

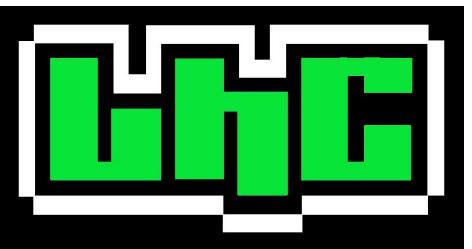
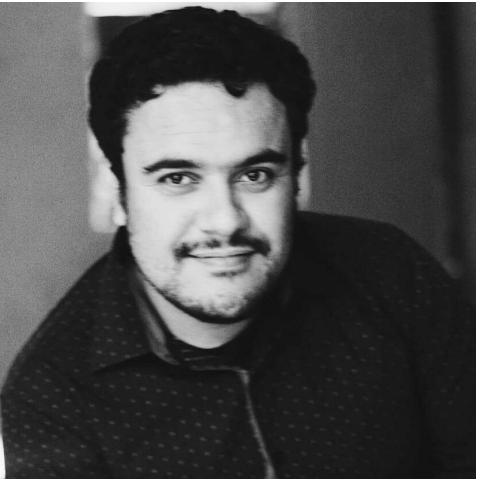


# Oficina de IoT

Conhecendo PlatformIO e ESP8266

Douglas Esteves

Brasília, 21 de Junho de 2019



# Sobre mim

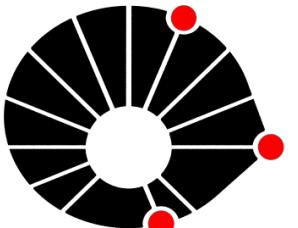
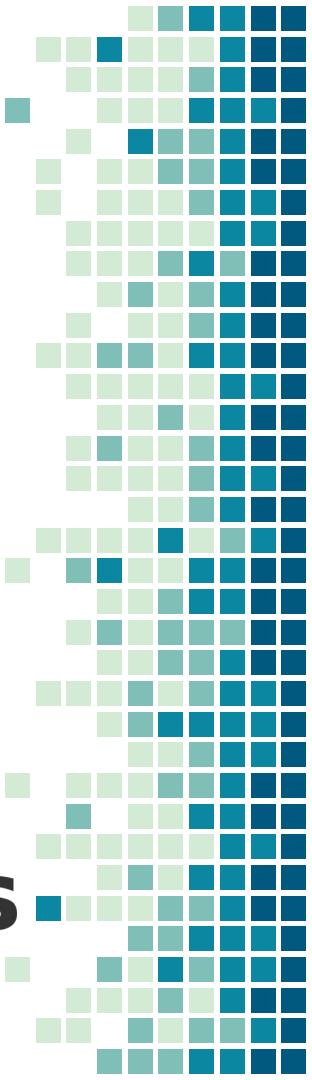
## Douglas Esteves

Engenheiro da Computação

Centro de Computação Unicamp

Co-fundador do IoTMakers

Membro do LHC Laboratório Hacker  
de Campinas



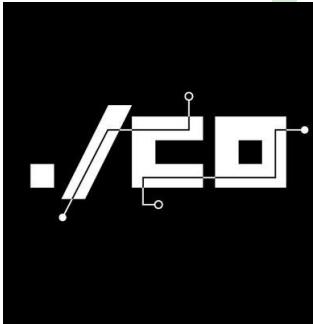
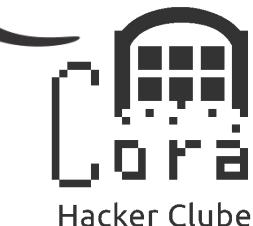
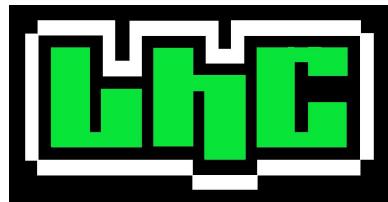
**UNICAMP**



**DUMONT**  
Hackerspace

**IoT****Makers**





# Oficina de IoT

Atividades em Hackerspaces, comunidades, grupos com interesse em entender e realizar atividade práticas.

Apresentações de plataformas, hardwares, projetos.

Mão na massa com diferentes tecnologias



# Agenda

- Módulos ESP8266 / **ESP32**
- Plataformas de desenvolvimentos
- Comunidades, onde buscar informações
- Ambientes para Oficina
- Mão na massa
- Referências





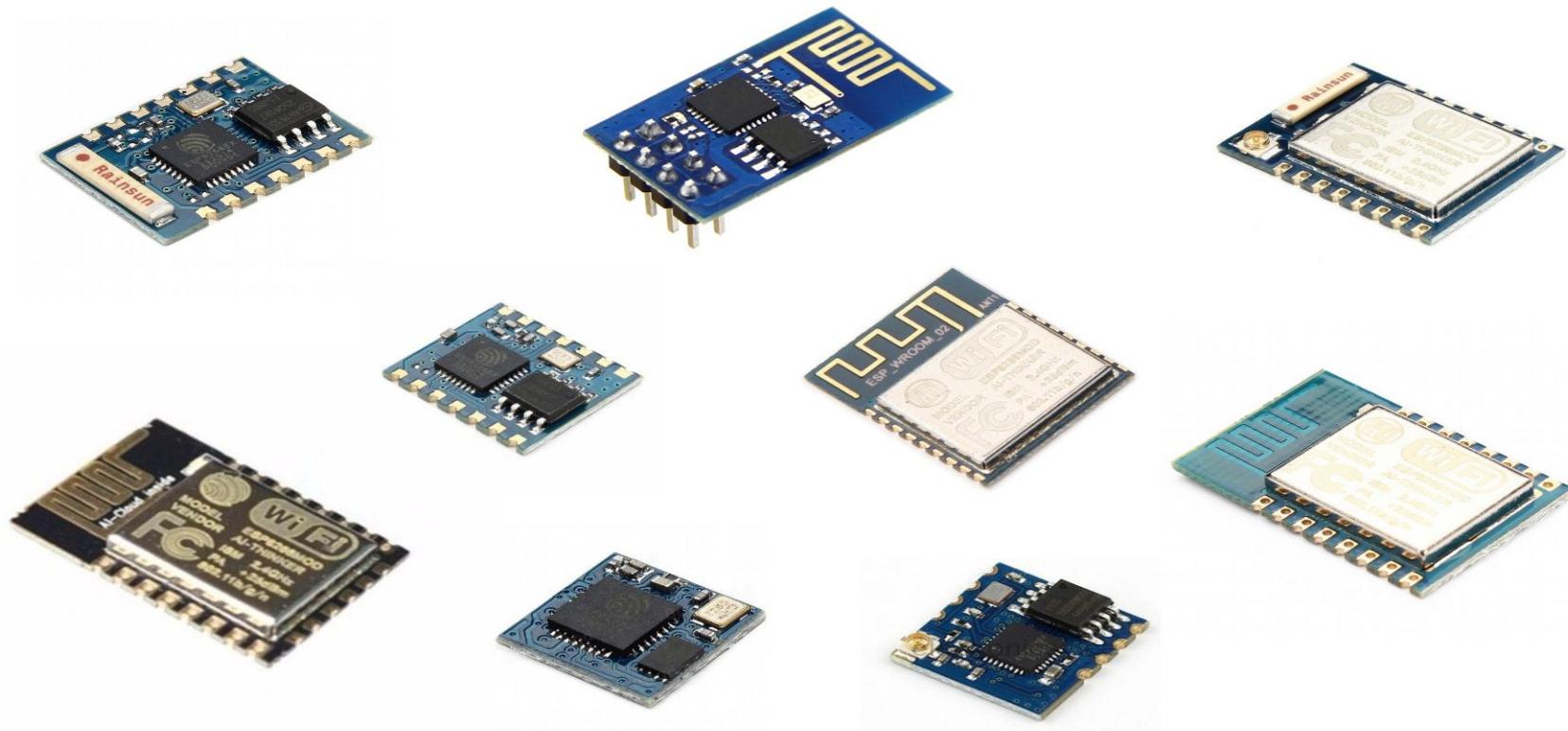
Imagen [cnx-software.com](http://cnx-software.com)

