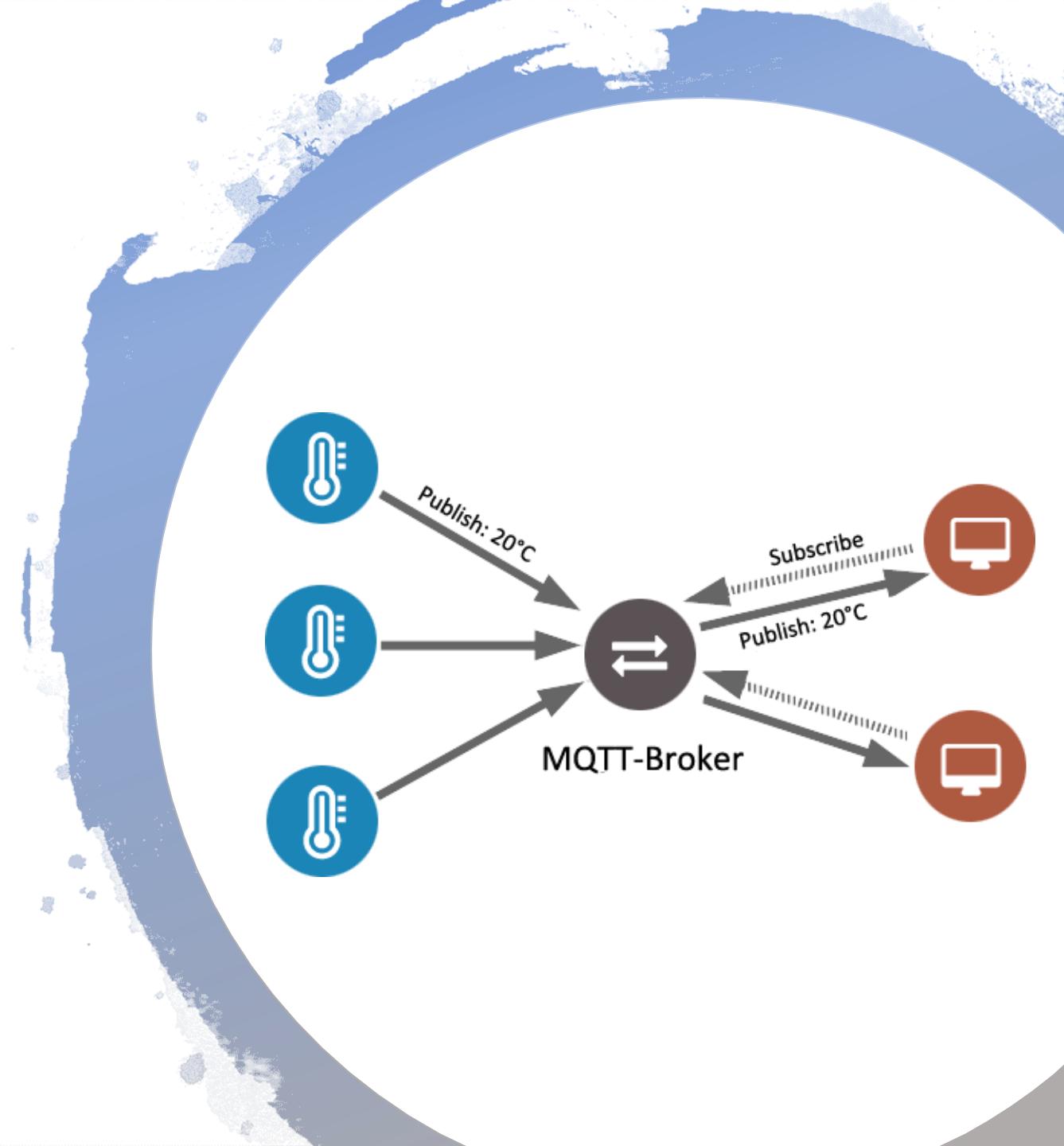
CHALLENGE



Desafio

Adicione mais saídas e
sensores ao web server

MQTT (Message Queue Telemetry Transport)



MQTT

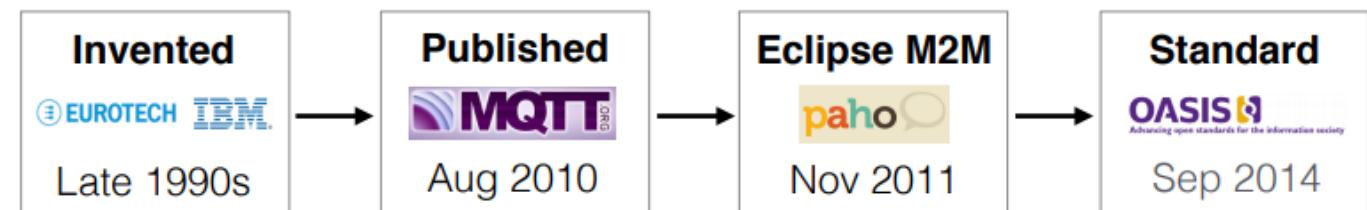
Criado pela IBM no final da década de 90, é um protocolo de mensagens leve, utilizado para comunicação Machine to Machine.

