

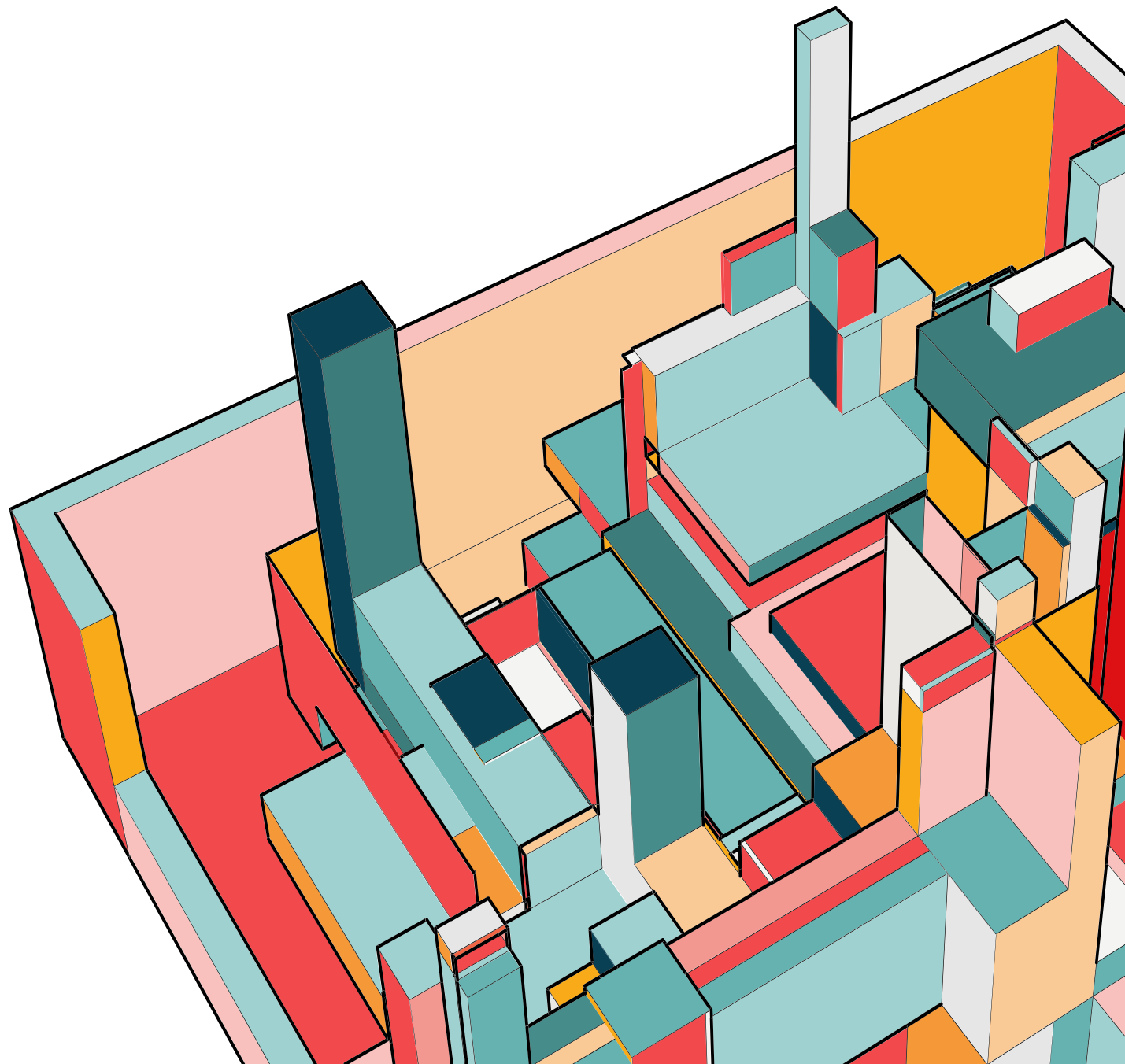


MINICURSO

**ARDUINO/NODEMCU
E IOT**