# ESP8266

Hardware



# 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

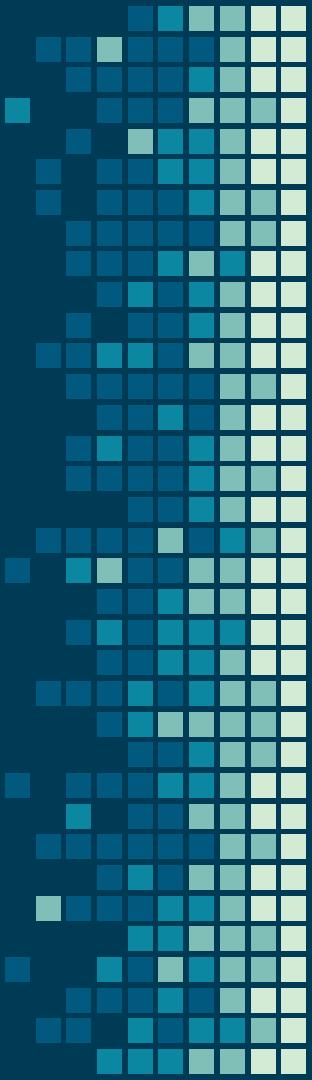
Categories	Items	Parameters
Software	Package size	QFN32-pin (5 mm x 5 mm)
	External interface	N/A
	Wi-Fi mode	station / softAP / SoftAP + station
	Security	WPA / WPA2
	Encryption	WEP / TKIP / AES
	Firmware upgrade	UART Download / OTA (via network)
	Software development	SDK for customised development / 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

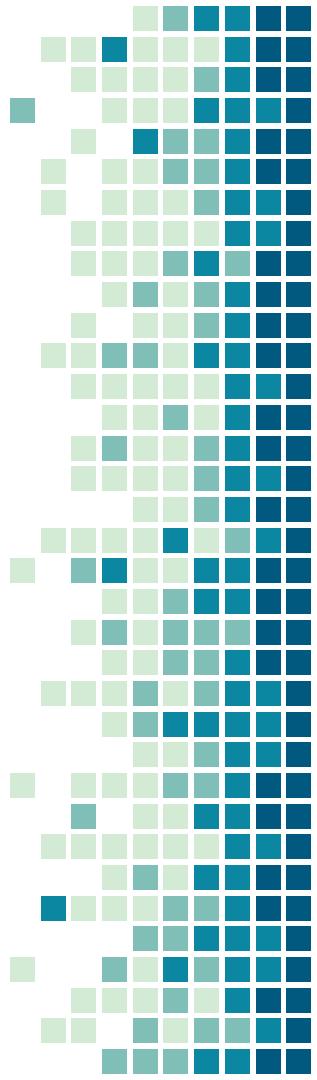
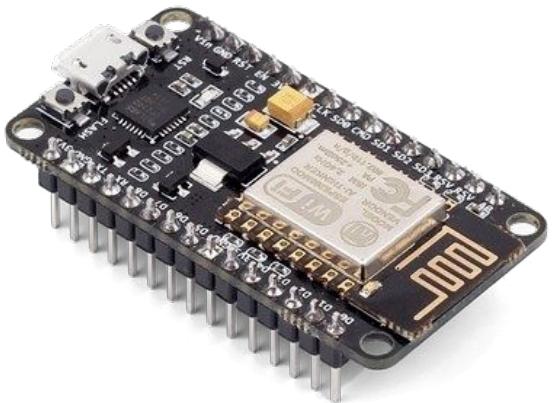
# Placas de Desenvolvimentos

# 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](http://nodemcu.com/index_en.html)

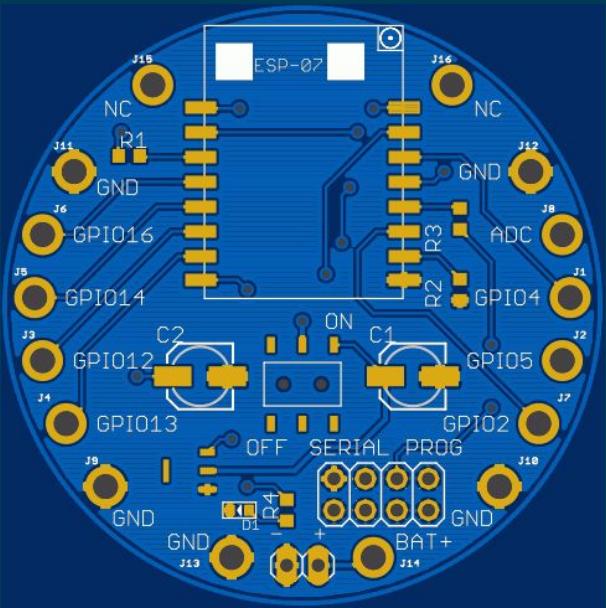


# nodeLHC



<https://hackaday.io/project/7763-nodelhc-esp8266-development-board>  
<https://lhc.net.br/wiki/NodeLHC>

