FirstSteps Documentation

Release 1.0.0

Automacao IOT

fev 14, 2019

Getting Started:

1 Guia Básico **3**
- 1.1 Pré-requisitos . . . 3

2 Cadastro **5**
- 2.1 Dispositivo . . . 5
- 2.2 Recurso . . . 6

3 API Automação-IOT **9**
- 3.1 Outras Plaformas . . . 9

4 MicroPython **13**
- 4.1 Documentação MicroPython . . . 13

5 Projeto **15**
- 5.1 Pré-requisitos . . . 15
- 5.2 Gravando Firmware . . . 15
- 5.3 Nodemcu Flash . . . 15
- 5.4 WebRepl Client . . . 16
- 5.5 Cliente Putty . . . 17
- 5.6 Site . . . 18
- 5.7 Criando arquivo main.py . . . 18
- 5.8 Configurando o ESP8266 . . . 19
- 5.9 Enviando arquivos . . . 19
- 5.10 Testando o Device e o Recurso . . . 20

6 Links Externos **23**
- 6.1 Links . . . 23

If you are looking for the english documentation go to here.

CAPÍTULO 1

Guia Básico

1.1 Pré-requisitos

Para iniciarmos nosso primeiro projeto na Automação-IOT, deveremos:

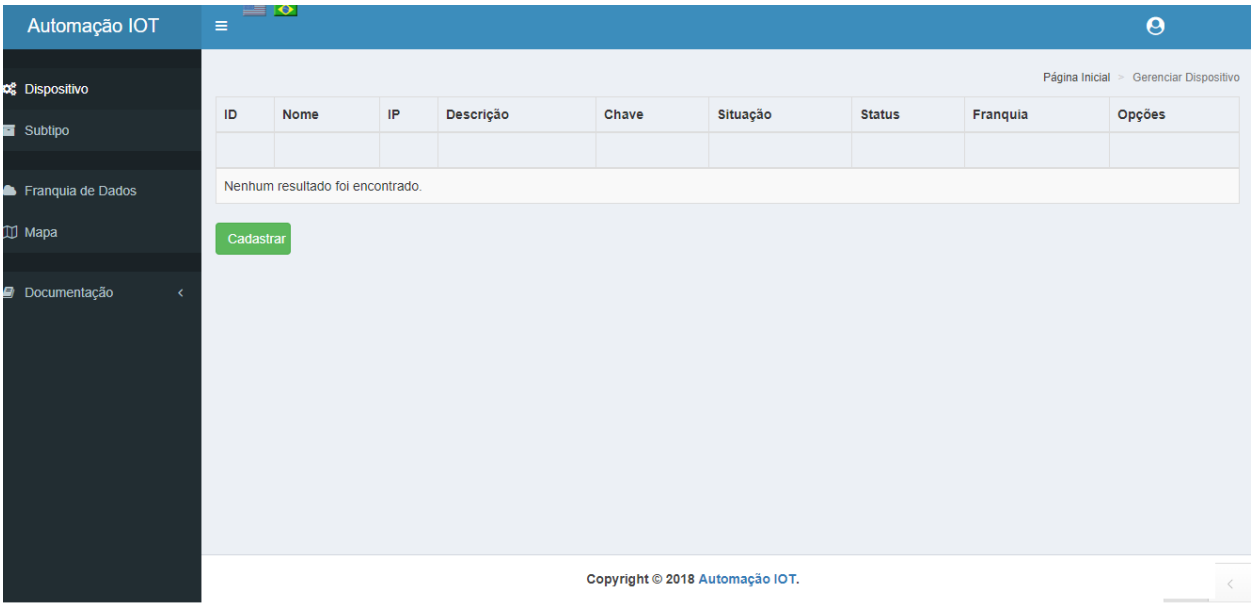
1. Criar uma conta no site Automacao-iot;
2. Dowload o binário do MicroPython ESP8266 ou ESP32 com as libs da Automação IoT;
3. Disponibilizar um ESP8266 para gravarmos o binário; e
4. Criar o projeto Rele para testarmos o firmware MicroPython IoT.