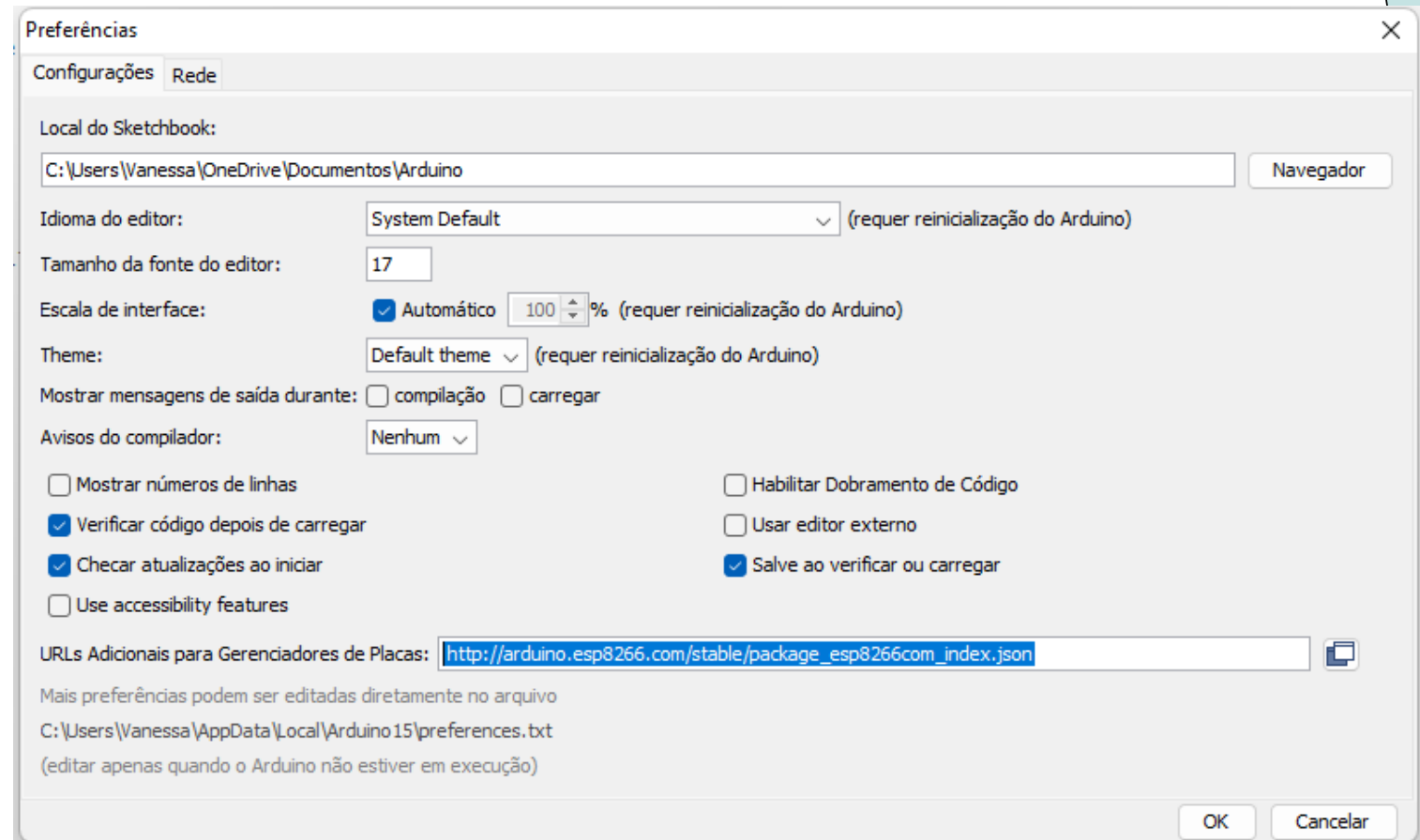
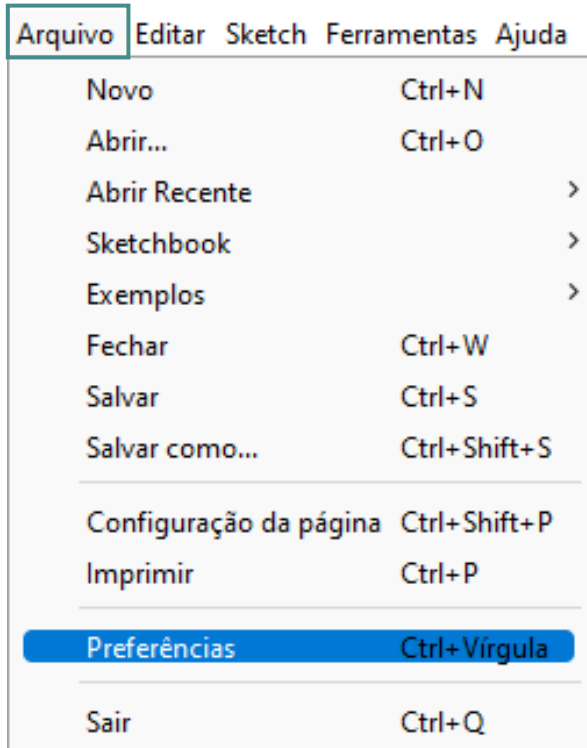
AULA 1