# IoT Makers ESP Wear

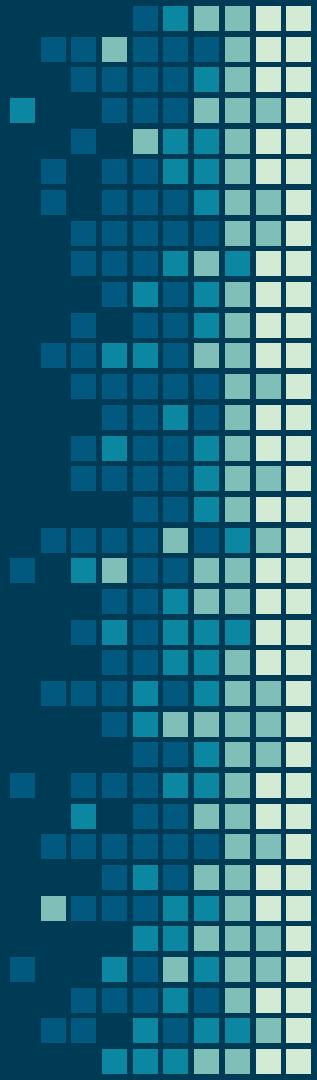


<https://hackaday.io/project/28790-espwear-esp8266-for-wearables>

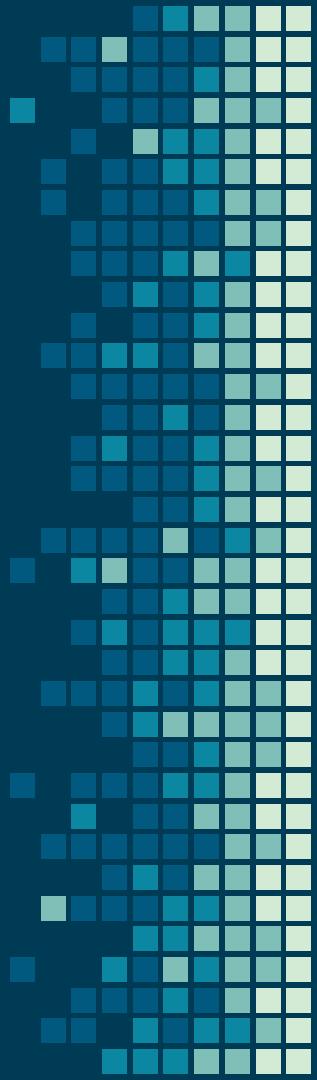
<http://iotmakers.com.br/esp8266/iot-for-wearables-espwear/>

