

# ESP8266: UMA INTRODUÇÃO AO IOT

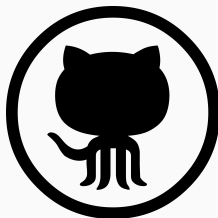
---

Gabriel Melo

October 4, 2018



Este material é um complemento do livro homônimo, disponível, junto com conteúdos de apoio do curso, em: [ESP8266: Uma introdução ao IoT](#), por Gabriel Melo.



# INTRODUÇÃO

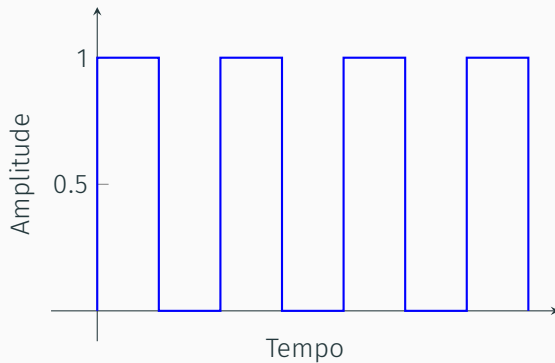
---

becomes

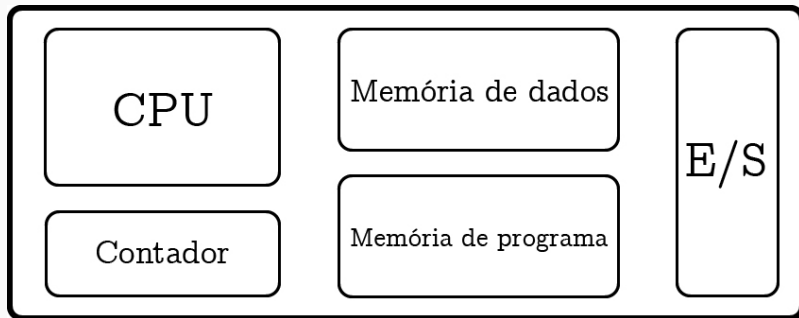


**Figure:** Sinal Analógico

becomes



**Figure:** Sinal Digital





Sensores

Atuadores



modos sta e ap (e o ap\_sta)

ESP8266

---

# O QUE É?







- \*boot.py script executado no boot do esp (possui configurações iniciais da placa, principalmente sobre comunicação)

- \*main.py script executado após o boot.py. Inicializa a árvore de execução do projeto











NodeMCU firmware

Lua

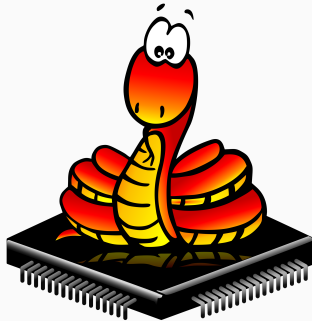
- Linguagem interpretada
- Padrão das placas NodeMCU
- Comunidade média



Arduino core firmware

Arduino/C++

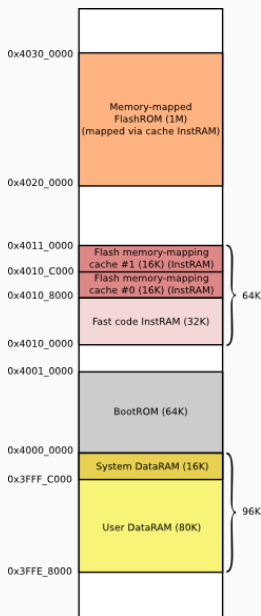
- Linguagem compilada
- Utilização da Arduino IDE
- Comunidade gigante



## Micropython Python

- Linguagem interpretada
- Linguagem acessível a iniciantes e poderosa para experientes
- Comunidade gigante
- upip (versão micro do pip)
- Otimização da memória RAM (???)