- ✓ Configuração da IDE
- ✓ Cadastro no Blynk
- ✓ Projeto Introdutório



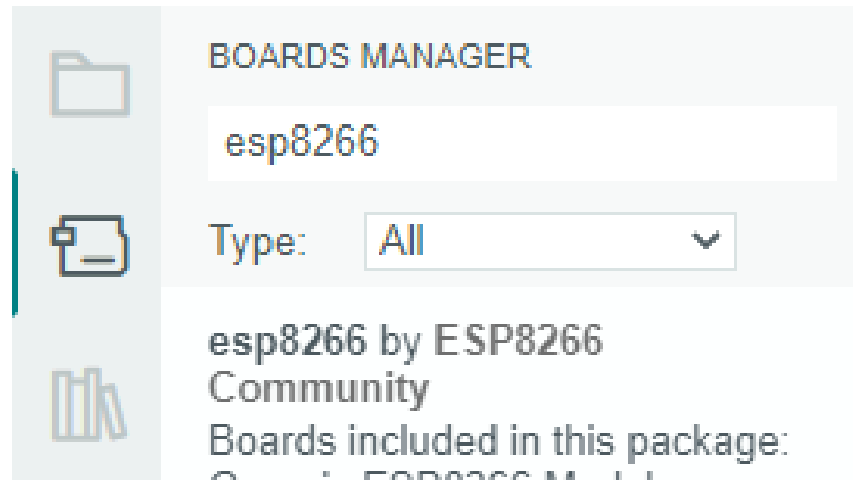
IDE - ARDUINO

http://arduino.esp8266.com/stable/package_esp8266com_index.json



ENCONTRAR A PLACA

Tools > Board > Boards Manager



Tools > Board

NodeMCU 0.9 (ESP-12 Module)

NodeMCU 1.0 (ESP-12E Module)

Olimex MOD-WIFI-ESP8266(-DEV)