# Placas de Desenvolvimentos

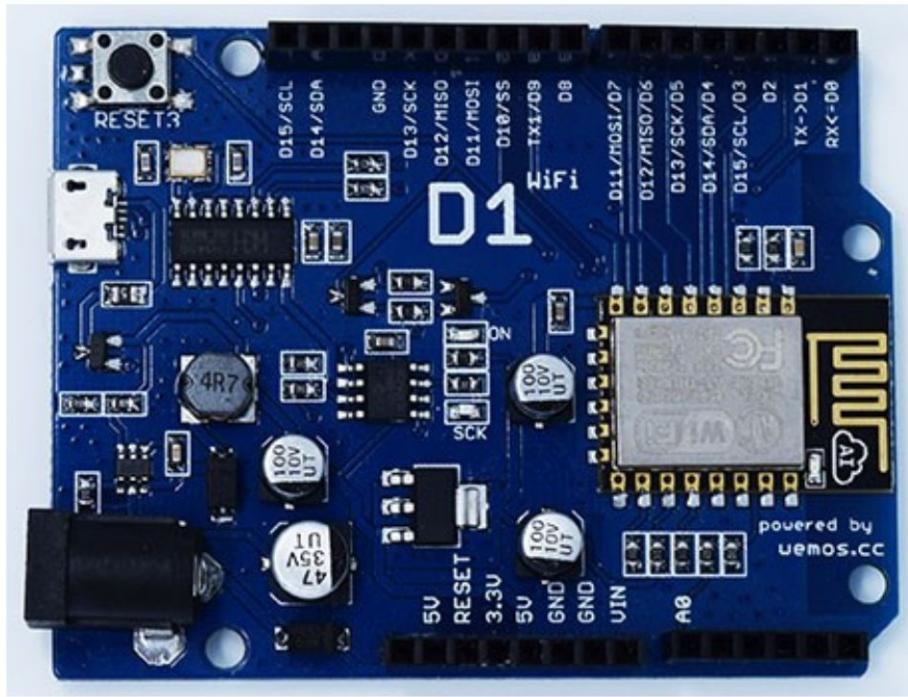
## Muitas opções



# Conhecendo a Wemos D1



# WeMos D1 (primeira versão)

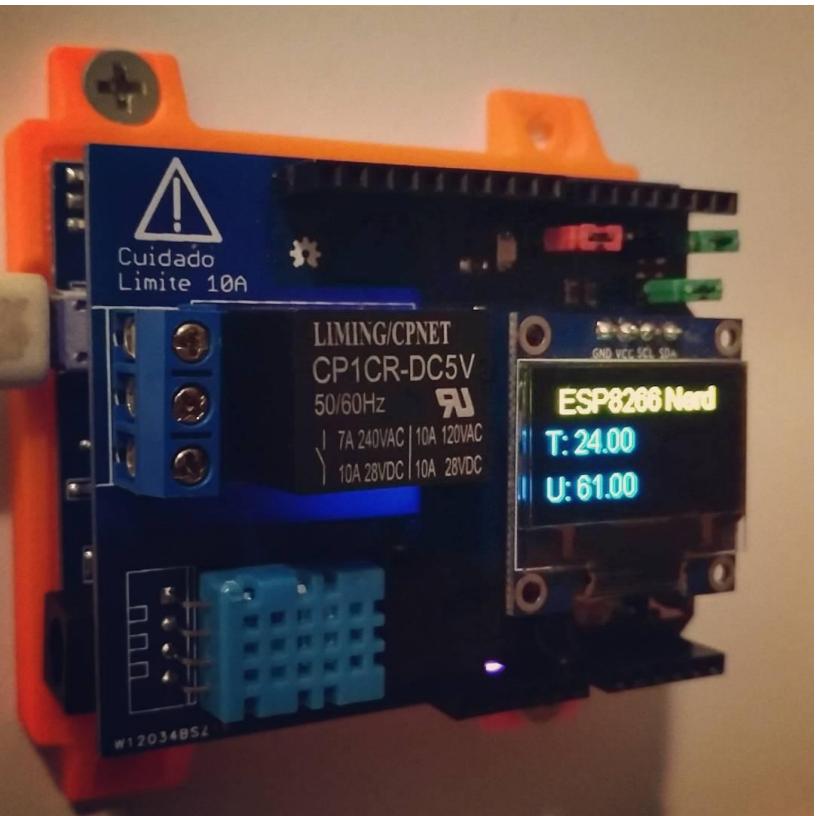


<http://pedrominatel.com.br/esp32/wemos-d1-o-esp8266-com-cara-de-arduino/>

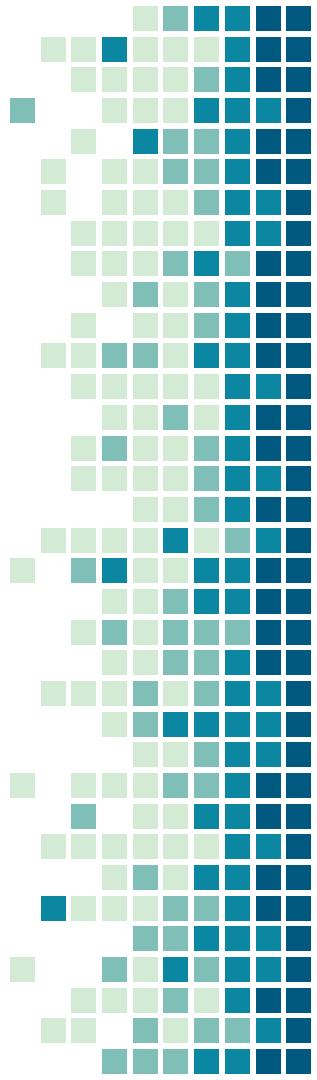
# WeMos D1

- 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

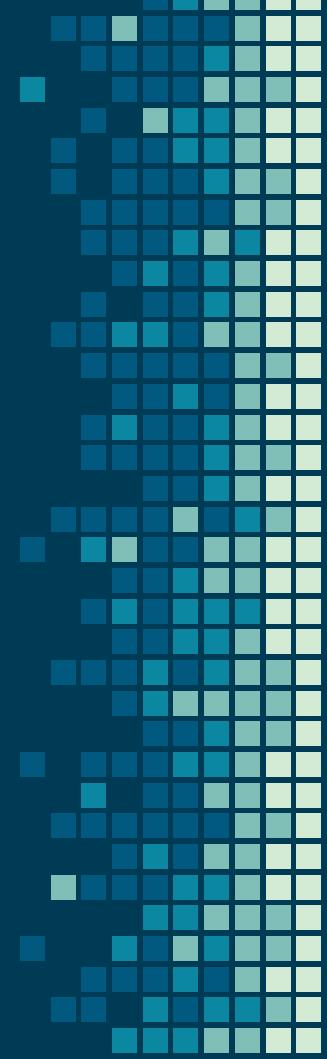
## IoT Shield



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

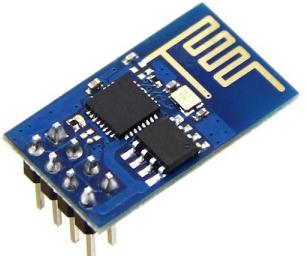


# ESP32



# 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://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

[Development Platforms](#)

[Frameworks](#)

[Embedded Boards](#)

[Project Examples](#)

[Libraries](#)

[Library Examples](#)

Professional development environment for

**ARM**mbed

**Atmel**

**ESPRESSIF SYSTEMS**

**intel**

**freescale**

**LATTICE SEMICONDUCTOR**

**MICROCHIP**

**NORDIC SEMICONDUCTOR**

**NXP**

**SILICON LABS**

**ST** life.augmented

**Teensy**

**TEXAS INSTRUMENTS**

**ARDUINO**

**Energía**

**libOpenCM3**



PlatformIO IDE 0.12.1  
Official PlatformIO IDE...  
Plat... Reload Install

IntelliJ IDEA K... 0.2.20  
Port of IntelliJ IDEA K...  
Keisuke Kato Install

PlatformIO 0.2.3  
PlatformIO for Visual ...  
Jun Han Install

vscode-flow-ide 1.2.0  
Visual studio code Flo...  
gcazaciu Install

OCaml and Rea... 1.0.24  
OCaml and Reason la...  
Darin Morrison Install

Polymer IDE 0.6.0  
Provides linting, autoc...  
polymer Install

JetBrains IDE K... 0.1.3  
JetBrains IDE Keymap...  
isodox Install

PureScript IDE 0.14.0  
PureScript IntelliSens...  
Nicholas Wolv... Install

Crystal IDE 0.0.4  
Syntax and error chec...  
Ryan L. Bell Install

Java IDE Pack 0.1.1  
All Java extensions to...  
Paul Verest Install

fish-ide 0.3.1  
IDE features for fish s...  
Sebastian Wie... Install

PocketMine IDE 0.0.6

PlatformIO IDE

PlatformIO.Org | ↗ 116,790 | ★★★★★ | Repository | License

Official PlatformIO IDE for IoT, Arduino, ARM mbed, Espressif (ESP8266/ESP32), STM32, PIC32, nRF51/nRF52, FPGA, CMSIS, SPL, ...

Reload Install

Details Contributions Changelog Dependencies

## PlatformIO IDE for VSCode

VS Marketplace v0.12.1 installs 116.79K rating 4.13/5 (15)

The next generation Integrated development environment for IoT

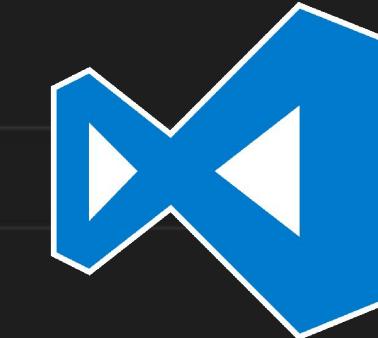
PlatformIO is an open source ecosystem for IoT development. Cross-platform build system and unified debugger. Remote unit testing and firmware updates.

**Platforms:** Atmel AVR, Atmel SAM, Espressif 32, Espressif 8266, Freescale Kinetis, Intel ARC32, Lattice ICE40, Maxim 32, Microchip PIC32, Nordic nRF51, Nordic nRF52, NXP LPC, Silicon Labs EFM32, ST STM32, Teensy, TI MSP430, TI Tiva, WIZNet W7500

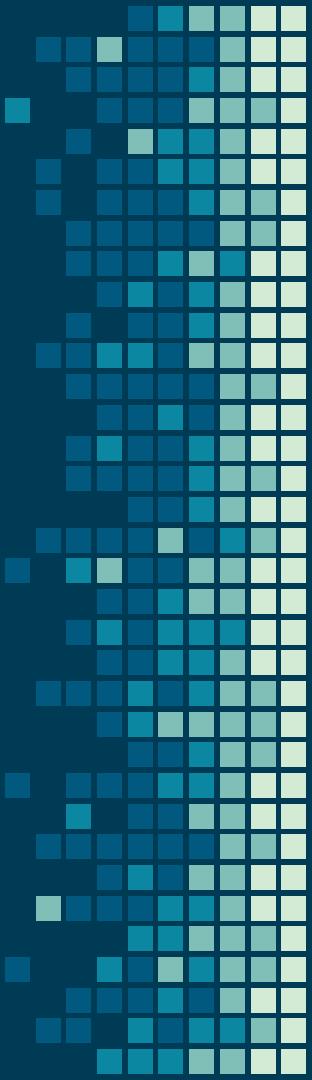
**Frameworks:** Arduino, ARTIK SDK, CMSIS, Energia, ESP-IDF, libOpenCM3, mbed, Pumbaa, Simba, SPL, STM32Cube, WiringPi

## Features

- Cross-platform code builder without external dependencies to a system software:
  - 450+ embedded boards
  - 20+ development platforms
  - 10+ frameworks
- PIO Remote™
- PIO Unified Debugger
- Unit Testing
- C/C++ Intelligent Code Completion



