

Campus Party Bahia

Oficina de IoT Conhecendo ESP8266



**Campus
Party™**



Douglas Esteves 20/05/2018



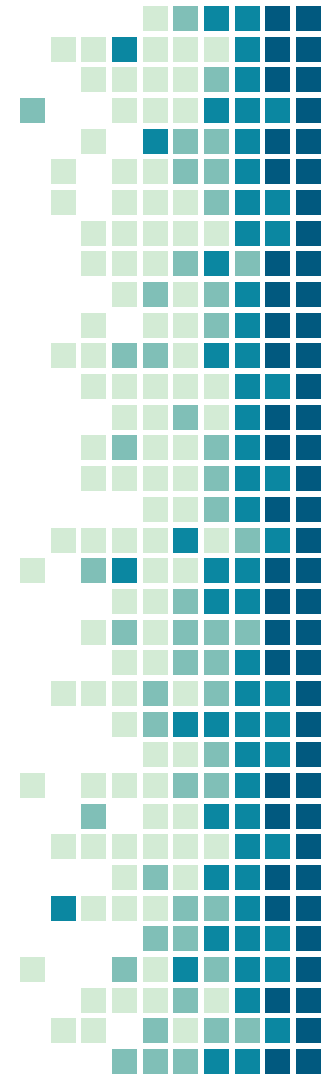
Sobre min

Douglas Esteves

Engenheiro da Computação

Co-fundador do @IoTmakers

Membro do LHC Laboratório Hacker
de Campinas



IoTmakers



Em breve! Cursos online de IoT

The logo for IoT Makers features the text "IoT Makers" in a bold, sans-serif font. The "IoT" portion is blue, while "Makers" is dark grey. A blue dot is positioned above the dot of the letter "i" in "Makers".