INSTALANDO BIBLIOTECAS DOS SENSORES

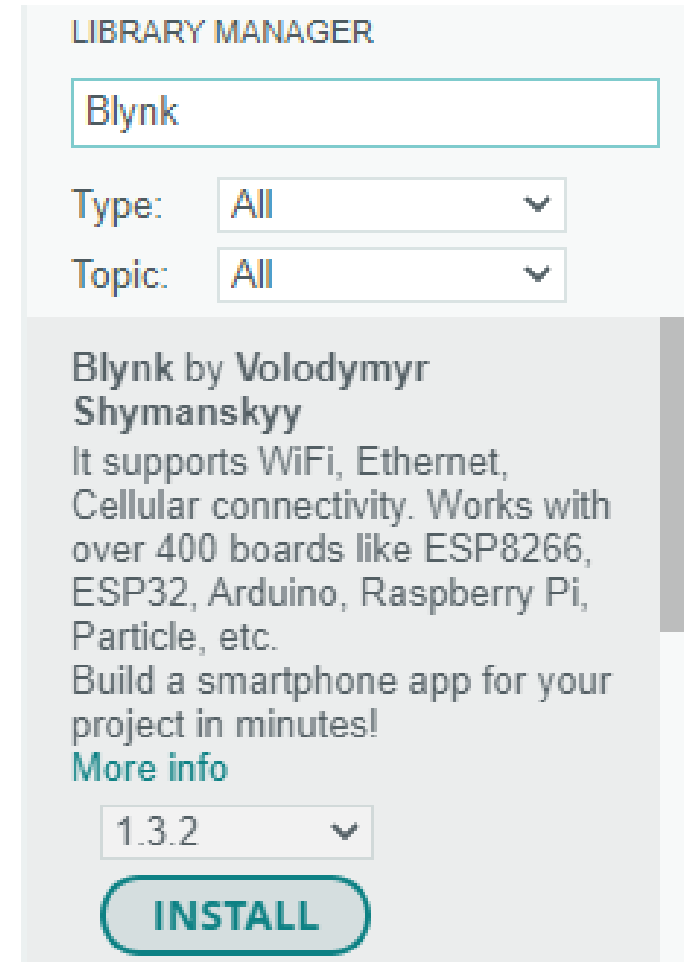


- Baixar do github
 - ✓ BMP180.ZIP
 - ✓ DHT-SENSOR-LIBRARY.ZIP

Sketch > Include Library > Add .ZIP Library

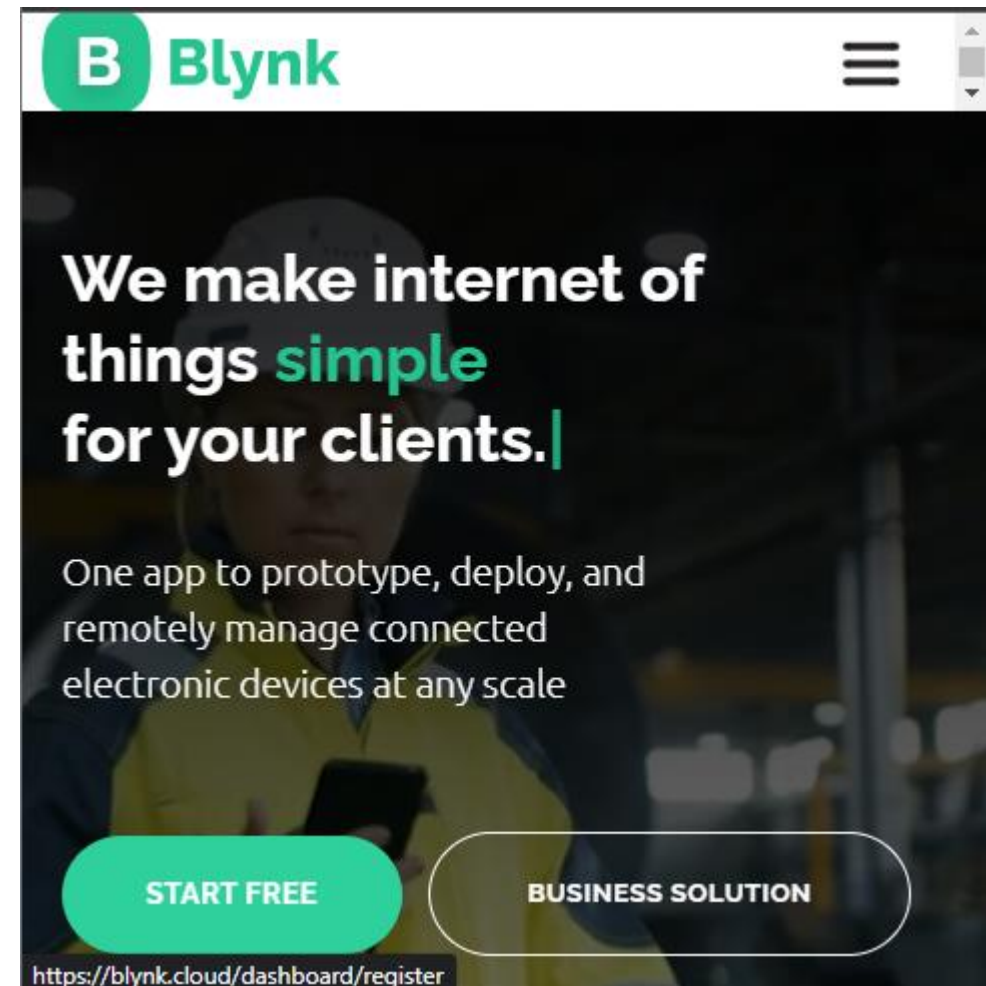
Sketch > Include Library > Manage Libraries