# Plataformas



# TagoIO

TagoIO | Developer

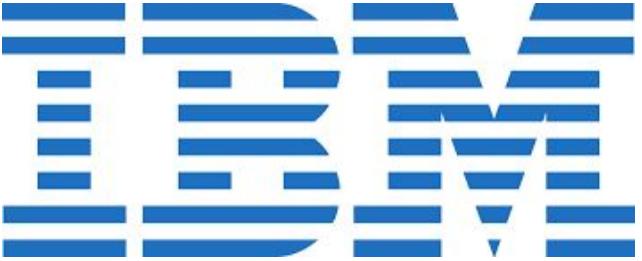
Home Devices Buckets Files Analysis Actions Explore Dashboards Chicago - Bus LHC-Temperatura

Devices

Devices are the link between your external things and the data buckets in your account. [Read more here.](#)

Name	Last Input	Last Output	Active	Bucket linked	Created at
wemos	N/A	N/A	Yes	wemos	2 months ago

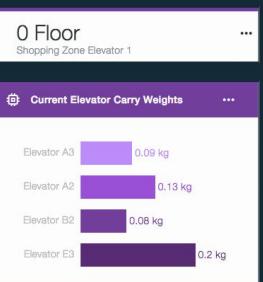
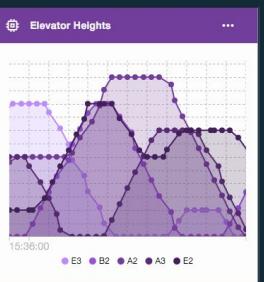
Device Emulator + Add Device



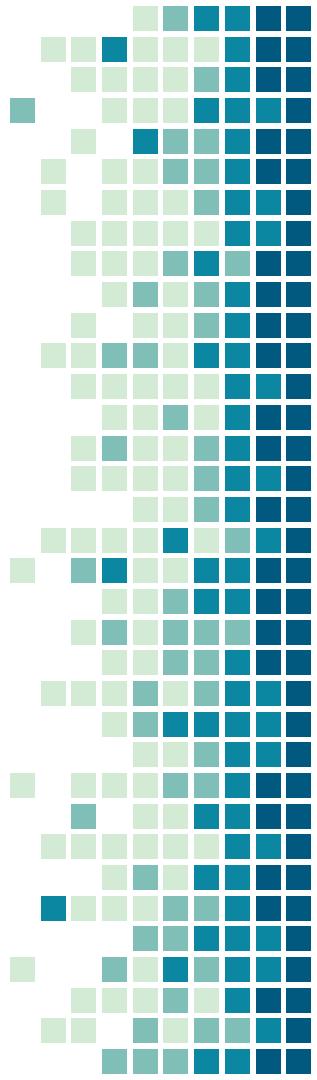
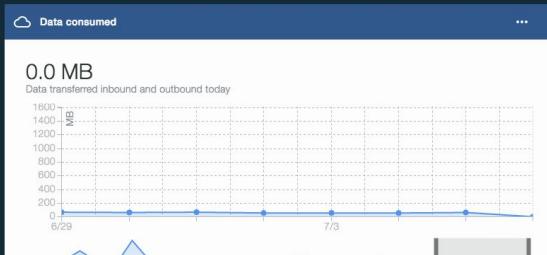
## Device Status Board

[+ Add New Card](#) [Settings](#)

- 1 Floor Shopping Zone Elevator 2
- 0 Floor East Corridor Elevator
- 3 Floor Emergency/Freight Elevator
- 0 Floor North Walkway Elevator