MQTT

a lightweight protocol for IoT messaging

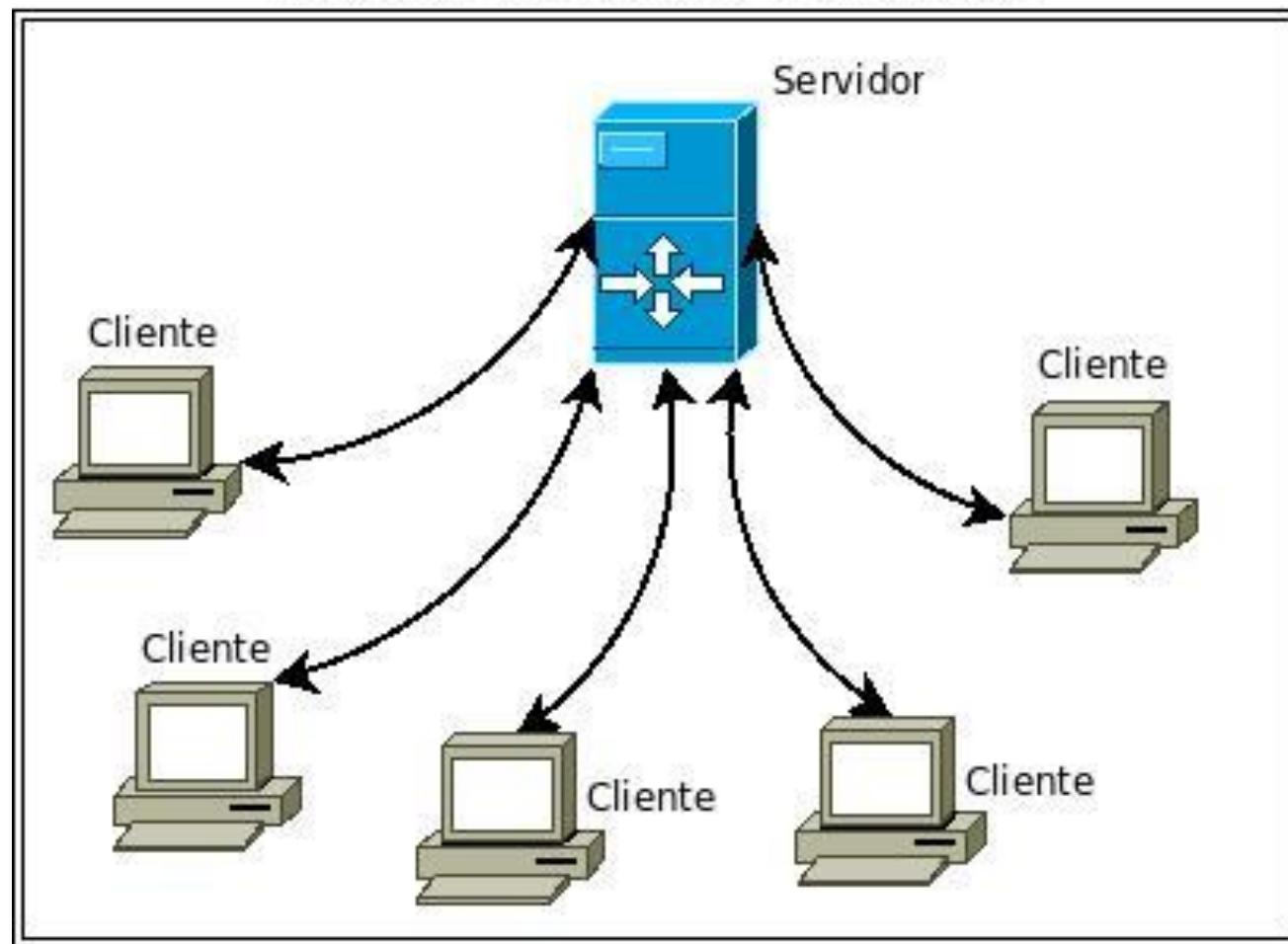
- **open**
- **lightweight**
- **reliable**
- **simple**