**** INSTALL ALL**

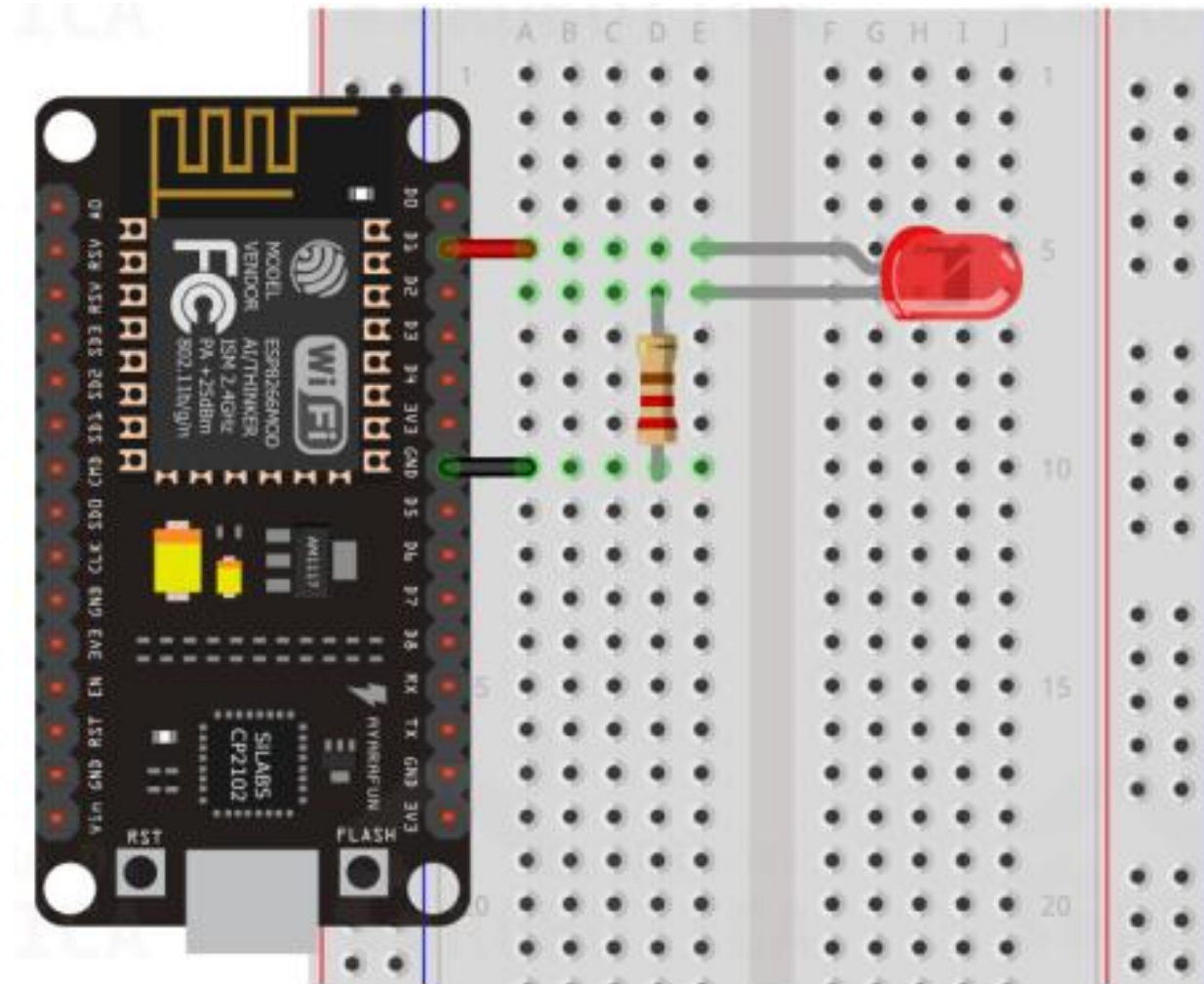


CRIAR CONTA NO BLYNK

<https://blynk.io/>



PROJETO 1 - PISCA LED COM BLYNK



CRIAR PROJETO NO BLYNK

The image shows the Blynk web interface. On the left is a sidebar with a green header containing a 'B' logo and a search icon. Below the header are links for 'My organization - 7480BJ', 'MY TEMPLATES', 'My Templates', 'BLUEPRINTS', and 'All Blueprints'. A 'BETA' badge is visible next to 'All Blueprints'. The main content area has a 'Templates' section with a '+ New Template' button and a search bar. To the right is a 'Create New Template' form with fields for 'NAME' (containing 'Semana do CEC'), 'HARDWARE' (a dropdown menu with 'ESP8266' selected), 'CONNECTION TYPE' (a dropdown menu with 'WiFi' selected), and 'DESCRIPTION' (containing 'Description'). Below the form is a navigation bar with tabs: 'Home', 'Datastreams' (highlighted with a green underline), 'Web Dashboard', 'Automations', 'Metadata', and 'Eve'. A small '0 / 128' character count is visible at the bottom right of the form. At the bottom of the page, there is a 'Cancel' button and a 'Done' button. A modal window is open over the 'Datastreams' tab, showing a list of datastream types: 'Virtual Pin', 'Enum', 'Location' (with an 'UPGRADE' button), 'Digital Pin', and 'Analog Pin'. Below the list is a '+ New Datastream' button. The modal also contains descriptive text about datastreams: 'Datastream regularly flows sensor data that is used for actuators.'

Templates [+ New Template](#)

Search Templates

Create New Template

NAME
Semana do CEC

HARDWARE
ESP8266

CONNECTION TYPE
WiFi

DESCRIPTION
Description

0 / 128

Cancel Done

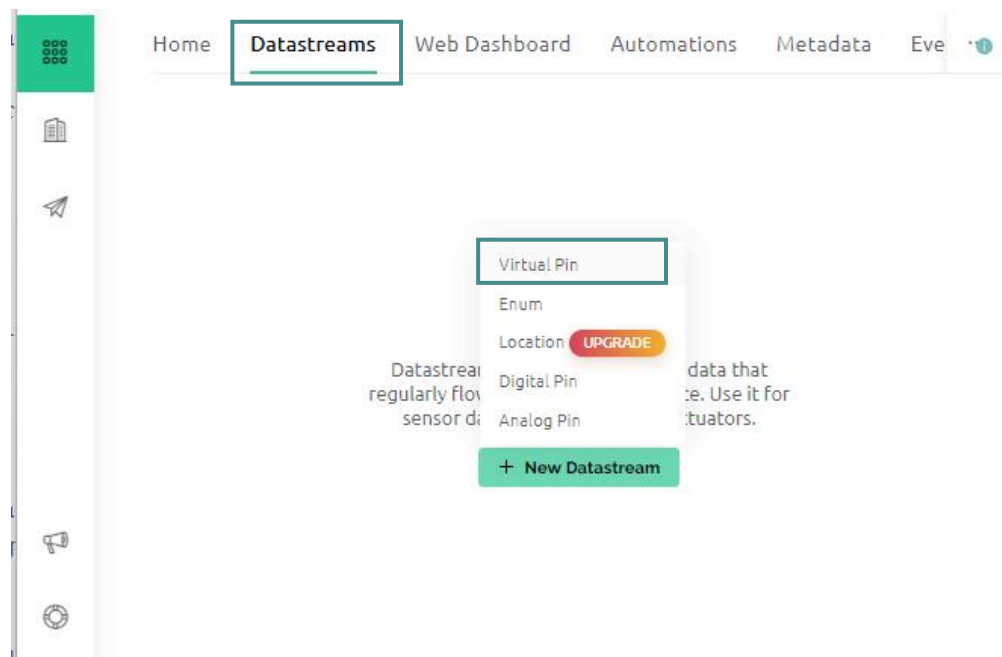
Home **Datastreams** Web Dashboard Automations Metadata Eve

Virtual Pin
Enum
Location **UPGRADE**
Digital Pin
Analog Pin

+ New Datastream

Datastream regularly flows sensor data that is used for actuators.