CAPÍTULO 2

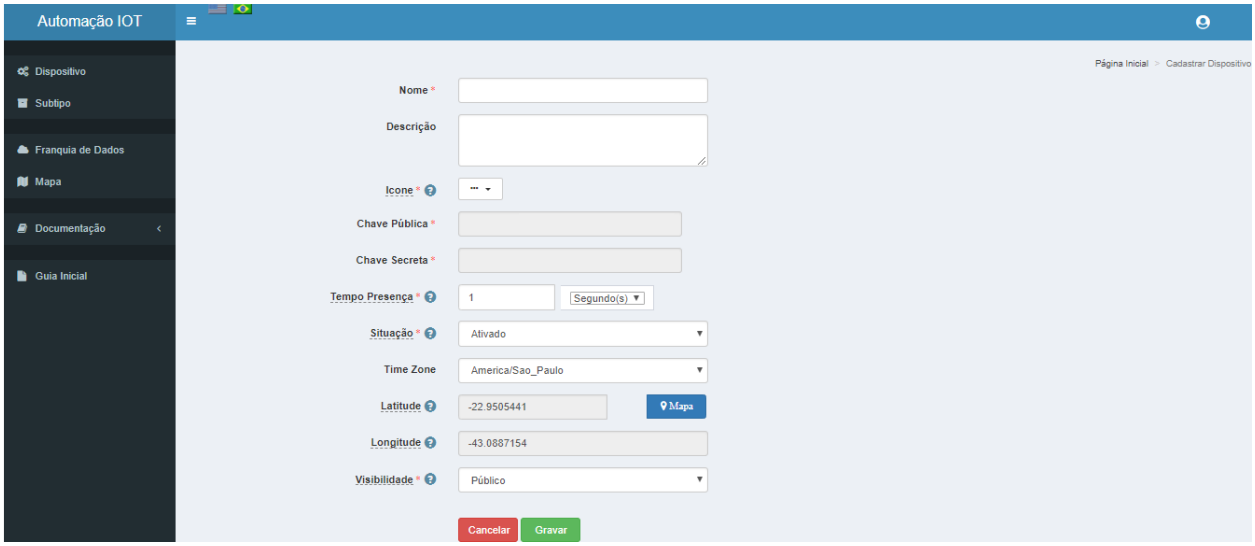
Cadastro

2.1 Dispositivo

Após termos criado a conta e realizado o login no site da Automação-IOT , veremos a Dashboard de Dispositivos.



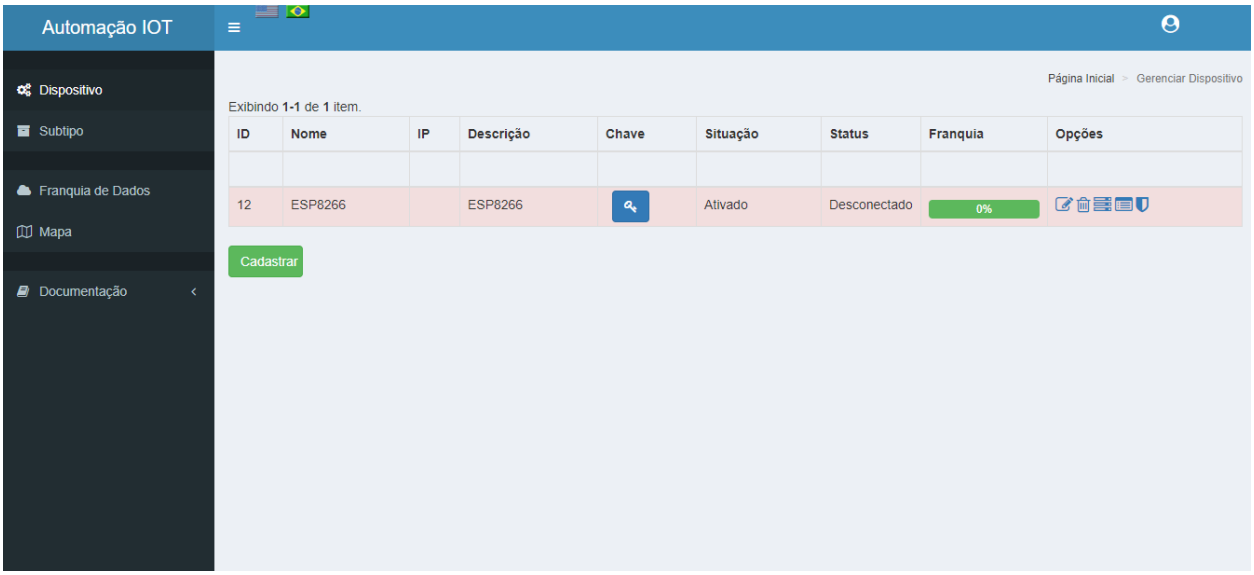
Deveremos criar nosso primeiro Dispositivo, opção **Cadastrar**, que será a princípio o ESP8266.