IoT Makers

cursosiotmakers.com.br

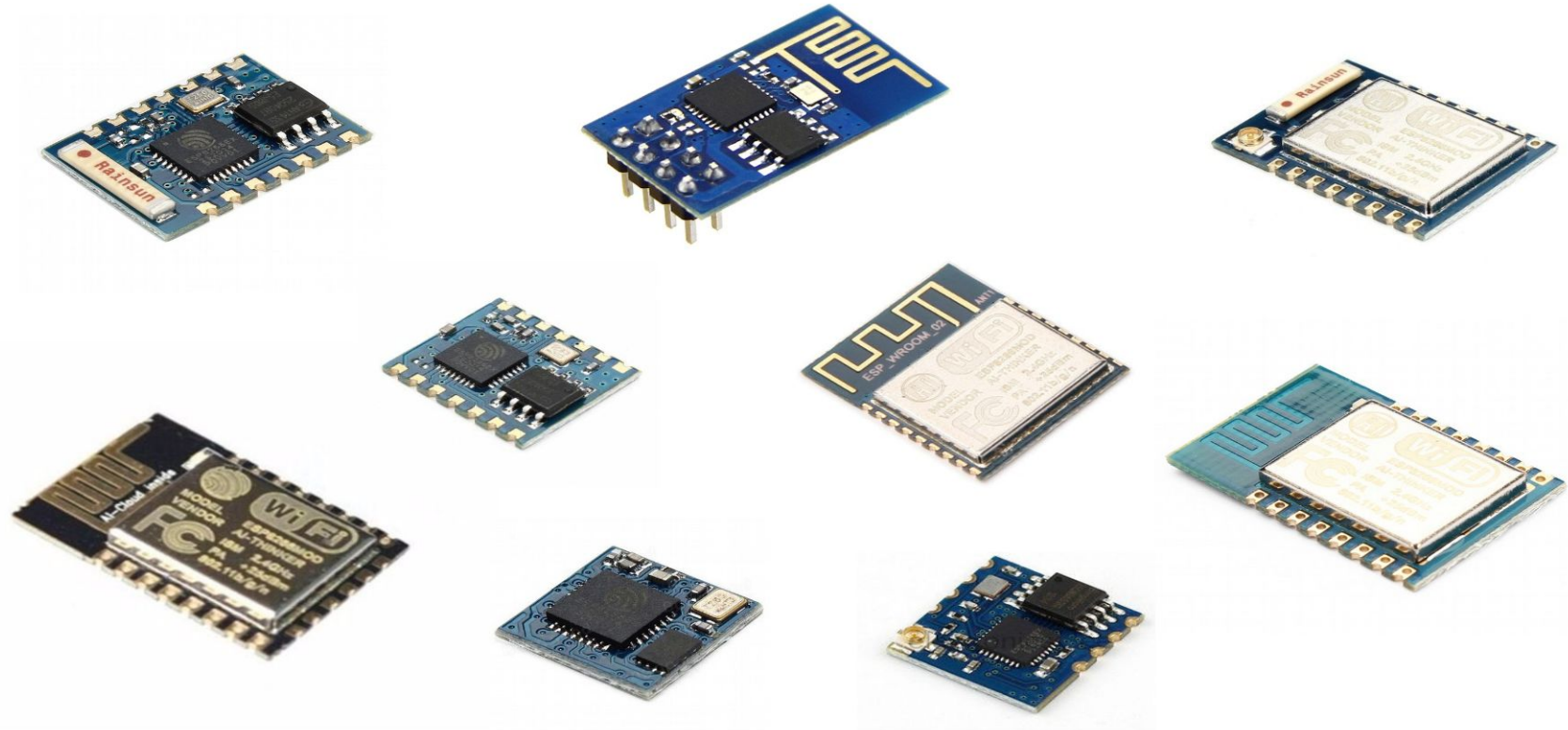
needinfo

ESP8266

Hardware e Software



Módulos com ESP8266



ESpecificações

Categories	Items	Parameters
Wi-Fi	Standard	CCC / FCC / CE / TELEC / SRRC
	Protocols	802.11 b/g/n
	Frequency Range	2.4 G ~ 2.5 G (2400 M ~ 2483.5 M)
	Tx power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Antenna	PCB on-board, external, IPEX connector, ceramic chip
Hardware	Peripheral interface	UART / SDIO / SPI / I2C / I2S / IR Remote Control
		GPIO / PWM
	Operating voltage	3.0 V ~ 3.6 V
	Operating current	Average: 80mA
	Operating temperature range	-40 °C ~ 125 °C
	Storage temperature range	-40 °C ~ 125 °C

ESPeCificações

Categories	Items	Parameters
	Package size	QFN32-pin (5 mm x 5 mm)
	External interface	N/A
Software	Wi-Fi mode	station / softAP / SoftAP + station
	Security	WPA / WPA2
	Encryption	WEP / TKIP / AES
	Firmware upgrade	UART Download / OTA (via network)
	Software development	SDK for customised development / cloud server development
	Network Protocols	IPv4, TCP / UDP / HTTP / FTP
	User configuration	AT Instruction Set, Cloud Server, Android/ iOS App

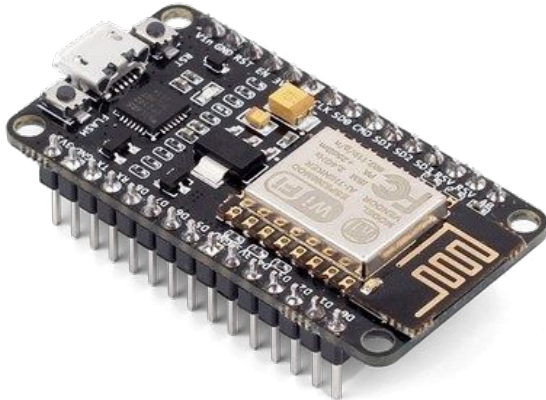
Funcionalidades do ESP e Hardware

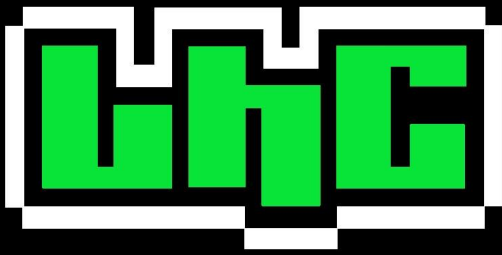
- O ESP é 3.3V. Em tensões maiores? Queima!
- O Mesmo se aplica as GPIOs e a Serial/UART
- O ESP consome, em picos, cerca de 250mA
- Ligar no máximo 12mA em cada GPIO
- Algumas GPIOs tem funções no boot!
- O ADC tem 10bits (0-1023 & 0-1V)
- Somente 4 PWMs