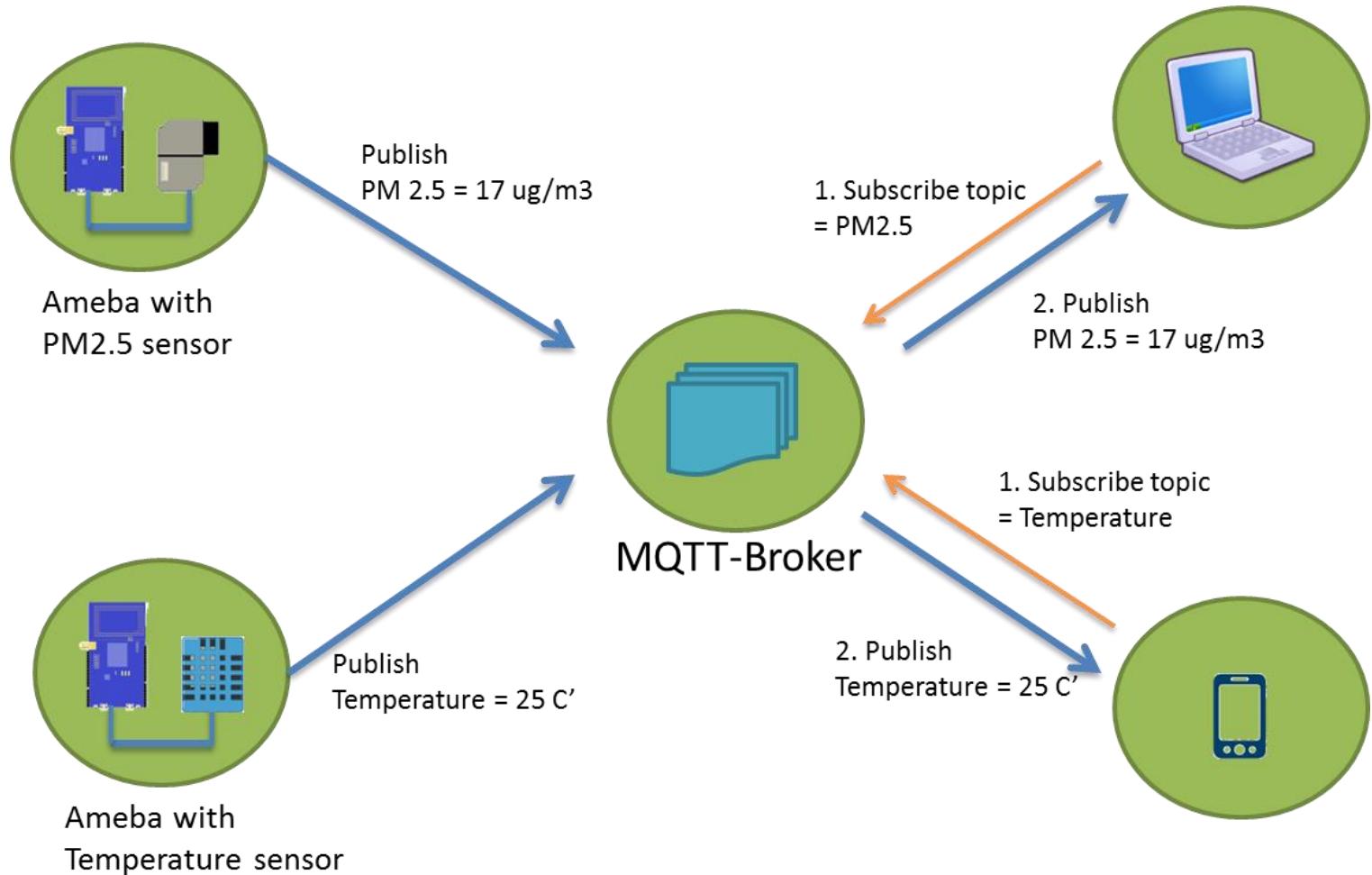
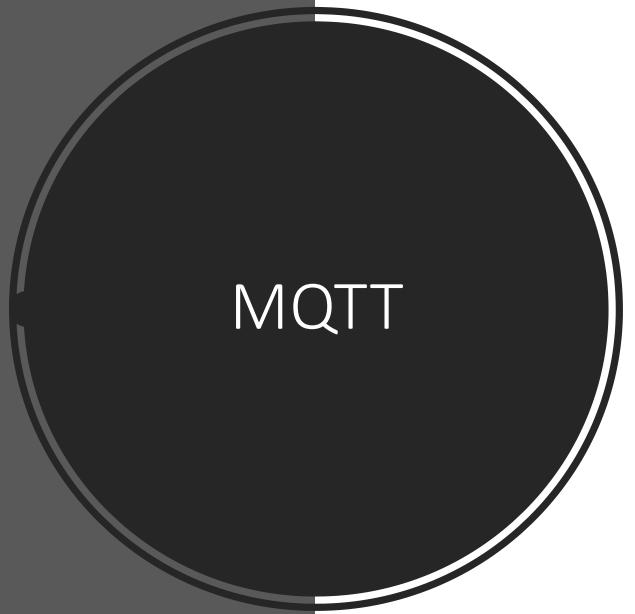
open spec, standard 40+ client implementations
minimal overhead efficient format tiny clients (kb)
QoS for reliability on unreliable networks
43-page spec connect + publish + subscribe



Lembrando
do modelo
Cliente-
Servidor

Modelo Cliente-Servidor





MQTT Publish

area/ID_da_area/sensor/ID_do_sensor/temperatura

area/ID_da_area/sensor/ID_do_sensor/umidade



MQTT
Subscriber

area/10/sensor/+/temperatura

area/20/sensor/#



Qualidade
do serviço

MQTT

Quality of Service for reliable messaging

QoS 0
at most once



PUBLISH

- doesn't survive failures
- never duplicated

QoS 1
at least once



PUBLISH
PUBACK

- survives connection loss
- can be duplicated

**MQTT
Broker**

PUBLISH
PUBREC
PUBREL
PUBCOMP



QoS 2
exactly once

- survives connection loss
- never duplicated



Acionamento de Lâmpada
com ESP8266 e MQTT

Criar uma
conta no
CloudMQTT



<https://www.cloudmqtt.com>

Crie
uma instância
no CloudMQTT

Create new instance

No credit card Please [add a credit card](#) if you want to subscribe to a paid plan

Name

Plan

Data center



Tags

Admins can [manage tag access control.](#)

[Create New Instance](#)

Plan



Cute Cat

See the [plan page](#) to learn about the different plans.

Preencha o nome, escolha o plano Free (Cute Cat) e clique em *Create a New Instance*.

Abra a
instância
criada

The screenshot shows a 'Details' page for a MQTT broker instance. At the top right is a green 'Statistics' button. On the right side, there's a section titled 'Active Plan' featuring a cartoon cat icon and the text 'Cute Cat'. Below it is a red 'Upgrade Instance' button. The main area displays connection settings:

- Server: m13.cloudmqtt.com
- User: [REDACTED]
- Password: [REDACTED]
- Port: 12857
- SSL Port: 22857
- Websockets Port (TLS only): 32857
- Connection limit: 10

Next to each setting is a small circular button with a gear icon, likely for editing or configuration.