Acesse a documentação do Site Automação-IOT, para obter maiores informações.

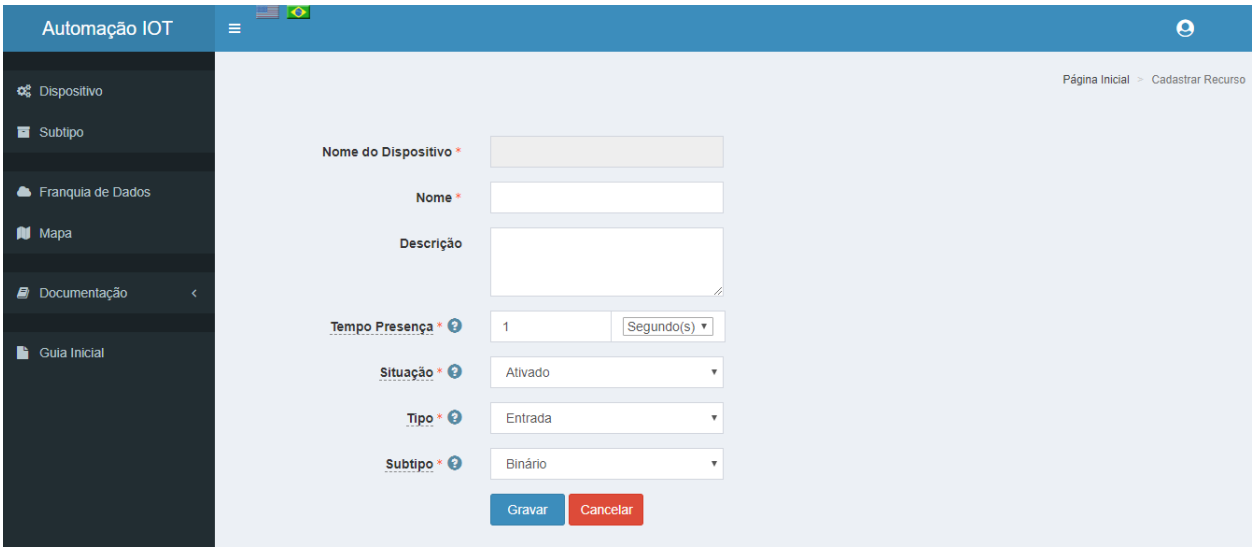
2.2 Recurso

Após criarmos nosso Dispositivo, veremos a Dashboard de Dispositivos, com o nosso Dispositivo criado:



Deveremos clicar no ícone de **Opções** denominado **Gerenciar Recursos**. Veremos a Dashboard de Recursos:

Deveremos criar o recurso **Rele**, opção cadastrar:



Após criarmos nosso Recurso, veremos a Dashboard de Recursos, com o Recurso **Rele** criado:

CAPÍTULO 3

API Automação-IOT

A Automaçao-IOT utiliza um conjunto de rotinas padronizadas de programação denominado de API que serve de acesso para a plataforma IOT baseado na WEB. Através da API que estabelecemos comunicação de entrada e/ou saída com os Dispositivos e/ou Recursos.

3.1 Outras Plaformas

A API da Automação_IOT, possibilita a conexão de outras plataformas a base de dados IOT. Vamos implementar o exemplo utilizado no SDK do ESP8266, de forma genérica para ser aplicado ao seu projeto em outras plataformas.

3.1.1 Get Device

Obtém informações sobre um determinado Dispositivo. Este EndPoint retorna dados informacionais do Dispositivo, tal como id, dono, nome e descrição do dispositivo

```
GET api/device/{PUBLIC_KEY}
```

Requisição

```
GET api/device/B07CFA0A9204228A30C68FB346C407E7  HTTP/1.1
Authorization: Bearer B8CE671CFE226CC190D5478E8E5A3CD7FB59D87F872278E59E4E6847B5E4B2F1
Host: https://automacao-iot.com.br/api
```

Resposta

```
{
        "success": true,
        "payload": {
                "id_device": 12,
                "id_user": 12,
                "name": "ESP8266",
```

(continues on next page)