CONFIGURAR O LED NO BLYNK



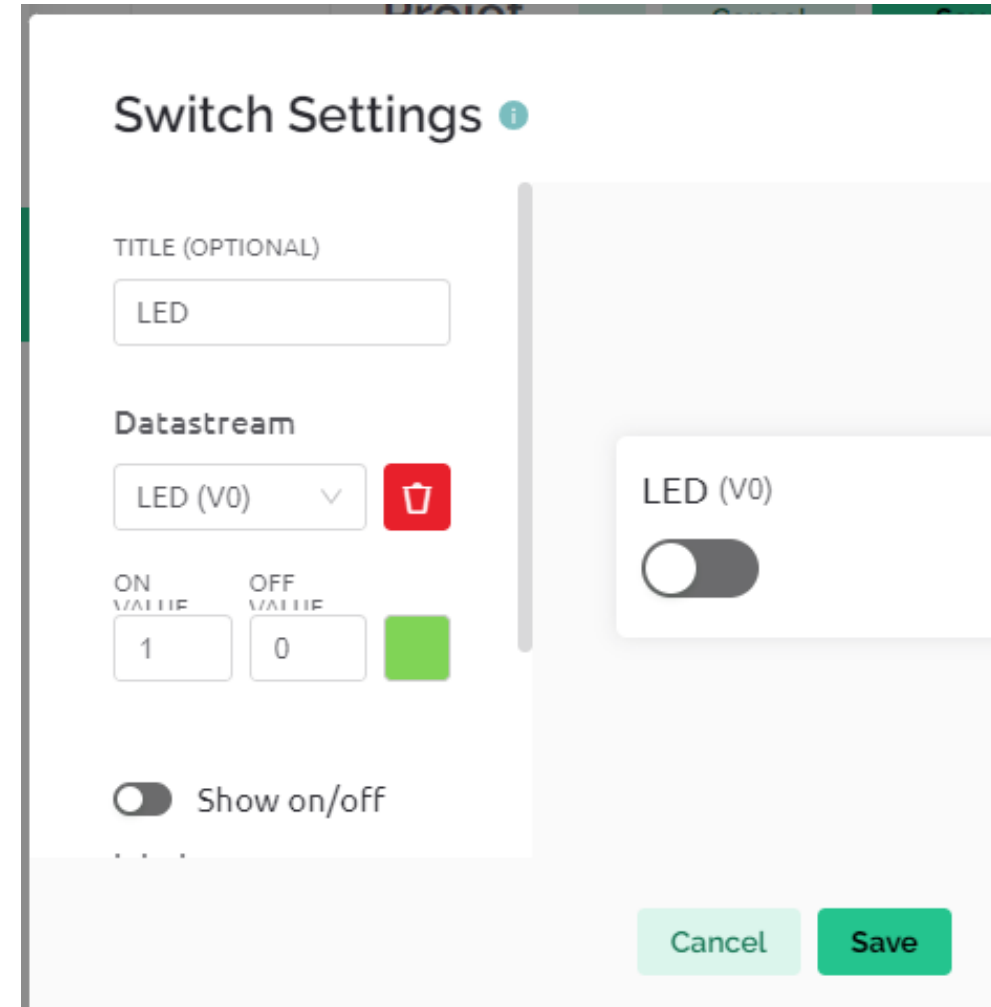
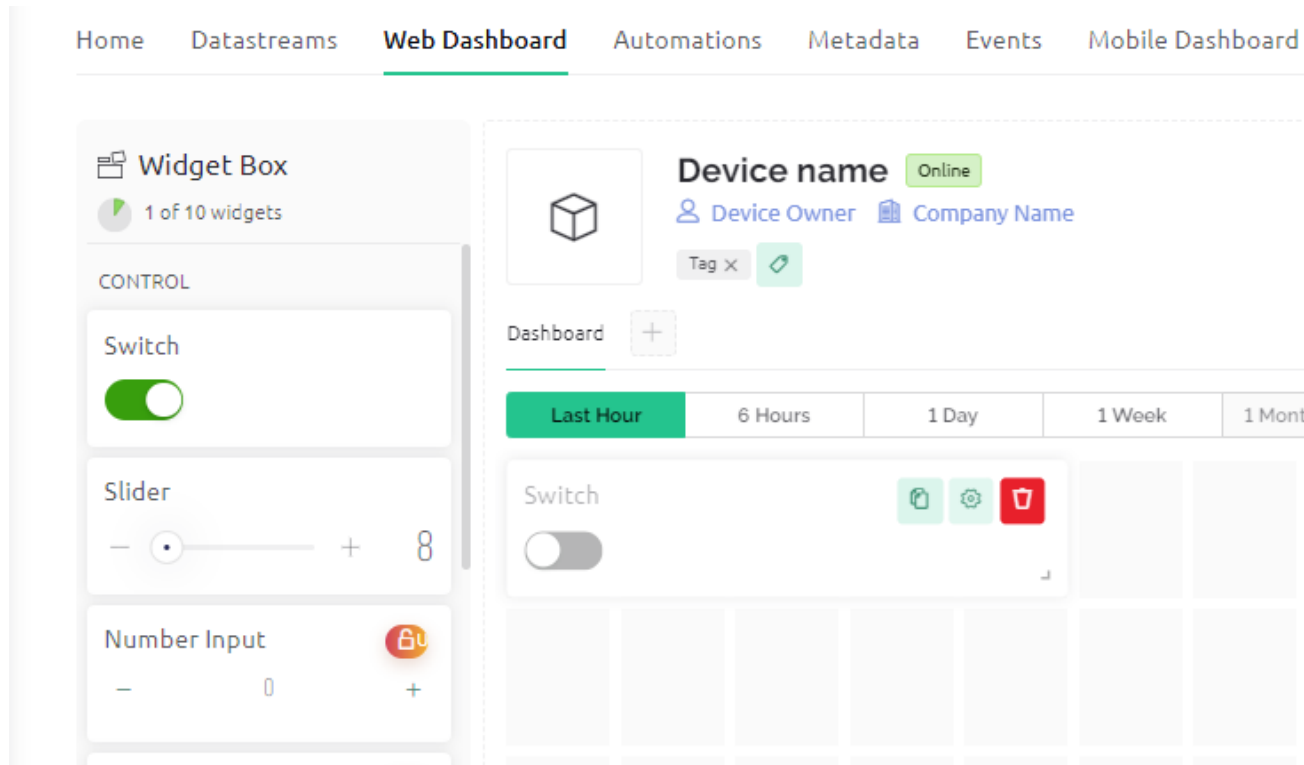
Virtual Pin Datastream