Aqui estão as informações necessárias para conexão com o Broker: Server, User, Password e Port. Vamos usar essas informações no código do ESP8266 e no aplicativo MQTT Dash, mais a frente.

Home



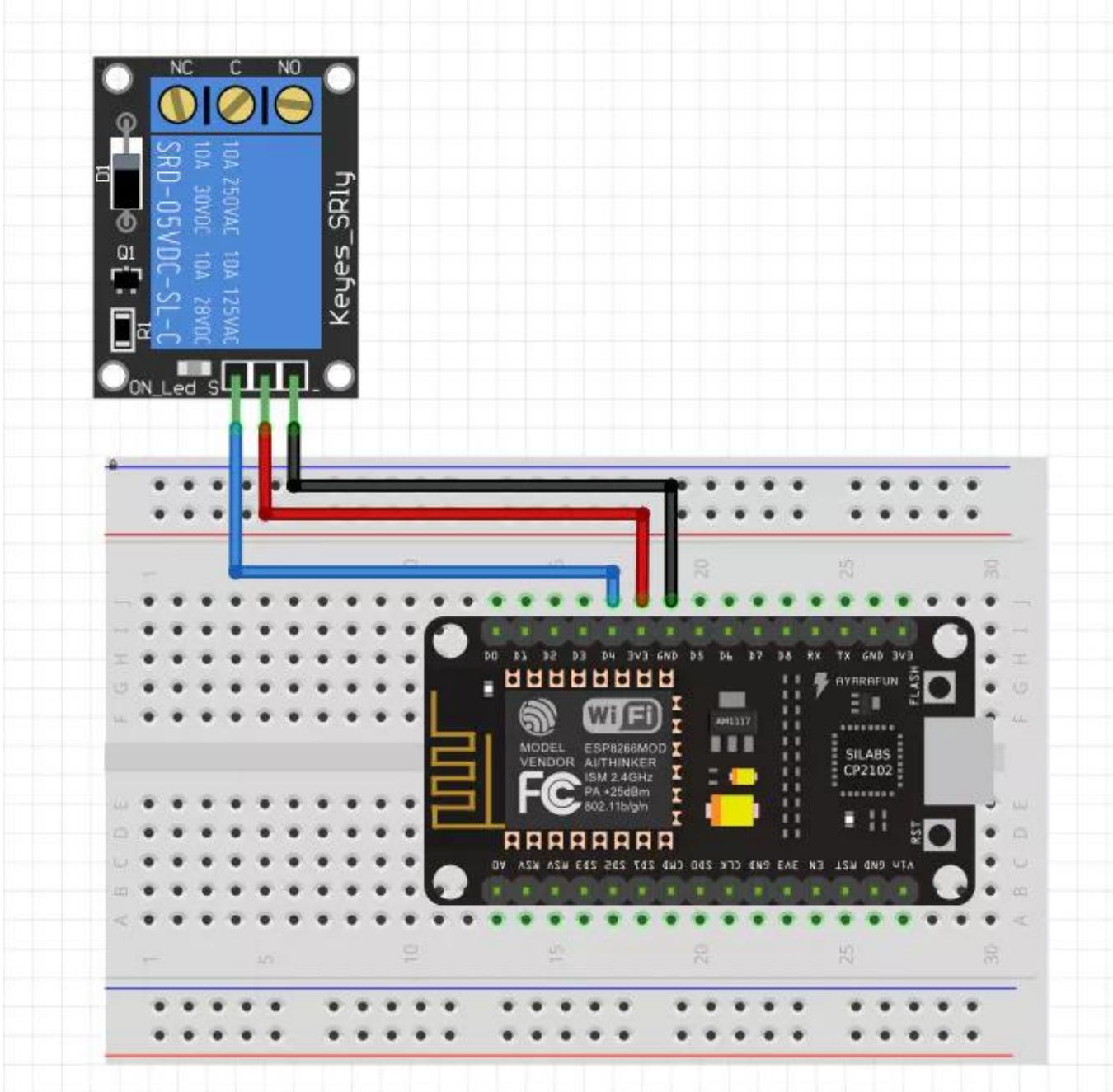
MQTT DASH

O **MQTT Dash** é um dos melhores aplicativos para interface gráfica no smartphone. Possui uma interface agradável, de fácil customização e configuração, sendo um dos melhores aplicativos que já utilizei para esse fim.

Você pode baixá-lo na [Google Play](#) e instalar no seu smartphone.

Deixe no jeito, logo vamos configurar nossa aplicação.