(continuação da página anterior)


```
                "icon_name": "fa-microchip",
                "description": "ESP8266",
                "ip": "10.1.1.11",
                "timezone": "America\/Sao_Paulo",
                "latitude": "-35.2837000",
                "longitude": "-47.5343000",
                "public_key": "B07CFA0A9204228A30C68FB346C407E7",
                "api_requests_usage": 30653,
                "api_network_usage": 4060992,
                "api_network_income_usage": 4060875,
                "api_network_outgoing_usage": 117,
                "lifetime": 15,
                "lifetime_updated_at": "2018-06-05 17:30:46",
                "created_at": "2018-06-04 19:09:39",
                "updated_at": "2018-06-04 19:09:39",
                "private": 0,
                "active": 1,
                "is_alive": false,
                "is_dead": true,
                "last_sys_call": null
        },
        "message": ""
}
```


3.1.2 Get Resource Last Feeds

Recupera o último Registro de Feeds do Recurso.

```
GET/api/device/{PUBLIC_KEY}/resource/{ID_RESOURCE}/feeds/last
```

Requisição

```
GET api/device/B07CFA0A9204228A30C68FB346C407E7/resource/6/feeds/last  HTTP/1.1
Authorization: Bearer B8CE671CFE226CC190D5478E8E5A3CD7FB59D87F872278E59E4E6847B5E4B2F1
Host: https://automacao-iot.com.br/api
```

Resposta


```
{
        "success": true,
        "payload": {
                "id_resource_feed": 14603,
                "id_resource": 6,
                "raw_data": "0",
                "created_at": "2018-06-05 17:13:06"
        },
        "message": ""
}
```


3.1.3 Create Resource Feeds

Insere um novo Registro de Feeds no recurso do Dispositivo.

```
POST /api/device/{PUBLIC_KEY}/resources/feeds
```

Requisição

```
POST api/device/B07CFA0A9204228A30C68FB346C407E7/resources/feeds  HTTP/1.1
Authorization: Bearer B8CE671CFE226CC190D5478E8E5A3CD7FB59D87F872278E59E4E6847B5E4B2F1
Host: https://automacao-iot.com.br/api

Body:
[
        {
                "id_resource": 6,
                "input": 1
        }
]
```

Resposta

```
{
        "success": true,
        "payload": {
                "success": [{
                        "raw_data": true,
                        "id_resource": 6,
                        "created_at": "2018-06-07 12:33:01",
                        "id_resource_feed": 14604
                }],
                "failed": []
        },
        "message": ""
}
```

CAPÍTULO 4

MicroPython

4.1 Documentação MicroPython

O MicroPython é executado em uma variedade de sistemas e plataformas de hardware. No link você poderá ler a documentação geral que se aplica a todos os sistemas, bem como informações específicas sobre as várias plataformas, também conhecidas como ports - nas quais o MicroPython é executado.

CAPÍTULO 5

Projeto

5.1 Pré-requisitos

Para iniciarmos a execução do nosso projeto Rele IoT, deveremos:

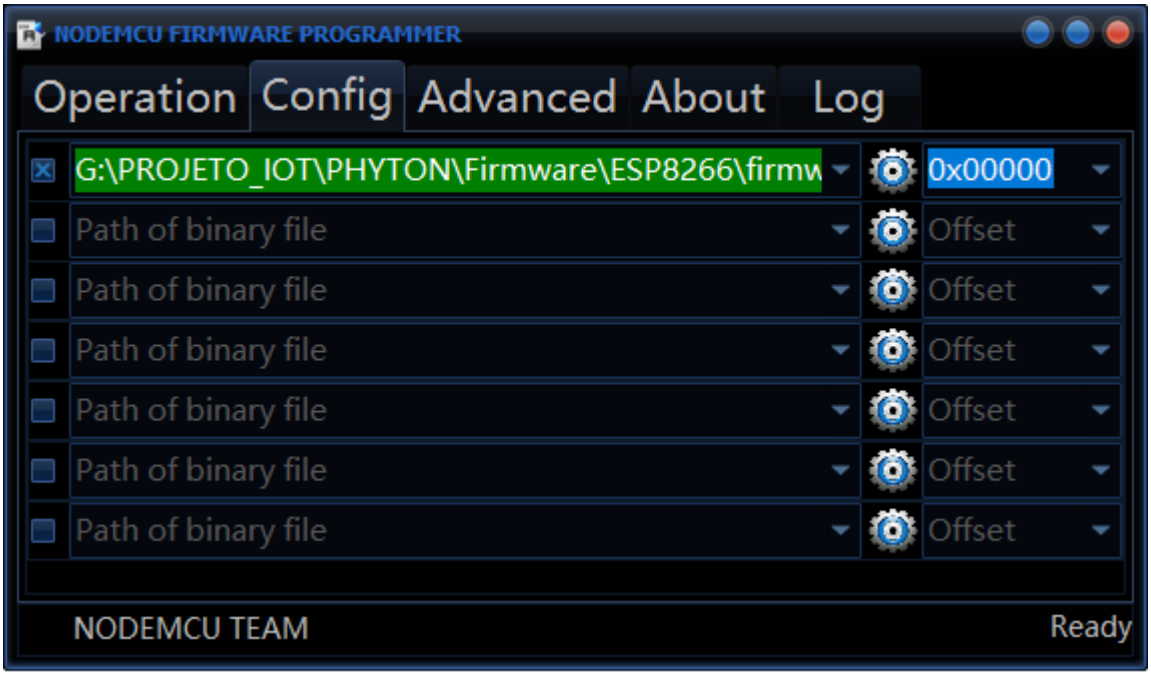
1. Dowload Firmware ESP8266;
2. Dowload Nodemcu Flash;
3. Dowload WebRepl Client;
4. Dowload SDK ESP8266/32;
5. Dowload Putty ; e
6. Dowload Zip ESP-SDK-MicroPython.

