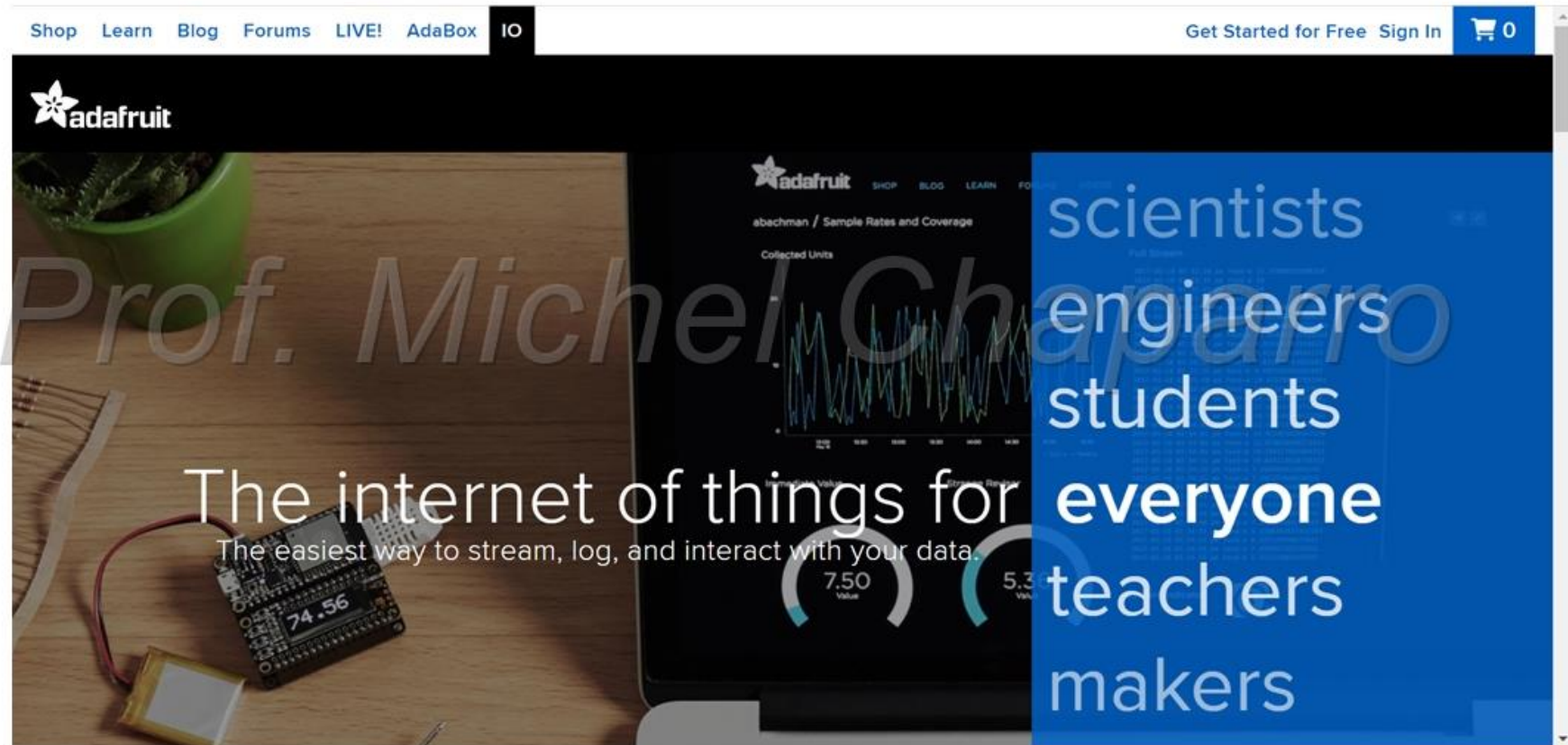


Plataforma IOT



Plataforma IOT

Realizar o Cadastro na Plataforma:

Prof. Michel Chaparro
<https://io.adafruit.com/>

Plataforma IOT

Criar um Feed:

Create a new Feed

Name

saida

Maximum length: 128 characters. Used: 5

Description

feed para a saida digital

Cancel

Create

Plataforma IOT

Criar uma Dashboard:

Create a new Dashboard

Name

Led

Description

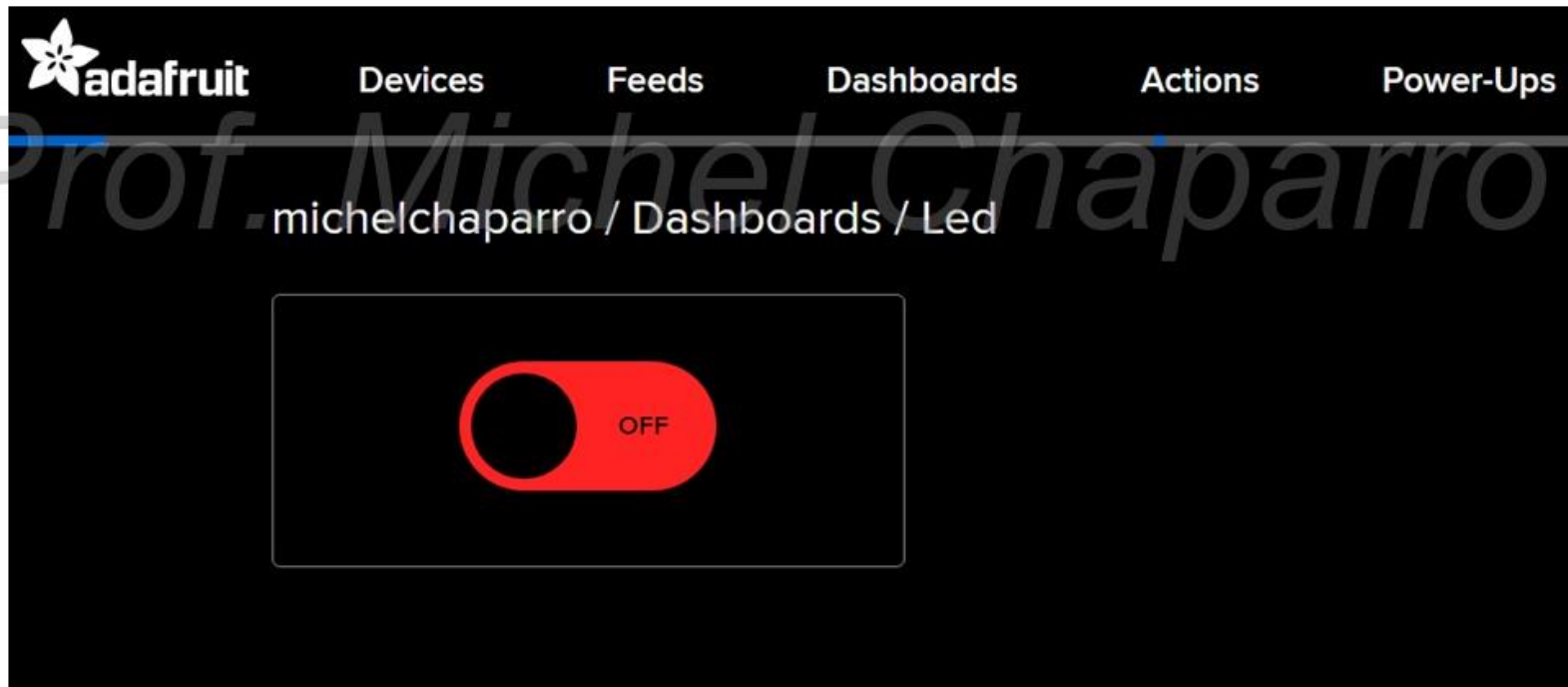
Status do Led

Cancel

Create

Plataforma IOT

Editor Dashboard:



My Watermark (unregistered version)

Plataforma IOT

API Key:

YOUR ADAFRUIT IO KEY ×

Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.



If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.

Username

Active Key

REGENERATE KEY

IDE Arduino

Instalar as Bibliotecas na IDE do Arduino:

✓ Adafruit IO Arduino

✓ Adafruit MQTT Library

Programação

```
// configurar com as credenciais do Adafruit IO
#include "config.h"

#define LED_PIN 2

// configura o feed
AdafruitIO_Feed *digital = io.feed("saida");
```


Programação

```
void setup()
{
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(115200);
  // espera o monitor serial se abrir
  while(! Serial);

  // conecta como io.adafruit.com
  Serial.print("Conectando com Adafruit IO");
  io.connect();
}
```

Programação

```
// configura um gerenciador de mensagens
// para o feed 'digital'.
// a função handleMessage será chamado sempre
// que uma mensagem for recebido de adafruit io.
digital->onMessage(handleMessage);
// aguarda a conexão
while(io.status() < AIO_CONNECTED)
{
    Serial.print(".");
    delay(500);
}
```

Programação

```
// estando conectado  
Serial.println();  
Serial.println(io.statusText());  
digital->get();  
}
```

Prof. Michel Chaparro

Programação

```
void loop()  
{  
    // mantém o cliente conectado  
    io.run();  
}
```

Prof. Michel Chaparro

Programação

```
// esta função é chamada sempre que uma mensagem
// de feed 'digital' é recebido da Adafruit IO
void handleMessage(AdafruitIO_Data *data)
{
    Serial.print("recebido <- ");
    if(data->toPinLevel() == HIGH)
        Serial.println("HIGH");
    else
        Serial.println("LOW");
    digitalWrite(LED_PIN, data->toPinLevel());
}
```


Arquivo config.h

```
// configuração Adafruit
#define IO_USERNAME  "michelchaparro"
#define IO_KEY       "aio_RcEK44LxGbqjv4MlopMEccvk3gzu"

// configuração wifi
#define WIFI_SSID "IOT"
#define WIFI_PASS "IOT20222"

#include "AdafruitIO_WiFi.h"
```


Arquivo config.h

```
#if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) || \
    defined(ADAFRUIT_PYPORTAL)
// Configure the pins used for the ESP32 connection
#if !defined(SPIWIFI_SS) // if the wifi definition isnt in the board variant
// Don't change the names of these #define's! they match the variant ones
#define SPIWIFI SPI
#define SPIWIFI_SS 10 // Chip select pin
#define NINA_ACK 9 // a.k.a BUSY or READY pin
#define NINA_RESETN 6 // Reset pin
#define NINA_GPIO0 -1 // Not connected
#endif
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS, SPIWIFI_SS,
                  NINA_ACK, NINA_RESETN, NINA_GPIO0, &SPIWIFI);
#else
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
#endif
```