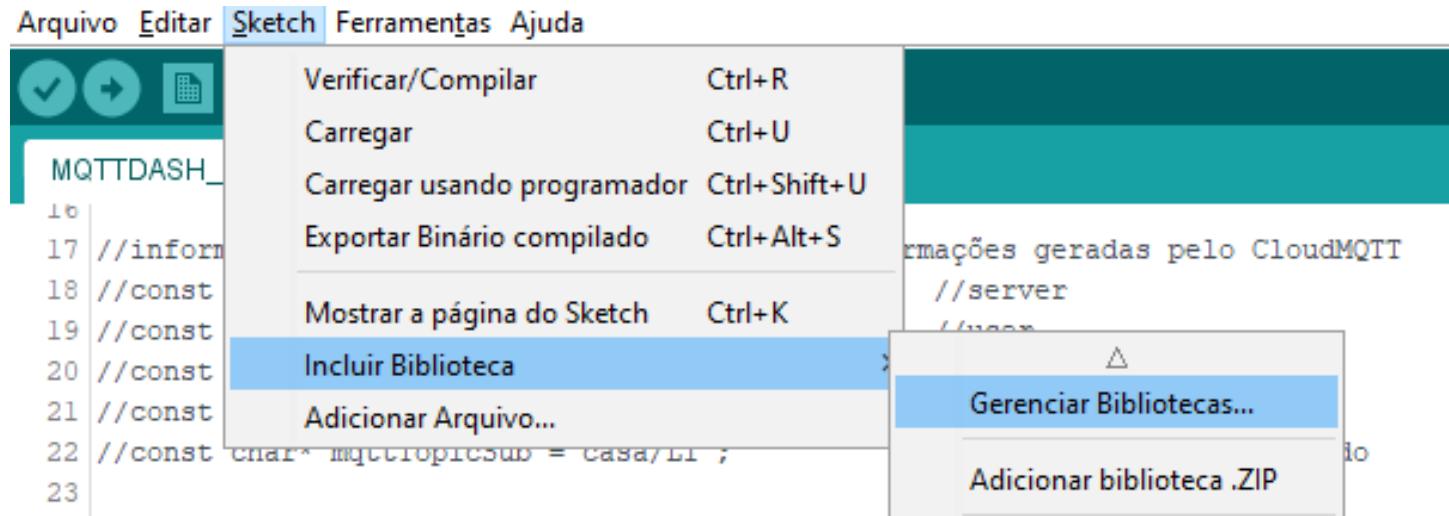
Circuito



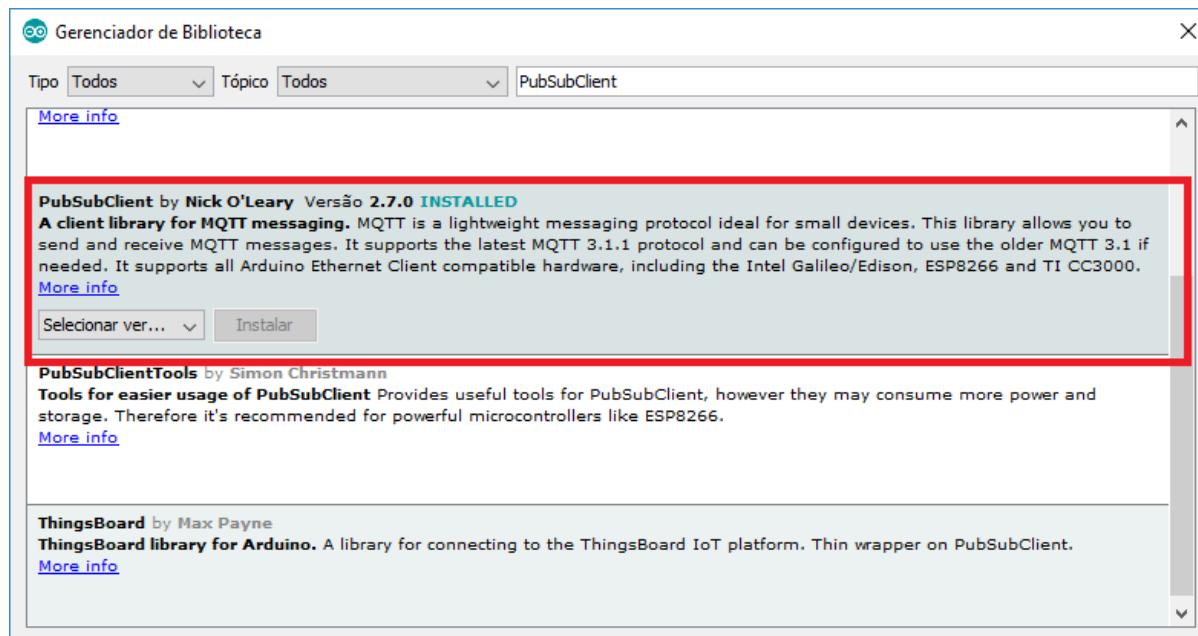
Programação

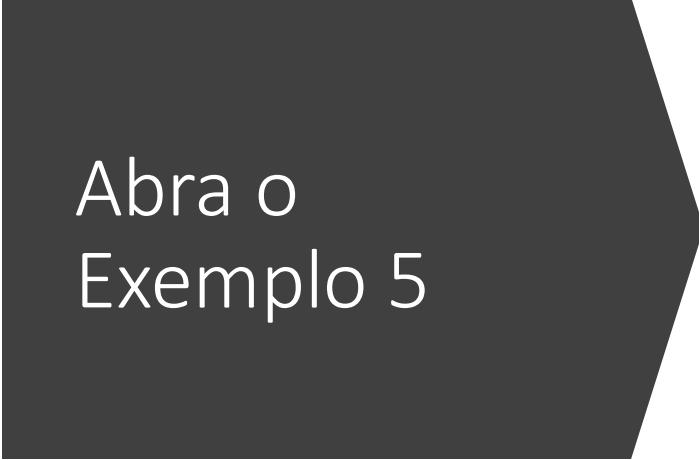
instalar a
biblioteca
PubSubClient.h

Acesse Sketch> Incluir Biblioteca> Gerenciar Bibliotecas:



Procure por PubSubClient e instale a seguinte opção:





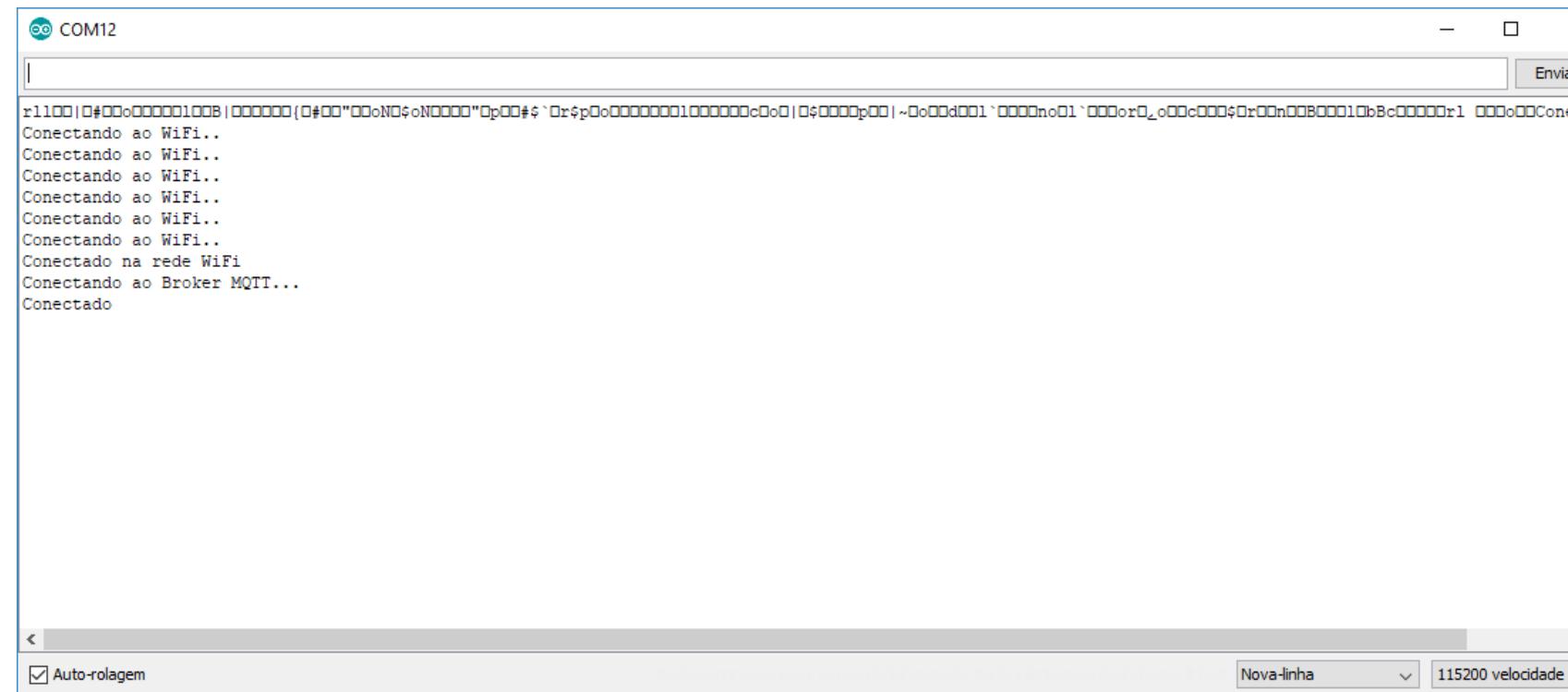
Abra o
Exemplo 5

Substitua com as informações do Broker

```
#define servidor_mqtt "xxxxxxxxxxxxxxxxxxxxxx" //URL do servidor MQTT
#define servidor_mqtt_porta  "xxxxxx" //Porta do servidor (a mesma deve ser informada na variável abaixo)
#define servidor_mqtt_usuario "xxxxxx" //Usuário
#define servidor_mqtt_senha   "xxxxxx" //Senha
```

Abra o terminal

Verifique se a placa conectou a rede WIFI e ao broker:



The screenshot shows a terminal window titled "COM12". The window displays the following text output:

```
r1100|0#0o00000100B|000000(0#00"0o0N0$0N0000"0p00#$`0r$p0o000000100000c0o|0f0000p00|~o00d001`0000no01`0000or0_0_o0c000$0r00n00B00010bBc00000r1 000o00Conectando ao WiFi..Conectando ao WiFi..Conectando ao WiFi..Conectando ao WiFi..Conectando ao WiFi..Conectando ao WiFi..Conectando na rede WiFiConectando ao Broker MQTT...Conectado
```

At the bottom of the window, there are several status indicators and settings: "Auto-rolagem" (checkbox checked), "Nova-linha" (dropdown menu), and "115200 velocidade" (dropdown menu).