# MICROPYTHON - OTIMIZAÇÃO DA RAM



- Limitação da **InstRAM**
- **BootRAM** possui uma biblioteca para rotinas de suporte
- Essa biblioteca pode substituir instruções presentes na **InstRAM**
- **Micropython** faz uso disso e possui um Test coverage de 92%

Instruções sobre instalação, bem como a preparação do ambiente, ferramentas, dicas, exemplos e documentação externa estão disponíveis no próprio repositório remoto do material:

[https://github.com/GabrielMMelo/esp8266\\_course](https://github.com/GabrielMMelo/esp8266_course)

Memória de programa: - 1M de FlashROM - 64K de BootROM (que armazena um bootloader, o RTOS de fato e uma BIBLIOTECA COM VÁRIAS ROTINAS DE SUPORTE)

Memória de dados: - 96K de DataRAM (-16K do RTOS, -20K estáticos do Wi-Fi, -20k de dinâmico, - 10K de constantes) \* Constantes são armazenadas na RAM pois a ROM é alocada por blocos de 32-bits e, desta forma, o acesso byte a byte de uma string, por exemplo, poderá gerar um erro - 32K de InstRAM (+ 32K da FlashROM, mas a leitura é cerca de 10x mais lenta)

Acontece que grande parte da InstRAM já é usada por operações de tempo real (como handling Wi-Fi) e por ROTINAS usadas frequentemente. Essas rotinas, em muitos casos, já se encontram no BootROM. O Micropython possui um alto valor de test coverage quanto a isso, otimizando o uso da InstRAM

fonte:

conectar no REPL (interface com python) com o picocom picocom /dev/ttySporta\_com -b115200 (tudo junto)





# APLICAÇÕES

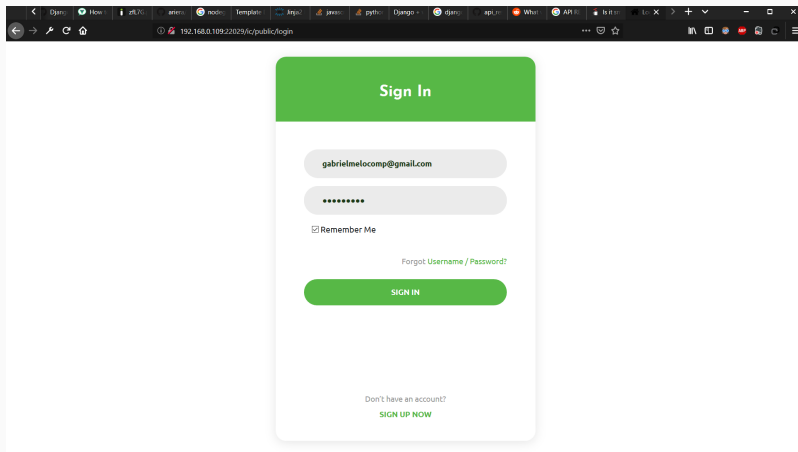
---

## SISTEMAS QUE ENVOLVAM ACESSO A REDE (*LOCAL OU INTERNET*)



<https://github.com/wendlers/micropython-mfrc522>

# SISTEMA DE CONTROLE RESIDENCIAL



The screenshot shows a web browser window with a dark theme. The address bar displays the URL `192.168.0.109:22029/ic/public/login`. The browser's tab bar shows several open tabs, including 'Django', 'How', '25.7%', 'arena', 'node', 'template', 'jquery', 'jazz', 'python', 'Django', 'django', 'api', 'What', 'API', 'Is it in', and 'Lo: X'. The main content area features a 'Sign In' form with a green header. The form includes two input fields: the first contains the email `gabrielmelcomp@gmail.com`, and the second contains masked characters `*****`. Below the password field is a checkbox labeled 'Remember Me'. A link 'Forgot Username / Password?' is positioned below the checkbox. A green 'SIGN IN' button is located at the bottom of the form. At the very bottom of the form, there is a link 'Don't have an account?' followed by a green 'SIGN UP NOW' button.