Primeiros passos com ESP8266

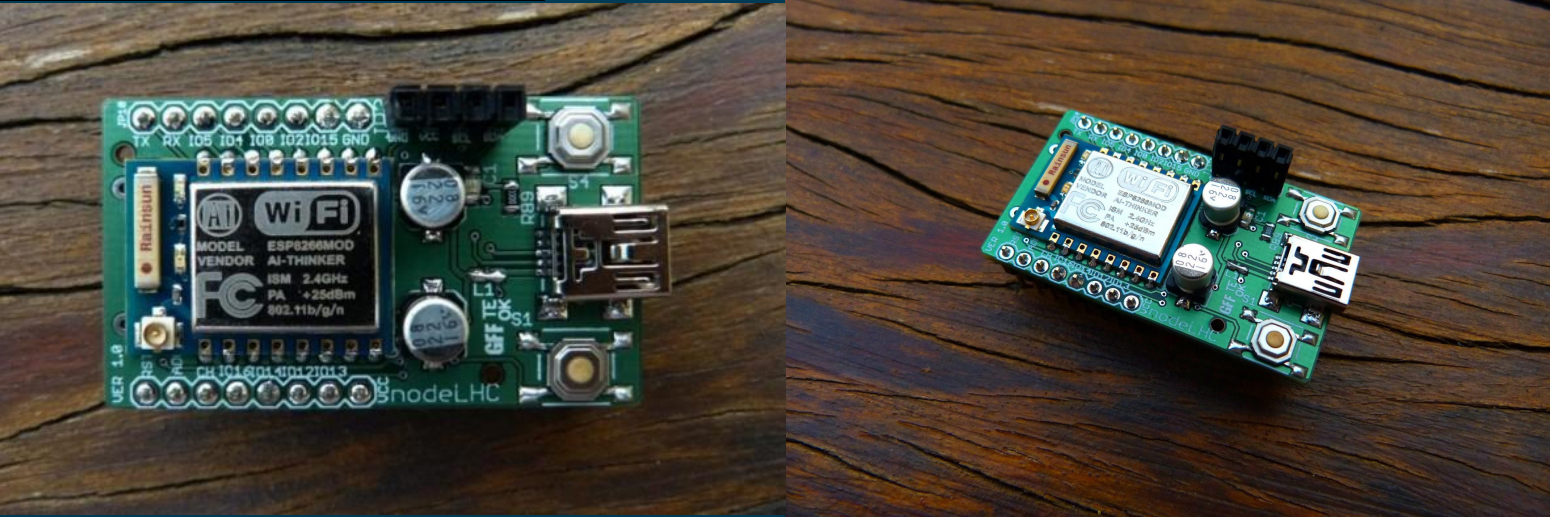
Programação com nodeMCU

- Programação em LUA script
- Curta curva de aprendizado
- Se faz um programa com poucas linhas de código
- http://nodemcu.com/index_en.html