Abra a guia
Websocket UI
no CloudMQTT

Envie uma mensagem no tópico casa/L1 pelo Websocket UI:

- 0 - Saída em nível LOW
- 1 - Saída em nível HIGH

websocket

Send message

Topic: casa/L1

Message: 1

Send

Received messages

Topic	Message
esp8266/pincmd	desliga
casa/L1	1
casa/L1	0
casa/L1	1
casa/L1	0
casa/L1	1

websocket

Send message

Topic: casa/L1

Message: 1

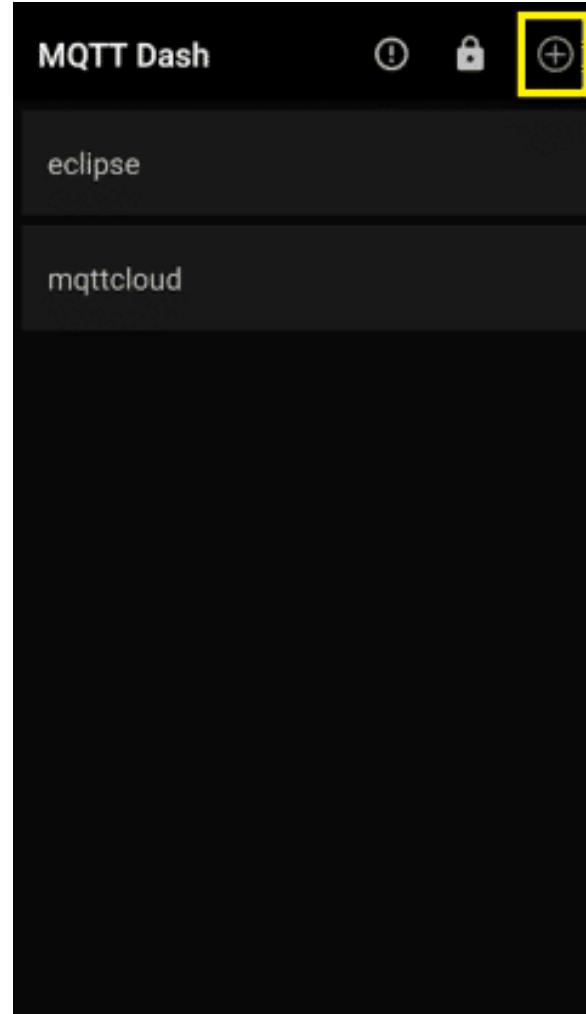
Send

Received messages

Topic	Message
esp8266/pincmd	desliga
casa/L1	1
casa/L1	0
casa/L1	1
casa/L1	0
casa/L1	1

Configurando a aplicação no MQTT Dash

Após instalado, clique
no sinal “+” na sua
tela inicial:



Insira as informações do Broker

Será aberta a configuração de uma nova conexão. Insira as seguinte informações:

- Name
- Adress
- Port
- User Name
- User Password

MQTT Dash 

Name

Address

Port

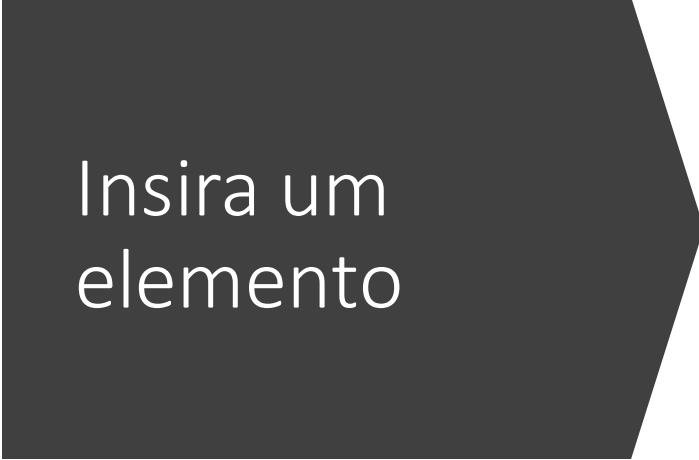
Enable connection encryption (SSL/TLS).
Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to your device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections.

This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk.