5.2 Gravando Firmware

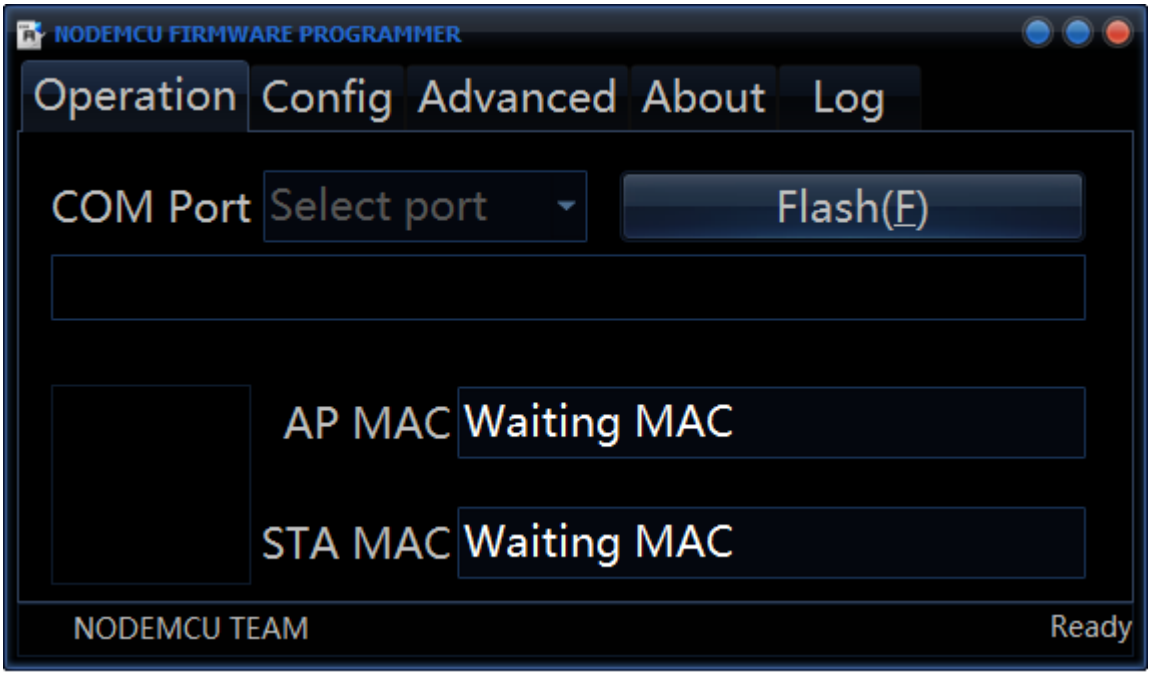
Antes de iniciarmos o projeto deveremos gravar o firmware específico, no nosso caso o ESP8266. Para isso deveremos baixar o programa de gravação de firmware o Nodemcu Flash.

5.3 Nodemcu Flash

Para iniciarmos a gravação do firmware no ESP8266, deveremos configurar o Nodemcu, para isso, selecione a aba Config, e configure o mesmo com a seleção do arquivo do firmware.



Deverá ser iniciado a gravação do firmware, conecte o ESP8266 na USB do computador (use o cabo microUSB x USB), selecione a porta de uso em COM Port e inicie a gravação em Flash(F).