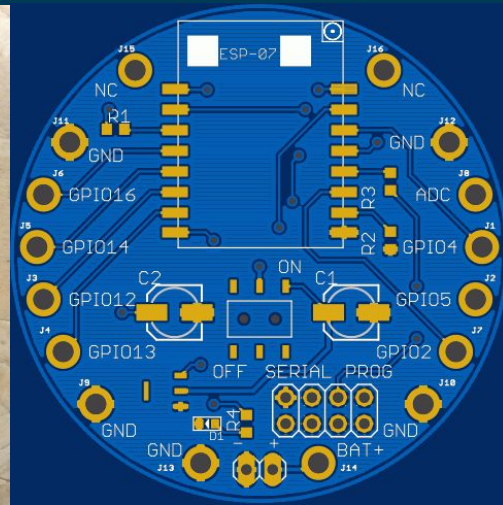
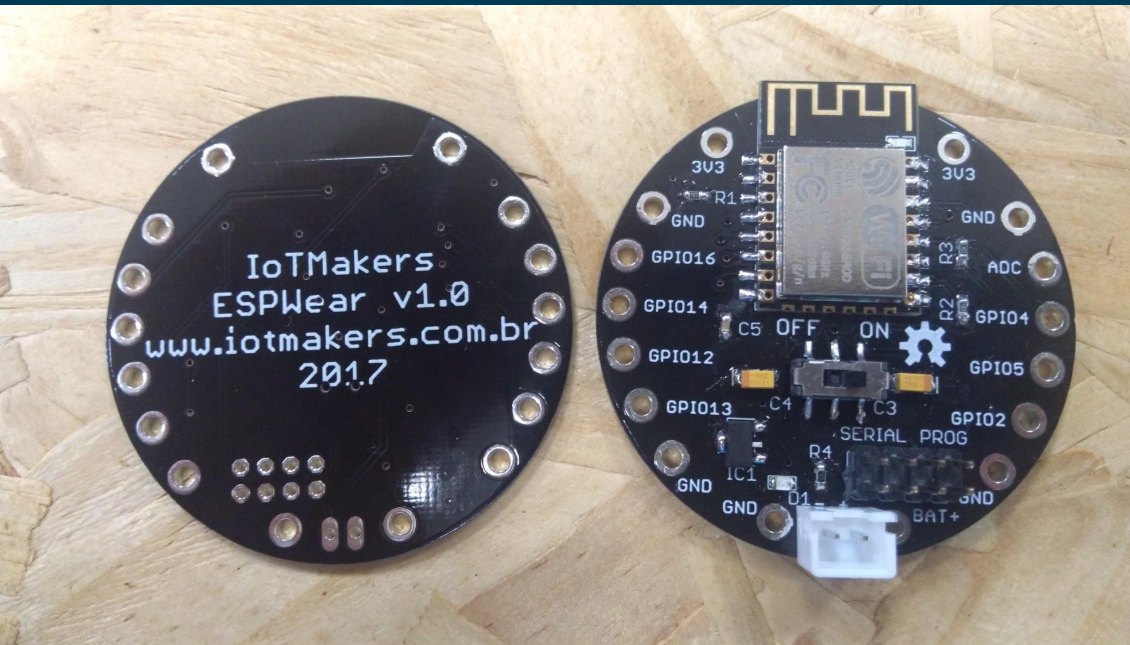