User name

User password

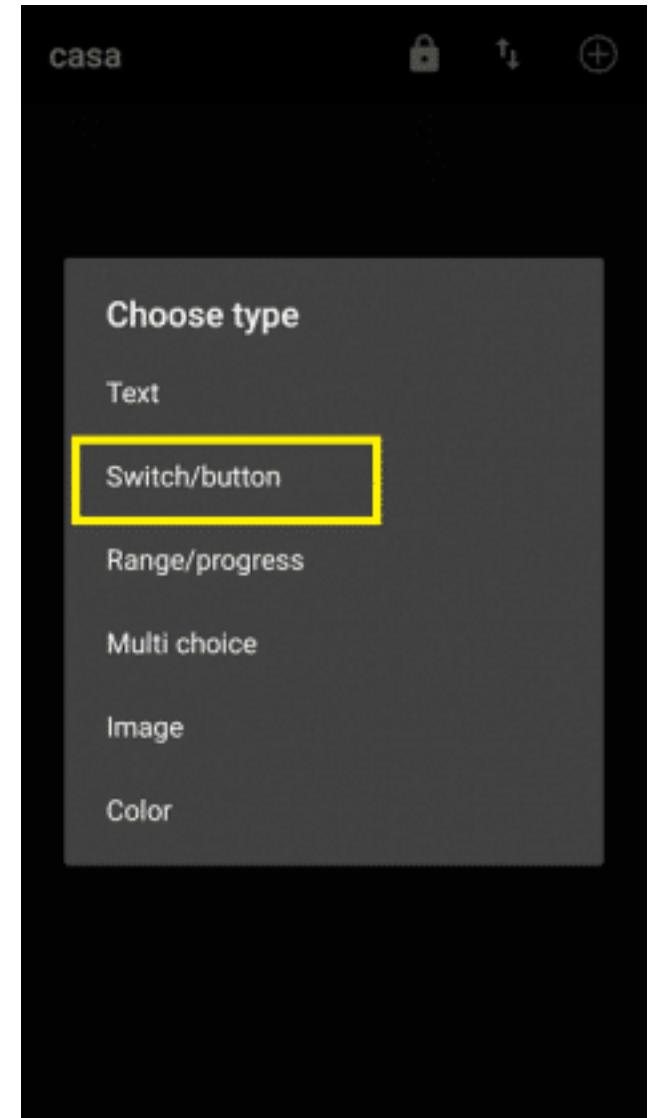
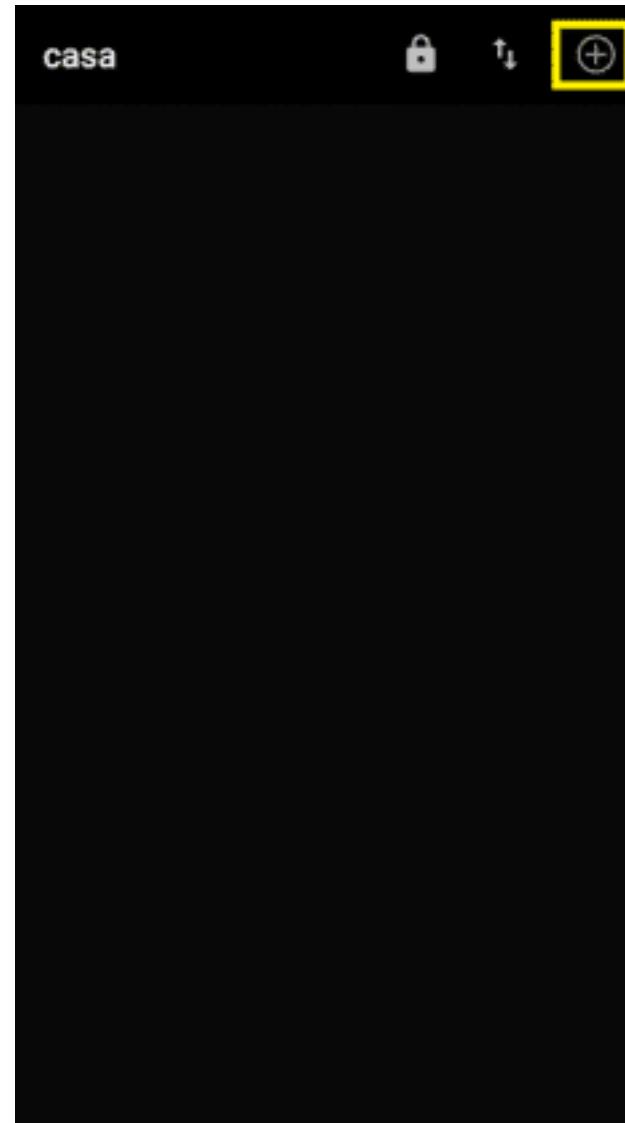
Client ID (must be unique)



Insira um
elemento

Com a conexão configurada
corretamente, clique no
sinal de “+” dentro do
dashboard criado:

Insira um Switch/button:



Configuração do Botão

- O botão será usado para acionamento da lâmpada. Você pode dar o nome L1 para ele, ou outro que achar melhor. Para o tópico, configure para “casa/L1”:
- Para a parte visual desse botão, configure para exibir ícones de uma lâmpada acesa e outra apagada. Para On, enviaremos o valor “1” e para Off, o valor “0”
- Por fim, selecione o nível de qualidade para QoS(1).

MQTT Dash

This metric is intended for state displaying and switching (e.g. light on/off). Or it can behave as a simple static button. Payload is expected to be string.

Name
L1

Topic (sub)
casa/L1

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>

Enable publishing

Topic (pub) - keep empty if the same as sub

Update metric on publish immediately (do not wait for incoming message to update visual state)

Payload and icons. If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same

MQTT Dash

Update metric on publish immediately (do not wait for incoming message to update visual state)

Payload and icons. If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same payload value.

On 1 Off 0

Other settings

QoS(0)
 QoS(1)
 QoS(2)

Retained

Blink tile to draw attention, if the expression evaluates to 'true'.
Expression can be any valid JavaScript expression which evaluates to boolean (true/false).

Websocket UI no CloudMQTT

Abra a guia Websocket UI no CloudMQTT e veja se as mensagens estão chegando ao pressionar o botão:

The screenshot shows the CloudMQTT Websocket UI. On the left, there's a 'Send message' section with fields for 'Topic' and 'Message', and a 'Send' button. Below it is a 'Clear session' button. On the right, there's a 'Received messages' section displaying a table of received messages. The table has columns for 'Topic' and 'Message'. The data in the table is as follows:

Topic	Message
casa/L1	0
casa/L1	1
casa/L1	1
casa/L1	0
casa/L1	0
casa/L1	1
casa/L1	0
casa/L1	1

Se placas estiverem conectadas saída também mudará o seu estado.
Faça o teste!

CHALLENGE



Desafio

Adicione mais saídas e
outros tópicos ao seu projeto

obrigado!

CONTATO

fs.embarcados@gmail.com

about.me/fabio.souza

@fabiosouza.io