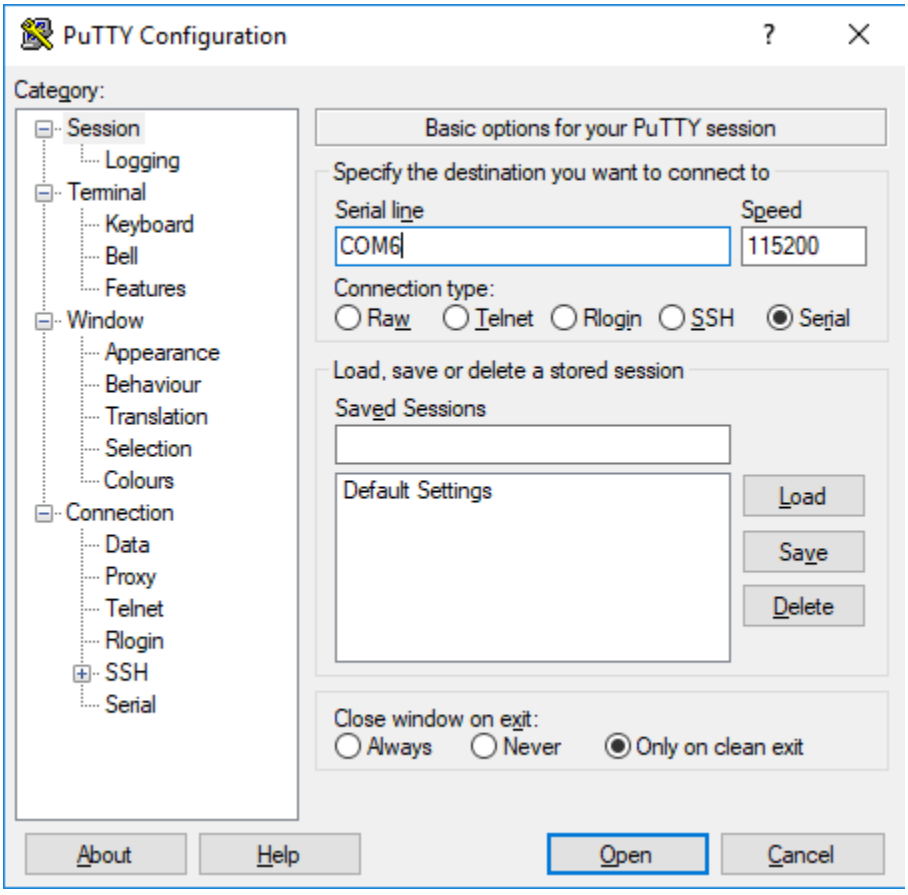


5.4 WebRepl Client

WebRepl é o cliente para a comunicação e envio/recebimento de arquivos sob WebSockets, com o MicroPython. Deveremos clonar o git para uso posteriore.

5.5 Cliente Putty

O PuTTY é um cliente SSH e telnet, desenvolvido para a plataforma Windows.



5.6 Site

Siga os passos do tópico Cadastro.

5.7 Criando arquivo main.py

Deveremos criar o arquivo **main.py**, conforme descrito abaixo:

```
from Device import Device
from Rele import Rele

device = Device("Public Key","Secret Key")
device.SYS_CPU_160MHZ()
device.setNetworkConfig('SSID','PWD')

rele1 = Rele(id do Recurso,device.GPIO02,Rele.OPEN,1000)
device.start()
```

Deverão ser copiados do site as chaves Pública e Secreta do Dispositivo, através do ícone do Dashboard **Chave**:

Deverá ser modificado a seguinte linha de código:

```
device = Device("Public Key","Secret Key")
```

Deverá ser copiado do site **Dashboard Recurso** o **id do Recurso**. No meu caso **id do Recurso** = 1

```
rele1 = Rele(1,device.GPIO02,Rele.OPEN,1000)
```

Para maiores informações consulte a documentação do SDK ESP8266.

5.8 Configurando o ESP8266

Para acessarmos o ESP através do WebSocket, deveremos utilizar o Putty através da comunicação serial (USB), assim que obtivermos o prompt deveremos executar os seguintes comandos abaixo:

```
import webrepl_setup
```

Deveremos informar a senha a ser utlizada e em seguida confirmar o reset do ESP.

```
import wifi_setup;
```

Deveremos informar SSID e PWD da rede WIFI.