nodeLHC



<https://hackaday.io/project/7763-nodelhc-esp8266-development-board>

IoT Makers ESPWear

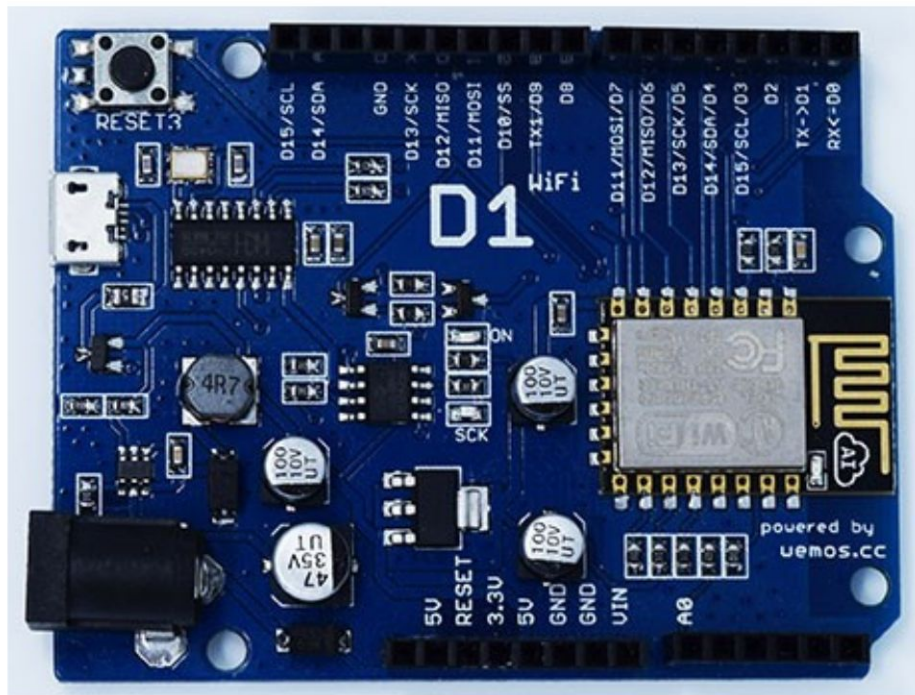


<https://hackaday.io/project/28790-espwear-esp8266-for-wearables>

<http://iotmakers.com.br/esp8266/iot-for-wearables-espwear/>

Conhecendo a **Wemos D1**

WeMos D1 (primeira versão)

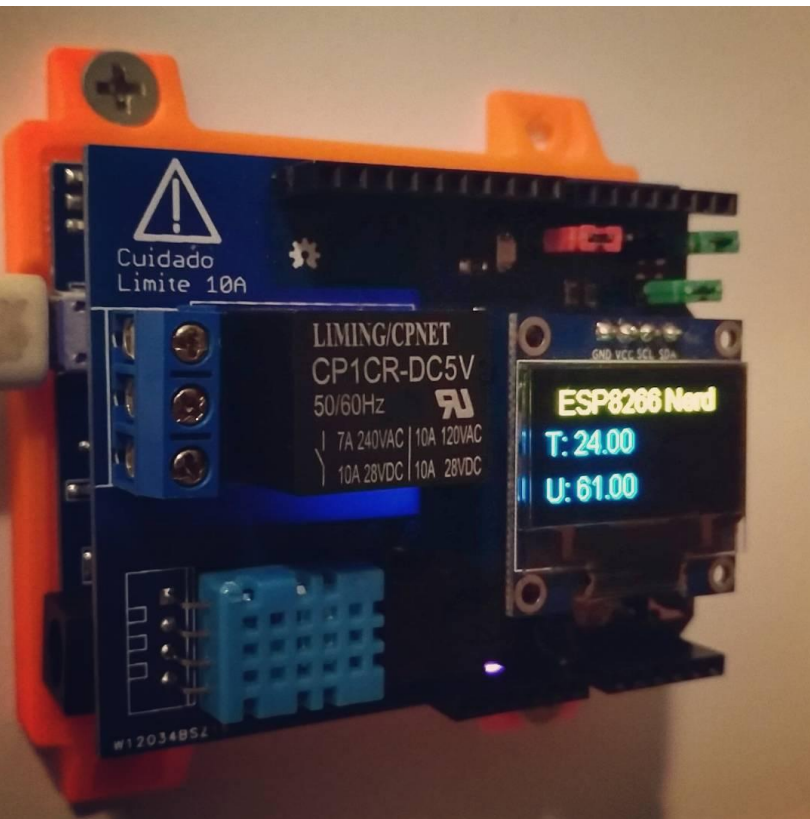


<http://pedrominatel.com.br/esp32/wemos-d1-o-esp8266-com-cara-de-arduino/>

WeMos D1

- 11 Pinos de IO
- 1 ADC (max 3.2V)
- Micro USB (CH340)
- Entrada DC 9-24V
- ESP12
- 3V3
- 4MB Flash
- Clock 80/160MHz
- Formato Arduino

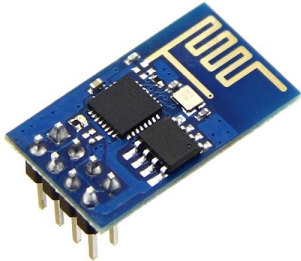




- Sensor de temperatura
- Sensor de umidade
- Rele (10Amps)
- LDR
- OLED Display
- Push button
- * LED RGB

Programação com Arduino IDE

- Programação similar ao Arduino
- Reuso de códigos já desenvolvidos no Arduino
- Reaproveitamento de conhecimento



Programação com Arduino IDE

- IDE 1.8.0 ou superior
 - Windows, Linux ou MAC
- Pacote ESP8266
 - Versão 2.3.0
 - http://arduino.esp8266.com/stable/package_esp8266com_index.json
 - <http://pedrominatel.com.br/pt/arduino/como-utilizar-o-esp8266-com-a-ide-arduino-instalando-o-modulo/>
 -
- Drive CH340 para alguns casos no windows
- GNU/LINUX
 - Manjaro, Ubuntu, Debian, Mint, ElementaryOS...