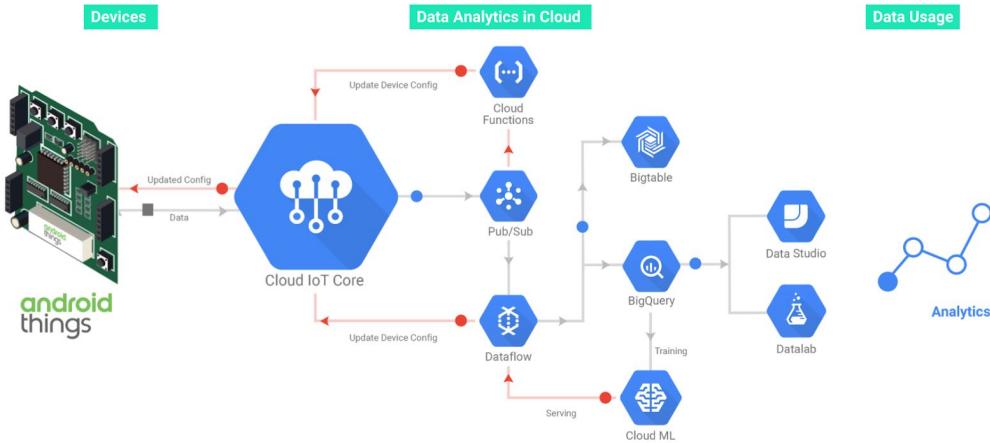


## Data Status



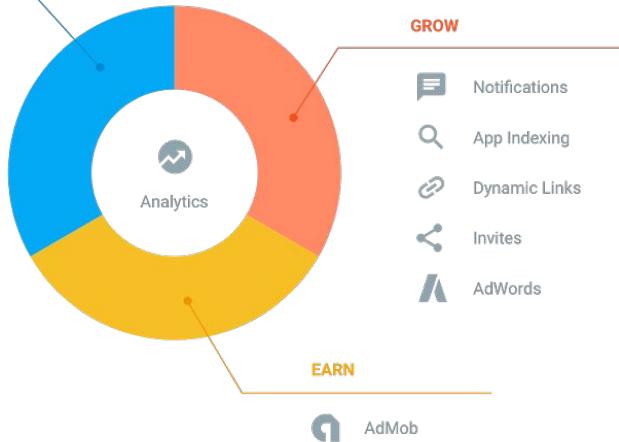


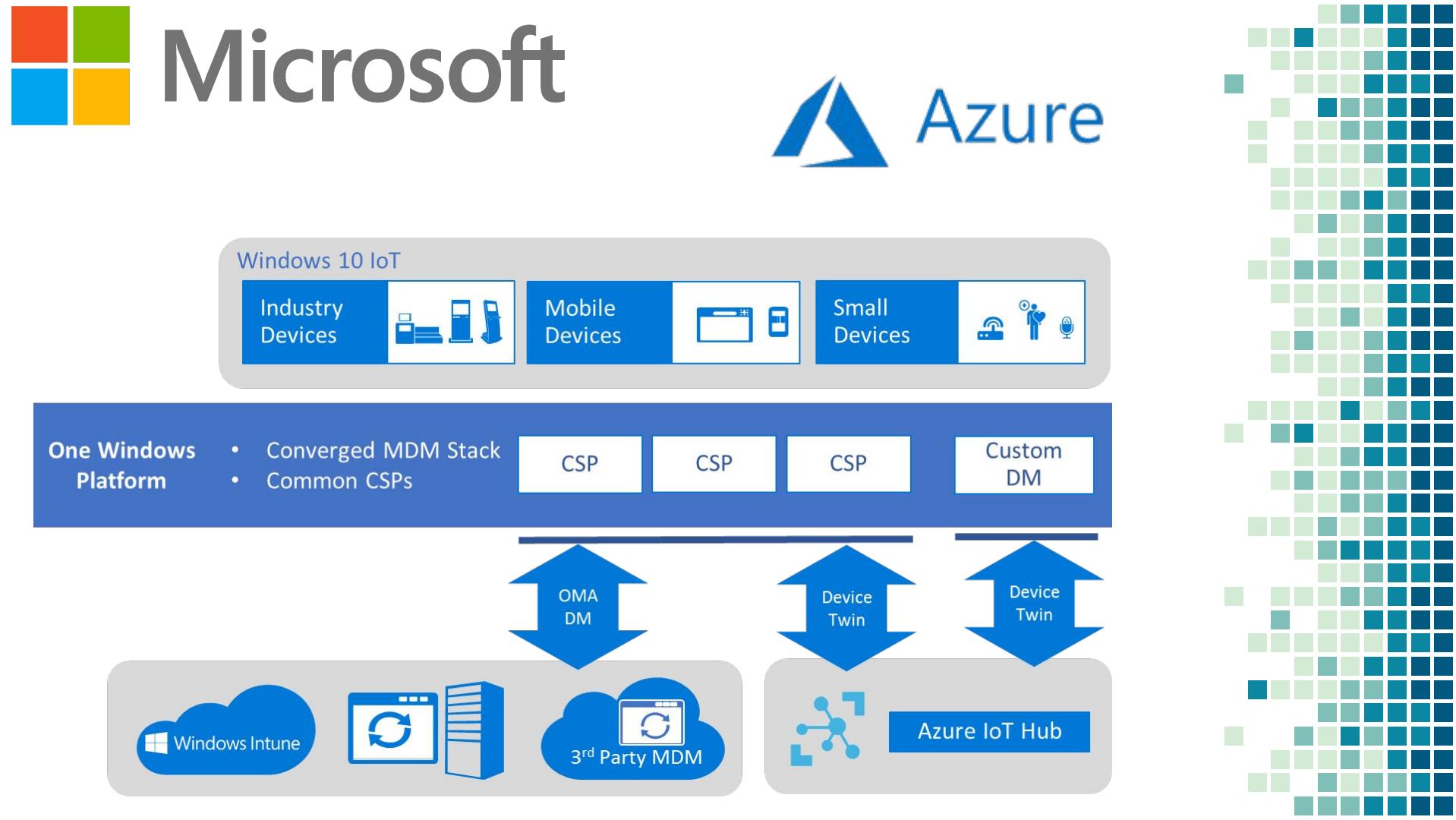
# Google Cloud



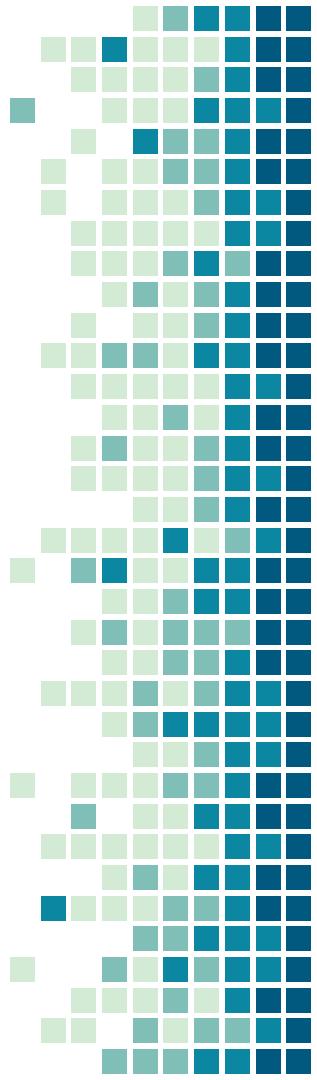
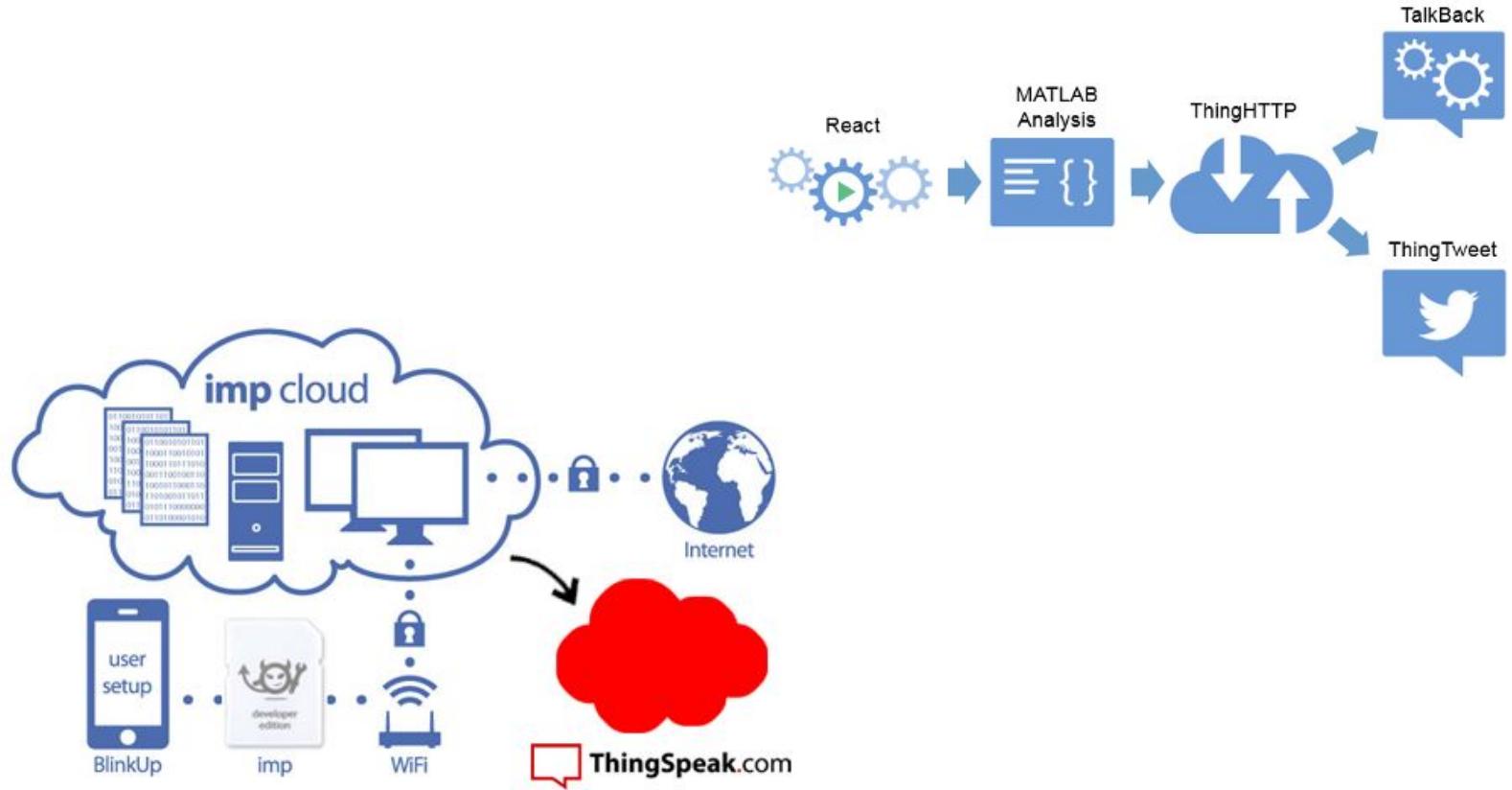
## DEVELOP