**Sign In**

`gabrielmelcomp@gmail.com`

`*****`

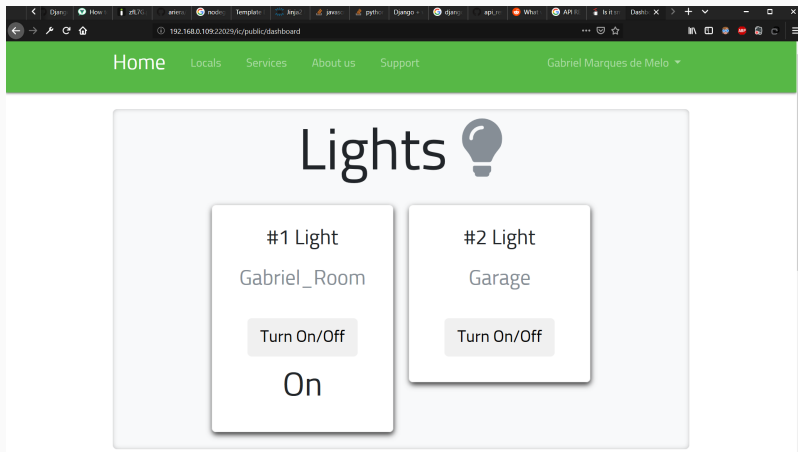
☒ Remember Me

[Forgot Username / Password?](#)

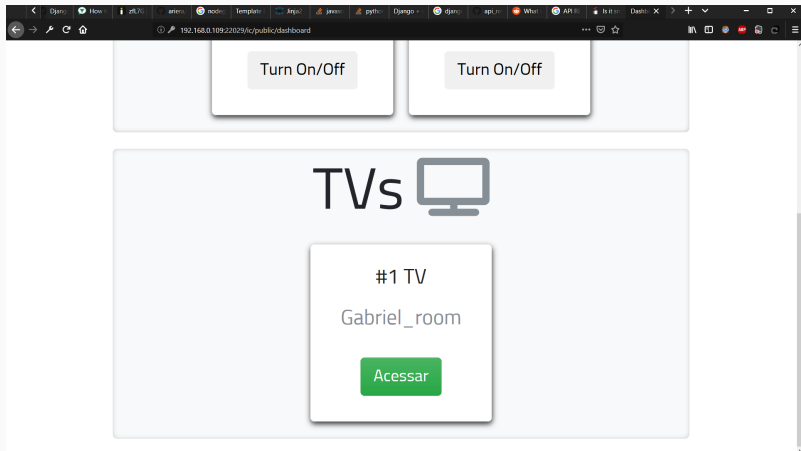
**SIGN IN**

[Don't have an account?](#)

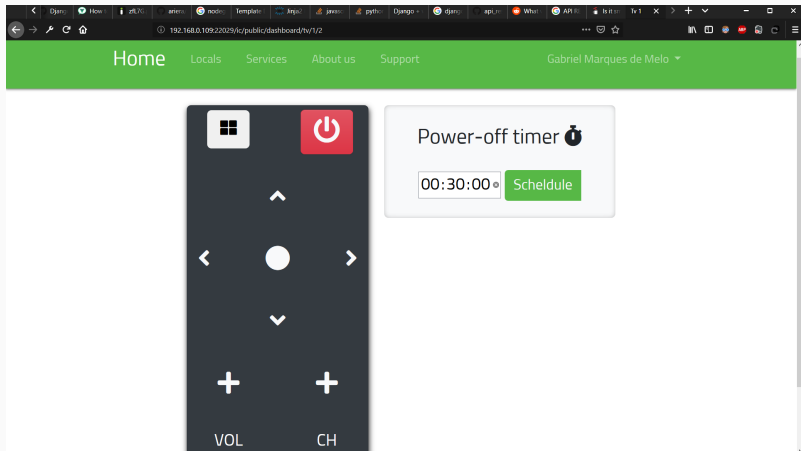
**SIGN UP NOW**



# SISTEMA DE CONTROLE RESIDENCIAL



# SISTEMA DE CONTROLE RESIDENCIAL





## REFERÊNCIAS

---

- **ESP8266: Uma introdução ao IoT**, disponível em:  
[https://github.com/GabrielMMelo/esp8266\\_course](https://github.com/GabrielMMelo/esp8266_course)
- **Documentação Micropython:**  
<https://docs.micropython.org/en/latest/esp8266/tutorial/intro.html>
- **Pacotes upip** (nem todos possuem suporte para port ESP8266):  
<https://pypi.org/search/?q=micropython->
- **Discussão acerca do uso do Micropython com ESP8266:**  
<https://www.kickstarter.com/projects/214379695/micropython-on-the-esp8266-beautifully-easy-iot/posts/1501224>

PERGUNTAS?