PlatformIO



PlatformIO is an open source ecosystem for IoT development

**Cross-platform IDE and unified debugger. Remote unit testing and
firmware updates**

[23 Development Platforms](#) [13 Frameworks](#) [423 Embedded Boards](#) [61 Project Examples](#) [1,777 Libraries](#) [8,251 Library Examples](#)

Professional development environment for

ARMmbed

Atmel



intel

freescale

LATTICE
SEMICONDUCTOR

MICROCHIP

NORDIC
SEMICONDUCTOR

NXP

SILICON LABS

life.augmented

Teensy

TEXAS
INSTRUMENTS

ARDUINO

Energia

libOpenCM3

The screenshot shows the PlatformIO IDE interface. On the left, a project tree for 'ESP8266' is visible, containing files like 'lib', 'readme.txt', 'src', 'platformio.ini', and 'wifuno'. The main editor area displays the 'platformio.ini' file with the following configuration:

```
[env:huzzah]
platform = espressif8266
board = huzzah
framework = arduino
```

A modal dialog titled 'PlatformIO: Serial Monitor' is open in the center. It has two dropdown menus: 'Port' (empty) and 'Baudrate' (set to '115200'). At the bottom of the dialog are 'Start' and 'Cancel' buttons, and a 'Toggle advanced settings' link.

Below the dialog, the terminal output shows the command 'platformio run' and its execution details:

```
platformio run

Verbose mode can be enabled via '-v, --verbose' option
Collected 23 compatible libraries
Looking for dependencies...
Project does not have dependencies
[INFO] Took 0.98 seconds
```

Below the terminal output is a list of available commands and their descriptions:

- boards: Embedded Board Explorer
- ci: Continuous Integration
- debug: PIO Unified Debugger
- device: Monitor device or list existing
- init: Initialize PlatformIO project or update existing
- lib: Library Manager
- platform: Platform Manager
- remote: PIO Remote
- run: Process project environments
- settings: Manage PlatformIO settings
- test: Local Unit Testing
- update: Update installed platforms, packages and libraries
- upgrade: Upgrade PlatformIO to the latest version

The terminal prompt shows the current directory: 'PS C:\Users\Doug\Documents\Github\IoTmakers\ESP8266>'.

On the right side of the image, there are two circular logos. The top one is a green circle with a white atomic symbol. The bottom one is an orange circle with a white Arduino Uno logo.



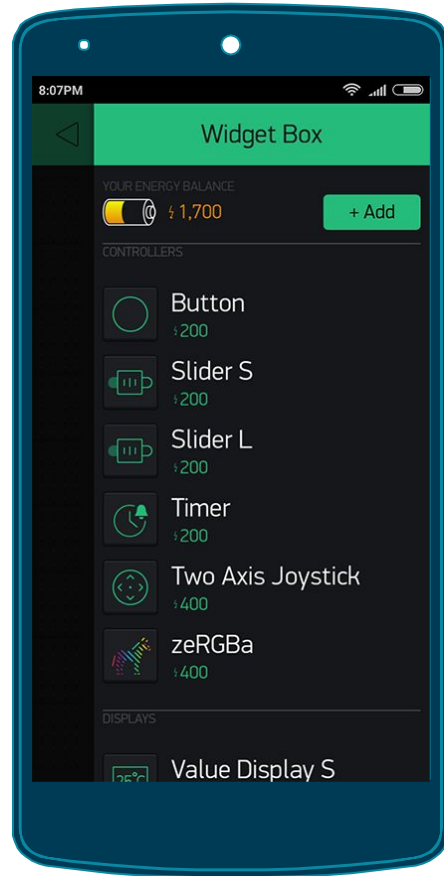
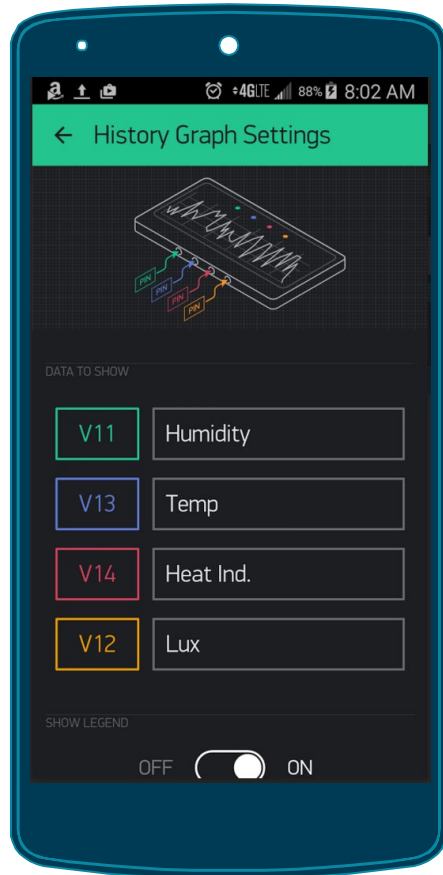
Blynk
Mobile