NAME	<input type="text" value="LED"/>	ALIAS	<input type="text" value="LED"/>	<input type="checkbox"/>
PIN	<input type="text" value="V0"/>	DATA TYPE	<input type="text" value="Integer"/>	
UNITS	<input type="text" value="None"/>			
MIN	<input type="text" value="0"/>	MAX	<input type="text" value="1"/>	DEFAULT VALUE
				<input type="text" value="0"/>
<input type="checkbox"/> ADVANCED SETTINGS				

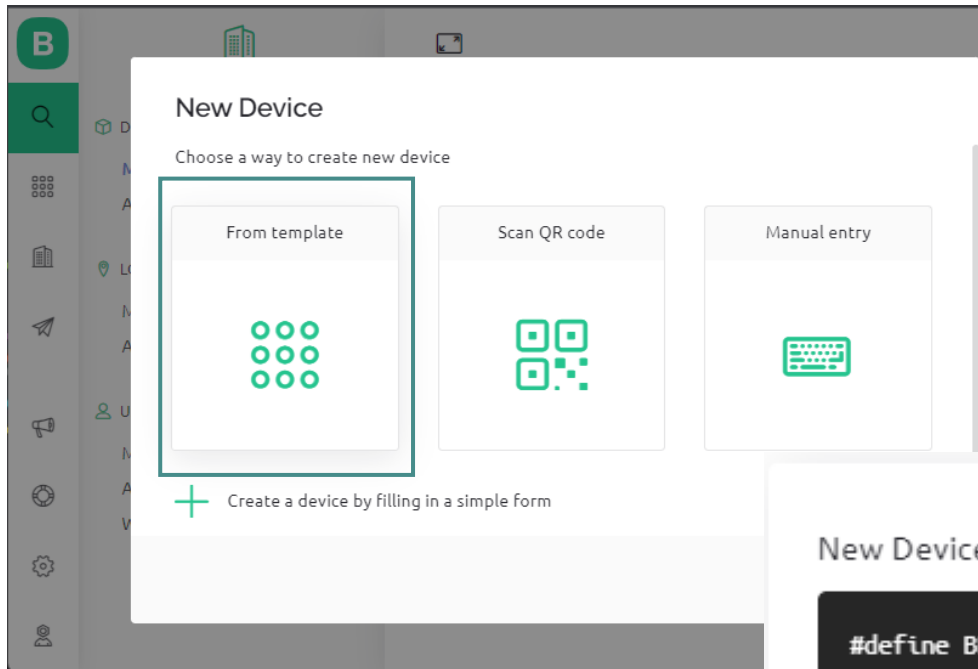
Cancel

Save

CRIAR E CONFIGURAR O DASHBOARD



CONECTAR O ESP8266 AO PROJETO BLYNK



New Device

Create new device by filling in the form below

TEMPLATE

DEVICE NAME

New Device Created!

```
#define BLYNK_TEMPLATE_ID "TMPL24z9RRgWr"  
#define BLYNK_TEMPLATE_NAME "Projeto 1 Semana do CEC"  
#define BLYNK_AUTH_TOKEN  
"FS02z58Xty3ERqAoW8AhM55Ivwt0o2UL"
```

Template ID, Device Name, and AuthToken should be declared at the very top of the firmware code.