5.9 Enviando arquivos

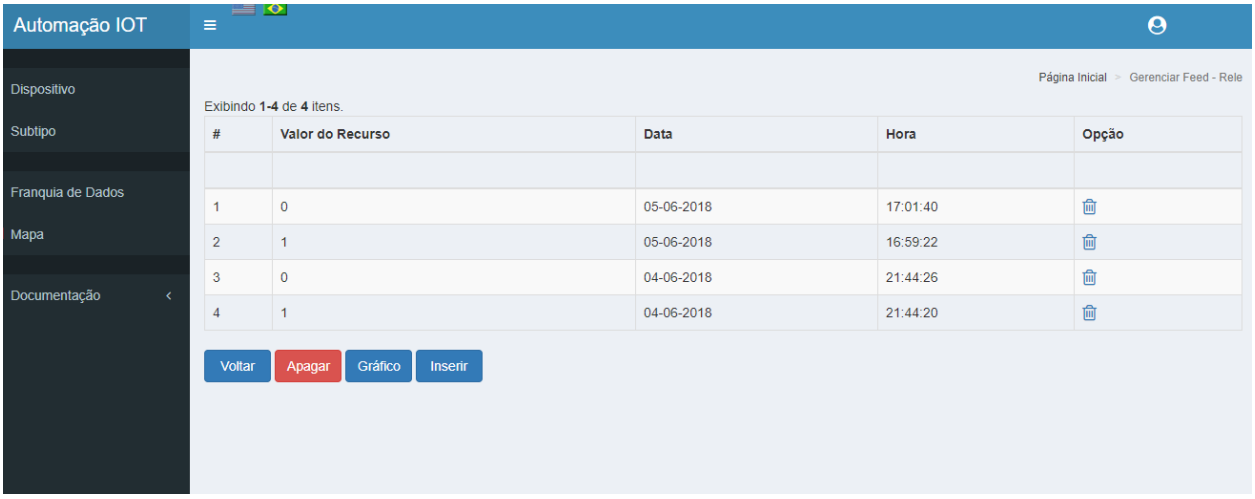
Após criarmos o arquivo main.py, deveremos enviá-lo para o ESP8266 bem como o arquivo Rele.py (ESP-SDK-MicroPython), através do WebRepl Client.

Configurar com o IP do ESP8266 que foi informado no tópico _Configurando o ESP8266, informe a senha solicitada que foi cadastrado (import webrepl_setup), em seguida devremos enviar os arquivos: main.py e Rele.py.

5.10 Testando o Device e o Recurso

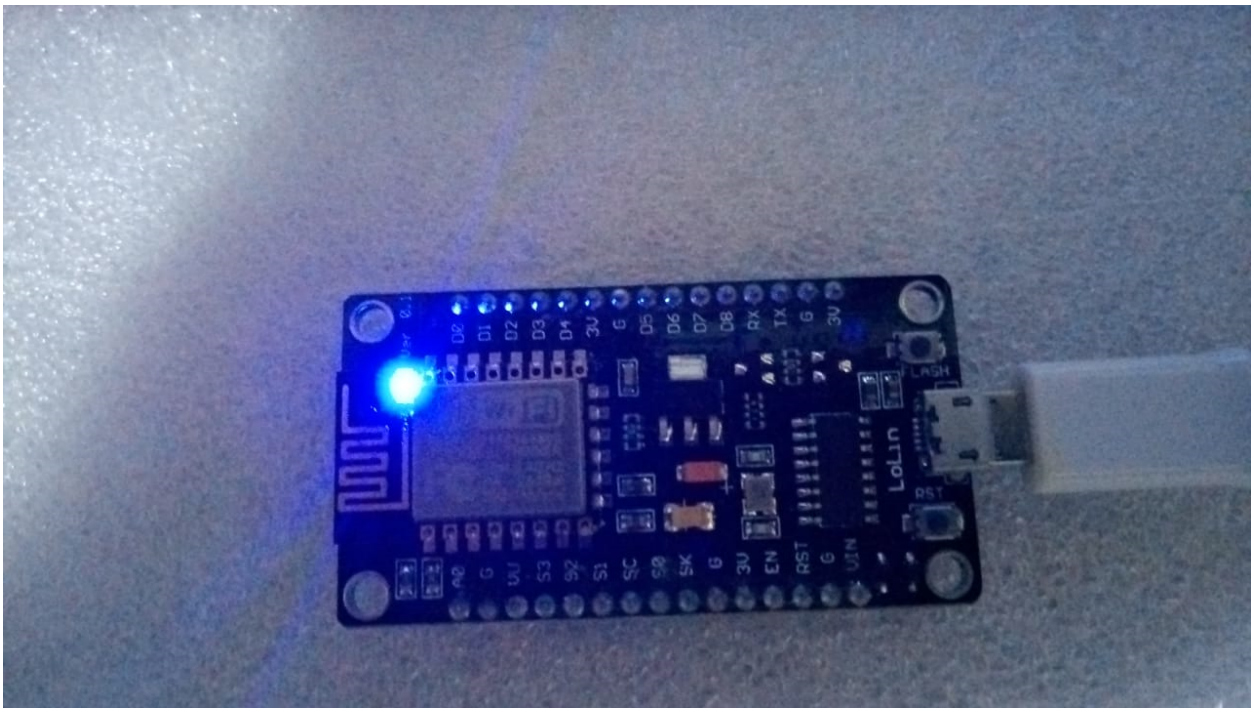
Após concluído as etapas listadas acima, deveremos resetar o ESP8266 e verificar no site o seu correto funcionamento.