- Desenvolvimento direto do APP
- Código inicial usando o internet do Arduino
- Rede Local
- Token
- Recursos visuais



Blynk

App



O que é um GPIO ?

“General Purpose Input/Output (GPIO) são portas programáveis de entrada e saída de dados que são utilizadas para prover uma interface entre os periféricos e os microcontroladores/microprocessadores”.

Exercício – blink

Implementar um blink (hello world) utilizando o LED

Tempo: 5 minutos.

blink.ino

```
void setup(void) {  
  //Configura o pino digital para saida/output  
  pinMode(5, OUTPUT);  
  digitalWrite(5, LOW);  
}
```

```
void loop(void) {  
  //envia o comando de escrita no pino digital  
  digitalWrite(5, HIGH);  
  //funcao de pausa/delay em mili-segundos  
  delay(1000);  
  digitalWrite(5, LOW);  
  delay(1000);  
}
```

Exercício – Push button

Implementar leitura do botão tact e acionar o LED.

Tempo: 5 minutos.

Extra : Implementar contato de selo

```
#define BOTAO D13
```

```
#define LED D10
```

```
Int botao_status = 0;
```

```
void setup(void) {  
  pinMode(BOTAO, INPUT);  
  digitalWrite(LED, OUTPUT);  
}
```

```
void loop(void) {  
  Botao_status = digitalRead(BOTAO);  
  digitalWrite(LED,botao_status);  
}
```