[Documentation](#) [Copy to clipboard](#)

CÓDIGO

```
// Colar os dados do projeto
#define BLYNK_TEMPLATE_ID "DADOS_DO_PROJETO"
#define BLYNK_TEMPLATE_NAME "DADOS_DO_PROJETO"
#define BLYNK_AUTH_TOKEN "DADOS_DO_PROJETO"

#include <ESP8266WiFi.h> //Biblioteca para conectar a uma rede Wi-Fi
#include <BlynkSimpleEsp8266.h> //Biblioteca para uso da plataforma Blynk

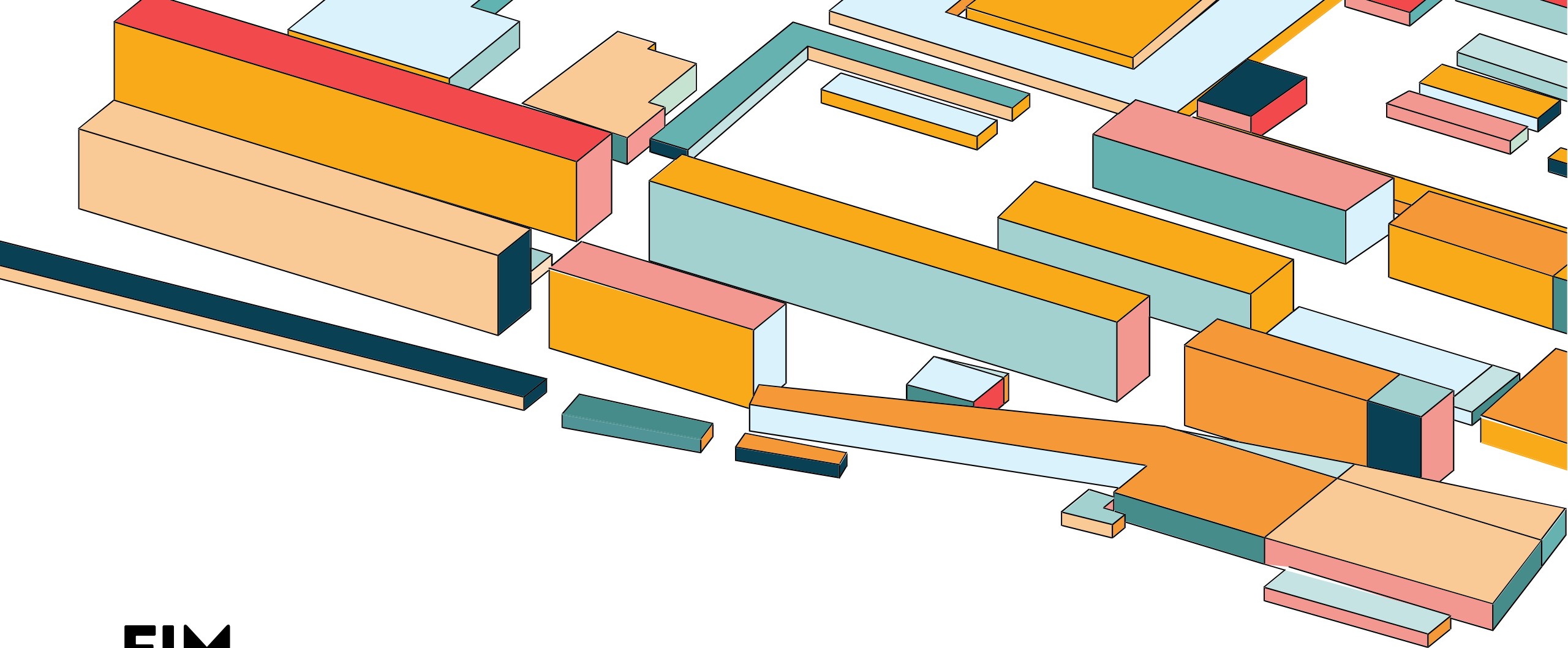
// Configurações do aplicativo e da rede Wi-Fi
char auth[] = BLYNK_AUTH_TOKEN; //Armazena o AuthToken no array auth
char ssid[] = "NOME_REDE"; //Rede Wi-Fi
char pass[] = "SENHA"; //Senha da rede Wi-Fi

int ledPin = 5; // Corresponde ao pino D1

BLYNK_WRITE(V0) {
    digitalWrite(ledPin, param.asInt());
}

void setup() {
    Blynk.begin(auth, ssid, pass); //Inicializa o Blynk passando parâmetros
    pinMode(ledPin, OUTPUT); // Define ledPin como de saída
}

void loop() {
    Blynk.run(); //Chama a função Blynk.run
}
```



FIM
OBRIGADA :)