- Realtime Database
- Authentication
- Cloud Messaging
- Storage
- Hosting
- Remote Config
- Test Lab
- Crash Reporting

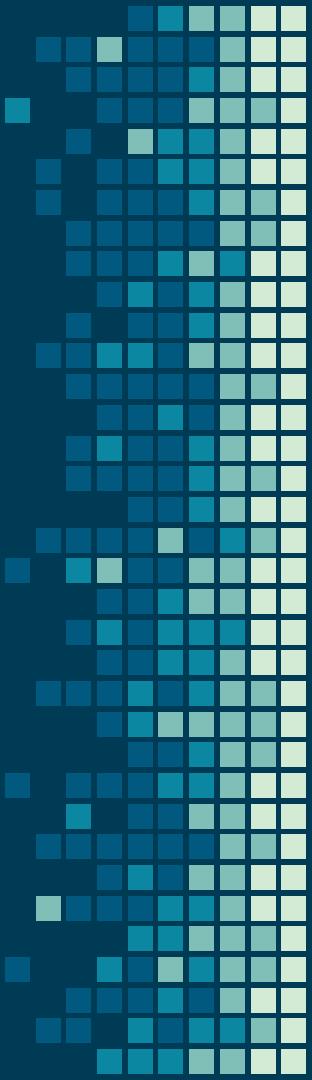




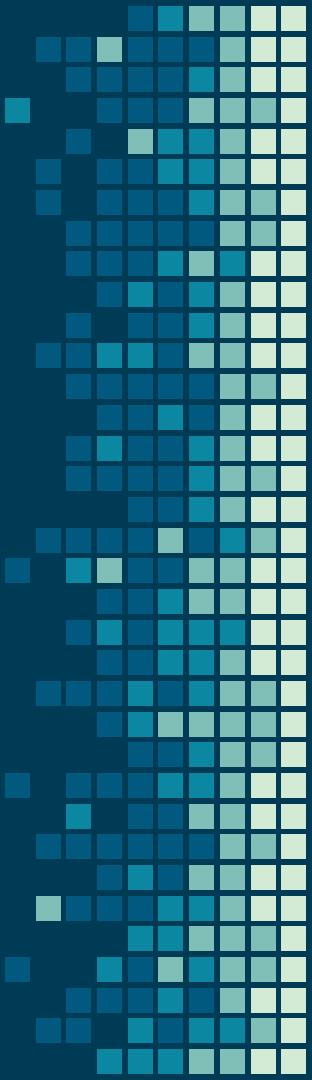
# ThingSpeak



# Mão na Massa!



# Máquina Virtual



<https://bit.ly/31LD4d6>

Fazer o Download da máquina Virtual

# Máquina Virtual

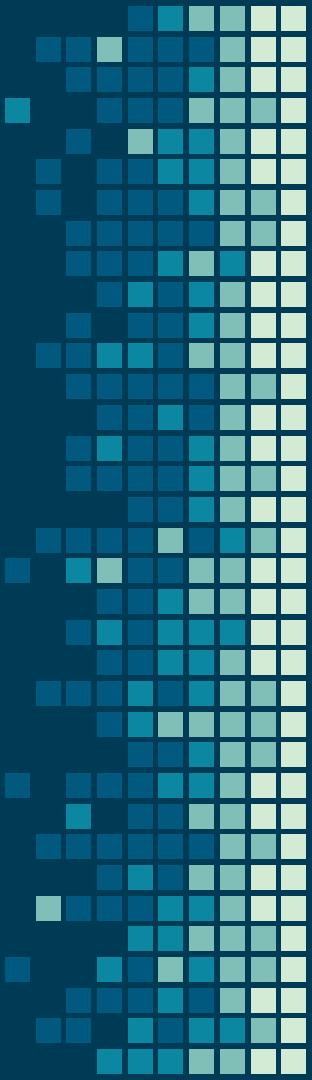
<https://www.virtualbox.org/>

Baixar a VM com a Oficina de IoT

Quem utiliza linux : user o comando

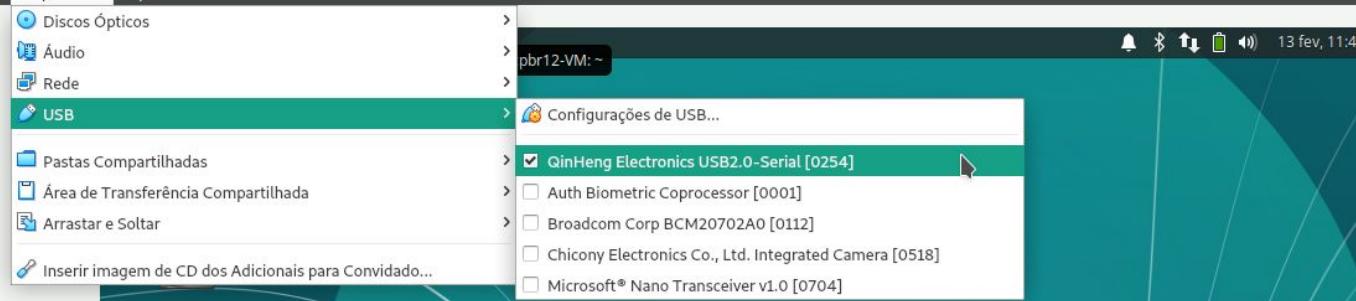
\$ usermod -a -G vboxusers seuusuário

# Máquina Virtual



Usuário : Oficina de IoT  
Senha : oficina

o Máquina Visualizar Dispositivos Ajuda



main.cpp - Untitled (Workspace) Terminal - cpbr12@cpbr12...