Modos do WiFi

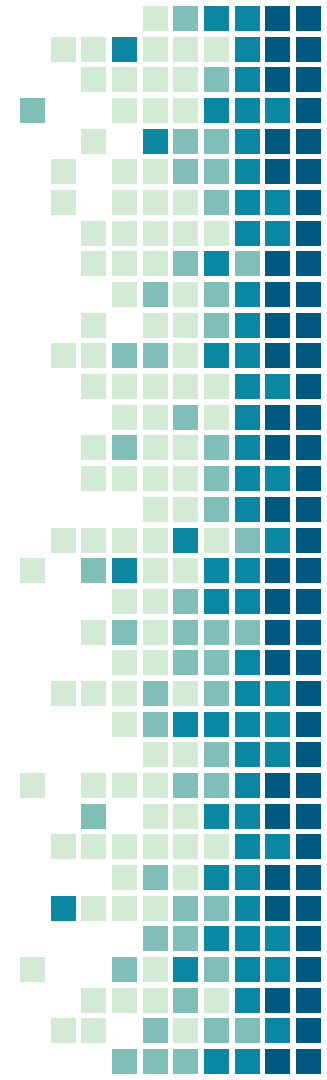
STATION

Utilizado para a conexão entre o ESP e um Access Point

SOFTAP

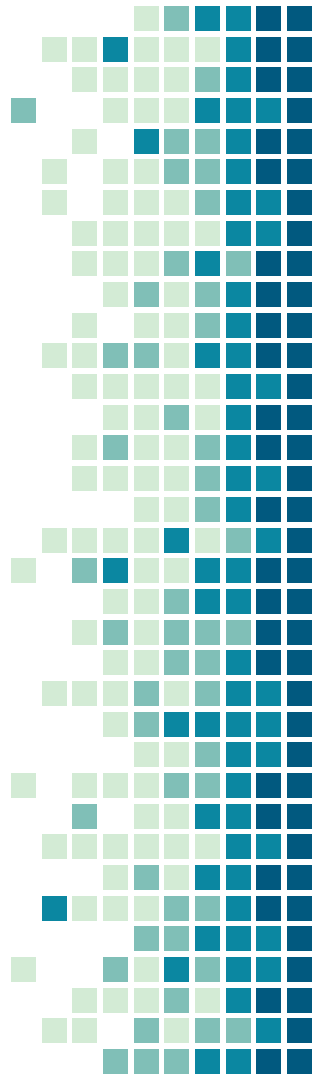
Utilizado para o ESP ser um Access Point

Bibliotecas : **ESP8266Wifi.h** & **ESP8266WIFIMulti.h**



GitHub

<https://github.com/IoTMakers/CPBA>
2



ESP8266-webserver.ino

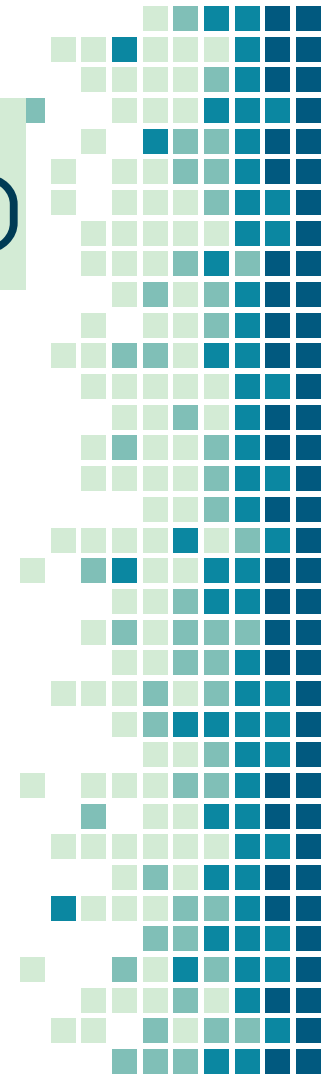
Página Web no ESP8266

ESP8266-site.ino

Conectar em uma plataforma online

ESP8266-SCAN.ino

Bônus



Referências

github

github.com/iotmakers

Pedro Minatel Blog

www.pedrominatel.com.br

Portal Embarcados

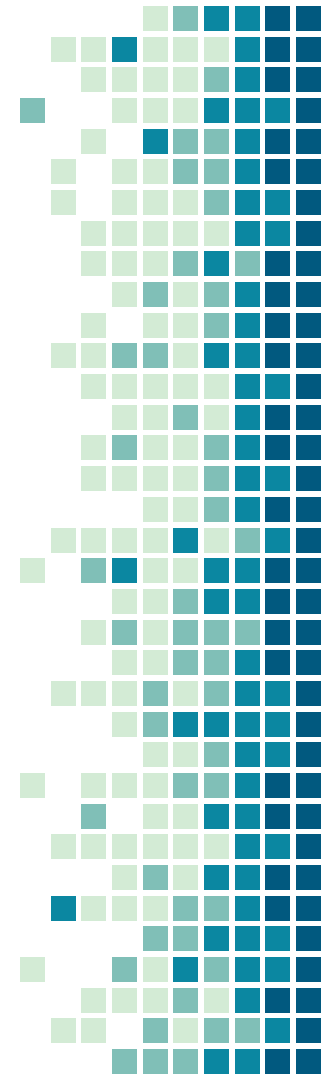
www.embarcados.com.br

Hackerspace LHC

www.lhc.net.br

Tlegram IoTmakers

<https://t.co/sUjdiltWBI>



Muito Obrigado

Contatos !



19 98230-3616



@_EstevesDouglas



douglas@iotmakers.com.br