Modificando o Feed do recurso:

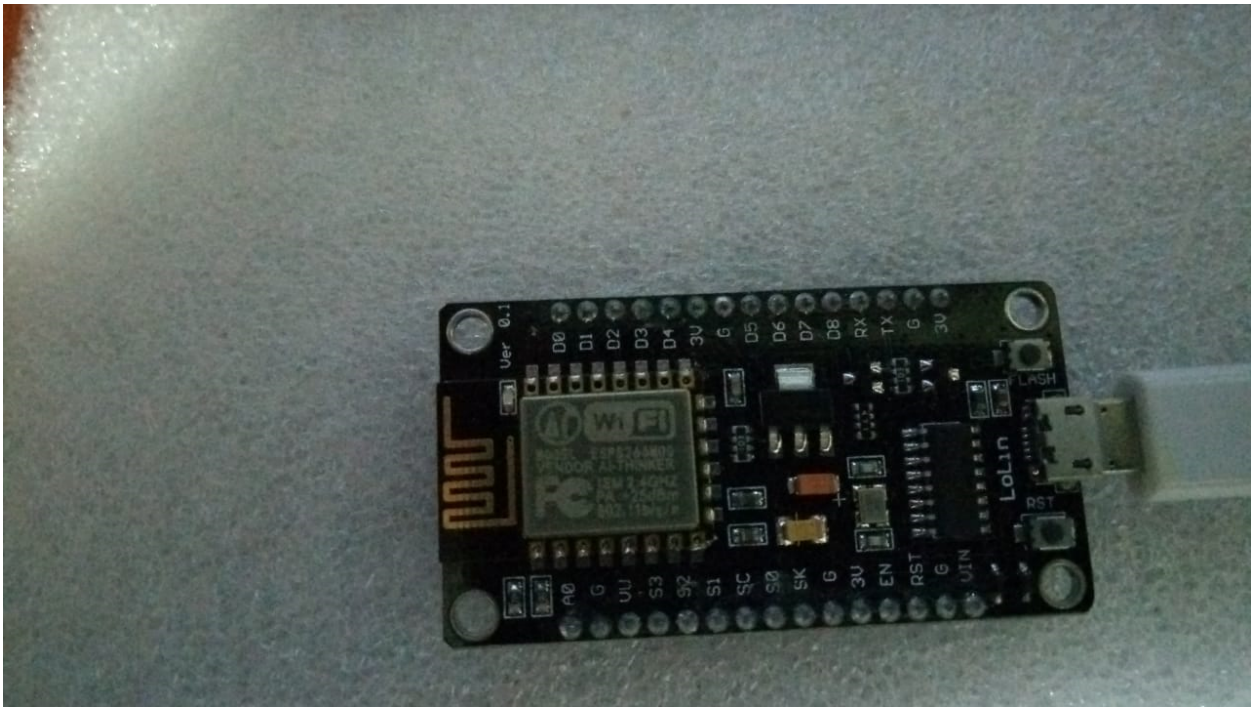


Selecionar o botão **Inserir**

Ao selecionarmos **Ativar** e Gravar, iremos observar o Led do ESP8266 acender.



Ao selecionarmos **Desativar** e Gravar, iremos observar o Led do ESP8266 apagar.



CAPÍTULO 6

Links Externos

6.1 Links

Documentação API Automação-IOT.

Documentação Site Automação-IOT.

Documentação Micropython.

Nodemcu Flash.