File Edit Selection View Go Debug Terminal Help main.cpp - Untitled (Workspace) - Visual Studio Code

EXPLORER main.cpp CPBR12-01-WiFi-Scan • src main.cpp CPBR12-04-tagoio • src PIO Home

OPEN EDITORS main.cpp CPBR12-01-WiFi-Scan • src main.cpp CPBR12-04-tagoio • src PIO Home

UNTITLED (WORKSPACE) main.cpp test .gitignore .travis.yml platformio.ini CPBR12-02-softap CPBR12-03-webserver CPBR12-04-tagoio .pioenvs .vscode include lib src main.cpp

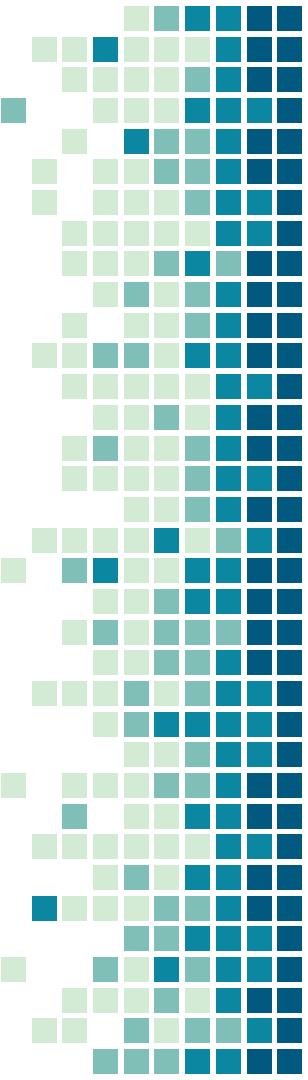
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Task - PlatformIO: M + - ^ x

```
/* Pedro Minatel - Sistemas Embarcados
 * Data: 16 de Setembro de 2015
 * Author: Pedro Minatel
 * Website: http://pedrominatec.com.br
 */
//Include da lib de Wifi do ESP8266
#include <ESP8266WiFi.h>
//Definir o SSID da rede WiFi
const char* ssid = "<<<YOUR_SSID>>>";
//Definir a senha da rede WiFi
const char* password = "<<<YOUR_PASSWORD>>>";
//Colocar a API Key para escrita neste campo
//Ela é fornecida no canal que foi criado na aba API Keys
String apiKey = "<<<YOUR_TAGO_KEY>>>";
const char* server = "api.tago.io";
int time_esp = 0;
WiFiClient client;
```

--- Miniterm on /dev/ttyUSB0 9600,8,N,1 ---  
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

Filter loop() setup() envia\_dados(v...) server

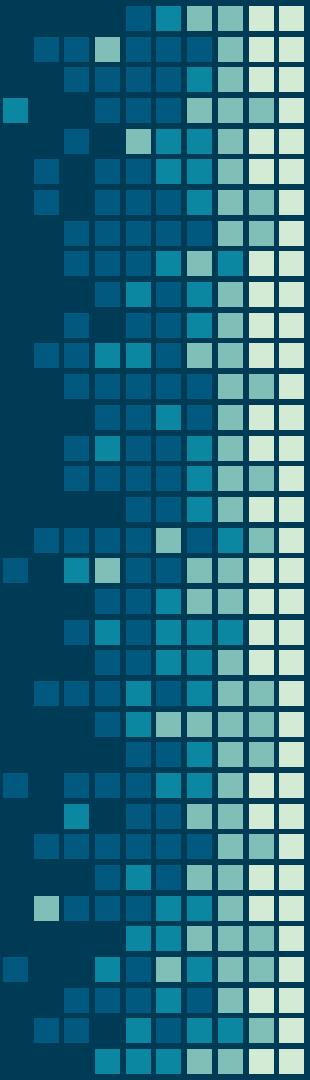
(Global Scope) Ln 11, Col 33 (9 selected) Spaces: 2 UTF-8 LF C++ Linux



# GitHub

<https://github.com/EstevesDouglas/Oficina-de-iot-cpbsb3>

# 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);  
}
```

# 01-wifi-scan.ino

Scaneamento de SSID.

# 02-softap.ino

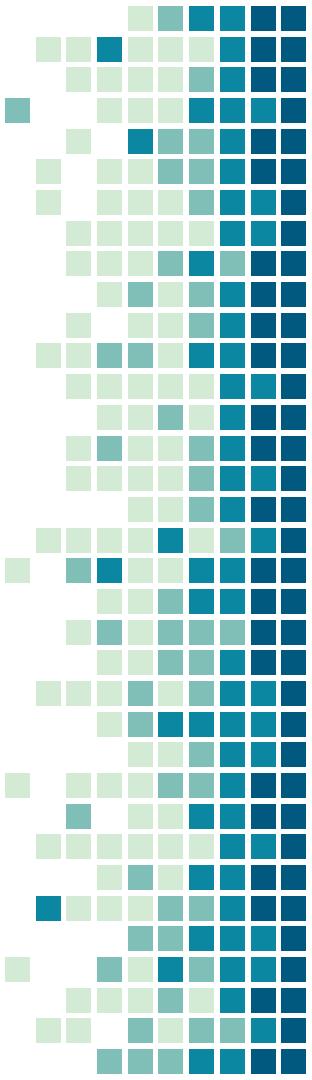
Monte seu AP

# 03-webserver.ino

Webserver de um ESP8266

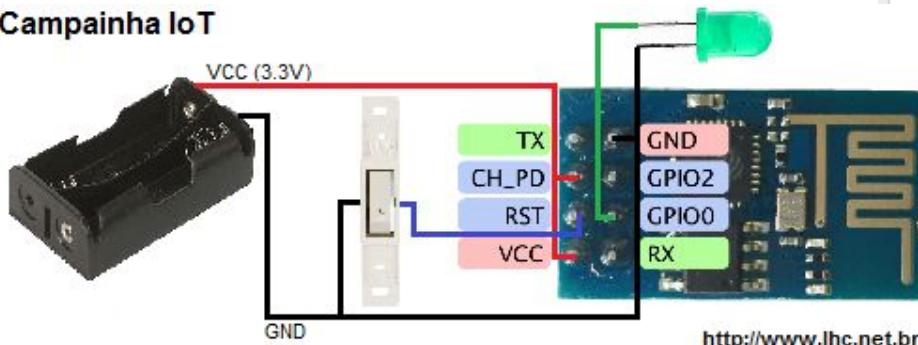
# 04-tagoio.ino

Conectando em uma plataforma



# Campainha IoT

Campainha IoT



Página do Projeto : [https://lhc.net.br/wiki/Campainha\\_IoT](https://lhc.net.br/wiki/Campainha_IoT)

Campainha LHC





# Referências

github

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

Hackerspace LHC

[www.lhc.net.br](http://www.lhc.net.br)

Pedro Minatel Blog

[www.pedrominatel.com.br](http://www.pedrominatel.com.br)

Portal Embarcados

[www.embarcados.com.br](http://www.embarcados.com.br)

Grupos makers

<https://t.co/sUjdiltWBI>



# Muito Obrigado

## Contatos !



19 98230-3616



@\_EstevesDouglas



douglas@iotmakers.com.br

